

MediaCMD 4 Low Level Docs

Drastic Technologies
Version 4
June 22, 2022

Table of Contents

Table of contents

MediaCmd SDK

Introduction

The Drastic MediaCmd SDK is the mechanism by which all the elements of Drastic's DDRs communicate with one another. This includes:

Controlling Drastic Titan and VVW Series DDR Servers, QuickClip locally, and QuickClip with network option remotely
Controlling 9 pin serial VTRs and Servers via Sony, Odetics or VDCP protocol
Receiving commands from 9 pin serial controller via Sony, Odetics or VDCP protocol
Receiving commands from Drastic GUIs, servers and controllers
Building HTML/Ajax status and control pages

MediaCmd is the communication method used within Drastic's DDR products. Any operation you see in a Drastic interface is available to your application through MediaCmd.

Overview

MediaCmd is a simple structure that supports a small, well defined set of commands for communicating transport, status and setup information between components in Drastic's DDR software. There are a number of fields in the structure, but the important fields are:

ctCmd - the primary command of this packet (Play, Pause, Stop, Record, etc)
ISpeed - the transport speed for any play commands (integer where 65520 = normal forward play)
dwPosition - the frame position for any play, pause or record commands
dwStart - the starting frame for any play or record commands (inclusive)
dwEnd - the ending frame for any play or record commands (exclusive)
arbID - clip name, file name or other string/binary data for the command
cFlags - denotes which fields above are valid and their meaning

With the standard initialization of the structure, you can quickly build commands in this structure by changing a few members and sending it. The primary motion commands are ctPlay, ctPause, ctStop, ctRecStop, ctRecord, ctEject and ctTransfer. To get the current state (position, speed, start and end, current clip), the command ctGetState will return a filled in MediaCmd. For setup and less common status (e.g. video input, audio rms level, genlock) there is ctGetValue and ctSetValue. This is documented in the Low Level Header Docs.

Hopefully, you will not have to deal with the MediaCmd structure directly. The SDK includes a series of simple commands that should provide 99% of what your application needs. These functions are simply wrappers that create and send MediaCmd structures. The source for all these functions is provided in the SDK under SRC/General/vvWiFi.cpp in case you need to modify or create new commands. The commands have slightly different names depending on which interface you use, but have the same root name, such as: Play(), PlayFromTo(), Stop(), Pause(), Seek(), Record() and UpdateStatus(). Commands are also included for getting clip lists (GetNextClip()) and EDL elements from '::VTR_TC' time code spaces (EDLResetToStart(), EDLGetEdit()). A selection of the most common settings are also included (SetVideoInput(),

SetAudioInput(), SetVideoGenlock(), GetAudioPeakRMS(), etc). This interface is documented in the MediaCmd Documentation (previously called iVWV Interface Specification).

Installation

To properly work with the MediaCmd SDK, you should have a copy of the QuickClip software installed on your development system. Even if your target application will only use a part of the QuickClip software, it should all be installed for the development phase. Before proceeding with the SDK you should familiarize yourself with QuickClip's operation and toolset. All the elements available within QuickClip are the same elements available to your application through the SDK. Once you have QuickClip installed, you should install the MediaCmd SDK. This will install the headers, libraries and source needed to control QuickClip from your application.

Choosing An Access Method

The SDK access method you should use depends on what you would like your application to do, what programming language you are using and how involved you would like to/need to get in the low level MediaCmd structures. No matter which method you choose, the MediaCmd structure packets are exactly the same. Here are the main access methods, with their pros and cons:

ActiveX

Type: Microsoft ActiveX/COM access method

Pros: Easy to program, 1:1 relationship with QuickClip/XO interface.

Cons: Uses same config as QuickClip/XO. Requires a local copy of QuickClip.

Setup: Register VVW.DLL using RegSvr32.exe in the QuickClip installation directory.

Issues: Difficult to use when communicating via TCP/IP within the same machine. Can be overcome by using the default pipe communication system, but this requires changes for remote network control.

Direct Link

Type: Direct link to VVW.DLL

Pros: No ActiveX layer, code compatible with Linux, Irix, macOS.

Cons: Uses default config from QuickClip/XO, application must be run in QuickClip directory. Requires a local copy of QuickClip.

Setup: Link to vvw.lib, include vvw.h. Copy application into the QuickClip directory before running

Issues: Needs access to VVW.dll and all its support DLLs/DIXs. Still needs to be setup by LocalConfig, DDRConfig or QuickClip/XO

HTTP AJAX/SOAP

Type: XML command interchange with the DDR internal HTTP server, optionally wrapped as SOAP

Pros: Standard, multi language, multi platform.

Cons: Not as fast or efficient as direct connection.

Setup: Standard on version 3 and 4 DrasticDDRs

Issues: Soap is optional, contact Drastic

Command Line App

Type: Network connection, command line sender

Pros: Standard, multi platform (windows, os-x, linux).

Cons: Single command or command list, not interactive.

Setup: Downloadable.

Issues: Command line.

Network DLL

Type: Direct line to vvwNet2.dll

Pros: Consistent interface between local/remote and various OSs. Does not require a local copy of QuickClip.

Cons: Requires vvwNet2.dll and support dlls

Setup: Link to vvwNet2.lib, include vvwNet2.h. Copy dll set from SDK/bin directory with your application

Issues: Use the netOpenLocal function to avoid QuickClip configuration issues. Requires a few DLLs to be added to your application installations. Does not run the client software automatically, so your application may need to start it, depending on what your application is doing.

Network Direct

Type: Direct compile of network sources in your app or your DLL.

Pros: No extra dlls. Easy to customize and modify. Lots of commands already written.

Cons: Your app needs to handle setup and may need to run QuickClip.exe/VVWServer.exe/QCRun.exe.

Setup: Copy source files from vvwNet2 into your project, modify and compile

Issues: Does not run the client software automatically, so your application may need to start it, depending on what your application is doing.

Manual

Type: Use the structures and defines to write your own communication and control layer.

Pros: This is required if you are using an unsupported development platform like PHP.

Cons: Everything has to be built and tested from the ground up.

Setup: None.

Issues: Unless you absolutely have to, this method is not recommended.

SDK Structure

The location of the SDK directories will depend on the location you choose during the installation, but the directories within there will always be the same:

/BIN - Copies of the minimum dll set from a QuickClip installation. /LIB - Libraries required to link the vvwNet2.dll, examples and your application /INC - Header files required to compile vvwNet2.dll, examples and your application /Src/vvwNet2 - The source to our vvwNet2.dll from QuickClip /Src/General - Useful source files that do not compile into examples directly. The most important would be [vwvIF.cpp](#) that is the code behind the SDK functions described below. /Sample - Broken down into sub directories based on access type o /ActiveX - Examples that use the ActiveX control o /Direct - Examples that link directly to DLLs o /Java - Java based examples o /HTTP - Ajax based examples (must use QuickClip HTTP server to run)

Main Documentation Links

PDF version of the MediaCmd Documentation
<http://www.drastic.tv/images/sdk/mediacmd/mediacmd4api.pdf>

Low Level Header Documentation Links

Doxygen PDF file version of the main MediaCmd headers
<http://www.drastic.tv/images/sdk/mediacmd/mediacmd4apilowlevel.pdf>

HTTP XML AJAX Documentation Links

PDF for HTTP/AJAX and PHP MediaCmd
<http://www.drastic.tv/images/sdk/mediacmd/mediacmd4ajaxhttp.pdf>

Command Line Documentation Links

PDF for Command Line MediaCmd
<http://www.drastic.tv/images/sdk/mediacmd/mediacmd4cmdline.pdf>

Older Version 3 Documentation Links

<http://www.drastic.tv/images/sdk/mediacmd/mediacmd3api.pdf>

MEDIACMD Fast Info Page

MEDIACMD Introduction

This document covers inter-device communication protocol used for control and information exchange across a Drastic Media machine, Intranet (LAN) or Internet (WAN). This document does not include information concerning low-level file structures, raw data access, or hardware specific variations.

MediaCMD uses an inter-process piping metaphor to allow the implementation of media distribution and control over a small or large sized installation. A media distribution system normally includes the following components:

- 1 Receiving and interpreting external time code based instructions (Sony/SMPTE/VDR serial protocols)
- 2 Receiving and implementing external file or clip based instructions (Odetics/Alamar)
- 3 Configuration and local control (Graphical User Interface)
- 4 Storage management (Clip/File/Time Code/LTC/VITC/User)
- 5 Transport control (On time/Pre load/Pathing/QOS)
- 6 Hardware audio/video/compression control (Local/Effective Remote)
- 7 of these components must be able to intercommunicate with one or more other components to maintain control and status information within the processes or systems. To allow transportation of these command elements across a local or wide area network, a protocol and transport method compatible with the running platforms and the networks connecting them must be used.
- 8 the protocol is in place, each of the drivers within a machine or network must have an established command set to inter communicate. The object of each component of the system is to receive media commands, send media commands or control media based on those commands. This requirement is met by the drastic MediaCmd structure, which is used for communication between all devices.
- 9 Transport
- 10 physical mechanisms for transporting commands throughout a system are encapsulated in the Drastic VVWNet library. The connection is in some ways similar to a Unix or NT, but does not actually use pipes. The VVWNet provides a simple interface for sending command packets between drivers within the same machine or different machines attached by a network. The VVWNet may use any underlying network protocol, but currently supports TCP/IP and IPX.

MEDIACMD QuickLinks

The structure: [MEDIACMD](#)

The commands: [cmdType](#)

The Flags: [cmdFlags](#)

Channels: [cmdVidChan](#), [cmdAudChan](#), [cmdinf](#)

GetSet Commands [cmdGetSetValue](#)

Basic cmdGetSetValue::gsTc

Clip cmdGetSetValue::gsGetNextClip

Channel cmdGetSetValue::gsAudChan

Audio cmdGetSetValue::gsAudInSelect

Video General cmdGetSetValue::gsVidFreeze

Video Input cmdGetSetValue::gsVidInSelect

Video TBC cmdGetSetValue::gsVidSetup

Video Output cmdGetSetValue::gsVidOutSelect

Signal Type cmdGetSetValue::gsSignalFormat

Compression cmdGetSetValue::gsCompType

Storage cmdGetSetValue::gsTotalStorageAvail

Control cmdGetSetValue::gsLocal

Info cmdGetSetValue::gsVVWVersion

Internal cmdGetSetValue::gsSetHwnds
Mode+Info cmdGetSetValue::gsChannelExist
Returns [GS_NOT_SUPPORTED](#)
Declaration Local [DECLARE_MEDIACMD](#)
Init Local [INIT_MEDIACMD](#)
Init Pointer [INIT_PMEDIACMD](#)
Sizes [SIZEOF_MEDIACMD_BASE](#)

MEDIACMD Sample Commands

The following are sample commands as sent through media command. The samples are taken from the VVW series of products, and as such are guaranteed to work with them. Other OEMs and manufacturers supporting the MediaCmd interface may vary from this specification.

Simple Play

[MEDIACMD::dwCmdID](#) = [MEDIACMD_CURRENT](#)
 [MEDIACMD::dwStructSize](#) = 28
[MEDIACMD::ctCmd](#) = [ctPlay](#)

Simple Pause

[MEDIACMD::dwCmdID](#) = [MEDIACMD_CURRENT](#)
 [MEDIACMD::dwStructSize](#) = 28
[MEDIACMD::ctCmd](#) = [ctPause](#)

Simple Stop

[MEDIACMD::dwCmdID](#) = [MEDIACMD_CURRENT](#)
 [MEDIACMD::dwStructSize](#) = 28
[MEDIACMD::ctCmd](#) = [ctStop](#)

Play From-To

[MEDIACMD::dwCmdID](#) = [MEDIACMD_CURRENT](#)
 [MEDIACMD::dwStructSize](#) = 60
[MEDIACMD::ctCmd](#) = [ctPlay](#)
 [MEDIACMD::cfFlags](#) = [cfUseStart](#) | [cfUseEnd](#)

[MEDIACMD::dwStart](#) = 1800 // 1 minute in frames NTSC NDF
[MEDIACMD::dwEnd](#) = 2100 // 1 minute 10 seconds NTSC NDF

Play DMC (Play at speed)

[MEDIACMD::dwCmdID](#) = [MEDIACMD_CURRENT](#)
 [MEDIACMD::dwStructSize](#) = 36
[MEDIACMD::ctCmd](#) = [ctPlay](#)
 [MEDIACMD::cfFlags](#) = [cfUseSpeed](#)

[MEDIACMD::lSpeed](#) = 32760 // Half play speed forward

Record Edit

[MEDIACMD::dwCmdID](#) = [MEDIACMD_CURRENT](#)
 [MEDIACMD::dwStructSize](#) = 60
[MEDIACMD::ctCmd](#) = [ctRecord](#)
 [MEDIACMD::cfFlags](#) = [cfUseStart](#) | [cfUseEnd](#) | [cfUsePresets](#)
 [MEDIACMD::dwVideoChannels](#) = 0x01 // Record V
[MEDIACMD::dwAudioChannels](#) = 0x03 // Record A1 & A2 (AA)
[MEDIACMD::dwInfoChannels](#) = 0x00

[MEDIACMD::dwStart](#) = 3000

[MEDIACMD::dwEnd](#) = 3200

Play Two Segments

[MEDIACMD::dwCmdID](#) = [MEDIACMD_CURRENT](#)
 [MEDIACMD::dwStructSize](#) = 60

[MEDIACMD::ctCmd](#) = [ctPlay](#)
 [MEDIACMD::cfFlags](#) = [cfUseStart](#) | [cfUseEnd](#)

[MEDIACMD::dwStart](#) = 2200

[MEDIACMD::dwEnd](#) = 2500

[MEDIACMD::dwCmdID](#) = [MEDIACMD_CURRENT](#)
 [MEDIACMD::dwStructSize](#) = 60

[MEDIACMD::ctCmd](#) = [ctPlay](#)
 [MEDIACMD::cfFlags](#) = [cfUseStart](#) | [cfUseEnd](#) | [cfDeferred](#)
 [MEDIACMD::dwStart](#) = 5300

[MEDIACMD::dwEnd](#) = 5450

Play A Named Clip

Char szName = "C:\Adir\Afile.OMF"

[MEDIACMD::dwCmdID](#) = [MEDIACMD_CURRENT](#)
 [MEDIACMD::dwStructSize](#) = 60 +
strlen(szName)

[MEDIACMD::ctCmd](#) = [ctPlay](#)
 [MEDIACMD::cfFlags](#) = [cfUseClipID](#)

strcpy(MEDIACMD::arbID , szName)

Seek To 1 Minute

[MEDIACMD::dwCmdID](#) = [MEDIACMD_CURRENT](#)
 [MEDIACMD::dwStructSize](#) = 48

[MEDIACMD::ctCmd](#) = [ctPause](#)
 [MEDIACMD::cfFlags](#) = [cfUsePosition](#)

[MEDIACMD::dwPosition](#) = 1800 // Frames NTSC NDF

Step Reverse 1 Frame

[MEDIACMD::dwCmdID](#) = [MEDIACMD_CURRENT](#)
 [MEDIACMD::dwStructSize](#) = 48

[MEDIACMD::ctCmd](#) = [ctPause](#)
 [MEDIACMD::cfFlags](#) = [cfUsePositionOffset](#)

[MEDIACMD::dwPosition](#) = (DWORD) -1

Record A 1 Minute Named Clip

Char szName = "C:\Adir\ArecordFile.MOV"

[MEDIACMD::dwCmdID](#) = [MEDIACMD_CURRENT](#)
 [MEDIACMD::dwStructSize](#) = 60 +
strlen(szName)

[MEDIACMD::ctCmd](#) = [ctRecord](#)
 [MEDIACMD::cfFlags](#) = [cfUseClipID](#) | [cfUseEnd](#)

[MEDIACMD::dwEnd](#) = 1800

strcpy(MEDIACMD::arbID , szName)

Record An Odetics or Louth Clip (ext)

(Note: The stop command is required for accuracy on some DDRs)

Char szName = "THISCLIP" // Max 8 char

MEDIACMD::dwCmdID = MEDIACMD_CURRENT
 MEDIACMD::dwStructSize = 60 +
strlen(szName)

MEDIACMD::ctCmd = ctStop // Record ready

MEDIACMD::cfFlags = cfUseClipID
 strcpy(MEDIACMD::arbID , szName)

MEDIACMD::dwCmdID = MEDIACMD_CURRENT
 MEDIACMD::dwStructSize = 60 +
strlen(szName)

MEDIACMD::ctCmd = ctRecord
 MEDIACMD::cfFlags = cfUseClipID

strcpy(MEDIACMD::arbID , szName)

Eject The Current Media

MEDIACMD::dwCmdID = MEDIACMD_CURRENT
 MEDIACMD::dwStructSize = 28

MEDIACMD::ctCmd = ctEject

Setup The Video To Nominal

MEDIACMD::dwCmdID = MEDIACMD_CURRENT
 MEDIACMD::dwStructSize = 60

MEDIACMD::ctCmd = ctSetValue
 MEDIACMD::dwCmdAlt = gsVidSetup

MEDIACMD::dwStart = 0

MEDIACMD::dwCmdID = MEDIACMD_CURRENT
 MEDIACMD::dwStructSize = 60

MEDIACMD::ctCmd = ctSetValue
 MEDIACMD::dwCmdAlt = gsVidVideo

MEDIACMD::dwStart = 0

MEDIACMD::dwCmdID = MEDIACMD_CURRENT
 MEDIACMD::dwStructSize = 60

MEDIACMD::ctCmd = ctSetValue
 MEDIACMD::dwCmdAlt = gsVidHue

MEDIACMD::dwStart = 0

MEDIACMD::dwCmdID = MEDIACMD_CURRENT
 MEDIACMD::dwStructSize = 60

MEDIACMD::ctCmd = ctSetValue
 MEDIACMD::dwCmdAlt = gsVidChroma

MEDIACMD::dwStart = 0

Check The Device Type

MEDIACMD::dwCmdID = MEDIACMD_CURRENT
 MEDIACMD::dwStructSize = 60

MEDIACMD::ctCmd = ctGetValue
 MEDIACMD::dwCmdAlt = gsVtrType

MEDIACMD::dwStart = 0

(Returns: VTR/DDR/Media type in .dwStart)

Change The Default TC Type To VITC

[MEDIACMD::dwCmdID](#) = [MEDIACMD_CURRENT](#)
 [MEDIACMD::dwStructSize](#) = 60
[MEDIACMD::ctCmd](#) = [ctSetValue](#)
 [MEDIACMD::dwCmdAlt](#) = gsViticTc
[MEDIACMD::dwStart](#) = GS_DEFAULT

Transfer From External VTR To Internal Channel

[MEDIACMD::dwCmdID](#) = [MEDIACMD_CURRENT](#)
 [MEDIACMD::dwStructSize](#) = 60
[MEDIACMD::ctCmd](#) = [ctTransfer](#)
 [MEDIACMD::cfFlags](#) = [cfUsePosition](#) | [cfUseStart](#) |
[cfUseEnd](#) | [cfUsePresets](#)
 [MEDIACMD::dwVideoChannels](#) = 0x01 // Capture Video
[MEDIACMD::dwAudioChannels](#) = 0x00 // No Audio
[MEDIACMD::dwInfoChannels](#) = 0x00
[MEDIACMD::dwCmdAlt](#) = hExternalChannel // The VTR channel
[MEDIACMD::dwPosition](#) = 2100 // Where to record to
[MEDIACMD::dwStart](#) = 10850 // Source In on VTR
[MEDIACMD::dwEnd](#) = 11000 // Source Out on VTR

Insert Media From File Over Current

Char szName = "C:\Adir\ArecordFile.MOV"
[MEDIACMD::dwCmdID](#) = [MEDIACMD_CURRENT](#)
 [MEDIACMD::dwStructSize](#) = 60 +
strlen(szName)
[MEDIACMD::ctCmd](#) = [ctInsert](#)
 [MEDIACMD::cfFlags](#) = [cfUsePosition](#) | [cfUseStart](#) |
[cfUseEnd](#) | [cfUseClipID](#)
 [MEDIACMD::dwPosition](#) = 2100 // Insert @ on target channel
[MEDIACMD::dwStart](#) = 0 // Start on source file
[MEDIACMD::dwEnd](#) = 180 // End on source file
strcpy([MEDIACMD::arbID](#) , szName) // Name of source file

Delete A Section Of Media (No Hole)

[MEDIACMD::dwCmdID](#) = [MEDIACMD_CURRENT](#)
 [MEDIACMD::dwStructSize](#) = 60
[MEDIACMD::ctCmd](#) = [ctDelete](#)
 [MEDIACMD::cfFlags](#) = [cfUseStart](#) | [cfUseEnd](#) |
[cfRipple](#) // Remove [cfRipple](#) to leave hole
[MEDIACMD::dwStart](#) = 140500 // Start on target media
[MEDIACMD::dwEnd](#) = 150010 // End on target media

Trim A Clip

[MEDIACMD::dwCmdID](#) = [MEDIACMD_CURRENT](#)
 [MEDIACMD::dwStructSize](#) = 60
[MEDIACMD::ctCmd](#) = [ctTrim](#)
 [MEDIACMD::cfFlags](#) = [cfUsePosition](#) | [cfUseStart](#) |
[cfUseEnd](#)
[MEDIACMD::dwPosition](#) = 2100 // Clip @ position
[MEDIACMD::dwStart](#) = +32 // Clip 32 frames from start

[MEDIACMD::dwEnd](#) = (DWORD) -12 // Clip 12 frame from end

Terminate Session (Restart At Default)

[MEDIACMD::dwCmdID](#) = [MEDIACMD_CURRENT](#)
 [MEDIACMD::dwStructSize](#) = 28

[MEDIACMD::ctCmd](#) = [ctTerminate](#)

Abort Current Command

[MEDIACMD::dwCmdID](#) = [MEDIACMD_CURRENT](#)
 [MEDIACMD::dwStructSize](#) = 28

[MEDIACMD::ctCmd](#) = [ctAbort](#)

MEDIACMD Sample Returns

Sample Returns

[MEDIACMD::dwCmdID](#) == [MEDIACMD_CURRENT](#)

[MEDIACMD::dwStructSize](#) == [SIZEOF_MEDIACMD](#)

[MEDIACMD::ctCmd](#) == cmdType::ctPlay

[MEDIACMD::cfFlags](#) == 0

Normal Play (100% Play Speed)

[MEDIACMD::dwCmdID](#) == [MEDIACMD_CURRENT](#)

[MEDIACMD::dwStructSize](#) == [SIZEOF_MEDIACMD](#)

[MEDIACMD::ctCmd](#) == cmdType::ctRecord

[MEDIACMD::cfFlags](#) == 0

Normal Record (Crash Record)

MEDIACMD DDR Setup

DDR Setup

DDR setup is achieved using cmdType::ctGetValue and cmdType::ctSetValue. Basically, the cmdType::ctGetValue will return the current setting in [MEDIACMD::dwPosition](#) and the bit wise available settings in [MEDIACMD::dwStart](#). To change the setting, send one of the supported bit wise settings in [MEDIACMD::dwPosition](#) using cmdType::ctSetValue. Here is a list of the main setup commands:

- 11 Video Input - cmdGetSetValue::gsVidInSelect : GS_VIDSELECT_???
- 12 Main Signal Format - cmdGetSetValue::gsSignalFormat : GS_SIGFORM_???
- 13 File Format - cmdGetSetValue::gsVideoEncodeFormat : VIDEOWRITETYPE_???: When you change the file format you will need to re populate the compression types and the bit depths. The compression type may no longer exist in the new file type, but the DDR will pick a new default in this case
- 14 Compression - cmdGetSetValue::gsCompType : GS_COMPTYPE_???: When you change the compression type, you will need to re-populate the bit depths, as they will have changed
- 15 Bit Depth/Count - cmdGetSetValue::gsCompChBitCount : GS_BITCOUNT_#
- 16 HDSDI Transfer/Camera - cmdGetSetValue::gsHDSDItransferType : GS_HDSIDIBAYER_???

- 17 Analog Monitor/Up/Down Convert - gsVidAnalogMonitorMethod :
GS_ANALOGMONITORMETHOD_???
- 18 Up Convert HD Type - cmdGetSetValue::gsVidAnalogMonitorHDType : GS_VIDSELECT_???
- 19 Down Convert SD Type - cmdGetSetValue::gsVidAnalogMonitorSDType :
GS_VIDSELECT_???
- 20 Set # Audio Channels - cmdGetSetValue::gsAudChan : Bitwise audio channels
- 21 Audio Input - cmdGetSetValue::gsAudInSelect : GS_AUDSELECT_???
- 22 Audio Bits - cmdGetSetValue::gsAudInputBitRate : 16, 20, 24, 32
- 23 Audio File Type - cmdGetSetValue::gsAudioEncodeFormat : AUDIOWRITETYPE_
- 24 Audio Monitor Pair - cmdGetSetValue::gsAudMonitorSelect : Bitwise audio pair
- 25 Enable/Disable Genlock/Reference - cmdGetSetValue::gsVidOutGenlock : 0/1
- 26 Select Genlock/Reference Source - cmdGetSetValue::gsVidOutGenlockSource :
GS_LOCKSRC_???

VVWTypes Fast Info Page

Introduction

[VVWTypes.h](#) are internal types used to pass information between VVW/QuickClip/MediaReactor modules when a [MEDIACMD](#) is not possible or appropriate. These are provided as a part of the mediacmd sdk to fill in any holes or references not directly available in [mediacmd.h](#).

To do List

Member [VVWXXX_GETSAMPLEFROMLENGTH](#) (__pvvw_)

Find and remove

Member [VVWXXX_SETSAMPLETOLENGTH](#) (__pvvw_, _length)

Find and remove

Class Index

Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

_GUID	14
_LARGE_INTEGER	15
_POINT	16
_RGB	17
DCLIP	18
DFRAME	20
DPOSSIZENAME	24
DRASTIC_CHANNEL	27
DTDIRECT_WAVEHDR	28
FRAME_INFO	31
LARGE_INTEGER_MEMBERS	33

MEDIACMD	34
RECT	38
RECT16	39
SIZE	40
tagBITMAPFILEHEADER	41
tagBITMAPINFO	42
tagBITMAPINFOHEADER	42
tagRGBQUAD	44
tagVFWX_CHANNEL	45
tWAVEFORMATEX	47
VFWAUDIO	48
vfwInflImageInfo	54
VFWINFO	58
VFWNETDIRECT_CHANNEL	80
VFWSYSTEM	81
VFWVIDEO	86
WAVEFORMATEXTENSIBLE	94

File Index

File List

Here is a list of all files with brief descriptions:

E:/drastic/api/mediacmd/src/dtnetdirect.cpp	95
E:/drastic/api/mediacmd/src/dtnetdirect.h	97
E:/drastic/api/mediacmd/src/dtsystemtypes.h	99
E:/drastic/api/mediacmd/src/mediacmd.h	120
E:/drastic/api/mediacmd/src/timecode.h	340
E:/drastic/api/mediacmd/src/vwif.cpp	354
E:/drastic/api/mediacmd/src/vwif.h	380
E:/drastic/api/mediacmd/src/vwtypes.h	406

Class Documentation

_GUID Struct Reference

```
#include <dtsystemtypes.h>
```

Public Attributes

- 27 [DWORD Data1](#)
- 28 [WORD Data2](#)
- 29 [WORD Data3](#)
- 30 [UCHAR Data4](#) [8]

Detailed Description

Definition at line 290 of file dsystemtypes.h.

Member Data Documentation

[DWORD _GUID::Data1](#)

Definition at line 292 of file dsystemtypes.h.

[WORD _GUID::Data2](#)

Definition at line 293 of file dsystemtypes.h.

[WORD _GUID::Data3](#)

Definition at line 294 of file dsystemtypes.h.

[UCHAR _GUID::Data4\[8\]](#)

Definition at line 295 of file dsystemtypes.h.

The documentation for this struct was generated from the following file:

31 E:/drastic/api/mediacmd/src/[dsystemtypes.h](#)

_LARGE_INTEGER Union Reference

```
#include <dsystemtypes.h>
```

Public Attributes

```
32 struct {  
33     DWORD LowPart  
34     LONG HighPart  
35 };  
36 unsigned long long QuadPart
```

Detailed Description

Definition at line 213 of file dsystemtypes.h.

Member Data Documentation

struct { ... }

[LONG _LARGE_INTEGER::HighPart](#)

Definition at line 218 of file dtssystemtypes.h.

[DWORD _LARGE_INTEGER::LowPart](#)

Definition at line 217 of file dtssystemtypes.h.

unsigned long long [_LARGE_INTEGER::QuadPart](#)

Definition at line 223 of file dtssystemtypes.h.

The documentation for this union was generated from the following file:

37 E:/drastic/api/mediacmd/src/[dtssystemtypes.h](#)

_POINT Struct Reference

```
#include <dtssystemtypes.h>
```

Public Attributes

38 [INT](#) x

39 [INT](#) y

Detailed Description

Definition at line 253 of file dtssystemtypes.h.

Member Data Documentation

[INT _POINT::x](#)

Definition at line 255 of file dtssystemtypes.h.

[INT _POINT::y](#)

Definition at line 256 of file dtssystemtypes.h.

The documentation for this struct was generated from the following file:

40 E:/drastic/api/mediacmd/src/[dtsystemtypes.h](#)

_RGB Struct Reference

```
#include <dtsystemtypes.h>
```

Public Attributes

41 [BYTE r](#)
42 [BYTE g](#)
43 [BYTE b](#)
44 [BYTE a](#)

Detailed Description

Definition at line 271 of file dtsystemtypes.h.

Member Data Documentation

[BYTE _RGB::a](#)

Definition at line 276 of file dtsystemtypes.h.

[BYTE _RGB::b](#)

Definition at line 275 of file dtsystemtypes.h.

[BYTE _RGB::g](#)

Definition at line 274 of file dtsystemtypes.h.

[BYTE _RGB::r](#)

Definition at line 273 of file dtsystemtypes.h.

The documentation for this struct was generated from the following file:

45 E:/drastic/api/mediacmd/src/[dtsystemtypes.h](#)

DCLIP Struct Reference

```
#include <vvwtypes.h>
```

Public Attributes

46 [DWORD dwClipIn](#)

Clip In - user defined start of clip, trimmed from dwClipStart.

47 [DWORD dwClipOut](#)

Clip In - user defined end of clip, trimmed from dwClipStart, max == Clip End (the outpoint is never included)

48 [DWORD dwClipStart](#)

Clip Start - the actual physical start of a clip (normally 0)

49 [DWORD dwClipEnd](#)

Clip End - the actual physical end of a clip + 1 (the outpoint is never included)

50 [DWORD dwClipAux](#)

Clip Auxillary - used to denote picon frame, internal key frame, or other dependant on clip type.

51 [DWORD dwClipID](#)

A Numeric Clip or Reel ID - Normally a reel id, but could also be a take or other info.

52 [DWORD dwVidChan](#)

Available video channels bit array.

53 [DWORD dwAudChan](#)

Available audio channels bit array.

54 [DWORD dwInfChan](#)

Available information channels bit array.

55 char * [szClipName](#)

The Clip Name - For louth/odetics compatibility, this should be 8 characters + 0 terminator long.

56 char * [szName](#)

The full name - Usually the file name of the media on disk.

57 char * [szComment](#)

User comment connected to this file. Free form, may contain other key information.

58 char * [szReel](#)

Reel ID or string denoting source tape or media.

Detailed Description

This is the basis of all our clip handling. If is used in tcspc, clipspc, clipctrl, edlxl and most other clip handling areas. It includes info on the clip, trim points, its position within a container, channels available, name, comment and reel.

Definition at line 312 of file vvwtypes.h.

Member Data Documentation

[DWORD DCLIP::dwAudChan](#)

Available audio channels bit array.
Definition at line 328 of file vvwtypes.h.

[DWORD DCLIP::dwClipAux](#)

Clip Auxillary - used to denote picon frame, internal key frame, or other dependant on clip type.
Definition at line 322 of file vvwtypes.h.

[DWORD DCLIP::dwClipEnd](#)

Clip End - the actual physical end of a clip + 1 (the outpoint is never included)
Definition at line 320 of file vvwtypes.h.

[DWORD DCLIP::dwClipID](#)

A Numeric Clip or Reel ID - Normally a reel id, but could also be a take or other info.
Definition at line 324 of file vvwtypes.h.

[DWORD DCLIP::dwClipIn](#)

Clip In - user defined start of clip, trimmed from dwClipStart.
Definition at line 314 of file vvwtypes.h.

[DWORD DCLIP::dwClipOut](#)

Clip In - user defined end of clip, trimmed from dwClipStart, max == Clip End (the outpoint is never included)
Definition at line 316 of file vvwtypes.h.

[DWORD DCLIP::dwClipStart](#)

Clip Start - the actual physical start of a clip (normally 0)
Definition at line 318 of file vvwtypes.h.

[DWORD DCLIP::dwInfChan](#)

Available information channels bit array.
Definition at line 330 of file vvwtypes.h.

DWORD DCLIP::dwVidChan

Available video channels bit array.
Definition at line 326 of file vvwtypes.h.

char* DCLIP::szClipName

The Clip Name - For louth/odetics compatibility, this should be 8 characters + 0 terminator long.
Definition at line 332 of file vvwtypes.h.

char* DCLIP::szComment

User comment connected to this file. Free form, may contain other key information.
Definition at line 336 of file vvwtypes.h.

char* DCLIP::szName

The full name - Usually the file name of the media on disk.
Definition at line 334 of file vvwtypes.h.

char* DCLIP::szReel

Reel ID or string denoting source tape or media.
Definition at line 338 of file vvwtypes.h.

The documentation for this struct was generated from the following file:

59 E:/drastic/api/mediacmd/src/[vwtypes.h](#)

DFRAME Struct Reference

```
#include <vwtypes.h>
```

Public Attributes

```
60 void * pPrev  
Copy From D_LNODE - DO NOT MODIFY.  
61 void * pNext  
Copy From D_LNODE - DO NOT MODIFY.  
62 void * pList  
Copy From D_LNODE - DO NOT MODIFY.  
63 DWORD dwSize
```

size of this structure (+ extra at end if required)

64 [DWORD dwType](#)

65 [DWORD dwReps](#)

Number of repeats of this frame. Uses to create slow motion effects, or save memory on still images.

66 [FRAME_INFO fi](#)

The external timing info for this frame (LTC/VITC/CTL timecode/userbits - See [FRAME_INFO](#))

67 [DWORD dwRepIndex](#)

Internal - Count down for rep usage in slow motion (AvHAL exclusive)

68 [DWORD dwSpeed](#)

The VVW Speed (65520 based) at which this frame is supposed to play. Always forward?? Should be long probably.

69 [DWORD dwExpireMS](#)

Internal - The expected time that this frame may be safely released and deallocated.

70 [DWORD dwPresentationMS](#)

Used to determine the time to display the frame to the viewer, also `dwTimeCaptured` in `AvHal::VfW`.

71 unsigned char * [lpData](#)

72 [DWORD dwBufferLength](#)

73 [DWORD dwBytesUsed](#)

74 [DWORD_PTR dwUser](#)

User var for VfW/Wave driver - internal - do not use.

75 [DWORD dwFlags](#)

76 [DWORD dwLoops](#)

77 [DWORD dwReserved](#) [4]

Internal - Do not use, hardware specific.

78 [WORD resdata](#)

Filler for alignment or extended user data.

Detailed Description

The [DFRAME](#) - A container/marker for a frame of information in memory

This type holds the internal timing, memory allocation, frame flags and dlist elements for one frame of video, audio or information being processed by VVW or MEDIAREACTOR. It should always be allocated by the PhysMem.DLL for maximum speed of memory manipulation. It is used just about everywhere and should not be changed without extremely careful consideration.

Definition at line 358 of file `vvwtypes.h`.

Member Data Documentation

[DWORD DFRAME::dwBufferLength](#)

The maximum length of the buffer pointed to by [DFRAME::lpData](#) This should set by PhysHeap.dll and left alone by the user. If more memory is required, allocate a new [DFRAME](#) and copy the current data into it. Should always be aligned to disk sector size or

4096 (whichever is greater).

Definition at line 483 of file vvwtypes.h.

DWORD DFRAME::dwBytesUsed

The current number of valid unsigned chars pointed to by [DFRAME::lpData](#). Must be less than [DFRAME::dwBufferLength](#). User adjustable as nec.

Definition at line 488 of file vvwtypes.h.

DWORD DFRAME::dwExpireMS

Internal - The expected time that this frame may be safely released and deallocated.

Definition at line 464 of file vvwtypes.h.

DWORD DFRAME::dwFlags

Flags including external flags [_PDFRAMEFLAGS_CLIPSTILL](#), [_PDFRAMEFLAGS_CLIPSTART](#), [_PDFRAMEFLAGS_CLIPEND](#), [_PDFRAMEFLAGS_FIRSTFRAME](#), [_PDFRAMEFLAGS_LASTFRAME](#) and AvHAL internal flags (stripped in AvHAL) [DTWHDR_DONE](#), [DTWHDR_PREPARED](#), [DTWHDR_ENDLOOP](#), [DTWHDR_INQUEUE](#) and [DTVHDR_DONE](#), [DTVHDR_PREPARED](#), [DTVHDR_INQUEUE](#), [DTVHDR_KEYFRAME](#)

Definition at line 556 of file vvwtypes.h.

DWORD DFRAME::dwLoops

Loop counter for WAVEHDR

Definition at line 559 of file vvwtypes.h.

DWORD DFRAME::dwPresentationMS

Used to determine the time to display the frame to the viewer, also dwTimeCaptured in AvHal::VfW.

Definition at line 466 of file vvwtypes.h.

DWORD DFRAME::dwReplIndex

Internal - Count down for rep usage in slow motion (AvHAL exclusive)

Definition at line 460 of file vvwtypes.h.

DWORD DFRAME::dwReps

Number of repeats of this frame. Uses to create slow motion effects, or save memory on still images.

Definition at line 456 of file vvwtypes.h.

DWORD DFRAME::dwReserved[4]

Internal - Do not use, hardware specific.

Definition at line 561 of file vvwtypes.h.

DWORD DFRAME::dwSize

size of this structure (+ extra at end if required)

Definition at line 367 of file vvwtypes.h.

DWORD DFRAME::dwSpeed

The VVW Speed (65520 based) at which this frame is supposed to play. Always forward??
Should be long probably.

Definition at line 462 of file vvwtypes.h.

DWORD DFRAME::dwType

Any combination of the DFRAME_ flags including (not
DFRAME_TYPE_UNCTYPE_MASK): DFRAME_TYPE_RECORD,
DFRAME_TYPE_PLAY, DFRAME_TYPE_PAUSE,
DFRAME_TYPE_UNCOMPRESSED, DFRAME_TYPE_UNCTYPE_MASK,
DFRAME_TYPE_UNCTYPE_YCBCR8, DFRAME_TYPE_UNCTYPE_YUY2,
DFRAME_TYPE_UNCTYPE_V210, DFRAME_TYPE_UNCTYPE_BGR,
DFRAME_TYPE_UNCTYPE_BGRA, DFRAME_TYPE_UNCTYPE_DPX10
#FRAME_TYPE_NOTPHYSHEAP, DFRAME_TYPE_INCLUDES_HEADER
DFRAME_TYPE_AUDIO, DFRAME_TYPE_VIDEO, DFRAME_PROGRESSIVE,
DFRAME_FIELD_INVERT, DFRAME_TIME_INVERT,
DFRAME_ORIENTATION_INVERT, #DFRAME_LARGE_AUDIO,
DFRAME_NEW_FORMAT, DFRAME_TYPE_KEYFRAME,
DFRAME_TYPE_KEYFRAME_I, DFRAME_TYPE_KEYFRAME_P,
DFRAME_TYPE_KEYFRAME_B, DFRAME_SKIP_FRAME

Definition at line 380 of file vvwtypes.h.

DWORD_PTR DFRAME::dwUser

User var for VfW/Wave driver - internal - do not use.

Definition at line 492 of file vvwtypes.h.

FRAME_INFO DFRAME::fi

The external timing info for this frame (LTC/VITC/CTL timecode/userbits - See
[FRAME_INFO](#))

Definition at line 458 of file vvwtypes.h.

unsigned char* [DFRAME::lpData](#)

Start - taken from VIDEOHDR, WAVEHEADER This is actually a WAVEHDR - VideoHdr includes a dwTimeCaptured after the dwunsigned chars used member! Leave it alone as we use the WAVEHDR directly with win32 wave... functions

Definition at line 475 of file vvwtypes.h.

void* [DFRAME::pList](#)

Copy From D_LNODE - DO NOT MODIFY.

Definition at line 365 of file vvwtypes.h.

void* [DFRAME::pNext](#)

Copy From D_LNODE - DO NOT MODIFY.

Definition at line 363 of file vvwtypes.h.

void* [DFRAME::pPrev](#)

Copy From D_LNODE - DO NOT MODIFY.

Definition at line 361 of file vvwtypes.h.

[WORD DFRAME::resdata](#)

Filler for alignment or extended user data.

Definition at line 566 of file vvwtypes.h.

The documentation for this struct was generated from the following file:

79 E:/drastic/api/mediacmd/src/[vwtypes.h](#)

DPOSSIZENAME Struct Reference

```
#include <vwtypes.h>
```

Public Attributes

```
80 void * pPrev  
    Copy From D_LNODE - DO NOT MODIFY.  
81 void * pNext  
    Copy From D_LNODE - DO NOT MODIFY.  
82 void * pList  
    Copy From D_LNODE - DO NOT MODIFY.  
83 DWORD dwSize
```


Size of this structure, must be set or request will be rejected.

84 [DWORD dwType](#)

Data type.

85 [DWORD dwReps](#)

Just for completeness.

86 [FRAME_INFO fi](#)

The external timing info for this frame (LTC/VITC/CTL timecode/userbits - See [FRAME_INFO](#))

87 [__int64 u64Position](#)

Position.

88 [DWORD dwBytesUsed](#)

Size.

89 char [szFileName](#) [_MAX_PATH]

Filename.

Detailed Description

Definition at line 578 of file vvwtypes.h.

Member Data Documentation

[DWORD DPOSSIZENAME::dwBytesUsed](#)

Size.

Definition at line 640 of file vvwtypes.h.

[DWORD DPOSSIZENAME::dwReps](#)

Just for completeness.

This frame is independent of other frames for decode see [DFRAME::dwType](#) This frame is independent of other frames for decode (an MPEG I Frame) see [DFRAME::dwType](#) This frame requires previous keyframe(s) (for MPEG a P Frame) see [DFRAME::dwType](#) This frame requires more than one frame to decode (for MPEG a B Frame) see [DFRAME::dwType](#) This frame should be skipped (decoded, but not displayed) - Used to reach seek frame on a non key frame from key frame see [DFRAME::dwType](#) Number of repeats of this frame. Uses to create slow motion effects, or save memory on still images

Definition at line 631 of file vvwtypes.h.

[DWORD DPOSSIZENAME::dwSize](#)

Size of this structure, must be set or request will be rejected.

Definition at line 587 of file vvwtypes.h.

DWORD DPOSSIZENAME::dwType

Data type.

Definition at line 589 of file vvwtypes.h.

FRAME_INFO DPOSSIZENAME::fi

The external timing info for this frame (LTC/VITC/CTL timecode/userbits - See [FRAME_INFO](#))

Definition at line 633 of file vvwtypes.h.

void* DPOSSIZENAME::pList

Copy From D_LNODE - DO NOT MODIFY.

Definition at line 585 of file vvwtypes.h.

void* DPOSSIZENAME::pNext

Copy From D_LNODE - DO NOT MODIFY.

Definition at line 583 of file vvwtypes.h.

void* DPOSSIZENAME::pPrev

Copy From D_LNODE - DO NOT MODIFY.

Definition at line 581 of file vvwtypes.h.

char DPOSSIZENAME::szFileName[_MAX_PATH]

Filename.

Definition at line 642 of file vvwtypes.h.

__int64 DPOSSIZENAME::u64Position

Position.

Definition at line 638 of file vvwtypes.h.

The documentation for this struct was generated from the following file:

90 E:/drastic/api/mediacmd/src/[vwtypes.h](#)

DRASTIC_CHANNEL Struct Reference

```
#include <vwtypes.h>
```

Public Attributes

91 [DWORD dwLastUpdate](#)

Last update time in ms per the local performance close (vsyncGetCurMs())

92 [FRAME_INFO fi](#)

The current ctl/ltc/vtc frame/userbits and other per frame data.

93 [MEDIACMD mCmd](#)

The current state of the channel as a [MEDIACMD](#) structure.

94 [DWORD * pdwFrame](#)

This will always point at one of the frame counts in the [frame_info](#) structure above.

95 void * [pOwnerHandle](#)

A handle to used by the owner of this struct (usually its base class address)

96 [DWORD dwChannelID](#)

The channel ID (numeric 0-255) of this structure.

Detailed Description

A channel memory area - This is the basis of the DSync system It is used to provide inter and intra channel information on current state in a timely fashion. Most importantly, it is used for the ctTransfer command to move media to and from external vtrs. This allows the transferring channel to get information on the other channel as quickly as possible. It was designed to operate effeciently when both channels are running on the same machine, but also to provide a single interface when a network or other transport is between the two channels.

Definition at line 270 of file vwtypes.h.

Member Data Documentation

[DWORD DRASTIC_CHANNEL::dwChannelID](#)

The channel ID (numeric 0-255) of this structure.

Definition at line 283 of file vwtypes.h.

[DWORD DRASTIC_CHANNEL::dwLastUpdate](#)

Last update time in ms per the local performance close (vsyncGetCurMs())

Definition at line 273 of file vwtypes.h.

[FRAME_INFO DRASTIC_CHANNEL::fi](#)

The current ctl/ltc/vtc frame/userbits and other per frame data.

Definition at line 275 of file vvwtypes.h.

[MEDIACMD_DRASTIC_CHANNEL::mCmd](#)

The current state of the channel as a [MEDIACMD](#) structure.

Definition at line 277 of file vvwtypes.h.

[DWORD* DRASTIC_CHANNEL::pdwFrame](#)

This will always point at one of the frame counts in the frame_info structure above.

Definition at line 279 of file vvwtypes.h.

[void* DRASTIC_CHANNEL::pOwnerHandle](#)

A handle to used by the owner of this struct (usually its base class address)

Definition at line 281 of file vvwtypes.h.

The documentation for this struct was generated from the following file:

97 E:/drastic/api/mediacmd/src/[vwtypes.h](#)

DTDIRECT_WAVEHDR Struct Reference

```
#include <vwtypes.h>
```

Public Attributes

98 [DWORD u32ChunkID](#)

99 [DWORD u32ChunkSize](#)

100 [DWORD u32Format](#)

101 [DWORD u32SubChunkIID](#)

102 [DWORD u32SubChunkISize](#)

103 [WORD wFormatTag](#)

20 2 *Format of audio data, uncompressed* == [DTWAVE_FORMAT_PCM](#) (windows [WAVE_FORMAT_PCM](#)) == 1

104 [WORD nChannels](#)

22 2 *Number of channels in this stream. Usually 1 or 2*

105 [DWORD nSamplesPerSec](#)

24 4 *Number of samples per second. eg. 44100, 22050, 48000 etc*

106 [DWORD nAvgBytesPerSec](#)

28 4 *Average unsigned chars Per Second (see also [VWVAUDIO::dwRate](#))* == [nBlockAlign](#) * [nSamplesPerSec](#)

107 [WORD nBlockAlign](#)

32 2 *Size of a sample group.* == to [nChannels](#) * ([wBitsPerSample](#) / 8) eg. Stereo 16bit = 4, Mono 16bit = 2, Stereo 8Bit = 2

- 108 [WORD wBitsPerSample](#)
34 2 Number of bits per sample, normally 8 or 16
109 [DWORD u32SubChunk2ID](#)
110 [DWORD u32SubChunk2Size](#)
-

Detailed Description

Special wave header for quickly creating wave file Simply fill in, write and then add audio samples afterward and then update the sizes

Definition at line 1459 of file vvwtypes.h.

Member Data Documentation

[DWORD DTDIRECT_WAVEHDR::nAvgBytesPerSec](#)

28 4 Average unsigned chars Per Second (see also [VVWAUDIO::dwRate](#)) == [nBlockAlign](#) * [nSamplesPerSec](#)

Definition at line 1500 of file vvwtypes.h.

[WORD DTDIRECT_WAVEHDR::nBlockAlign](#)

32 2 Size of a sample group. == to [nChannels](#) * ([wBitsPerSample](#) / 8) eg. Stereo 16bit = 4, Mono 16bit = 2, Stereo 8Bit = 2

Definition at line 1502 of file vvwtypes.h.

[WORD DTDIRECT_WAVEHDR::nChannels](#)

22 2 Number of channels in this stream. Usually 1 or 2

Definition at line 1496 of file vvwtypes.h.

[DWORD DTDIRECT_WAVEHDR::nSamplesPerSec](#)

24 4 Number of samples per second. eg. 44100, 22050, 48000 etc

Definition at line 1498 of file vvwtypes.h.

[DWORD DTDIRECT_WAVEHDR::u32ChunkID](#)

0 4 ChunkID Contains the letters "RIFF" in ASCII form (0x52494646 big-endian form).

Definition at line 1465 of file vvwtypes.h.

[DWORD DTDIRECT_WAVEHDR::u32ChunkSize](#)

4 4 ChunkSize 36 + SubChunk2Size, or more precisely: 4 + (8 + SubChunk1Size) + (8 + SubChunk2Size) This is the size of the rest of the chunk following this number. This is the size of the entire file in bytes minus 8 bytes for the two fields not included in this count:

ChunkID and ChunkSize.

Definition at line 1474 of file vvwtypes.h.

DWORD DTDIRECT_WAVEHDR::u32Format

8 4 Format Contains the letters "WAVE" (0x57415645 big-endian form).

Definition at line 1479 of file vvwtypes.h.

DWORD DTDIRECT_WAVEHDR::u32SubChunk1ID

12 4 Subchunk1ID Contains the letters "fmt " (0x666d7420 big-endian form).

Definition at line 1488 of file vvwtypes.h.

DWORD DTDIRECT_WAVEHDR::u32SubChunk1Size

16 4 Subchunk1Size 16 for PCM. This is the size of the rest of the Subchunk which follows this number.

Definition at line 1492 of file vvwtypes.h.

DWORD DTDIRECT_WAVEHDR::u32SubChunk2ID

36 4 Subchunk2ID Contains the letters "data" (0x64617461 big-endian form).

Definition at line 1513 of file vvwtypes.h.

DWORD DTDIRECT_WAVEHDR::u32SubChunk2Size

40 4 Subchunk2Size == NumSamples * NumChannels * BitsPerSample/8 This is the number of bytes in the data. You can also think of this as the size of the read of the subchunk following this number.

Definition at line 1520 of file vvwtypes.h.

WORD DTDIRECT_WAVEHDR::wBitsPerSample

34 2 Number of bits per sample, normally 8 or 16

Definition at line 1504 of file vvwtypes.h.

WORD DTDIRECT_WAVEHDR::wFormatTag

20 2 Format of audio data, uncompressed == [DTWAVE_FORMAT_PCM](#) (windows WAVE_FORMAT_PCM) == 1

Definition at line 1494 of file vvwtypes.h.

The documentation for this struct was generated from the following file:

111 E:/drastic/api/mediacmd/src/[vwtypes.h](#)

FRAME_INFO Struct Reference

```
#include <vvwtypes.h>
```

Public Attributes

- 112 [DWORD dwFrame](#)
Current time code, Control (CTL), our internal time code.
- 113 [DWORD dwLtcFrame](#)
Current LTC time code from original tape or internal generation.
- 114 [DWORD dwLtcUb](#)
Current LTC user Bits from original tape or internal generation.
- 115 [DWORD dwVitcFrame](#)
Current VITC time code from original tape or internal generation.
- 116 [DWORD dwVitcUb](#)
Current VITC user bits from original tape or internal generation.
- 117 [BYTE arbKeyCode](#) [8]
Version 4: Key code value (all used)
- 118 [BYTE arbInkCode](#) [8]
Version 4: Ink code value (6-8 = count, upper 1-5 = code)
- 119 [DWORD dwDataFlags](#)
- 120 [DWORD dwDataSize](#)
- 121 [BYTE arbData](#) [FRAMEINFO_MAX_DATA_SIZE]
- 122 [size_t dwVitcAux](#)
- 123 [DWORD dwCCData](#)

Detailed Description

A Frame Info Packet - Part of the DFrame structure maintained with each memory frame passed through the MR or VVW system DFrame maintains the memory and internal timing info for the system, whereas frame_info maintains the external meta data from the original source.

Definition at line 168 of file vvwtypes.h.

Member Data Documentation

[BYTE FRAME_INFO::arbData](#)[FRAMEINFO_MAX_DATA_SIZE]

Version 4: extra space for EIA/CIE-708 captioning. The old SD 608 could only manage 4 characters per frame (2 per field). The HD 708 supports up to 40 characters per second. Version 4 Mark 2: Need enough room for VBI FRAME ELEMENT data from 436. Max we will allow is 3 lines of 10 bit hd 1920x1080. The mapping is always Y only, so worst case is 1920 samples / 3 samples per dword (at 10 bit) * 4 bytes per dword = 7680 bytes. Round up to 8K

Definition at line 208 of file vvwtypes.h.

[BYTE FRAME_INFO::arblnkCode](#)[8]

Version 4: Ink code value (6-8 = count, upper 1-5 = code)

Definition at line 182 of file vvwtypes.h.

BYTE FRAME_INFO::arbKeyCode[8]

Version 4: Key code value (all used)

Definition at line 180 of file vvwtypes.h.

DWORD FRAME_INFO::dwCCData

PRE V4!: Captured close caption data from line 21 or if DFRAME_TYPE_FI_PTR_DATA then this contains the size of the data area pointed to by dwVitcAux. NOTE: When the DFRAME_TYPE_FI_PTR_DATA is up in the [DFRAME::dwType](#) member, then this is the size of the data area. It is set initially to (DFRAME_MAX_EXTRA_DATA_SIZE+1) which is illegal. If you have dwVitcAux!=0 and dwCCData != 0 && dwCCData <= DFRAME_MAX_EXTRA_DATA_SIZE then you probably have a valid data area. If the DFRAME_TYPE_FI_PTR_DATA in [DFRAME::dwType](#) is available and up and the above conditions are met, then you are definately looking at valid data.

Definition at line 236 of file vvwtypes.h.

DWORD FRAME_INFO::dwDataFlags

Version 4: Flags of what is in the data space

Definition at line 193 of file vvwtypes.h.

DWORD FRAME_INFO::dwDataSize

Version 4: Size used in the data space

Definition at line 196 of file vvwtypes.h.

DWORD FRAME_INFO::dwFrame

Current time code, Control (CTL), our internal time code.

Definition at line 170 of file vvwtypes.h.

DWORD FRAME_INFO::dwLtcFrame

Current LTC time code from original tape or internal generation.

Definition at line 172 of file vvwtypes.h.

DWORD FRAME_INFO::dwLtcUb

Current LTC user Bits from original tape or internal generation.

Definition at line 174 of file vvwtypes.h.

size_t FRAME_INFO::dwVitcAux

Pre V4!: Current VITC aux for extra vitc time code (3 line) or other info, or if

DFRAME_TYPE_FI_PTR_DATA is dwType of PDRFRAME is set, then this is a pointer to the extended data area. This are should exist at the end of the [DFRAME](#) structure. NOTE: When the DFRAME_TYPE_FI_PTR_DATA is up in the [DFRAME::dwType](#) member, then this is the size of the data area. It is set initially to (DFRAME_MAX_EXTRA_DATA_SIZE+1) which is illegal. If you have dwVitcAux!=0 and dwCCData != 0 && dwCCData <= DFRAME_MAX_EXTRA_DATA_SIZE then you probably have a valid data area. If the DFRAME_TYPE_FI_PTR_DATA in [DFRAME::dwType](#) is available and up and the above conditions are met, then you are definately looking at valid data. ===== Expanded to 64 bits to allow for pointers

Definition at line 223 of file vvwtypes.h.

[DWORD FRAME_INFO::dwVitcFrame](#)

Current VITC time code from original tape or internal generation.

Definition at line 176 of file vvwtypes.h.

[DWORD FRAME_INFO::dwVitcUb](#)

Current VITC user bits from original tape or internal generation.

Definition at line 178 of file vvwtypes.h.

The documentation for this struct was generated from the following file:

124 E:/drastic/api/mediacmd/src/[vwtypes.h](#)

LARGE_INTEGER_MEMBERS Struct Reference

```
#include <dtsystemtypes.h>
```

Public Attributes

125 [DWORD LowPart](#)

126 [LONG HighPart](#)

Detailed Description

Definition at line 209 of file dtsystemtypes.h.

Member Data Documentation

[LONG LARGE_INTEGER_MEMBERS::HighPart](#)

Definition at line 211 of file dtsystemtypes.h.

[DWORD LARGE_INTEGER MEMBERS::LowPart](#)

Definition at line 210 of file dtssystemtypes.h.

The documentation for this struct was generated from the following file:

127 E:/drastic/api/mediacmd/src/[dtssystemtypes.h](#)

MEDIACMD Struct Reference

```
#include <mediacmd.h>
```

Public Attributes

128 void * [pPrev](#)
INTERNAL dlist.dll Link List Pointers - do not use.

129 void * [pNext](#)
INTERNAL dlist.dll Link List Pointers - do not use.

130 void * [pList](#)
INTERNAL dlist.dll Link List Pointers - do not use.

131 [DWORD dwPrevPtr32Fill](#)

132 [DWORD dwNextPtr32Fill](#)

133 [DWORD dwListPtr32Fill](#)

134 [DWORD dwCmdID](#)

135 [DWORD dwStructSize](#)

136 [DWORD dwChannel](#)

137 enum [cmdType](#) [ctCmd](#)

138 [DWORD cfFlags](#)

139 [LONG lSpeed](#)

140 [DWORD dwVideoChannels](#)

141 [DWORD dwAudioChannels](#)

142 [DWORD dwInfoChannels](#)

143 [DWORD dwCmdAlt](#)

144 [DWORD dwPosition](#)

145 [DWORD dwStart](#)

146 [DWORD dwEnd](#)

147 unsigned char [arbID](#) [CMD_MAX_CLIP_ID_LEN+2+2]

Detailed Description

Definition at line 6799 of file mediacmd.h.

Member Data Documentation

unsigned char [MEDIACMD::arbID](#)[CMD_MAX_CLIP_ID_LEN+2+2]

Free form memory area mostly used for string and clip handling. Valid if the [cfFlags](#) member

includes `cmdFlags::cfUseClipID` Basic Clip Name: first 8 unsigned charS alpha number, may include spaces, terminated by 0

Two Clip Names: 'Basic Clip Name' followed immediately by another 'Basic Clip Name'
Clip+File Name: 'Basic Clip Name' followed by a null terminated file/path name. String: Null terminated ANSI string.

Definition at line 6947 of file `mediacmd.h`.

DWORD MEDIACMD::cfFlags

The `cfFlags` member contains flags that modify the operation of the other structure members. These flags are of the type [cmdFlags](#). The flags normally specify which other members are to be used for the command as well as modifiers for delaying or otherwise augmenting commands. Basic flags include: `cmdFlags::cfDeferred`, `cmdFlags::cfUseSpeed`, `cmdFlags::cfUsePosition`, `cmdFlags::cfUseStart`, `cmdFlags::cfUseEnd`, `cmdFlags::cfUsePositionOffset`, `cmdFlags::cfUseClipID`.

Definition at line 6862 of file `mediacmd.h`.

enum cmdType MEDIACMD::ctCmd

The `ctCmd` member contains the basic or overall command. It is of the type [cmdType](#). It includes transport and setup commands that may be immediate or delayed depending on the rest of the structure. Basic commands include: `cmdType::ctPlay`, `cmdType::ctStop`, `cmdType::ctPause` (or `seek`), `cmdType::ctRecord`, `cmdType::ctGetState`, `cmdType::ctGetValue` and `cmdType::ctSetValue`. CAUTION: This is an enum and we send this structure as a binary between various systems. Fortunately MSVC 2005/2008 and gcc 4.x both still make enums 32 bits, so this works. If we ever port to a 64 bit compiler that changes this, it will break the binary compatibility.

Definition at line 6853 of file `mediacmd.h`.

DWORD MEDIACMD::dwAudioChannels

AUDIO Bit array of channels where

1 = first channel

2 = second channel

4 = third channel

8 = fourth channel

Also, [cmdAudChan](#) enum may be used

Definition at line 6905 of file `mediacmd.h`.

DWORD MEDIACMD::dwChannel

The target channel. For direct connect channels (e.g. in the same machine) this member is ignored. For serial/network/piped channels, this allows a single transport to access multiple channels. It is also used for kernel<->user mode commands for `DCoMctrl.sys` and `vvwCtl.dll` as well as `vvwNet.dll`.

Definition at line 6841 of file `mediacmd.h`.

DWORD MEDIACMD::dwCmdAlt

This member may have many meaning depending on the rest of the [MEDIACMD](#) structure.

if [ctCmd](#) is `cmdType::ctGetValue`, `cmdType::ctSetValue` or `cmdType::ctValueSupported` then it contains the [cmdGetSetValue](#) to use if [ctCmd](#) is `cmdType::ctTransfer` it contains the source channel for the transfer (the command is always set to the target) if [cfFlags](#) includes `cmdFlags::cfUseCmdAlt` and `cmdFlags::cfTimeMs` is contains a millisecond version of the performance clock if [ctCmd](#) is `cmdType::ctRecord` then it may be used as a millisecond offset to start of record if [ctCmd](#) is `cmdType::ctPlay` then it may be used as a millisecond offset to start of playback

Definition at line 6920 of file `mediacmd.h`.

DWORD MEDIACMD::dwCmdID

The command identifier is used to confirm that the command is properly formatted and of a version that the receiver understands. This member should always be set to [MEDIACMD_CURRENT](#)

Definition at line 6825 of file `mediacmd.h`.

DWORD MEDIACMD::dwEnd

For most commands, this will be the ending frame counter position for the command.

Check of `cmdFlags::cfUseEnd` or `cmdFlags::cfUseEndOffset` (becomes long against current position) to make sure this member is valid.

For `cmdType::ctGetValue`, `cmdType::ctSetValue` or `cmdType::ctValueSupported` this is secondary set and return member.

Definition at line 6938 of file `mediacmd.h`.

DWORD MEDIACMD::dwInfoChannels

INFO Bit array if channels. See [cmdinf](#) for possible channels including `cmdinf::LTC`, `cmdinf::VITC`, `cmdinf::Copyright`, etc.

Definition at line 6910 of file `mediacmd.h`.

DWORD MEDIACMD::dwListPtr32Fill

Definition at line 6819 of file `mediacmd.h`.

DWORD MEDIACMD::dwNextPtr32Fill

Definition at line 6819 of file `mediacmd.h`.

DWORD MEDIACMD::dwPosition

For most commands, this will be the current or target frame counter position for the command.

Check of `cmdFlags::cfUsePosition` or `cmdFlags::cfUsePositionOffset` (becomes long against current position) to make sure this member is valid.

For `cmdType::ctGetValue`, `cmdType::ctSetValue` or `cmdType::ctValueSupported` this is primary set and return member.

Definition at line 6926 of file `mediacmd.h`.

DWORD MEDIACMD::dwPrevPtr32Fill

We need to make sure the MediaCMD structure is the same size in 64 bit and 32 bit compiles. The 3 x 64 bit pointers are 24 bytes in total. The 3 x 32 pointers are 12, so we have 24 bytes added in the xxxPtr32Fills below.

Definition at line 6819 of file mediacmd.h.

DWORD MEDIACMD::dwStart

For most commands, this will be the starting frame counter position for the command.

Check of cmdFlags::cfUseStart or cmdFlags::cfUseStartOffset (becomes long against current position) to make sure this member is valid.

For cmdType::ctGetValue, cmdType::ctSetValue or cmdType::ctValueSupported this is secondary set and return member.

Definition at line 6932 of file mediacmd.h.

DWORD MEDIACMD::dwStructSize

This member contains the entire size of the structure being sent. Certain commands may not require all fields to be completely understood. Most commands will send the bulk of the structure and remove only unused parts of arbID[], though this should not be counted on. It may be assumed that all commands will include the [pPrev](#), [pNext](#), [pList](#), [dwCmdID](#), [dwStructSize](#), [dwChannel](#), [ctCmd](#), and [cfFlags](#) members with every command so dwStructSize must be at least 32 unsigned chars in length.

Definition at line 6834 of file mediacmd.h.

DWORD MEDIACMD::dwVideoChannels

VIDEO Bit array of channels where

- 1 = first channel
- 2 = second channel
- 4 = third channel
- 8 = fourth channel

Also, [cmdVidChan](#) enum may be used

Definition at line 6896 of file mediacmd.h.

LONG MEDIACMD::lSpeed

This member controls the speed of a command. Normally it is used with cmdType::ctPlay and required cmdFlags::cfUseSpeed to be set in the [cfFlags](#) member to be used. The defines [SPD_FWD_PLAY](#), [SPD_PAUSE](#), [SPD_REV_PLAY](#), [SPD_FWD_MAX](#), [SPD_REV_MAX](#) can be used, or any other valid speed. This table outlines the basic linear speed changes.

SPD_REV_MAX	SPD_REV_PLAY	SPD_PAUSE	SPD_FWD_PLAY	SPD_FWD_MAX
-5896800	-65520	0	65520	5896800
-90x	-1x	0	1x	90x
-9000%	-100%	0	100%	9000%
-90.0	-1.0	0	1.0	90.0

Some Normal Speeds (for reverse, set to minus)

10 times play	10.0	1000%	655200	
5 times play	5.0	500%	327600	
2 times Play	2.0	200%	131040	
Play Normal	1.0	100%	65520	

Two Third	0.66	66%	43680
Half Play	0.5	50%	32760
One Third	0.33	33%	21840

note: Speed table is linear (log like serial control must be converted)

Definition at line 6887 of file mediacmd.h.

void* [MEDIACMD::pList](#)

INTERNAL dlist.dll Link List Pointers - do not use.

Definition at line 6807 of file mediacmd.h.

void* [MEDIACMD::pNext](#)

INTERNAL dlist.dll Link List Pointers - do not use.

Definition at line 6805 of file mediacmd.h.

void* [MEDIACMD::pPrev](#)

INTERNAL dlist.dll Link List Pointers - do not use.

Definition at line 6803 of file mediacmd.h.

The documentation for this struct was generated from the following file:

148 E:/drastic/api/mediacmd/src/[mediacmd.h](#)

RECT Struct Reference

```
#include <dtsystemtypes.h>
```

Public Attributes

149 int [left](#)
150 int [top](#)
151 int [right](#)
152 int [bottom](#)

Detailed Description

Definition at line 242 of file dtsystemtypes.h.

Member Data Documentation

int [RECT::bottom](#)

Definition at line 247 of file dtssystemtypes.h.

int [RECT::left](#)

Definition at line 244 of file dtssystemtypes.h.

int [RECT::right](#)

Definition at line 246 of file dtssystemtypes.h.

int [RECT::top](#)

Definition at line 245 of file dtssystemtypes.h.

The documentation for this struct was generated from the following file:

153 E:/drastic/api/mediacmd/src/[dtssystemtypes.h](#)

RECT16 Struct Reference

```
#include <vfwtypes.h>
```

Public Attributes

154 short [left](#)

left side of rect - see microsoft [RECT](#) struct

155 short [top](#)

top of rect - see microsoft [RECT](#) struct

156 short [right](#)

right side of rect - see microsoft [RECT](#) struct

157 short [bottom](#)

bottom of rect - see microsoft [RECT](#) struct

Detailed Description

Those stupid idiots at Microsoft forgot to change this structure when making the VfW headers for Win32. The result is that the AVIStreamHeader is correct in 16-bit Windows, but the 32-bit headers define [RECT](#) as longs instead of shorts. This creates invalid AVI files!!!!

Definition at line 141 of file vfwtypes.h.

Member Data Documentation

short [RECT16::bottom](#)

bottom of rect - see microsoft [RECT](#) struct
Definition at line 149 of file vvwtypes.h.

short [RECT16::left](#)

left side of rect - see microsoft [RECT](#) struct
Definition at line 143 of file vvwtypes.h.

short [RECT16::right](#)

right side of rect - see microsoft [RECT](#) struct
Definition at line 147 of file vvwtypes.h.

short [RECT16::top](#)

top of rect - see microsoft [RECT](#) struct
Definition at line 145 of file vvwtypes.h.

The documentation for this struct was generated from the following file:

158 E:/drastic/api/mediacmd/src/[vwtypes.h](#)

SIZE Struct Reference

```
#include <dtssystemtypes.h>
```

Public Attributes

159 int [cx](#)
160 int [cy](#)

Detailed Description

Definition at line 262 of file dtssystemtypes.h.

Member Data Documentation

int [SIZE::cx](#)

Definition at line 264 of file dtssystemtypes.h.

int [SIZE::cy](#)

Definition at line 265 of file dtssystemtypes.h.

The documentation for this struct was generated from the following file:

161 E:/drastic/api/mediacmd/src/[dtssystemtypes.h](#)

tagBITMAPFILEHEADER Struct Reference

```
#include <dtssystemtypes.h>
```

Public Attributes

162 [WORD bfType](#)
163 [DWORD bfSize](#)
164 [WORD bfReserved1](#)
165 [WORD bfReserved2](#)
166 [DWORD bfOffBits](#)

Detailed Description

Definition at line 346 of file dtssystemtypes.h.

Member Data Documentation

[DWORD tagBITMAPFILEHEADER::bfOffBits](#)

Definition at line 351 of file dtssystemtypes.h.

[WORD tagBITMAPFILEHEADER::bfReserved1](#)

Definition at line 349 of file dtssystemtypes.h.

[WORD tagBITMAPFILEHEADER::bfReserved2](#)

Definition at line 350 of file dtssystemtypes.h.

[DWORD tagBITMAPFILEHEADER::bfSize](#)

Definition at line 348 of file dtssystemtypes.h.

[WORD tagBITMAPFILEHEADER::bfType](#)

Definition at line 347 of file dtssystemtypes.h.

The documentation for this struct was generated from the following file:

167 E:/drastic/api/mediacmd/src/[dtssystemtypes.h](#)

tagBITMAPINFO Struct Reference

```
#include <dtssystemtypes.h>
```

Public Attributes

168 [BITMAPINFOHEADER bmiHeader](#)

169 [RGBQUAD bmiColors](#) [1]

Detailed Description

Definition at line 341 of file dtssystemtypes.h.

Member Data Documentation

[RGBQUAD tagBITMAPINFO::bmiColors](#)[1]

Definition at line 343 of file dtssystemtypes.h.

[BITMAPINFOHEADER tagBITMAPINFO::bmiHeader](#)

Definition at line 342 of file dtssystemtypes.h.

The documentation for this struct was generated from the following file:

170 E:/drastic/api/mediacmd/src/[dtssystemtypes.h](#)

tagBITMAPINFOHEADER Struct Reference

```
#include <dtsystemtypes.h>
```

Public Attributes

- 171 [DWORD biSize](#)
 - 172 [LONG biWidth](#)
 - 173 [LONG biHeight](#)
 - 174 [WORD biPlanes](#)
 - 175 [WORD biBitCount](#)
 - 176 [DWORD biCompression](#)
 - 177 [DWORD biSizeImage](#)
 - 178 [LONG biXPelsPerMeter](#)
 - 179 [LONG biYPelsPerMeter](#)
 - 180 [DWORD biClrUsed](#)
 - 181 [DWORD biClrImportant](#)
-

Detailed Description

Definition at line 327 of file dtsystemtypes.h.

Member Data Documentation

[WORD tagBITMAPINFOHEADER::biBitCount](#)

Definition at line 332 of file dtsystemtypes.h.

[DWORD tagBITMAPINFOHEADER::biClrImportant](#)

Definition at line 338 of file dtsystemtypes.h.

[DWORD tagBITMAPINFOHEADER::biClrUsed](#)

Definition at line 337 of file dtsystemtypes.h.

[DWORD tagBITMAPINFOHEADER::biCompression](#)

Definition at line 333 of file dtsystemtypes.h.

[LONG tagBITMAPINFOHEADER::biHeight](#)

Definition at line 330 of file dtsystemtypes.h.

[WORD tagBITMAPINFOHEADER::biPlanes](#)

Definition at line 331 of file dtsystemtypes.h.

[DWORD tagBITMAPINFOHEADER::biSize](#)

Definition at line 328 of file dtsystemtypes.h.

[DWORD tagBITMAPINFOHEADER::biSizeImage](#)

Definition at line 334 of file dtsystemtypes.h.

[LONG tagBITMAPINFOHEADER::biWidth](#)

Definition at line 329 of file dtsystemtypes.h.

[LONG tagBITMAPINFOHEADER::biXPelsPerMeter](#)

Definition at line 335 of file dtsystemtypes.h.

[LONG tagBITMAPINFOHEADER::biYPelsPerMeter](#)

Definition at line 336 of file dtsystemtypes.h.

The documentation for this struct was generated from the following file:

182 E:/drastic/api/mediacmd/src/[dtsystemtypes.h](#)

tagRGBQUAD Struct Reference

```
#include <dtsystemtypes.h>
```

Public Attributes

183 [BYTE rgbBlue](#)
184 [BYTE rgbGreen](#)
185 [BYTE rgbRed](#)
186 [BYTE rgbReserved](#)

Detailed Description

Definition at line 281 of file dtsystemtypes.h.

Member Data Documentation

[BYTE tagRGBQUAD::rgbBlue](#)

Definition at line 282 of file dtssystemtypes.h.

[BYTE tagRGBQUAD::rgbGreen](#)

Definition at line 283 of file dtssystemtypes.h.

[BYTE tagRGBQUAD::rgbRed](#)

Definition at line 284 of file dtssystemtypes.h.

[BYTE tagRGBQUAD::rgbReserved](#)

Definition at line 285 of file dtssystemtypes.h.

The documentation for this struct was generated from the following file:

187 E:/drastic/api/mediacmd/src/[dtssystemtypes.h](#)

tagVWX_CHANNEL Struct Reference

Public Attributes

188 long [lRefs](#)

Number of references to this channel.

189 void * [hvvw](#)

Regular vvw channel, port, etc.

190 char [szAddress](#) [_MAX_PATH]

String (host name for network, or nothing)

191 unsigned long [dwPort](#)

Port for network.

192 unsigned long [dwChannel](#)

Channel on target machine for network.

193 unsigned long [dwFlags](#)

Is the channel open and in use.

194 unsigned long [dwMsPerFrame](#)

Milliseconds per frame.

195 [MEDIACMD mCmdState](#)

Last known state.

196 [MEDIACMD mCmdEDL](#)

Last known state.

Detailed Description

Definition at line 124 of file vvwif.cpp.

Member Data Documentation

unsigned long [tagVWX_CHANNEL::dwChannel](#)

Channel on target machine for network.

Definition at line 134 of file vvwif.cpp.

unsigned long [tagVWX_CHANNEL::dwFlags](#)

Is the channel open and in use.

Definition at line 136 of file vvwif.cpp.

unsigned long [tagVWX_CHANNEL::dwMsPerFrame](#)

Milliseconds per frame.

Definition at line 141 of file vvwif.cpp.

unsigned long [tagVWX_CHANNEL::dwPort](#)

Port for network.

Definition at line 132 of file vvwif.cpp.

void* [tagVWX_CHANNEL::hvwx](#)

Regular vwx channel, port, etc.

Definition at line 128 of file vvwif.cpp.

long [tagVWX_CHANNEL::lRefs](#)

Number of references to this channel.

Definition at line 126 of file vvwif.cpp.

MEDIACMD [tagVWX_CHANNEL::mCmdEDL](#)

Last known state.

Definition at line 145 of file vvwif.cpp.

MEDIACMD [tagVWX_CHANNEL::mCmdState](#)

Last known state.

Definition at line 143 of file vvwif.cpp.

char [tagVWIX_CHANNEL::szAddress\[_MAX_PATH\]](#)

String (host name for network, or nothing)

Definition at line 130 of file vvwif.cpp.

The documentation for this struct was generated from the following file:

197 E:/drastic/api/mediacmd/src/[vwif.cpp](#)

tWAVEFORMATEX Struct Reference

```
#include <dtsystemtypes.h>
```

Public Attributes

198 [WORD wFormatTag](#)
199 [WORD nChannels](#)
200 [DWORD nSamplesPerSec](#)
201 [DWORD nAvgBytesPerSec](#)
202 [WORD nBlockAlign](#)
203 [WORD wBitsPerSample](#)
204 [WORD cbSize](#)

Detailed Description

Definition at line 302 of file dtsystemtypes.h.

Member Data Documentation

[WORD tWAVEFORMATEX::cbSize](#)

Definition at line 310 of file dtsystemtypes.h.

[DWORD tWAVEFORMATEX::nAvgBytesPerSec](#)

Definition at line 307 of file dtsystemtypes.h.

[WORD tWAVEFORMATEX::nBlockAlign](#)

Definition at line 308 of file dtsystemtypes.h.

[WORD tWAVEFORMATEX::nChannels](#)

Definition at line 305 of file dtssystemtypes.h.

[DWORD tWAVEFORMATEX::nSamplesPerSec](#)

Definition at line 306 of file dtssystemtypes.h.

[WORD tWAVEFORMATEX::wBitsPerSample](#)

Definition at line 309 of file dtssystemtypes.h.

[WORD tWAVEFORMATEX::wFormatTag](#)

Definition at line 304 of file dtssystemtypes.h.

The documentation for this struct was generated from the following file:

205 E:/drastic/api/mediacmd/src/[dtssystemtypes.h](#)

VVWAUDIO Struct Reference

```
#include <vvwtypes.h>
```

Public Attributes

206 [WORD wFormatTag](#)

Format of audio data, uncompressed == [DTWAVE_FORMAT_PCM](#) (windows WAVE_FORMAT_PCM) == 1.

207 [WORD nChannels](#)

Number of channels in this stream. Usually 1 or 2.

208 [DWORD nSamplesPerSec](#)

Number of samples per second. eg. 44100, 22050, 48000 etc.

209 [DWORD nAvgBytesPerSec](#)

*Average unsigned chars Per Second (see also [VVWAUDIO::dwRate](#)) == [nBlockAlign](#) * [nSamplesPerSec](#).*

210 [WORD nBlockAlign](#)

*Size of a sample group. == to [nChannels](#) * ([wBitsPerSample](#) / 8) eg. Stereo 16bit = 4, Mono 16bit = 2, Stereo 8Bit = 2.*

211 [WORD wBitsPerSample](#)

Number of bits per sample, normally 8 or 16.

212 [WORD cbSize](#)

Size of the dwReserved area used. For PCM this will be zero. For other compressors, it may be anything < 256 DWORDs.

213 [DWORD dwReserved](#) [[_VWXXX_RESERVED_SIZE](#)]
Extra info storage for audio codecs.

214 [DWORD fccType](#)
For [VWVAUDIO](#) structure this is always `streamtypeAUDIO == 'auds'`.

215 [DWORD fccHandler](#)
Codec type, see `fccDef.h` Normally the same as [VWVAUDIO::wFormatTag](#) but not always.

216 [DWORD dwFlags](#)

217 [DWORD dwCaps](#)
Not sure. See [VWVSYSTEM::dwCaps](#) for possible interp if something is set. MS Doc: currently unused.

218 [WORD wPriority](#)
Priority of stream (<-MSDoc in relation to other streams in the file I suppose)

219 [WORD wLanguage](#)
Language of stream (<-MSDoc but no language id defines)

220 [DWORD dwScale](#)
For PCM == to [VWVAUDIO::nBlockAlign](#). See [VWVSYSTEM::dwScale](#) for more info and table example.

221 [DWORD dwRate](#)
For PCM == to [VWVAUDIO::nAvgBytesPerSec](#). See [VWVSYSTEM::dwRate](#) for more info and table example.

222 [DWORD dwStart](#)

223 [DWORD dwLength](#)

224 [DWORD dwInitialFrames](#)

225 [DWORD dwSuggestedBufferSize](#)

226 [DWORD dwQuality](#)

227 [DWORD dwSampleSize](#)

228 [RECT rcFrame](#)

229 [DWORD dwEditCount](#)
Number of times the stream has been edited. The stream handler maintains this count.

230 [DWORD dwFormatChangeCount](#)
Number of times the stream format has changed. The stream handler maintains this count.

231 char [szName](#) [[_VWXXX_NAME_SIZE](#)]
Null-terminated string containing a description of the stream.

232 [LONG biUnused](#)
No used currently.

233 [DWORD dwDrFlags](#)

234 [DWORD dwFileType](#)
Source File Type.

235 [DWORD dwResDrastic](#)
Reserved, init to zero and leave alone.

Detailed Description

The audio structure is a combination of a WAVEFORMATEX and a AVISTREAMINFO structure. The top half should be treated as a WAVEFORMATEX and the bottom as a related but independant structure. These are as they appear in an AVI file and are manipulated to fill other file types like OMF and MOV.

Definition at line 1293 of file vvwtypes.h.

Member Data Documentation

[LONG VVWAUDIO::biUnused](#)

No used currently.

Definition at line 1390 of file vvwtypes.h.

[WORD VVWAUDIO::cbSize](#)

Size of the dwReserved area used. For PCM this will be zero. For other compressors, it may be anything < 256 DWORDs.

Definition at line 1311 of file vvwtypes.h.

[DWORD VVWAUDIO::dwCaps](#)

Not sure. See [VVWSYSTEM::dwCaps](#) for possible interp if something is set. MS Doc: currently unused.

Definition at line 1331 of file vvwtypes.h.

[DWORD VVWAUDIO::dwDrFlags](#)

Our internal flags including: [DRFLAGS_HAS_KEYFRAMES](#),
[DRFLAGS_FCC_USE_INTERN](#), [DRFLAGS_FCC_USE_OT](#),
[DRFLAGS_FCC_USE_ICM](#), [DRFLAGS_CODECPRIVATEDATA_AVI](#),
[DRFLAGS_CODECPRIVATEDATA_MOV](#), [DRFLAGS_CODECPRIVATEDATA_OME](#),
[DTVVW_PREVIEW](#)

Definition at line 1398 of file vvwtypes.h.

[DWORD VVWAUDIO::dwEditCount](#)

Number of times the stream has been edited. The stream handler maintains this count.

Definition at line 1382 of file vvwtypes.h.

[DWORD VVWAUDIO::dwFileType](#)

Source File Type.

Definition at line 1400 of file vvwtypes.h.

[DWORD VVWAUDIO::dwFlags](#)

Definition at line 1329 of file vvwtypes.h.

DWORD VVWAUDIO::dwFormatChangeCount

Number of times the stream format has changed. The stream handler maintains this count.

Definition at line 1384 of file vvwtypes.h.

DWORD VVWAUDIO::dwInitialFrames

Amount of audio in the file before video commences. For offset files, typically 0.75 sec converted to units per [VVWAUDIO::dwRate](#)/[VVWAUDIO::dwScale](#). For high end files, always zero as audio and video are sent without scew (except premiere, which uses 'rec ' chunks and audio skew)

Definition at line 1355 of file vvwtypes.h.

DWORD VVWAUDIO::dwLength

Length of the video stream in units per [VVWAUDIO::dwRate](#)/[VVWAUDIO::dwScale](#) (for video - frames)

Definition at line 1349 of file vvwtypes.h.

DWORD VVWAUDIO::dwQuality

Quality used by the compressor. Between 0 and 10,000 or -1 if default quality. For some compressors, the -1 can also mean the quality info is encoded into the frame or in the [dwReserved](#) or other private data area.

Definition at line 1366 of file vvwtypes.h.

DWORD VVWAUDIO::dwRate

For PCM == to [VVWAUDIO::nAvgBytesPerSec](#). See [VVSYSYSTEM::dwRate](#) for more info and table example.

Definition at line 1339 of file vvwtypes.h.

DWORD VVWAUDIO::dwResDrastic

Reserved, init to zero and leave alone.

Definition at line 1402 of file vvwtypes.h.

DWORD VVWAUDIO::dwReserved[_VVWXXX_RESERVED_SIZE]

Extra info storage for audio codecs.

Definition at line 1314 of file vvwtypes.h.

DWORD VVWAUDIO::dwSampleSize

Size, in unsigned chars, of a single data sample. If the value of this member is zero, the samples can vary in size and each data sample (such as a video frame) must be in a separate chunk. A nonzero value indicates that multiple samples of data can be grouped into a single chunk within the file. For video streams, this number is typically zero, although it can be

nonzero if all video frames are the same size. For audio streams, this number should be the same as the `nBlockAlign` member of the `WAVEFORMAT` or `WAVEFORMATEX` structure describing the audio.

Definition at line 1375 of file `vvwtypes.h`.

DWORD VVWAUDIO::dwScale

For PCM == to [VVWAUDIO::nBlockAlign](#). See `VVSYSTEM::dwScale` for more info and table example.

Definition at line 1337 of file `vvwtypes.h`.

DWORD VVWAUDIO::dwStart

Delay in units per [VVWAUDIO::dwRate](#)/`VVWAUDIO::dwScale` (for video - frames) for this stream to start in the playback of the file. NOTE AVI v1.0 and simple avi readers will choke or play incorrectly if this is not 0, so be careful.

Definition at line 1345 of file `vvwtypes.h`.

DWORD VVWAUDIO::dwSuggestedBufferSize

Recommended buffer size based on the largest single chunk in the file. Set by writer, so often incorrect or 0.

Definition at line 1360 of file `vvwtypes.h`.

DWORD VVWAUDIO::fccHandler

Codec type, see `fccDef.h` Normally the same as [VVWAUDIO::wFormatTag](#) but not always.

Definition at line 1321 of file `vvwtypes.h`.

DWORD VVWAUDIO::fccType

For [VVWAUDIO](#) structure this is always `streamtypeAUDIO == 'auds'`.

Definition at line 1319 of file `vvwtypes.h`.

DWORD VVWAUDIO::nAvgBytesPerSec

Average unsigned chars Per Second (see also [VVWAUDIO::dwRate](#)) == [nBlockAlign](#) * [nSamplesPerSec](#).

Definition at line 1305 of file `vvwtypes.h`.

WORD VVWAUDIO::nBlockAlign

Size of a sample group. == to [nChannels](#) * ([wBitsPerSample](#) / 8) eg. Stereo 16bit = 4, Mono 16bit = 2, Stereo 8Bit = 2.

Definition at line 1307 of file `vvwtypes.h`.

WORD VVWAUDIO::nChannels

Number of channels in this stream. Usually 1 or 2.

Definition at line 1301 of file vvwtypes.h.

DWORD VVWAUDIO::nSamplesPerSec

Number of samples per second. eg. 44100, 22050, 48000 etc.

Definition at line 1303 of file vvwtypes.h.

RECT VVWAUDIO::rcFrame

NOT USED BY [VVWAUDIO](#). Dimensions of the video destination rectangle. The values represent the coordinates of upper left corner, the height, and the width of the rectangle.

Definition at line 1380 of file vvwtypes.h.

char VVWAUDIO::szName[_VWXXX_NAME_SIZE]

Null-terminated string containing a description of the stream.

Definition at line 1386 of file vvwtypes.h.

WORD VVWAUDIO::wBitsPerSample

Number of bits per sample, normally 8 or 16.

Definition at line 1309 of file vvwtypes.h.

WORD VVWAUDIO::wFormatTag

Format of audio data, uncompressed == [DTWAVE_FORMAT_PCM](#) (windows WAVE_FORMAT_PCM) == 1.

Definition at line 1299 of file vvwtypes.h.

WORD VVWAUDIO::wLanguage

Language of stream (<-MSDoc but no language id defines)

Definition at line 1335 of file vvwtypes.h.

WORD VVWAUDIO::wPriority

Priority of stream (<-MSDoc in relation to other streams in the file I suppose)

Definition at line 1333 of file vvwtypes.h.

The documentation for this struct was generated from the following file:

236 E:/drastic/api/mediacmd/src/[vwwtypes.h](#)

vwwInflmageInfo Struct Reference

```
#include <vwwtypes.h>
```

Public Attributes

237 [DWORD dwStructSize](#)

How much do we have, effective version.

238 [DWORD ardwSMPTE_UMID](#) [8]

SMPTE_UMID.

239 [DWORD dwCurveType](#)

Base curve type (may be overridden below, but simple version here.

240 float [fGamma](#)

Gamma value.

241 float [fR](#)

Bayer white balance offset.

242 float [fG1](#)

Bayer white balance offset.

243 float [fG2](#)

Bayer white balance offset.

244 float [fB](#)

Bayer white balance offset.

245 [DWORD dwWhiteBalance](#)

White balance.

246 [DWORD dwExposure](#)

247 [DWORD dwBlackLevel](#)

248 [DWORD dwWhiteLevel](#)

249 [DWORD dwHorizontalFlip](#)

Horizontal flip.

250 [DWORD dwVerticalFlip](#)

Vertical flip.

251 [DWORD dwStereoHorizontalFlip](#)

Stereo Horizontal flip.

252 [DWORD dwStereoVerticalFlip](#)

Stereo Vertical flip.

253 [DWORD dwExposureTime](#)

Exposure time in micro seconds (10⁻⁶ seconds)

254 [DWORD dwShuttleAngle](#)

*Shutter angle in degree * 1000. Shutter Angle = (ExposureTime * SensorFps)/1E9 * 360.0.*

255 [DWORD dwX](#)

Camera position.

256 [DWORD dwY](#)

Camera position.
257 [DWORD dwZ](#)
Camera position.
258 [DWORD dwPan](#)
Camera position.
259 [DWORD dwTilt](#)
Camera position.
260 [DWORD dwRoll](#)
Camera position.
261 bool [fCustomMatrix](#)
Do we have a custom matrix.
262 float [matrix](#) [3][3]
Colour matrix.
263 char [szLookFile](#) [32]

Detailed Description

Structure to pass image information from the file to the render. Always add at end.
Definition at line 1541 of file vvwtypes.h.

Member Data Documentation

[DWORD vvwInflmageInfo::ardwSMPTE_UMID\[8\]](#)

SMPTE_UMID.
Definition at line 1545 of file vvwtypes.h.

[DWORD vvwInflmageInfo::dwBlackLevel](#)

Definition at line 1563 of file vvwtypes.h.

[DWORD vvwInflmageInfo::dwCurveType](#)

Base curve type (may be overridden below, but simple version here).
Definition at line 1547 of file vvwtypes.h.

[DWORD vvwInflmageInfo::dwExposure](#)

Definition at line 1561 of file vvwtypes.h.

[DWORD vvwInflmageInfo::dwExposureTime](#)

Exposure time in micro seconds (10^{-6} seconds)

Definition at line 1575 of file vvwtypes.h.

DWORD vvwInflmageInfo::dwHorizontalFlip

Horizontal flip.

Definition at line 1567 of file vvwtypes.h.

DWORD vvwInflmageInfo::dwPan

Camera position.

Definition at line 1585 of file vvwtypes.h.

DWORD vvwInflmageInfo::dwRoll

Camera position.

Definition at line 1589 of file vvwtypes.h.

DWORD vvwInflmageInfo::dwShuttleAngle

Shutter angle in degree * 1000. Shutter Angle = (ExposureTime * SensorFps)/1E9 * 360.0.

Definition at line 1577 of file vvwtypes.h.

DWORD vvwInflmageInfo::dwStereoHorizontalFlip

Stereo Horizontal flip.

Definition at line 1571 of file vvwtypes.h.

DWORD vvwInflmageInfo::dwStereoVerticalFlip

Stereo Vertical flip.

Definition at line 1573 of file vvwtypes.h.

DWORD vvwInflmageInfo::dwStructSize

How much do we have, effective version.

Definition at line 1543 of file vvwtypes.h.

DWORD vvwInflmageInfo::dwTilt

Camera position.

Definition at line 1587 of file vvwtypes.h.

DWORD vvwInflmageInfo::dwVerticalFlip

Vertical flip.

Definition at line 1569 of file vvwtypes.h.

DWORD vvwInflmageInfo::dwWhiteBalance

White balance.

Definition at line 1559 of file vvwtypes.h.

DWORD vvwInflmageInfo::dwWhiteLevel

Definition at line 1565 of file vvwtypes.h.

DWORD vvwInflmageInfo::dwX

Camera position.

Definition at line 1579 of file vvwtypes.h.

DWORD vvwInflmageInfo::dwY

Camera position.

Definition at line 1581 of file vvwtypes.h.

DWORD vvwInflmageInfo::dwZ

Camera position.

Definition at line 1583 of file vvwtypes.h.

float vvwInflmageInfo::fB

Bayer white balance offset.

Definition at line 1557 of file vvwtypes.h.

bool vvwInflmageInfo::fCustomMatrix

Do we have a custom matrix.

Definition at line 1591 of file vvwtypes.h.

float vvwInflmageInfo::fG1

Bayer white balance offset.

Definition at line 1553 of file vvwtypes.h.

float [vwwInflmageInfo::fG2](#)

Bayer white balance offset.

Definition at line 1555 of file vvwtypes.h.

float [vwwInflmageInfo::fGamma](#)

Gamma value.

Definition at line 1549 of file vvwtypes.h.

float [vwwInflmageInfo::fR](#)

Bayer white balance offset.

Definition at line 1551 of file vvwtypes.h.

float [vwwInflmageInfo::matrix\[3\]\[3\]](#)

Colour matrix.

Definition at line 1593 of file vvwtypes.h.

char [vwwInflmageInfo::szLookFile\[32\]](#)

Definition at line 1595 of file vvwtypes.h.

The documentation for this struct was generated from the following file:

264 E:/drastic/api/mediacmd/src/[vwtypes.h](#)

VWINFO Struct Reference

```
#include <vwtypes.h>
```

Public Attributes

265 char * [szFileName](#)
266 char * [szNativeLocator](#)
267 char * [szUniversalName](#)
268 char * [szIP](#)
269 char * [szSourceLocator](#)
270 char * [szChannel](#)
271 char * [szChannelName](#)
272 char * [szChannelDescription](#)
273 char * [szTitle](#)

274 char * [szSubject](#)
275 char * [szCategory](#)
276 char * [szKeywords](#)
277 char * [szRatings](#)
278 char * [szComments](#)
279 char * [szDoNotUse](#)
280 char * [szOwner](#)
281 char * [szEditor](#)
282 char * [szSupplier](#)
283 char * [szSource](#)
284 char * [szProject](#)
285 char * [szStatus](#)
286 char * [szAuthor](#)
287 char * [szRevisionNumber](#)
288 char * [szProduced](#)
289 char * [szAlbum](#)
290 char * [szArtist](#)
291 char * [szComposer](#)
292 char * [szCopyright](#)
293 char * [szCreationData](#)
294 char * [szDescription](#)
295 char * [szDirector](#)
296 char * [szDisclaimer](#)
297 char * [szEncodedBy](#)
298 char * [szFullName](#)
299 char * [szGenre](#)
300 char * [szHostComputer](#)
301 char * [szInformation](#)
302 char * [szMake](#)
303 char * [szModel](#)
304 char * [szOriginalArtist](#)
305 char * [szOriginalFormat](#)
306 char * [szPerformers](#)
307 char * [szProducer](#)
308 char * [szProduct](#)
309 char * [szSoftware](#)
310 char * [szSpecialPlaybackRequirements](#)
311 char * [szTrack](#)
312 char * [szWarning](#)
313 char * [szURLLink](#)
314 char * [szEditData1](#)
315 char * [szEditData2](#)
316 char * [szEditData3](#)
317 char * [szEditData4](#)
318 char * [szEditData5](#)
319 char * [szEditData6](#)
320 char * [szEditData7](#)
321 char * [szEditData8](#)
322 char * [szEditData9](#)
323 char * [szVersionString](#)
324 char * [szManufacturer](#)
325 char * [szLanguage](#)
326 char * [szFormat](#)
327 char * [szInputDevice](#)
328 char * [szDeviceModelNum](#)
329 char * [szDeviceSerialNum](#)

330 char * [szReel](#)
331 char * [szShot](#)
332 char * [szTake](#)
333 char * [szSlateInfo](#)
334 char * [szFrameAttribute](#)
335 char * [szEpisode](#)
336 char * [szScene](#)
337 char * [szDailyRoll](#)
338 char * [szCamRoll](#)
339 char * [szSoundRoll](#)
340 char * [szLabRoll](#)
341 char * [szKeyNumberPrefix](#)
342 char * [szInkNumberPrefix](#)
343 char * [szPictureIcon](#)
344 char * [szProxyFile](#)
345 char * [szCustomMetadataBlockPointer](#)
346 char * [szImageInfo](#)
347 char * [szUMID](#)
348 char * [szNU_84](#)
349 char * [szNU_85](#)
350 char * [szNU_86](#)
351 char * [szNU_87](#)
352 char * [szNU_88](#)
353 char * [szNU_89](#)
354 char * [szNU_90](#)
355 char * [szNU_91](#)
356 char * [szNU_92](#)
357 char * [szNU_93](#)
358 char * [szNU_94](#)
359 char * [szNU_95](#)
360 char * [szNU_96](#)
361 char * [szNU_97](#)
362 char * [szNU_98](#)
363 char * [szNU_99](#)
364 [DWORD dwReservedMustBe0](#)
365 [DWORD dwTimeCode](#)
366 [DWORD dwUserBits](#)
367 [DWORD dwVITCTimeCode](#)
368 [DWORD dwVITCUserBits](#)
369 [DWORD dwVITCLine3](#)
370 [DWORD dwPosterFrame](#)
371 [DWORD dwAFrame](#)
372 [DWORD dwAspectRatio](#)
373 [DWORD dwOriginalRate](#)
374 [DWORD dwOriginalScale](#)
375 [DWORD dwConversions](#)
376 [DWORD dwVersionNumber](#)
377 [DWORD dwFileSize](#)
378 [DWORD dwFileDate](#)
379 [DWORD dwFileTime](#)
380 [DWORD dwSequenceNumber](#)
381 [DWORD dwTotalStreams](#)
382 [DWORD dwTotalLength](#)
383 [DWORD dwFilmManufacturerCode](#)
384 [DWORD dwFilmTypeCode](#)
385 [DWORD dwWhitePoint](#)

386 [DWORD dwBlackPoint](#)
387 [DWORD dwBlackGain](#)
388 [DWORD dwBreakPoint](#)
389 [DWORD dwGamma1000](#)
390 [DWORD dwTagNumber](#)
391 [DWORD dwFlags](#)
392 [DWORD dwTimeCodeType](#)
393 [DWORD dwLTCTimeCodeType](#)
394 [DWORD dwVITCTimeCodeType](#)
395 [DWORD dwProdDate](#)
396 [DWORD dwUniqueID](#)
397 [DWORD dwCustomMetadataBlockType](#)
398 [DWORD dwCustomMetadataBlockSize](#)
399 [DWORD dwNorthSouthEastWest](#)
400 [DWORD dwLatitude](#)
401 [DWORD dwLongitude](#)
402 [DWORD dwNU_38](#)
403 [DWORD dwNU_39](#)
404 [DWORD dwNU_40](#)
405 [DWORD dwNU_41](#)
406 [DWORD dwNU_42](#)
407 [DWORD dwNU_43](#)
408 [DWORD dwNU_44](#)
409 [DWORD dwNU_45](#)
410 [DWORD dwNU_46](#)
411 [DWORD dwNU_47](#)
412 [DWORD dwNU_48](#)
413 [DWORD dwNU_49](#)
414 [DWORD dwFileType](#)

Source File Type.

415 [DWORD dwResDrastic](#)

Set to 0 on allocate and leave alone.

Detailed Description

The info structure is used to hold the metadata for one clip It is a combination of a custom info identifiers from all the file types we currently support. It is always written as a side bar xml file, and also insert into/read from the actual file whenever possible. NOTE: There are other fields included in the xml that are not contained here. There are contained in [VVWSYSTEM](#), [VVWVIDEO](#) or [VVWAUDIO](#). Data Rules (needs updating)

416 --- **LOCATOR STRINGS**

417 File Name (base name+ext)

418 Native Locator

419 Universal Name Convention Locator

420 IP

421 Source Locator (Universal)

422 --- **Inferred**

423 Channel

424 Channel Length

425 Channel Name

426 Channel Description
427 Channel Poster Frame

428 --- **STRINGS**

429 Title W (AAF:Name) (TIF:DocumentName)
430 Subject W
431 Category W
432 Keywords W
433 Rating AVI
434 Comments W QT
435 Owner AVI
436 Editor AVI
437 Supplier AVI
438 Source W
439 Project AVI
440 Status
441 AVI Other, Final, Proof, Review, Edit, In Progress, Draft, Preliminary, New Normal
442 Author W QT
443 Revision Number W
444 Produced AVI
445 Album QT
446 Artist QT TIF
447 Composer QT
448 Copyright QT TIF
449 Creation Data QT
450 Description QT AAF (TIF:ImageDescription)
451 Director QT
452 Disclaimer QT
453 Encoded By QT
454 Full Name QT
455 Genre QT
456 Host Computer QT TIF
457 Information QT
458 Make QT TIF
459 Model QT TIF
460 Original Artist QT
461 Original Format QT
462 Original Source QT
463 Performers QT
464 Producer QT
465 Product QT
466 Software QT TIF
467 Special Playback Requirements QT
468 Track QT
469 Warning QT
470 URL Link QT
471 Edit Data 1..9 QT
472 Version String AAF
473 Manufacturer AAF
474 Language DublinCore.org
475 Format DublinCore.org
476

477 **Numeric Entries**

478 Timecode (LTC)

479 UserBits (LTC)
480 VITC Timecode
481 VITC Userbits
482 VITC Line 3
483 Poster Frame
484 'A' Frame
485 Aspect Ratio
486 OrginalScale
487 OrginalRate
488 Conversions
489 Version Number AAF
490 File Size (64 Bits)
491 File Date TIF
492 File Time TIF
493 Sequence Number AVI

Definition at line 2143 of file vvwtypes.h.

Member Data Documentation

[DWORD VVWINFO::dwAFrame](#)

A-Frame if 2/3 else 0

XML tag <A-Frame>

Definition at line 2988 of file vvwtypes.h.

[DWORD VVWINFO::dwAspectRatio](#)

Upperword/LowerWord (e.g. 0x00040003 = 4/3)

XML tag <AspectRatio>

Definition at line 2996 of file vvwtypes.h.

[DWORD VVWINFO::dwBlackGain](#)

Log capture Black gain

XML tag <BlackGain>

Definition at line 3118 of file vvwtypes.h.

[DWORD VVWINFO::dwBlackPoint](#)

Log captured Black point (also CCIR black for 8/10 YCbCr)

XML tag <BlackPoint>

Definition at line 3110 of file vvwtypes.h.

[DWORD VVWINFO::dwBreakPoint](#)

Log captured break point

XML tag <BreakPoint>

Definition at line 3126 of file vvwtypes.h.

DWORD VVWINFO::dwConversions

Number of conversion

XML tag <TotalConversions>

Definition at line 3020 of file vvwtypes.h.

DWORD VVWINFO::dwCustomMetadataBlockSize

Custom metadata block size

Definition at line 3201 of file vvwtypes.h.

DWORD VVWINFO::dwCustomMetadataBlockType

Custom metadata block type

Definition at line 3194 of file vvwtypes.h.

DWORD VVWINFO::dwFileDate

Date stamp of file

Date as YYYYMMDD (BCD: 1 digit -> 4 bits).

XML tag <FileDate>

Definition at line 3045 of file vvwtypes.h.

DWORD VVWINFO::dwFileSize

Size of file

XML tag <FileSize>

Definition at line 3036 of file vvwtypes.h.

DWORD VVWINFO::dwFileTime

Time of file

Time as HHMMSScc (BCD: 1 digit -> 4 bits) The last byte 'cc' are centiseconds.

XML tag <FileTime>

Definition at line 3054 of file vvwtypes.h.

DWORD VVWINFO::dwFileType

Source File Type.

Definition at line 3312 of file vvwtypes.h.

DWORD VVWINFO::dwFilmManufacturerCode

Film manufacturer's code

XML tag <FileManufacturerCode>

Definition at line 3086 of file vvwtypes.h.

DWORD VVWINFO::dwFilmTypeCode

Film type code

XML tag <FileTypeCode>
Definition at line 3094 of file vvwtypes.h.

DWORD VVWINFO::dwFlags

TBD
Definition at line 3148 of file vvwtypes.h.

DWORD VVWINFO::dwGamma1000

Gamma * 1000 (e.g. 1.7 == 17000)
XML tag <Gamma1000>
Definition at line 3134 of file vvwtypes.h.

DWORD VVWINFO::dwLatitude

Latitude value in decimal degrees * 10000000 (uses bit 1 for north/south)
Definition at line 3215 of file vvwtypes.h.

DWORD VVWINFO::dwLongitude

Longitude value in decimal degrees * 10000000
Definition at line 3222 of file vvwtypes.h.

DWORD VVWINFO::dwLTCTimeCodeType

LTC Time code type for the counter/ctl NDF/30, DF/29, PAL/24, FILM/24, NTSCFILM/23
Definition at line 3164 of file vvwtypes.h.

DWORD VVWINFO::dwNorthSouthEastWest

GPS coordinates - North=2/South=0, East=1/West=0
Definition at line 3208 of file vvwtypes.h.

DWORD VVWINFO::dwNU_38

TBD
Definition at line 3229 of file vvwtypes.h.

DWORD VVWINFO::dwNU_39

TBD
Definition at line 3236 of file vvwtypes.h.

DWORD VVWINFO::dwNU_40

TBD
Definition at line 3243 of file vvwtypes.h.

DWORD VVWINFO::dwNU_41

TBD
Definition at line 3250 of file vvwtypes.h.

[DWORD VVWINFO::dwNU_42](#)

TBD

Definition at line 3257 of file vvwtypes.h.

[DWORD VVWINFO::dwNU_43](#)

TBD

Definition at line 3264 of file vvwtypes.h.

[DWORD VVWINFO::dwNU_44](#)

TBD

Definition at line 3271 of file vvwtypes.h.

[DWORD VVWINFO::dwNU_45](#)

TBD

Definition at line 3278 of file vvwtypes.h.

[DWORD VVWINFO::dwNU_46](#)

TBD

Definition at line 3285 of file vvwtypes.h.

[DWORD VVWINFO::dwNU_47](#)

TBD

Definition at line 3292 of file vvwtypes.h.

[DWORD VVWINFO::dwNU_48](#)

TBD

Definition at line 3299 of file vvwtypes.h.

[DWORD VVWINFO::dwNU_49](#)

TBD

Definition at line 3306 of file vvwtypes.h.

[DWORD VVWINFO::dwOriginalRate](#)

Original file rate

XML tag <OriginalFileRate>

Definition at line 3004 of file vvwtypes.h.

[DWORD VVWINFO::dwOriginalScale](#)

Original file scale

XML tag <OriginalFileScale>

Definition at line 3012 of file vvwtypes.h.

DWORD VVWINFO::dwPosterFrame

Poster (picon) frame number
XML tag <PosterFrame>
Definition at line 2980 of file vvwtypes.h.

DWORD VVWINFO::dwProdDate

RP-215 ProdDate 4 bytes
Definition at line 3179 of file vvwtypes.h.

DWORD VVWINFO::dwResDrastic

Set to 0 on allocate and leave alone.
Definition at line 3315 of file vvwtypes.h.

DWORD VVWINFO::dwReservedMustBe0

Special reserved place holder for end of strings
Definition at line 2934 of file vvwtypes.h.

DWORD VVWINFO::dwSequenceNumber

Sequence number
XML tag <SequenceNumber>
Definition at line 3062 of file vvwtypes.h.

DWORD VVWINFO::dwTagNumber

TBD
Definition at line 3141 of file vvwtypes.h.

DWORD VVWINFO::dwTimeCode

Starting time code (LTC)
XML tag <TimeCode>
Definition at line 2940 of file vvwtypes.h.

DWORD VVWINFO::dwTimeCodeType

Time code type for the counter/ctl NDF/30, DF/29, PAL/24, FILM/24, NTSCFILM/23
Definition at line 3156 of file vvwtypes.h.

DWORD VVWINFO::dwTotalLength

Total frames of longest stream in file
XML tag <TotalLength>
Definition at line 3078 of file vvwtypes.h.

DWORD VVWINFO::dwTotalStreams

Total number of streams in file
XML tag <TotalStreams>
Definition at line 3070 of file vvwtypes.h.

DWORD VVWINFO::dwUniqueID

TBD
Definition at line 3187 of file vvwtypes.h.

DWORD VVWINFO::dwUserBits

Starting user bits (LTC)
XML tag <UserBits>
Definition at line 2948 of file vvwtypes.h.

DWORD VVWINFO::dwVersionNumber

Version number
XML tag <VersionNumber>
Definition at line 3028 of file vvwtypes.h.

DWORD VVWINFO::dwVITCLine3

Extra VITC data
XML tag <VITCExtraData>
Definition at line 2972 of file vvwtypes.h.

DWORD VVWINFO::dwVITCTimeCode

Starting VITC time code
XML tag <VITCTimeCode>
Definition at line 2956 of file vvwtypes.h.

DWORD VVWINFO::dwVITCTimeCodeType

VITC Time code type for the counter/ctl NDF/30, DF/29, PAL/24, FILM/24, NTSCFILM/23
Definition at line 3172 of file vvwtypes.h.

DWORD VVWINFO::dwVITCUserBits

Starting VITC user bits
XML tag <VITCUserBits>
Definition at line 2964 of file vvwtypes.h.

DWORD VVWINFO::dwWhitePoint

Log captured White point (also CCIR white for 8/10 YCbCr)
XML tag <WhitePoint>
Definition at line 3102 of file vvwtypes.h.

char* [VWINFO::szAlbum](#)

Album Name

XML tag <Album>

Definition at line 2353 of file vvwtypes.h.

char* [VWINFO::szArtist](#)

Artist Name

XML tag <Artist>

Definition at line 2361 of file vvwtypes.h.

char* [VWINFO::szAuthor](#)

REG Default Author's name

XML tag <Author>

Definition at line 2329 of file vvwtypes.h.

char* [VWINFO::szCamRoll](#)

RP-215 camera roll namd/id

Definition at line 2744 of file vvwtypes.h.

char* [VWINFO::szCategory](#)

Category of media (AVI/Mov Format)

XML tag <Category>

Definition at line 2235 of file vvwtypes.h.

char* [VWINFO::szChannel](#)

Channel identifier

XML tag <ChannelIdentifier>

Definition at line 2195 of file vvwtypes.h.

char* [VWINFO::szChannelDescription](#)

Description of channel

XML tag <ChannelDescription>

Definition at line 2211 of file vvwtypes.h.

char* [VWINFO::szChannelName](#)

Name of channel

XML tag <ChannelName>

Definition at line 2203 of file vvwtypes.h.

char* [VWINFO::szComments](#)

Free form comments

XML tag <Comments>

Definition at line 2259 of file vvwtypes.h.

char* [VWINFO::szComposer](#)

Composer's Name

XML tag <Composer>

Definition at line 2369 of file vvwtypes.h.

char* [VWINFO::szCopyright](#)

REG Default Copyright message

XML tag <Copyright>

Definition at line 2378 of file vvwtypes.h.

char* [VWINFO::szCreationData](#)

Creation Data

XML tag <CreationData>

Definition at line 2386 of file vvwtypes.h.

char* [VWINFO::szCustomMetadataBlockPointer](#)

CustomMetadataBlockPointer

Definition at line 2793 of file vvwtypes.h.

char* [VWINFO::szDailyRoll](#)

RP-215 daily roll namd/id

Definition at line 2737 of file vvwtypes.h.

char* [VWINFO::szDescription](#)

Description of contents

XML tag <Description>

Definition at line 2394 of file vvwtypes.h.

char* [VWINFO::szDeviceModelNum](#)

Input device Model (32 char)

XML tag <DeviceModelNum>

Definition at line 2668 of file vvwtypes.h.

char* [VWINFO::szDeviceSerialNum](#)

Input device serial number (32 char)

XML tag <DeviceSerialNum>

Definition at line 2676 of file vvwtypes.h.

char* [VWINFO::szDirector](#)

Director's Name

XML tag <Director>
Definition at line 2402 of file vvwtypes.h.

char* [VWINFO::szDisclaimer](#)

Disclaimer
XML tag <Disclaimer>
Definition at line 2410 of file vvwtypes.h.

char* [VWINFO::szDoNotUse](#)

??? - This was (sz)SequenceNumber which was in conflict with (dw)SequenceNumber below. Leave empty for compatibility
XML tag <DoNotUse>
Definition at line 2268 of file vvwtypes.h.

char* [VWINFO::szEditData1](#)

User Data
XML tag <EditData1>
Definition at line 2556 of file vvwtypes.h.

char* [VWINFO::szEditData2](#)

User Data
XML tag <EditData2>
Definition at line 2564 of file vvwtypes.h.

char* [VWINFO::szEditData3](#)

User Data
XML tag <EditData3>
Definition at line 2572 of file vvwtypes.h.

char* [VWINFO::szEditData4](#)

User Data
XML tag <EditData4>
Definition at line 2580 of file vvwtypes.h.

char* [VWINFO::szEditData5](#)

User Data
XML tag <EditData5>
Definition at line 2588 of file vvwtypes.h.

char* [VWINFO::szEditData6](#)

User Data
XML tag <EditData6>

Definition at line 2596 of file vvwtypes.h.

char* [VWINFO::szEditData7](#)

User Data

XML tag <EditData7>

Definition at line 2604 of file vvwtypes.h.

char* [VWINFO::szEditData8](#)

User Data

XML tag <EditData8>

Definition at line 2612 of file vvwtypes.h.

char* [VWINFO::szEditData9](#)

User Data

XML tag <EditData9>

Definition at line 2620 of file vvwtypes.h.

char* [VWINFO::szEditor](#)

REG Default Editor of file

XML tag <Editor>

Definition at line 2286 of file vvwtypes.h.

char* [VWINFO::szEncodedBy](#)

REG Default Encoders Name

XML tag <Encoded By>

Definition at line 2419 of file vvwtypes.h.

char* [VWINFO::szEpisode](#)

RP-215 episode name

Definition at line 2723 of file vvwtypes.h.

char* [VWINFO::szFileName](#)

File Name and Extension (no path info) Created

XML tag <FileName>

Definition at line 2149 of file vvwtypes.h.

char* [VWINFO::szFormat](#)

File format

XML tag <Format>

Definition at line 2652 of file vvwtypes.h.

char* [VWINFO::szFrameAttribute](#)
Default frame attribute string for Cin/DPix (32 char)
XML tag <FrameAttribute>
Definition at line 2716 of file vvwtypes.h.

char* [VWINFO::szFullName](#)
Full name of media
XML tag <FullName>
Definition at line 2427 of file vvwtypes.h.

char* [VWINFO::szGenre](#)
Genre (type) of content
XML tag <Genre>
Definition at line 2435 of file vvwtypes.h.

char* [VWINFO::szHostComputer](#)
Computer hosting content
XML tag <HostComputer>
Definition at line 2443 of file vvwtypes.h.

char* [VWINFO::szImageInfo](#)
Pointer to ImageInfo structure
Definition at line 2800 of file vvwtypes.h.

char* [VWINFO::szInformation](#)
Free form information
XML tag <Information>
Definition at line 2451 of file vvwtypes.h.

char* [VWINFO::szInkNumberPrefix](#)
Prefix (0 feet) of the ink number from the film
Definition at line 2772 of file vvwtypes.h.

char* [VWINFO::szInputDevice](#)
Input device (Cin/DPix) telecine name (64 char)
XML tag <InputDevice>
Definition at line 2660 of file vvwtypes.h.

char* [VWINFO::szIP](#)
IP Address (192.168.0.3) Created
XML tag <TCP-IPAddress>
Definition at line 2177 of file vvwtypes.h.

char* [VWINFO::szKeyNumberPrefix](#)

Prefix (0 feet) of the key number from the film
Definition at line 2765 of file vvwtypes.h.

char* [VWINFO::szKeywords](#)

Search keywords
XML tag <Keywords>
Definition at line 2243 of file vvwtypes.h.

char* [VWINFO::szLabRoll](#)

RP-215 lab roll namd/id
Definition at line 2758 of file vvwtypes.h.

char* [VWINFO::szLanguage](#)

Language
XML tag <Language>
Definition at line 2644 of file vvwtypes.h.

char* [VWINFO::szMake](#)

Make - not used
XML tag <Make>
Definition at line 2459 of file vvwtypes.h.

char* [VWINFO::szManufacturer](#)

Manufacturer (DEF: Drastic Technologies Ltd)
XML tag <Manufacturer>
Definition at line 2636 of file vvwtypes.h.

char* [VWINFO::szModel](#)

Not Used
XML tag <Model>
Definition at line 2467 of file vvwtypes.h.

char* [VWINFO::szNativeLocator](#)

Native path + file + ext (C:/Test/Junk.avi) Created
XML tag <NativeLocator>
Definition at line 2159 of file vvwtypes.h.

char* [VWINFO::szNU_84](#)

TBD
Definition at line 2821 of file vvwtypes.h.

char* [VWINFO::szNU_85](#)

TBD

Definition at line 2828 of file vwtypes.h.

char* [VWINFO::szNU_86](#)

TBD

Definition at line 2835 of file vwtypes.h.

char* [VWINFO::szNU_87](#)

TBD

Definition at line 2842 of file vwtypes.h.

char* [VWINFO::szNU_88](#)

TBD

Definition at line 2849 of file vwtypes.h.

char* [VWINFO::szNU_89](#)

TBD

Definition at line 2856 of file vwtypes.h.

char* [VWINFO::szNU_90](#)

TBD

Definition at line 2863 of file vwtypes.h.

char* [VWINFO::szNU_91](#)

TBD

Definition at line 2870 of file vwtypes.h.

char* [VWINFO::szNU_92](#)

TBD

Definition at line 2877 of file vwtypes.h.

char* [VWINFO::szNU_93](#)

TBD

Definition at line 2884 of file vwtypes.h.

char* [VWINFO::szNU_94](#)

TBD

Definition at line 2891 of file vwtypes.h.

char* [VWINFO::szNU_95](#)

TBD

Definition at line 2898 of file vwtypes.h.

char* [VWINFO::szNU_96](#)

TBD

Definition at line 2905 of file vwtypes.h.

char* [VWINFO::szNU_97](#)

TBD

Definition at line 2912 of file vwtypes.h.

char* [VWINFO::szNU_98](#)

TBD

Definition at line 2919 of file vwtypes.h.

char* [VWINFO::szNU_99](#)

TBD

Definition at line 2926 of file vwtypes.h.

char* [VWINFO::szOriginalArtist](#)

Original artist's name

XML tag <OriginalArtist>

Definition at line 2475 of file vwtypes.h.

char* [VWINFO::szOriginalFormat](#)

Original format of content

XML tag <OriginalFormat>

Definition at line 2483 of file vwtypes.h.

char* [VWINFO::szOwner](#)

REG Default Owner of content

XML tag <Owner>

Definition at line 2277 of file vwtypes.h.

char* [VWINFO::szPerformers](#)

Performers in content

XML tag <Performers>

Definition at line 2491 of file vwtypes.h.

char* [VWINFO::szPictureIcon](#)

Small jpg frame capture

Definition at line 2779 of file vwtypes.h.

char* [VWINFO::szProduced](#)

Producer company name

XML tag <Produced>
Definition at line 2345 of file vvwtypes.h.

char* [VWINFO::szProducer](#)
Producer's Name
XML tag <Producer>
Definition at line 2499 of file vvwtypes.h.

char* [VWINFO::szProduct](#)
Targeted associated product
XML tag <Product>
Definition at line 2507 of file vvwtypes.h.

char* [VWINFO::szProject](#)
REG Default Project Name
XML tag <Project>
Definition at line 2312 of file vvwtypes.h.

char* [VWINFO::szProxyFile](#)
Low res proxy file
Definition at line 2786 of file vvwtypes.h.

char* [VWINFO::szRatings](#)
Parental Ratings
XML tag <Ratings>
Definition at line 2251 of file vvwtypes.h.

char* [VWINFO::szReel](#)
Reel this was recorded to
XML tag <Reel>
Definition at line 2684 of file vvwtypes.h.

char* [VWINFO::szRevisionNumber](#)
Revision number as string
XML tag <RevisionNumber>
Definition at line 2337 of file vvwtypes.h.

char* [VWINFO::szScene](#)
RP-215 scene
Definition at line 2730 of file vvwtypes.h.

char* [VWINFO::szShot](#)

Shot name (or number in string)

XML tag <Shot>

Definition at line 2692 of file vvwtypes.h.

char* [VWINFO::szSlateInfo](#)

Info from the original slate (200 char)

XML tag <SlateInfo>

Definition at line 2708 of file vvwtypes.h.

char* [VWINFO::szSoftware](#)

Software in use (DEF: "MediaReactor")

XML tag <Software>

Definition at line 2515 of file vvwtypes.h.

char* [VWINFO::szSoundRoll](#)

RP-215 sound roll namd/id

Definition at line 2751 of file vvwtypes.h.

char* [VWINFO::szSource](#)

Source of file (VTR, Betacam, MPEG, Satellite)

XML tag <Source>

Definition at line 2303 of file vvwtypes.h.

char* [VWINFO::szSourceLocator](#)

Source locator (either szUnversalName, szNativeLocator, szFileName available in this order)
Created

XML tag <SourceLocator>

Definition at line 2186 of file vvwtypes.h.

char* [VWINFO::szSpecialPlaybackRequirements](#)

Special playback requirements

XML tag <SpecialPlaybackRequirements>

Definition at line 2523 of file vvwtypes.h.

char* [VWINFO::szStatus](#)

Status

XML tag <Status>

Definition at line 2320 of file vvwtypes.h.

char* [VWINFO::szSubject](#)

Subject

XML tag <Subject>
Definition at line 2227 of file vvwtypes.h.

char* [VWINFO::szSupplier](#)

REG Default Supplier
XML tag <Supplier>
Definition at line 2295 of file vvwtypes.h.

char* [VWINFO::szTake](#)

Take name (or number in string)
XML tag <Take>
Definition at line 2700 of file vvwtypes.h.

char* [VWINFO::szTitle](#)

Title
XML tag <Title>
Definition at line 2219 of file vvwtypes.h.

char* [VWINFO::szTrack](#)

Track Name
XML tag <Track>
Definition at line 2531 of file vvwtypes.h.

char* [VWINFO::szUMID](#)

The SMPTE 330M Unique Material Identifier (UMID) is a standard for providing a stand-alone method for generating a unique label designed to be used to attach to media files and streams.

There are two types of UMIDs:

The Basic UMID contains the minimal components necessary for the unique identification (the essential metadata) The length of the basic UMID is 32 octets.

The Extended UMID provides information on the creation time and date, recording location and the name of the organisation and the maker as well as the components of the Basic UMID. The length of the Extended UMID is 64 octets. This data may be parsed to extract specific information produced at the time it was generated or simply used as a unique label.

Definition at line 2814 of file vvwtypes.h.

char* [VWINFO::szUniversalName](#)

UNC (/TestBox/C/Test/Junk.avi) Created
XML tag <UniversalFileName>
Definition at line 2168 of file vvwtypes.h.

char* [VWINFO::szURLLink](#)

REG Default Link to site (<http://www.drastictech.com>)

XML tag <URL>

Definition at line 2548 of file vvwtypes.h.

char* [VWINFO::szVersionString](#)

Version string

XML tag <VersionString>

Definition at line 2628 of file vvwtypes.h.

char* [VWINFO::szWarning](#)

Warning info

XML tag <Warning>

Definition at line 2539 of file vvwtypes.h.

The documentation for this struct was generated from the following file:

494 E:/drastic/api/mediacmd/src/[vwtypes.h](#)

VWNETDIRECT_CHANNEL Struct Reference

Public Attributes

495 char [szAddress](#) [256]

496 unsigned long [dwPort](#)

497 unsigned long [dwChannel](#)

498 long [lRefs](#)

499 long [lSocketOk](#)

500 CRITICAL_SECTION [crit](#)

501 int [sockfd](#)

502 struct sockaddr_in [my_addr](#)

Detailed Description

Definition at line 54 of file dtnetdirect.cpp.

Member Data Documentation

CRITICAL_SECTION [VWNETDIRECT_CHANNEL::crit](#)

Definition at line 61 of file dtnetdirect.cpp.

unsigned long [VWNETDIRECT_CHANNEL::dwChannel](#)

Definition at line 57 of file dtnetdirect.cpp.

unsigned long [VWNETDIRECT_CHANNEL::dwPort](#)

Definition at line 56 of file dtnetdirect.cpp.

long [VWNETDIRECT_CHANNEL::IRefs](#)

Definition at line 58 of file dtnetdirect.cpp.

long [VWNETDIRECT_CHANNEL::ISocketOk](#)

Definition at line 59 of file dtnetdirect.cpp.

struct sockaddr_in [VWNETDIRECT_CHANNEL::my_addr](#)

Definition at line 64 of file dtnetdirect.cpp.

int [VWNETDIRECT_CHANNEL::sockfd](#)

Definition at line 63 of file dtnetdirect.cpp.

char [VWNETDIRECT_CHANNEL::szAddress\[256\]](#)

Definition at line 55 of file dtnetdirect.cpp.

The documentation for this struct was generated from the following file:

503 E:/drastic/api/mediacmd/src/[dtnetdirect.cpp](#)

VVWSYSTEM Struct Reference

```
#include <vvwtypes.h>
```

Public Attributes

504 [DWORD dwMicroSecPerFrame](#)
505 [DWORD dwMaxBytesPerSec](#)
506 [DWORD dwPaddingGranularity](#)
507 [DWORD dwFlags](#)
508 [DWORD dwTotalFrames](#)
509 [DWORD dwInitialFrames](#)
510 [DWORD dwStreams](#)
511 [DWORD dwSuggestedBufferSize](#)
512 [DWORD dwWidth](#)
513 [DWORD dwHeight](#)
514 [DWORD dwReserved](#) [4]
515 [DWORD dwCaps](#)

516 [DWORD dwScale](#)
517 [DWORD dwRate](#)
518 [DWORD dwLength](#)
519 [DWORD dwEditCount](#)
520 char [szFileType](#) [[_VWXXX_NAME_SIZE](#)]
521 [DWORD dwType](#)
522 [DWORD dwMfCaps](#)
523 [DWORD dwVidStandard](#)
524 [DWORD dwDrFlags](#)
525 [DWORD dwFileType](#)
Source File Type.
526 [DWORD dwResDrastic](#)

Detailed Description

The [VWSYSTEM](#) structure holds the overview of a media stream including basic time (rate/scale), number of tracks, buffering info, msperframe, width/height of one or more video streams, maximum length in frames and (rate/scale). It is binary compatible with the AVI system header, but translated as nec to OMF MOV or any other supported format. Testing has not been thorough enough to entirely trust this structure, but it is a good place to start.

Definition at line 681 of file [vwtypes.h](#).

Member Data Documentation

[DWORD VWSYSTEM::dwCaps](#)

Capability flags from video for windows including:

AVIFILECAPS_CANREAD	0x00000001	- File may be read
AVIFILECAPS_CANWRITE	0x00000002	- File may be written to
AVIFILECAPS_ALLKEYFRAMES	0x00000010	- File contains frames that do not require interframe data to decode (eg. JPEG)
AVIFILECAPS_NOCOMPRESSION	0x00000020	- The frames in the file are uncompressed

Do not trust these flags. Look at the [VWVIDEO](#) and [VWVAUDIO](#) areas to confirm types of streams.

Definition at line 776 of file [vwtypes.h](#).

[DWORD VWSYSTEM::dwDrFlags](#)

Our internal flags including:

DRFLAGS_ZERO_FIELD_DOMINANT ,	DRFLAGS_ZERO_FIELD_DOMINANT ,
DRFLAGS_FIRST_FIELD_DOMINANT ,	DRFLAGS_HAS_KEYFRAMES ,
DRFLAGS_FCC_MJPG_DIGISUITE ,	DRFLAGS_FCC_MJPG_DCx0 ,
DRFLAGS_FCC_MJPG_DSEDIT ,	DRFLAGS_FCC_MJPG_JPGDIB ,
DRFLAGS_FCC_MJPG_JFIF ,	DRFLAGS_FCC_USE_INTERN ,
DRFLAGS_FCC_USE_OT ,	DRFLAGS_FCC_USE_ICM ,
DRFLAGS_CODECPRIVATEDATA_AVI ,	DRFLAGS_CODECPRIVATEDATA_MOV ,
DRFLAGS_CODECPRIVATEDATA_OMF ,	DTVWVW_PREVIEW

Definition at line 855 of file [vwtypes.h](#).

DWORD VVWSYSTEM::dwEditCount

The number of times this file has been edited. Intended for tracing QuickTime style edits within an AVI file, but good editing capabilities were never added to video for windows, so this is always 0, 1 or a random number with little or no meaning.

Definition at line 828 of file vvwtypes.h.

DWORD VVWSYSTEM::dwFileType

Source File Type.

Definition at line 857 of file vvwtypes.h.

DWORD VVWSYSTEM::dwFlags

AVIFILEINFO_ flags such as

AVIFILEINFO_HASINDEX included in the file	0x00000010 - there is an index to the frames
AVIFILEINFO_MUSTUSEINDEX frames are out of order on disk	0x00000020 - index is required, normally means
AVIFILEINFO_ISINTERLEAVED same file (usually audio and video)	0x00000100 - includes multiple channels in
AVIFILEINFO_WASCAPTUREFILE optimally written in the heat of the moment (eg avicap)	0x00010000 - from live source, may not be
AVIFILEINFO_COPYRIGHTED	0x00020000 - early microsoft attempt at screwing us. not used to my knowledge

Definition at line 719 of file vvwtypes.h.

DWORD VVWSYSTEM::dwHeight

Basic or largest height of a video stream within this file. Caution, for some media streams this value can be negative indicating the stream is vertically inverted. If a positive value is required, abs this before using. In theory, this value should always be positive, but it isn't.

Definition at line 758 of file vvwtypes.h.

DWORD VVWSYSTEM::dwInitialFrames

The initial number of frames that should be loaded or, esp if using 'rec ' index chunks, the number of pre-buffer audio frames before video appears (usually 10~25). We usually ignore this value as it will only help with extremely slow drives. It doesn't even appear to help with 10bT network connections significantly.

Definition at line 734 of file vvwtypes.h.

DWORD VVWSYSTEM::dwLength

Length of the longest stream in samples. This could be audio or video samples, but will always relate to Rate/Scale above. For streams with video, this is a frame count and should be the same as dwTotalFrames.

Definition at line 822 of file vvwtypes.h.

DWORD VVWSYSTEM::dwMaxBytesPerSec

The maximum unsigned chars per second this media will generate. Should assume all channels active and in use. Should be set by writer, but often isn't. Not used internally, so

regard with some suspicion.

Definition at line 698 of file vvwtypes.h.

DWORD VVWSYSTEM::dwMfCaps

MediaFile Capability flags. Not used currently.

Definition at line 841 of file vvwtypes.h.

DWORD VVWSYSTEM::dwMicroSecPerFrame

Number of microseconds for a frame duration. Normally refers to video and will be set to values like 33367 (NTSC) or 40000 (PAL) or possibly 0 if not specified. When the system structure is used in AvHAL or elsewhere internally, this value must always be valid as some timing calculations are based on it. As good practice, make sure it is correct for media files as well. Related to [VVWSYSTEM::dwRate](#) and [VVWSYSTEM::dwScale](#) in that $\text{Rate} = 1000000 / (\text{Scale} / \text{Rate})$

Definition at line 692 of file vvwtypes.h.

DWORD VVWSYSTEM::dwPaddingGranularity

The size of pad used to write the file. With bad writers, this is set to 0/512/1024/2048/4096 and then the file is written without any regard for its setting. With decent writers, it is set to the disk sector size the writer is writing to and it is respected and correct at least for video. With our writers, it is set to 4096 (largest general multiple of sector size and most common on NTFS systems) and we write to 4096. If the granularity is bad, we can usually still use sector aligned reads, by reading a little extra garbage and ignoring it, except for the last frame if we run out of file.

Definition at line 708 of file vvwtypes.h.

DWORD VVWSYSTEM::dwRate

Rate - the frame second length divided by [VVWSYSTEM::dwScale](#) to get the frame frame rate

Scale	Rate	FrameRate	MsPerFrame	
1	60	60		HD 60FPS (720p)
1001	60000	59.94		HD 59.94FPS (720p)
1	50	50		HD 50FPS (720p)
1	30	30		HD 30FPS
1	25	25	40	(PAL, 1080i)
1001	30000	29.97	33.667	(NTSC, 1080i)
1	24	24	41.667	(FILM, 1080, DCinema)
1001	24000	23.98		23.98 psf/p
66000	1000000	15.152	65.998	(NTSC->Multimedia)
1	15	15	66.667	(MultiMedia)
1	22050	22.050kHz	--	(Audio 22kHz 8 Bit Mono)
4	176400	44.1kHz	--	(Audio 44.1kHz 16 Bit Stereo)
2	48000	48kHz	--	(Audio 48kHz 16 Mono --OR-- 48kHz 8 Bit Stereo)

Definition at line 816 of file vvwtypes.h.

DWORD VVWSYSTEM::dwResDrastic

Reserved. Set to 0 on allocate and do not touch.

Definition at line 861 of file vvwtypes.h.

DWORD VVWSYSTEM::dwReserved[4]

As it says, reserved, do not touch.

Definition at line 762 of file vvwtypes.h.

DWORD VVWSYSTEM::dwScale

Scale - the frame divisor into [VVWSYSTEM::dwRate](#) to get the frame frame rate

Scale	Rate	FrameRate	MsPerFrame	
1	60	60		HD 60FPS (720p)
1001	60000	59.94		HD 59.94FPS (720p)
1	50	50		HD 50FPS (720p)
1	30	30		HD 30FPS
1	25	25	40	(PAL, 1080i)
1001	30000	29.97	33.667	(NTSC, 1080i)
1	24	24	41.667	(FILM, 1080, DCinema)
1001	24000	23.98		23.98 psf/p
66000	1000000	15.152	65.998	(NTSC->Multimedia)
1	15	15	66.667	(MultiMedia)
1	22050	22.050kHz	--	(Audio 22kHz 8 Bit Mono)
4	176400	44.1kHz	--	(Audio 44.1kHz 16 Bit Stereo)
2	48000	48kHz	--	(Audio 48kHz 16 Mono --OR-- 48kHz 8 Bit Stereo)

Definition at line 796 of file vvwtypes.h.

DWORD VVWSYSTEM::dwStreams

The total number of audio, video and information streams in the media file. This does not include channels within stereo audio streams which are considered to be one channel.

Definition at line 739 of file vvwtypes.h.

DWORD VVWSYSTEM::dwSuggestedBufferSize

Suggested buffer size is set by the writer and is supposed to be the size of the largest 'chunk' (audio or video) of data that will have to be read at once. More important in systems where memory is at a premium. With Win32 we only use this to determine the correct audio buffer size if it is greater than half a second of stereo audio at the audio streams sample rate and bit size.

Definition at line 747 of file vvwtypes.h.

DWORD VVWSYSTEM::dwTotalFrames

This is the total number of actual video frames (counted as a frame count) of the longest video stream in the file. If there is no video stream in the file, the audio will be broken up into frames per ([VVWSYSTEM::dwRate](#) / [VVWSYSTEM::dwScale](#)) and if they are set to 0 then, we normally fill it with an NTSC value of frames by default. This is the place we normally count on to be correct for conversion source length and clip length in VVW.

Definition at line 727 of file vvwtypes.h.

DWORD VVWSYSTEM::dwType

VVW and ME internal flags - Add Flags Here MF_TYPE_???, AVH_TYPE_???

Definition at line 837 of file vvwtypes.h.

DWORD VVWSYSTEM::dwVidStandard

Basic video standard. Was PAL/NTSC, but should now use [GS_SIGFORM_NTSC](#) defines in [mediacmd.h](#)

Definition at line 845 of file vvwtypes.h.

DWORD VVWSYSTEM::dwWidth

Basic or largest width of a video stream within this file.

Definition at line 751 of file vvwtypes.h.

char VVWSYSTEM::szFileType[_VVWXXX_NAME_SIZE]

A marker for the time of media file or module that created this structure is Free form, used to present to user when problems occur.

Definition at line 833 of file vvwtypes.h.

The documentation for this struct was generated from the following file:

527 E:/drastic/api/mediacmd/src/[vwtypes.h](#)

VWVIDEO Struct Reference

```
#include <vwtypes.h>
```

Public Attributes

528 [DWORD biSize](#)

Size of the BITMAPINFOHEADER portion of this structure (sizeof(BITMAPINFOHEADER) + any [VWVIDEO::dwReserved](#) used)

529 [LONG biWidth](#)

Width of the video frame.

530 [LONG biHeight](#)

Height of the video frame. CAUTION: For vertically inverted frames this WILL be negative.

531 [WORD biPlanes](#)

Number of RGB groups (like photoshop layers) - Always 1 for our purposes.

532 [WORD biBitCount](#)

Number of bits per pixel (eg. YUV422=16, RGB=24, RGBA=32)

533 [DWORD biCompression](#)

Compression - a fourcc usually, but not always equal to fccHandler. Denotes compression type of frame - see fccDef.h.

534 [DWORD biSizeImage](#)

*Size of the image. For uncompressed $biWidth * abs(biHeight) * (biBitCount / 8)$ in unsigned chars. For compressed, variable.*

535 [LONG biXPelsPerMeter](#)

Horizontal picture elements per meter - normally 0.

536 [LONG biYPelsPerMeter](#)

Vertical picture elements per meter - normally 0.

537 [DWORD biClrUsed](#)

For colour tables in dwReserved, the number of RGBQUAD elements used.

538 [DWORD biClrImportant](#)

For colour tables in dwReserved, the number of RGBQUAD elements that are critical to display (for windows palette wars in < 256 colour mode)

539 [DWORD dwReserved](#) [_VWXXX_RESERVED_SIZE]

540 [DWORD fccType](#)

For [VWVIDEO](#) structure this is always streamtypeVIDEO == 'vids'.

541 [DWORD fccHandler](#)

Codec type, see fccDef.h Normally the same as [VWVIDEO::biCompression](#) but not always.

542 [DWORD dwFlags](#)

543 [DWORD dwCaps](#)

Not sure. See [VWSYSTEM::dwCaps](#) for possible interp if something is set. MS Doc: currently unused.

544 [WORD wPriority](#)

Priority of stream (<-MSDoc in relation to other streams in the file I suppose)

545 [WORD wLanguage](#)

Language of stream (<-MSDoc but no language id defines)

546 [DWORD dwScale](#)

dwRate/dwScale = frame rate. See [VSYSTEM::dwScale](#) for more info and table example

547 [DWORD dwRate](#)

dwRate/dwScale = frame rate. See [VSYSTEM::dwRate](#) for more info and table example

548 [DWORD dwStart](#)

549 [DWORD dwLength](#)

550 [DWORD dwInitialFrames](#)

551 [DWORD dwSuggestedBufferSize](#)

552 [DWORD dwQuality](#)

553 [DWORD dwSampleSize](#)

554 [RECT rcFrame](#)

555 [DWORD dwEditCount](#)

Number of times the stream has been edited. The stream handler maintains this count.

556 [DWORD dwFormatChangeCount](#)

Number of times the stream format has changed. The stream handler maintains this count.

557 char [szName](#) [_VWXXX_NAME_SIZE]

Null-terminated string containing a description of the stream.

558 [LONG biPitch](#)

The number of unsigned chars in a row of pixels. Allows for unsigned char/WORD/DWORD alignment of lines as nec for format.

559 [DWORD dwDrFlags](#)

560 [DWORD dwFileType](#)

Source File Type.

561 [DWORD dwResDrastic](#)

Reserved, init to zero and leave alone.

Detailed Description

The video structure is a combination of a BITMAPINFOHEADER and a AVISTREAMINFO structure. The top half should be treated as a BITMAPINFOHEADER and the bottom as a related but independent structure. These are as they appear in an AVI file and are manipulated to fill other file types like OMF and MOV.

Definition at line 947 of file vvwtypes.h.

Member Data Documentation

[WORD VVWVIDEO::biBitCount](#)

Number of bits per pixel (eg. YUV422=16, RGB=24, RGBA=32)

Definition at line 958 of file vvwtypes.h.

[DWORD VVWVIDEO::biClrImportant](#)

For color tables in dwReserved, the number of RGBQUAD elements that are critical to display (for windows palette wars in < 256 color mode)

Definition at line 970 of file vvwtypes.h.

[DWORD VVWVIDEO::biClrUsed](#)

For colour tables in dwReserved, the number of RGBQUAD elements used.

Definition at line 968 of file vvwtypes.h.

[DWORD VVWVIDEO::biCompression](#)

Compression - a fourcc usually, but not always equal to fccHandler. Denotes compression type of frame - see fccDef.h.

Definition at line 960 of file vvwtypes.h.

[LONG VVWVIDEO::biHeight](#)

Height of the video frame. CAUTION: For vertically inverted frames this WILL be negative.

Definition at line 954 of file vvwtypes.h.

[LONG VVWVIDEO::biPitch](#)

The number of unsigned chars in a row of pixels. Allows for unsigned char/WORD/DWORD alignment of lines as nec for format.

Definition at line 1134 of file vvwtypes.h.

WORD VVWVIDEO::biPlanes

Number of RGB groups (like photoshop layers) - Always 1 for our purposes.

Definition at line 956 of file vvwtypes.h.

DWORD VVWVIDEO::biSize

Size of the BITMAPINFOHEADER portion of this structure (sizeof(BITMAPINFOHEADER) + any [VVWVIDEO::dwReserved](#) used)

Definition at line 950 of file vvwtypes.h.

DWORD VVWVIDEO::biSizeImage

Size of the image. For uncompressed $biWidth * abs(biHeight) * (biBitCount / 8)$ in unsigned chars. For compressed, variable.

Definition at line 962 of file vvwtypes.h.

LONG VVWVIDEO::biWidth

Width of the video frame.

Definition at line 952 of file vvwtypes.h.

LONG VVWVIDEO::biXPelsPerMeter

Horizontal picture elements per meter - normally 0.

Definition at line 964 of file vvwtypes.h.

LONG VVWVIDEO::biYPelsPerMeter

Vertical picture elements per meter - normally 0.

Definition at line 966 of file vvwtypes.h.

DWORD VVWVIDEO::dwCaps

Not sure. See [VVWSYSTEM::dwCaps](#) for possible interp if something is set. MS Doc: currently unused.

Definition at line 1072 of file vvwtypes.h.

DWORD VVWVIDEO::dwDrFlags

Our internal flags including:

DRFLAGS_ZERO_FIELD_DOMINANT ,	DRFLAGS_ZERO_FIELD_DOMINANT ,
DRFLAGS_FIRST_FIELD_DOMINANT ,	DRFLAGS_HAS_KEYFRAMES ,
DRFLAGS_FCC_MJPG_DIGISUITE ,	DRFLAGS_FCC_MJPG_DCx0 ,
DRFLAGS_FCC_MJPG_DSEDT ,	DRFLAGS_FCC_MJPG_JPGDIB ,
DRFLAGS_FCC_MJPG_JFIF ,	DRFLAGS_FCC_USE_INTERN ,

[DRFLAGS_FCC_USE_QT](#), [DRFLAGS_FCC_USE_ICM](#),
[DRFLAGS_CODECPRIVATEDATA_AVI](#), [DRFLAGS_CODECPRIVATEDATA_MOV](#),
[DRFLAGS_CODECPRIVATEDATA_OMF](#), [DTV_VW_PREVIEW](#)

Definition at line 1197 of file vvwtypes.h.

DWORD VVWVIDEO::dwEditCount

Number of times the stream has been edited. The stream handler maintains this count.

Definition at line 1123 of file vvwtypes.h.

DWORD VVWVIDEO::dwFileType

Source File Type.

Definition at line 1199 of file vvwtypes.h.

DWORD VVWVIDEO::dwFlags

Definition at line 1070 of file vvwtypes.h.

DWORD VVWVIDEO::dwFormatChangeCount

Number of times the stream format has changed. The stream handler maintains this count.

Definition at line 1125 of file vvwtypes.h.

DWORD VVWVIDEO::dwInitialFrames

Amount of audio in the file before video commences. For offset files, typically 0.75 sec converted to units per [VVWVIDEO::dwRate](#)/[VVWVIDEO::dwScale](#). For high end files, always zero as audio and video are sent without skew (except Premiere, which uses 'rec ' chunks and audio skew)

Definition at line 1096 of file vvwtypes.h.

DWORD VVWVIDEO::dwLength

Length of the video stream in units per [VVWVIDEO::dwRate](#)/[VVWVIDEO::dwScale](#) (for video - frames)

Definition at line 1090 of file vvwtypes.h.

DWORD VVWVIDEO::dwQuality

Quality used by the compressor. Between 0 and 10,000 or -1 if default quality. For some compressors, the -1 can also mean the quality info is encoded into the frame or in the [dwReserved](#) or other private data area.

Definition at line 1107 of file vvwtypes.h.

DWORD VVWVIDEO::dwRate

dwRate/dwScale = frame rate. See VVSYSTEM::dwRate for more info and table example
 Definition at line 1080 of file vvwtypes.h.

DWORD VVWVIDEO::dwResDrastic

Reserved, init to zero and leave alone.
 Definition at line 1201 of file vvwtypes.h.

DWORD VVWVIDEO::dwReserved[_VVWXXX_RESERVED_SIZE]

The dwReserved may hold many things. Whatever it holds, the amount used can be determined by subtracting sizeof(BITMAPINFOHEADER) from [VVWVIDEO::biSize](#). Here are some possible uses

```

Table of struct RGBQUAD { unsigned char rgbBlue, rgbGreen, rgbRed,
rgbReserved; };

typedef struct tagJPEGINFOHEADER {
    // compression-specific fields
    // these fields are defined for 'JPEG' and 'MJPG'
    DWORD      JPEGSize;
    DWORD      JPEGProcess;

    // Process specific fields
    DWORD      JPEGColorSpaceID;
    DWORD      JPEGBitsPerSample;
    DWORD      JPEGHSubSampling;
    DWORD      JPEGVSubSampling;
} JPEGINFOHEADER;

typedef struct tagVIDEOINFOHEADER {
    RECT      rcSource;          // The bit we really want
to use
    RECT      rcTarget;        // Where the video should
go
    DWORD     dwBitRate;       // Approximate bit data
rate
    DWORD     dwBitErrorRate;  // Bit error rate for this
stream
    REFERENCE\_TIME AvgTimePerFrame; // Average time per frame
(100ns units)

    BITMAPINFOHEADER bmiHeader;

} VIDEOINFOHEADER;

typedef struct tagMPEG1VIDEOINFO {
    VIDEOINFOHEADER hdr;          // Compatible with
VIDEOINFO
    DWORD     dwStartTimeCode;   // 25-bit Group of
pictures time code
// at start of data
    DWORD     cbSequenceHeader;  // Length in unsigned
chars of bSequenceHeader
    unsigned char bSequenceHeader[1]; // Sequence
header including
// quantization matrices if any

```

```

        } MPEG1VIDEOINFO;

        typedef struct tagAnalogVideoInfo {
            RECT rcSource; // Width max is 720,
height varies w/ TransmissionStd
            RECT rcTarget; // Where the video should
go
            DWORD dwActiveWidth; // Always 720 (CCIR-601
active samples per line)
            DWORD dwActiveHeight; // 483 for NTSC, 575 for
PAL/SECAM
            REFERENCE_TIME AvgTimePerFrame; // Normal ActiveMovie
units (100 nS)
        } ANALOGVIDEOINFO;

        typedef struct tagMPEG2VIDEOINFO {
            VIDEOINFOHEADER2 hdr;
            DWORD dwStartTimeCode; // ?? not used
for DVD ??
            DWORD cbSequenceHeader; // is 0 for DVD
(no sequence header)
            DWORD dwProfile; // use enum
MPEG2Profile
            DWORD dwLevel; // use enum
MPEG2Level
            DWORD dwFlags; // use AMMPEG2_*
defines. Reject connection if undefined bits are not 0
            DWORD dwSequenceHeader[1]; // DWORD instead
of unsigned char for alignment purposes

// For MPEG-2, if a sequence_header is included, the sequence_extension
// should also be included
        } MPEG2VIDEOINFO;

        typedef struct tagVIDEOINFOHEADER2 {
            RECT rcSource;
            RECT rcTarget;
            DWORD dwBitRate;
            DWORD dwBitErrorRate;
            REFERENCE_TIME AvgTimePerFrame;
            DWORD dwInterlaceFlags; // use AMINTERLACE_*
defines. Reject connection if undefined bits are not 0
            DWORD dwCopyProtectFlags; // use AMCOPYPROTECT_*
defines. Reject connection if undefined bits are not 0
            DWORD dwPictAspectRatioX; // X dimension of
picture aspect ratio, e.g. 16 for 16x9 display
            DWORD dwPictAspectRatioY; // Y dimension of
picture aspect ratio, e.g. 9 for 16x9 display
            DWORD dwReserved1; // must be 0; reject
connection otherwise
            DWORD dwReserved2; // must be 0; reject
connection otherwise
            BITMAPINFOHEADER bmiHeader;
        } VIDEOINFOHEADER2;

```

Definition at line 1055 of file vvwtypes.h.

[DWORD VVVVIDEO::dwSampleSize](#)

Size, in unsigned chars, of a single data sample. If the value of this member is zero, the samples can vary in size and each data sample (such as a video frame) must be in a separate chunk. A nonzero value indicates that multiple samples of data can be grouped into a single chunk within the file. For video streams, this number is typically zero, although it can be nonzero if all video frames are the same size. For audio streams, this number should be the

same as the nBlockAlign member of the WAVEFORMAT or WAVEFORMATEX structure describing the audio.

Definition at line 1116 of file vvwtypes.h.

DWORD VVWVIDEO::dwScale

dwRate/dwScale = frame rate. See VVSYSTEM::dwScale for more info and table example

Definition at line 1078 of file vvwtypes.h.

DWORD VVWVIDEO::dwStart

Delay in units per [VVWVIDEO::dwRate](#)/VVWVIDEO::dwScale (for video - frames) for this stream to start in the playback of the file. NOTE AVI v1.0 and simple avi readers will choke or play incorrectly if this is not 0, so be careful.

Definition at line 1086 of file vvwtypes.h.

DWORD VVWVIDEO::dwSuggestedBufferSize

Recommended buffer size based on the largest single chunk in the file. Set by writer, so often incorrect or 0.

Definition at line 1101 of file vvwtypes.h.

DWORD VVWVIDEO::fccHandler

Codec type, see fccDef.h Normally the same as [VVWVIDEO::biCompression](#) but not always.

Definition at line 1062 of file vvwtypes.h.

DWORD VVWVIDEO::fccType

For [VVWVIDEO](#) structure this is always streamtypeVIDEO == 'vids'.

Definition at line 1060 of file vvwtypes.h.

RECT VVWVIDEO::rcFrame

Dimensions of the video destination rectangle. The values represent the coordinates of upper left corner, the height, and the width of the rectangle.

Definition at line 1121 of file vvwtypes.h.

char VVWVIDEO::szName[_VVWXXX_NAME_SIZE]

Null-terminated string containing a description of the stream.

Definition at line 1127 of file vvwtypes.h.

WORD VVWVIDEO::wLanguage

Language of stream (<-MSDoc but no language id defines)

Definition at line 1076 of file vvwtypes.h.

[WORD VVWVIDEO::wPriority](#)

Priority of stream (<-MSDoc in relation to other streams in the file I suppose)

Definition at line 1074 of file vvwtypes.h.

The documentation for this struct was generated from the following file:

562 E:/drastic/api/mediacmd/src/[vwtypes.h](#)

WAVEFORMATEXTENSIBLE Struct Reference

```
#include <dtsystemtypes.h>
```

Public Attributes

```
563 WAVEFORMATEX Format  
564 union {  
565 WORD wValidBitsPerSample  
566 WORD wSamplesPerBlock  
567 WORD wReserved  
568 } Samples  
569 DWORD dwChannelMask  
570 GUID SubFormat
```

Detailed Description

Definition at line 314 of file dtsystemtypes.h.

Member Data Documentation

[DWORD WAVEFORMATEXTENSIBLE::dwChannelMask](#)

Definition at line 321 of file dtsystemtypes.h.

[WAVEFORMATEX WAVEFORMATEXTENSIBLE::Format](#)

Definition at line 315 of file dtsystemtypes.h.

union { ... } [WAVEFORMATEXTENSIBLE::Samples](#)

[GUID WAVEFORMATEXTENSIBLE::SubFormat](#)

Definition at line 323 of file dtssystemtypes.h.

[WORD WAVEFORMATEXTENSIBLE::wReserved](#)

Definition at line 319 of file dtssystemtypes.h.

[WORD WAVEFORMATEXTENSIBLE::wSamplesPerBlock](#)

Definition at line 318 of file dtssystemtypes.h.

[WORD WAVEFORMATEXTENSIBLE::wValidBitsPerSample](#)

Definition at line 317 of file dtssystemtypes.h.

The documentation for this struct was generated from the following file:

571 E:/drastic/api/mediacmd/src/[dtssystemtypes.h](#)

File Documentation

E:/drastic/api/mediacmd/src/dtnetdirect.cpp File Reference

```
#include <stdlib.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <unistd.h>
#include <stdio.h>
#include "dtssystemtypes.h"
#include "mediacmd.h"
#include "dtnetdirect.h"
#include "vwif.h"
#include <dtutil.h>
```

Classes

572 struct [VWNETDIRECT_CHANNEL](#)

Defines

573 #define [_USE_DTOS_LOCKING](#)

Functions

```
574 unsigned long __stdcall vwInit (void)
575 unsigned long __stdcall vwStart (void)
576 unsigned long __stdcall vwQuit (void)
577 unsigned long __stdcall vwPing (char *szAddress, DWORD dwPort, int nTimeOut)
578 VVWHANDLE __stdcall vwOpen (char *szAddress, unsigned long dwPort, unsigned long
    dwChannel)
579 unsigned long __stdcall vwConnectionStatus (VVWHANDLE hVWV)
```

580 unsigned long __stdcall [vwwCmd](#) (VVWHANDLE hVWV, [PMEDIACMD](#) pCmd)
581 unsigned long __stdcall [vwwConfigure](#) (unsigned long dwChannelIndex, [HANDLE](#) hWnd)
582 unsigned long __stdcall [vwwClose](#) (VVWHANDLE hVWV)

Variables

583 unsigned long [gdwUnique](#) = 1
584 unsigned long [gdwInit](#) = 0

Define Documentation

#define _USE_DTOS_LOCKING

Definition at line 35 of file dtnetdirect.cpp.

Function Documentation

unsigned long __stdcall vwwClose (VVWHANDLE hVWV)

Definition at line 470 of file dtnetdirect.cpp.

unsigned long __stdcall vwwCmd (VVWHANDLE hVWV, [PMEDIACMD](#) pCmd)

Send a media command at the VVW channel

Definition at line 293 of file dtnetdirect.cpp.

unsigned long __stdcall vwwConfigure (unsigned long dwChannelIndex, [HANDLE](#) hWnd)

Open the selected VVW channel

Definition at line 459 of file dtnetdirect.cpp.

unsigned long __stdcall vwwConnectionStatus (VVWHANDLE hVWV)

Is the socket open and connected (0 = ok, else error)

Definition at line 275 of file dtnetdirect.cpp.

unsigned long __stdcall vwwInit (void)

Init - called to initialize the subsystem before any Start, Load, Open, Cmd, Configure or Close Called once for the entire system

Definition at line 79 of file dtnetdirect.cpp.

VVWHANDLE __stdcall vwwOpen (char * szAddress, unsigned long dwPort, unsigned long dwChannel)

Open the selected VVW channel

Definition at line 200 of file dtnetdirect.cpp.

unsigned long __stdcall vvwPing (char * szAddress, [DWORD](#) dwPort, int nTimeOut)

Determine if the target machine is online

Definition at line 138 of file dtnetdirect.cpp.

unsigned long __stdcall vvwQuit (void)

Quit - called just before the DLL is unloaded. All channels should be closed and the operations complete before this is called. If the application crashes, DllMain should close all open channels before calling this function on the unload

Definition at line 112 of file dtnetdirect.cpp.

unsigned long __stdcall vvwStart (void)

Start - called once after all local channels are opened to complete any processing or start any global threads

Definition at line 97 of file dtnetdirect.cpp.

Variable Documentation

unsigned long [gdwInit](#) = 0

Definition at line 68 of file dtnetdirect.cpp.

unsigned long [gdwUnique](#) = 1

Definition at line 67 of file dtnetdirect.cpp.

E:/drastic/api/mediacmd/src/dtnetdirect.h File Reference

Functions

585 unsigned long __stdcall [vwvInit](#) (void)

586 unsigned long __stdcall [vwvStart](#) (void)

587 unsigned long __stdcall [vwvQuit](#) (void)

588 unsigned long __stdcall [vwvPing](#) (char *szAddress, [DWORD](#) dwPort, int nTimeOut)

589 VVWHANDLE __stdcall [vwvOpen](#) (char *szAddress, unsigned long dwPort, unsigned long dwChannel)

590 unsigned long __stdcall [vwvConnectionStatus](#) (VVWHANDLE hVvw)

591 unsigned long __stdcall [vwvCmd](#) (VVWHANDLE hVvw, [PMEDIACMD](#) pCmd)

592 unsigned long __stdcall [vwvConfigure](#) (unsigned long dwChannelIndex, [HANDLE](#) hWnd)

593 unsigned long __stdcall [vwvClose](#) (VVWHANDLE hVvw)

Function Documentation

unsigned long __stdcall vvwClose (VVWHANDLE hVvw)

Definition at line 470 of file dtnetdirect.cpp.

unsigned long __stdcall vvwCmd (VVWHANDLE hVvw, [PMEDIACMD](#) pCmd)

Send a media command at the VVW channel

Definition at line 293 of file dtnetdirect.cpp.

unsigned long __stdcall vvwConfigure (unsigned long dwChannelIndex, [HANDLE](#) hWnd)

Open the selected VVW channel

Definition at line 459 of file dtnetdirect.cpp.

unsigned long __stdcall vvwConnectionStatus (VVWHANDLE hVvw)

Is the socket open and connected (0 = ok, else error)

Definition at line 275 of file dtnetdirect.cpp.

unsigned long __stdcall vvwInit (void)

Init - called to initialize the subsystem before any Start, Load, Open, Cmd, Configure or Close Called once for the entire system

Definition at line 79 of file dtnetdirect.cpp.

VVWHANDLE __stdcall vvwOpen (char * szAddress, unsigned long dwPort, unsigned long dwChannel)

Open the selected VVW channel

Definition at line 200 of file dtnetdirect.cpp.

unsigned long __stdcall vvwPing (char * szAddress, [DWORD](#) dwPort, int nTimeOut)

Determine if the target machine is online

Definition at line 138 of file dtnetdirect.cpp.

unsigned long __stdcall vvwQuit (void)

Quit - called just before the DLL is unloaded. All channels should be closed and the operations complete before this is called. If the application crashes, DllMain should close all open channels before calling this function on the unload

Definition at line 112 of file dtnetdirect.cpp.

unsigned long __stdcall vvwStart (void)

Start - called once after all local channels are opened to complete any processing or start any global threads

Definition at line 97 of file dtnetdirect.cpp.

E:/drastic/api/mediacmd/src/dtsystemtypes.h File Reference

```
#include <stdint.h>
#include <unistd.h>
#include <limits.h>
```

Classes

```
594 struct LARGE\_INTEGER\_MEMBERS
595 union LARGE\_INTEGER
596 struct RECT
597 struct POINT
598 struct SIZE
599 struct \_RGB
600 struct tagRGBQUAD
601 struct \_GUID
602 struct tWAVEFORMATEX
603 struct WAVEFORMATEXTENSIBLE
604 struct tagBITMAPINFOHEADER
605 struct tagBITMAPINFO
606 struct tagBITMAPFILEHEADER
```

Defines

```
607 #define TRACEKI(...)
608 #define DRASTICCONFIG\_DIR "Drastic"
609 #define DT\_TARGET\_OS\_LINUX
610 #define DTFILESEPARATOR /
611 #define DTFILESEPARATOR\_CHAR '/'
612 #define DTFILESEPARATOR\_STRING "/"
613 #define \_GNU\_SOURCE
614 #define NEWLINE\_STR "\n\r"
615 #define DT\_TARGET\_COMP\_GNU
616 #define DTSYSTEM\_SHARED\_LIB\_PREFIX "lib"
617 #define DTSYSTEM\_SHARED\_LIB\_EXT ".so"
618 #define DTSYSTEM\_SHARED\_PLUGIN\_EXT ".so"
619 #define xFFFFFFFFFFFFFFFF 0xFFFFFFFFFFFFFFFFLL
620 #define I64\(\_x\_\) __x_##LL
621 #define \_MAX\_NET\_PATH 4096
622 #define WCHAR short
623 #define LOBYTE(w) ((BYTE)(w))
624 #define HIBYTE(w) (((WORD)(w) >> 8) & 0xFF)
625 #define MAKELONG(low, high) (((WORD)((BYTE)(low)) | (((WORD)((BYTE)(high))) <<
8)))
626 #define LOWORD(l) ((WORD)(DWORD)(l))
627 #define HIWORD(l) ((WORD)(((DWORD)(l) >> 16) & 0xFFFF))
628 #define LOWORD(l) ((WORD)(DWORD)(l))
629 #define HISWORD(l) ((WORD)(((DWORD)(l) >> 16) & 0xFFFF))
630 #define MAKELONG(low, high) (((DWORD)((WORD)(low)) | (((DWORD)((WORD)(high)))
<< 16)))
631 #define GetRValue(rgb) ((BYTE)(rgb))
632 #define GetGValue(rgb) ((BYTE)((WORD)(rgb) >> 8))
633 #define GetBValue(rgb) ((BYTE)((rgb) >> 16))
634 #define MakeRGB(r, g, b)
635 #define COLORREF RGBQUAD
636 #define \_WAVEFORMATEXTENSIBLE
637 #define \_ASSERT
```

```

638 #define ASSERT
639 #define STDMETHODIMP DWORD
640 #define STDAPI
641 #define WINAPI
642 #define CALLBACK
643 #define __stdcall
644 #define __cdecl
645 #define VOID void
646 #define FAR
647 #define PASCAL
648 #define _T
649 #define closesocket close
650 #define INVALID_SOCKET 0
651 #define SOCKET_ERROR -1
652 #define ERROR_SUCCESS 0
653 #define TRUE 1
654 #define FALSE 0
655 #define NULL 0
656 #define WAIT_OBJECT_0 0x00000000
657 #define WAIT_ABANDONED 0x00000080L
658 #define WAIT_TIMEOUT 0x00000102
659 #define STATUS_PENDING ((DWORD)0x00000103L)
660 #define BI_RGB 0
661 #define BI_BITFIELDS 0x00000003
662 #define IDOK 0
663 #define MB_OK 0
664 #define S_OK 0
665 #define _MAX_EXT NAME_MAX
666 #define _MAX_FNAME NAME_MAX
667 #define _MAX_DIR PATH_MAX
668 #define _MAX_DRIVE 4
669 #define _MAX_PATH PATH_MAX
670 #define ERROR_IO_PENDING 997
671 #define MOVEFILE_REPLACE_EXISTING 0x00000001
672 #define MOVEFILE_COPY_ALLOWED 0x00000002
673 #define MOVEFILE_DELAY_UNTIL_REBOOT 0x00000004
674 #define MOVEFILE_WRITE_THROUGH 0x00000008
675 #define SD_BOTH 2
676 #define vsprintf sprintf
677 #define strlen
678 #define strcat
679 #define strcpy
680 #define strcmp strcmp
681 #define strncmp strncmp
682 #define Sleep(__millisec_) usleep((__millisec_)*1000)
683 #define MoveFile rename
684 #define RenameFile rename
685 #define RemoveDir rmdir
686 #define ZeroMemory(__where_, __howmuch_) ZeroMemoryFast(__where_, __howmuch_)
687 #define FillMemory(__where_, __howmuch_, __what_)
        memset(__where_, __what_, __howmuch_)
688 #define CopyMemory CopyMemoryFast
689 #define MoveMemory memmove
690 #define ioctlsocket ioctl
691 #define MAX(a, b) (((a) > (b)) ? (a) : (b))
692 #define MIN(a, b) (((a) < (b)) ? (a) : (b))

```

```
693 #define ABS(a) (((a) < 0) ? -(a) : (a))
694 #define CLAMP(x, low, high) (((x) > (high)) ? (high) : (((x) < (low)) ? (low) : (x)))
695 #define VVWHANDLE void*
```

Typedefs

```
696 typedef size_t DTHANDLE
697 typedef uint32_t DWORD
698 typedef int32_t LONG
699 typedef DWORD FOURCC
700 typedef LONG INT
701 typedef unsigned short WORD
702 typedef short SHORT
703 typedef unsigned char UCHAR
704 typedef unsigned char BYTE
705 typedef unsigned char byte
706 typedef char CHAR
707 typedef char TCHAR
708 typedef char _TCHAR
709 typedef LONG BOOL
710 typedef LONG BOOLEAN
711 typedef unsigned int UINT
712 typedef BYTE * PBYTE
713 typedef WORD * PWORD
714 typedef short * PSHORT
715 typedef DWORD * PDWORD
716 typedef LONG * PLONG
717 typedef BYTE * LPBYTE
718 typedef WORD * LPWORD
719 typedef DWORD * LPDWORD
720 typedef LONG * LPLONG
721 typedef size_t LONG_PTR
722 typedef size_t * PLONG_PTR
723 typedef size_t ULONG_PTR
724 typedef size_t * PULONG_PTR
725 typedef ULONG_PTR DWORD_PTR
726 typedef ULONG_PTR * PDWORD_PTR
727 typedef unsigned long long DWORDLONG
728 typedef long long LONGLONG
729 typedef DWORDLONG QWORD
730 typedef DWORDLONG QUADWORD
731 typedef union _LARGE_INTEGER LARGE_INTEGER
732 typedef LARGE_INTEGER * PLARGE_INTEGER
733 typedef RECT * PRECT
734 typedef RECT * LPRECT
735 typedef struct _POINT POINT
736 typedef POINT * PPOINT
737 typedef POINT * LPPOINT
738 typedef SIZE * PSIZE
739 typedef SIZE * LPSIZE
740 typedef struct _RGB RGB
741 typedef RGB * PRGB
742 typedef RGB * LPRGB
743 typedef struct tagRGBQUAD RGBQUAD
744 typedef struct _GUID GUID
745 typedef struct _GUID * REFGUID
```

746 typedef struct [_GUID](#) * [LPGUID](#)
747 typedef struct [tWAVEFORMATEX](#) [WAVEFORMATEX](#)
748 typedef struct [tWAVEFORMATEX](#) * [PWAVEFORMATEX](#)
749 typedef struct
750 [WAVEFORMATEXTENSIBLE](#) * [PWAVEFORMATEXTENSIBLE](#)
751 typedef struct [tagBITMAPINFOHEADER](#) [BITMAPINFOHEADER](#)
752 typedef struct
753 [tagBITMAPINFOHEADER](#) * [LPBITMAPINFOHEADER](#)
754 typedef struct [tagBITMAPINFO](#) [BITMAPINFO](#)
755 typedef struct [tagBITMAPFILEHEADER](#) [BITMAPFILEHEADER](#)
756 typedef struct
757 [tagBITMAPFILEHEADER](#) * [PBITMAPFILEHEADER](#)
758 typedef long long int [__int64](#)
759 typedef unsigned long long int [__uint64](#)
760 typedef [DWORD](#) [WPARAM](#)
761 typedef long [LPARAM](#)
762 typedef signed long [SInt32](#)
763 typedef [__int64](#) [HANDLE](#)
764 typedef size_t [HINSTANCE](#)
765 typedef size_t [HMODULE](#)
766 typedef size_t [HWND](#)
767 typedef int32_t [HRESULT](#)
768 typedef [HRESULT](#) [MMRESULT](#)
769 typedef size_t [HDC](#)
770 typedef size_t [HBRUSH](#)
771 typedef size_t [PAINTSTRUCT](#)
772 typedef long [LRESULT](#)
773 typedef int [SOCKET](#)
774 typedef struct sockaddr_in [SOCKADDR_IN](#)
775 typedef struct sockaddr_in * [PSOCKADDR_IN](#)
776 typedef struct sockaddr_in FAR * [LPSOCKADDR_IN](#)
777 typedef struct sockaddr [SOCKADDR](#)
778 typedef struct sockaddr * [PSOCKADDR](#)
779 typedef struct sockaddr FAR * [LPSOCKADDR](#)
780 typedef char * [PSTR](#)
781 typedef char * [PTSTR](#)
782 typedef void * [PVOID](#)
783 typedef char * [LPSTR](#)
784 typedef char * [LPTSTR](#)
785 typedef void * [LPVOID](#)
786 typedef const char * [LPCSTR](#)
787 typedef const char * [LPCTSTR](#)
788 typedef const void * [LPCVOID](#)
789 typedef void *(* [LPTHREAD_START_ROUTINE](#))(void *)
790 typedef char * [Ptr](#)
791 typedef [Ptr](#) * [Handle](#)
792 typedef long [Size](#)
793 typedef signed char [SInt8](#)
794 typedef unsigned short [UInt16](#)
795 typedef signed short [SInt16](#)
796 typedef unsigned long [UInt32](#)
797 typedef unsigned long [FourCharCode](#)
798 typedef [FourCharCode](#) [OSType](#)
799 typedef [FourCharCode](#) [ResType](#)
800 typedef [OSType](#) * [OSTypePtr](#)
801 typedef [ResType](#) * [ResTypePtr](#)

802 typedef signed short [OSErr](#)
803 typedef signed long [OSStatus](#)

Define Documentation

#define __cdecl

Definition at line 408 of file dtssystemtypes.h.

#define __stdcall

Definition at line 407 of file dtssystemtypes.h.

#define _ASSERT

Definition at line 398 of file dtssystemtypes.h.

#define _GNU_SOURCE

Definition at line 91 of file dtssystemtypes.h.

#define _MAX_DIR PATH_MAX

Definition at line 462 of file dtssystemtypes.h.

#define _MAX_DRIVE 4

Definition at line 463 of file dtssystemtypes.h.

#define _MAX_EXT NAME_MAX

Definition at line 460 of file dtssystemtypes.h.

#define _MAX_FNAME NAME_MAX

Definition at line 461 of file dtssystemtypes.h.

#define _MAX_NET_PATH 4096

Definition at line 150 of file dtssystemtypes.h.

#define _MAX_PATH PATH_MAX

Definition at line 464 of file dtssystemtypes.h.

#define _T

Definition at line 413 of file dtssystemtypes.h.

#define _WAVEFORMATEXTENSIBLE_

Definition at line 325 of file dtssystemtypes.h.

#define ABS(a) (((a) < 0) ? -(a) : (a))

Definition at line 649 of file dtssystemtypes.h.

#define ASSERT

Definition at line 401 of file dtssystemtypes.h.

#define BI_BITFIELDS 0x00000003

Definition at line 455 of file dtssystemtypes.h.

#define BI_RGB 0

Definition at line 454 of file dtssystemtypes.h.

#define CALLBACK

Definition at line 406 of file dtssystemtypes.h.

#define CLAMP(x, low, high) (((x) > (high)) ? (high) : (((x) < (low)) ? (low) : (x)))

Definition at line 652 of file dtssystemtypes.h.

#define closesocket close

Definition at line 423 of file dtssystemtypes.h.

#define COLORREF [RGBQUAD](#)

Definition at line 288 of file dtssystemtypes.h.

#define CopyMemory CopyMemoryFast

Definition at line 494 of file dtssystemtypes.h.

#define DRASTICCONFIG_DIR "Drastic"

This is the directory we store our settings in. For application data it is /Drastic. For user data, is it .Drastic on unix and Drastic in windows. The dirs are Windows: APP: /ProgramData/Drastic/ USER: /Documents and Settings/[username]/Drastic Linux: APP: /etc/Drastic/ USER: /Home/[username]/.Drastic OS-X: APP: /Library/Applications Support/Drastic USER: /Home/[username]/.Drastic

to get the correct path use dtOSGetConfigDir(sz, false) for application directory and dtOSGetConfigDir(sz, true) for the user directory.

Definition at line 63 of file dtssystemtypes.h.

#define DT_TARGET_COMP_GNU

Definition at line 119 of file dtssystemtypes.h.

#define DT_TARGET_OS_LINUX

Find the DT_TARGET_OS_... and DT_TARGET_COMP_... types

Current OS: DT_TARGET_OS_LINUX, DT_TARGET_OS_WIN32,
DT_TARGET_OS_UNIX, DT_TARGET_OS_DTNIX32

Current Compiler: DT_TARGET_COMP_MSVC, DT_TARGET_COMP_GNU,
DT_TARGET_COMP_MINGW

Definition at line 83 of file dtssystemtypes.h.

#define DTFILESEPARATOR /

Definition at line 87 of file dtssystemtypes.h.

#define DTFILESEPARATOR_CHAR '/'

Definition at line 88 of file dtssystemtypes.h.

#define DTFILESEPARATOR_STRING "/"

Definition at line 89 of file dtssystemtypes.h.

#define DTSYSTEM_SHARED_LIB_EXT ".so"

Definition at line 138 of file dtssystemtypes.h.

#define DTSYSTEM_SHARED_LIB_PREFIX "lib"

Definition at line 137 of file dtssystemtypes.h.

#define DTSYSTEM_SHARED_PLUGIN_EXT ".so"

Definition at line 139 of file dtssystemtypes.h.

#define ERROR_IO_PENDING 997

Definition at line 467 of file dtssystemtypes.h.

#define ERROR_SUCCESS 0

Definition at line 438 of file dtssystemtypes.h.

#define FALSE 0

Definition at line 443 of file dtssystemtypes.h.

#define FAR

Definition at line 410 of file dtssystemtypes.h.

**#define FillMemory(__where_, __howmuch_, __what_)
memset(__where_,__what_,__howmuch_)**

Definition at line 493 of file dtssystemtypes.h.

#define GetBValue(rgb) ((BYTE)((rgb) >> 16))

Definition at line 238 of file dtssystemtypes.h.

#define GetGValue(rgb) ((BYTE)(((WORD)(rgb)) >> 8))

Definition at line 237 of file dtssystemtypes.h.

#define GetRValue(rgb) ((BYTE)(rgb))

Definition at line 236 of file dtssystemtypes.h.

#define HIBYTE(w) ((BYTE)(((WORD)(w)) >> 8) & 0xFF)

Definition at line 229 of file dtssystemtypes.h.

#define HISWORD(l) ((WORD)(((DWORD)(l)) >> 16) & 0xFFFF)

Definition at line 234 of file dtssystemtypes.h.

#define HIWORD(I) ((WORD)((((DWORD)(I)) >> 16) & 0xFFFF))

Definition at line 232 of file dtssystemtypes.h.

#define I64(__x_) __x_##LL

Definition at line 147 of file dtssystemtypes.h.

#define IDOK 0

Definition at line 456 of file dtssystemtypes.h.

#define INVALID_SOCKET 0

Definition at line 424 of file dtssystemtypes.h.

#define ioctlsocket ioctl

Definition at line 496 of file dtssystemtypes.h.

#define LOBYTE(w) ((BYTE)(w))

Definition at line 228 of file dtssystemtypes.h.

#define LOSWORD(I) ((SWORD)(DWORD)(I))

Definition at line 233 of file dtssystemtypes.h.

#define LOWORD(I) ((WORD)(DWORD)(I))

Definition at line 231 of file dtssystemtypes.h.

#define lstrcat strcat

Definition at line 482 of file dtssystemtypes.h.

#define lstrcpy strcpy

Definition at line 483 of file dtssystemtypes.h.

#define lstrlen strlen

Definition at line 481 of file dtssystemtypes.h.

```
#define MAKELONG( low, high) ((DWORD)((WORD)(low)) | (((DWORD)((WORD)(high))) << 16)))
```

Definition at line 235 of file dtssystemtypes.h.

```
#define MakeRGB( r, g, b)
```

```
Value: (((DWORD) ((BYTE) (r))) | ((DWORD) ((BYTE) (g)) << 8) \ | ((DWORD) ((BYTE) (b)) << 16))
```

Definition at line 239 of file dtssystemtypes.h.

```
#define MAKEWORD( low, high) ((WORD)((BYTE)(low)) | (((WORD)((BYTE)(high))) << 8)))
```

Definition at line 230 of file dtssystemtypes.h.

```
#define MAX( a, b) (((a) > (b)) ? (a) : (b))
```

General utility macros

Definition at line 643 of file dtssystemtypes.h.

```
#define MB_OK 0
```

Definition at line 457 of file dtssystemtypes.h.

```
#define MIN( a, b) (((a) < (b)) ? (a) : (b))
```

Definition at line 646 of file dtssystemtypes.h.

```
#define MoveFile rename
```

Definition at line 487 of file dtssystemtypes.h.

```
#define MOVEFILE_COPY_ALLOWED 0x00000002
```

Definition at line 470 of file dtssystemtypes.h.

```
#define MOVEFILE_DELAY_UNTIL_REBOOT 0x00000004
```

Definition at line 471 of file dtssystemtypes.h.

```
#define MOVEFILE_REPLACE_EXISTING 0x00000001
```

Definition at line 469 of file dtssystemtypes.h.

```
#define MOVEFILE_WRITE_THROUGH 0x00000008
```

Definition at line 472 of file dtssystemtypes.h.

#define MoveMemory memmove

Definition at line 495 of file dtssystemtypes.h.

#define NEWLINE_STR "\n\r"

Definition at line 93 of file dtssystemtypes.h.

#define NULL 0

Definition at line 446 of file dtssystemtypes.h.

#define PASCAL

Definition at line 411 of file dtssystemtypes.h.

#define RemoveDir rmdir

Definition at line 491 of file dtssystemtypes.h.

#define RenameFile rename

Definition at line 489 of file dtssystemtypes.h.

#define S_OK 0

Definition at line 458 of file dtssystemtypes.h.

#define SD_BOTH 2

Definition at line 475 of file dtssystemtypes.h.

#define Sleep(__millisec__) usleep((__millisec__)*1000)

Definition at line 486 of file dtssystemtypes.h.

#define SOCKET_ERROR -1

Definition at line 425 of file dtssystemtypes.h.

#define STATUS_PENDING (([DWORD](#))0x00000103L)

Definition at line 452 of file dtssystemtypes.h.

#define STDAPI

Definition at line 404 of file dtssystemtypes.h.

#define STDMETHODIMP [DWORD](#)

Definition at line 403 of file dtssystemtypes.h.

#define stricmp strcasecmp

Definition at line 484 of file dtssystemtypes.h.

#define strnicmp strncasecmp

Definition at line 485 of file dtssystemtypes.h.

#define TRACEKI(...)

Definition at line 21 of file dtssystemtypes.h.

#define TRUE 1

Definition at line 440 of file dtssystemtypes.h.

#define VOID void

Definition at line 409 of file dtssystemtypes.h.

#define VVWHANDLE void*

This is an internal channel identifier

Definition at line 747 of file dtssystemtypes.h.

#define WAIT_ABANDONED 0x00000080L

Definition at line 449 of file dtssystemtypes.h.

#define WAIT_OBJECT_0 0x00000000

Definition at line 448 of file dtssystemtypes.h.

#define WAIT_TIMEOUT 0x00000102

Definition at line 450 of file dtssystemtypes.h.

#define WCHAR short

Definition at line 183 of file dtssystemtypes.h.

#define WINAPI

Definition at line 405 of file dtssystemtypes.h.

#define vsprintf sprintf

Definition at line 480 of file dtssystemtypes.h.

#define xFFFFFFFFFFFFFFFF 0xFFFFFFFFFFFFFFFFLL

Definition at line 146 of file dtssystemtypes.h.

#define ZeroMemory(__where_, __howmuch_) ZeroMemoryFast(__where_, __howmuch_)

Definition at line 492 of file dtssystemtypes.h.

Typedef Documentation

typedef long long int [__int64](#)

Definition at line 359 of file dtssystemtypes.h.

typedef unsigned long long int [__uint64](#)

Definition at line 360 of file dtssystemtypes.h.

typedef char [_TCHAR](#)

Definition at line 174 of file dtssystemtypes.h.

typedef struct [tagBITMAPFILEHEADER](#) [BITMAPFILEHEADER](#)

typedef struct [tagBITMAPINFO](#) [BITMAPINFO](#)

typedef struct [tagBITMAPINFOHEADER](#) [BITMAPINFOHEADER](#)

typedef [LONG](#) [BOOL](#)

Definition at line 178 of file dtssystemtypes.h.

typedef [LONG BOOLEAN](#)

Definition at line 180 of file dtssystemtypes.h.

typedef unsigned char [BYTE](#)

Definition at line 170 of file dtssystemtypes.h.

typedef unsigned char [byte](#)

Definition at line 171 of file dtssystemtypes.h.

typedef char [CHAR](#)

Definition at line 172 of file dtssystemtypes.h.

typedef size_t [DTHANDLE](#)

Not compiling for Microsoft Windows 32

Definition at line 156 of file dtssystemtypes.h.

typedef uint32_t [DWORD](#)

Definition at line 162 of file dtssystemtypes.h.

typedef [ULONG_PTR](#) [DWORD_PTR](#)

Definition at line 196 of file dtssystemtypes.h.

typedef unsigned long long [DWORDLONG](#)

Definition at line 202 of file dtssystemtypes.h.

typedef [DWORD](#) [FOURCC](#)

Definition at line 165 of file dtssystemtypes.h.

typedef unsigned long [FourCharCode](#)

Definition at line 631 of file dtssystemtypes.h.

typedef struct [_GUID](#) [GUID](#)

typedef [__int64](#) [HANDLE](#)

Definition at line 376 of file dtssystemtypes.h.

typedef [Ptr*](#) [Handle](#)

Definition at line 618 of file dtssystemtypes.h.

typedef [size_t](#) [HBRUSH](#)

Definition at line 393 of file dtssystemtypes.h.

typedef [size_t](#) [HDC](#)

Definition at line 392 of file dtssystemtypes.h.

typedef [size_t](#) [HINSTANCE](#)

Definition at line 383 of file dtssystemtypes.h.

typedef [size_t](#) [HMODULE](#)

Definition at line 384 of file dtssystemtypes.h.

typedef [int32_t](#) [HRESULT](#)

Definition at line 389 of file dtssystemtypes.h.

typedef [size_t](#) [HWND](#)

Definition at line 385 of file dtssystemtypes.h.

typedef [LONG](#) [INT](#)

Definition at line 166 of file dtssystemtypes.h.

typedef union [_LARGE_INTEGER](#) [LARGE_INTEGER](#)

typedef [int32_t](#) [LONG](#)

Definition at line 164 of file dtssystemtypes.h.

typedef size_t [LONG_PTR](#)

Definition at line 194 of file dtssystemtypes.h.

typedef long long [LONGLONG](#)

Definition at line 203 of file dtssystemtypes.h.

typedef long [LPARAM](#)

Definition at line 364 of file dtssystemtypes.h.

typedef struct [tagBITMAPINFOHEADER](#) * [LPBITMAPINFOHEADER](#)

typedef [BYTE](#)* [LPBYTE](#)

Definition at line 190 of file dtssystemtypes.h.

typedef const char* [LPCSTR](#)

Definition at line 434 of file dtssystemtypes.h.

typedef const char* [LPCTSTR](#)

Definition at line 435 of file dtssystemtypes.h.

typedef const void* [LPCVOID](#)

Definition at line 436 of file dtssystemtypes.h.

typedef [DWORD](#)* [LPDWORD](#)

Definition at line 192 of file dtssystemtypes.h.

typedef struct [_GUID](#)* [LPGUID](#)

typedef [LONG](#)* [LPLONG](#)

Definition at line 193 of file dtssystemtypes.h.

typedef [POINT](#)* [LPPOINT](#)

Definition at line 259 of file dtssystemtypes.h.

typedef [RECT](#)* [LPRECT](#)

Definition at line 250 of file dtsystemtypes.h.

typedef [RGB](#)* [LPRGB](#)

Definition at line 279 of file dtsystemtypes.h.

typedef [SIZE](#)* [LPSIZE](#)

Definition at line 268 of file dtsystemtypes.h.

typedef struct sockaddr FAR* [LPSOCKADDR](#)

Definition at line 422 of file dtsystemtypes.h.

typedef struct sockaddr_in FAR* [LPSOCKADDR_IN](#)

Definition at line 419 of file dtsystemtypes.h.

typedef char* [LPSTR](#)

Definition at line 429 of file dtsystemtypes.h.

typedef void*(* [LPTHREAD_START_ROUTINE](#))(void *)

Definition at line 478 of file dtsystemtypes.h.

typedef char* [LPTSTR](#)

Definition at line 430 of file dtsystemtypes.h.

typedef void* [LPVOID](#)

Definition at line 431 of file dtsystemtypes.h.

typedef [WORD](#)* [LPWORD](#)

Definition at line 191 of file dtsystemtypes.h.

typedef long [LRESULT](#)

Definition at line 395 of file dtsystemtypes.h.

typedef [HRESULT](#) [MMRESULT](#)

Definition at line 391 of file dtsystemtypes.h.

typedef signed short [OSErr](#)

Definition at line 636 of file dtsystemtypes.h.

typedef signed long [OSStatus](#)

Definition at line 637 of file dtsystemtypes.h.

typedef [FourCharCode](#) [OSType](#)

Definition at line 632 of file dtsystemtypes.h.

typedef [OSType](#)* [OSTypePtr](#)

Definition at line 634 of file dtsystemtypes.h.

typedef size_t [PAINTSTRUCT](#)

Definition at line 394 of file dtsystemtypes.h.

typedef struct [tagBITMAPFILEHEADER](#) * [PBITMAPFILEHEADER](#)

typedef [BYTE](#)* [PBYTE](#)

Definition at line 185 of file dtsystemtypes.h.

typedef [DWORD](#)* [PDWORD](#)

Definition at line 188 of file dtsystemtypes.h.

typedef [ULONG_PTR](#) * [PDWORD_PTR](#)

Definition at line 196 of file dtsystemtypes.h.

typedef [LARGE_INTEGER](#)* [PLARGE_INTEGER](#)

Definition at line 226 of file dtsystemtypes.h.

typedef [LONG](#)* [PLONG](#)

Definition at line 189 of file dtssystemtypes.h.

typedef size_t * [PLONG_PTR](#)

Definition at line 194 of file dtssystemtypes.h.

typedef struct [_POINT](#) [POINT](#)

typedef [POINT](#)* [PPOINT](#)

Definition at line 258 of file dtssystemtypes.h.

typedef [RECT](#)* [PRECT](#)

Definition at line 249 of file dtssystemtypes.h.

typedef [RGB](#)* [PRGB](#)

Definition at line 278 of file dtssystemtypes.h.

typedef short* [PSHORT](#)

Definition at line 187 of file dtssystemtypes.h.

typedef [SIZE](#)* [PSIZE](#)

Definition at line 267 of file dtssystemtypes.h.

typedef struct sockaddr* [PSOCKADDR](#)

Definition at line 421 of file dtssystemtypes.h.

typedef struct sockaddr_in* [PSOCKADDR_IN](#)

Definition at line 418 of file dtssystemtypes.h.

typedef char* [PSTR](#)

Definition at line 426 of file dtssystemtypes.h.

typedef char* [Ptr](#)

Not compiling for Linux For newer AJA code, should already be defined in macOS Definition at line 617 of file dtssystemtypes.h.

typedef char* [PTSTR](#)

Definition at line 427 of file dtssystemtypes.h.

typedef size_t * [PULONG_PTR](#)

Definition at line 195 of file dtssystemtypes.h.

typedef void* [PVOID](#)

Definition at line 428 of file dtssystemtypes.h.

typedef struct [IWAVEFORMATEX](#) * [PWAVEFORMATEX](#)

typedef struct [WAVEFORMATEXTENSIBLE](#) * [PWAVEFORMATEXTENSIBLE](#)

typedef [WORD](#)* [PWORD](#)

Definition at line 186 of file dtssystemtypes.h.

typedef [DWORDLONG](#) [QUADWORD](#)

Definition at line 206 of file dtssystemtypes.h.

typedef [DWORDLONG](#) [QWORD](#)

Definition at line 205 of file dtssystemtypes.h.

typedef struct [_GUID](#)* [REFGUID](#)

typedef [FourCharCode](#) [ResType](#)

Definition at line 633 of file dtssystemtypes.h.

typedef [ResType](#)* [ResTypePtr](#)

Definition at line 635 of file dtssystemtypes.h.

typedef struct [_RGB](#) [RGB](#)

typedef struct [tagRGBQUAD](#) [RGBQUAD](#)

typedef short [SHORT](#)

Definition at line 168 of file dtssystemtypes.h.

typedef signed short [SInt16](#)

Definition at line 622 of file dtssystemtypes.h.

typedef signed long [SInt32](#)

Definition at line 369 of file dtssystemtypes.h.

typedef signed char [SInt8](#)

Definition at line 620 of file dtssystemtypes.h.

typedef long [Size](#)

Definition at line 619 of file dtssystemtypes.h.

typedef struct sockaddr [SOCKADDR](#)

Definition at line 420 of file dtssystemtypes.h.

typedef struct sockaddr_in [SOCKADDR_IN](#)

Definition at line 417 of file dtssystemtypes.h.

typedef int [SOCKET](#)

Definition at line 416 of file dtssystemtypes.h.

typedef char [TCHAR](#)

Definition at line 173 of file dtssystemtypes.h.

typedef unsigned char [UCHAR](#)

Definition at line 169 of file dtssystemtypes.h.

typedef unsigned int [UINT](#)

Definition at line 184 of file dtssystemtypes.h.

typedef unsigned short [UInt16](#)

Definition at line 621 of file dtssystemtypes.h.

typedef unsigned long [UInt32](#)

Definition at line 628 of file dtssystemtypes.h.

typedef size_t [ULONG_PTR](#)

Definition at line 195 of file dtssystemtypes.h.

typedef struct [tWAVEFORMATEX](#) [WAVEFORMATEX](#)

typedef unsigned short [WORD](#)

Definition at line 167 of file dtssystemtypes.h.

typedef [DWORD](#) [WPARAM](#)

Definition at line 363 of file dtssystemtypes.h.

E:/drastic/api/mediacmd/src/mediacmd.h File Reference

Classes

804 struct [MEDIACMD](#)

Defines

805 #define [MEDIACMD_VERSION_MAJOR_X32](#) 0x0101UL

Major command versioning for upgrades to the command set. See [MEDIACMD::dwCmdID](#).

806 #define [MEDIACMD_VERSION_MINOR_X32](#) 0x0003UL

Minor command versioning for upgrades to the command set. See [MEDIACMD::dwCmdID](#).

807 #define [MEDIACMD_VERSION_MAJOR](#) 0x0200UL

Major command versioning for upgrades to the command set. See [MEDIACMD::dwCmdID](#).

808 #define [MEDIACMD_VERSION_MINOR](#) 0x0000UL

Minor command versioning for upgrades to the command set. See [MEDIACMD::dwCmdID](#).

809 #define [MEDIACMD_VERSION_MASK](#) 0xFFFFUL

Mask for checking the command set version. See [MEDIACMD::dwCmdID](#).

810 #define [MEDIACMD_CHECK_VER](#) 0xFA250000UL

Permanent magic number of command id. See [MEDIACMD::dwCmdID](#).

811 #define [MEDIACMD_CHECK_MASK](#) 0xFFFF0000UL

Mask for permanent magic number of command id. See [MEDIACMD::dwCmdID](#).

812 #define [MEDIACMD_CURRENT](#) (MEDIACMD_VERSION_MAJOR |
MEDIACMD_VERSION_MINOR | MEDIACMD_CHECK_VER)

Current version and magic number. Place in [MEDIACMD::dwCmdID](#).

813 #define [SPD_FWD_PLAY](#) 65520L

Forward play speed (normal) in VVW (65520) see [MEDIACMD::lSpeed](#).

814 #define [SPD_PAUSE](#) 0L
Pause speed (0%) in VVW (0) see [MEDIACMD::lSpeed](#).

815 #define [SPD_REV_PLAY](#) (-SPD_FWD_PLAY)
Reverse play speed (-100%) in VVW (-65520) see [MEDIACMD::lSpeed](#).

816 #define [SPD_FWD_MAX](#) 5896800
Maximum possible play speed in VVW. See [MEDIACMD::lSpeed](#).

817 #define [SPD_REV_MAX](#) (-SPD_FWD_MAX)
Minimum possible play speed in VVW. See [MEDIACMD::lSpeed](#).

818 #define [SPD_FAST_BUMP](#) 114660L
Max speed for bumping.

819 #define [SPD_SLOW_BUMP](#) 32760L
Min Sped for bumping.

820 #define [SPD_ILLEGAL](#) 2147483647L
Illegal speed, set [MEDIACMD::lSpeed](#) to this value if not used.

821 #define [TC_ILLEGAL](#) 0xFFFFFFFF
Illegal time code reference, set [MEDIACMD::dwPosition](#), [MEDIACMD::dwStart](#), [MEDIACMD::dwEnd](#) to this if not used.

822 #define [CHAN_ILLEGAL](#) 0xFFFFFFFF
Illegal channel, or All Channels. Set [MEDIACMD::dwAudioChannels](#), [MEDIACMD::dwVideoChannels](#), [MEDIACMD::dwInfoChannels](#) to this if not used.

823 #define [CMD_MAX_CLIP_ID_LEN](#) (1024+8+4)

824 #define [cfOverrideDeferred](#) cfOverrideDeferred
Spelling.

825 #define [vidChanAll](#) 0xFFFFFFFFUL

826 #define [audChanAll](#) 0xFFFFFFFFUL

827 #define [infChanAll](#) 0xFFFFFFFFUL

828 #define [GS_TCSOURCE_LTC](#) 1
For cmdGetSetValue::gsTcSource - Using LTC.

829 #define [GS_TCSOURCE_VITC](#) 2
For cmdGetSetValue::gsTcSource - Using VITC.

830 #define [GS_TCSOURCE_CTL](#) 4
For cmdGetSetValue::gsTcSource - Using CTL.

831 #define [GS_TCSOURCE_CLIP](#) 7
For cmdGetSetValue::gsTcSource - Using absolute clip.

832 #define [GS_TCSOURCE_IRIG](#) 8
For cmdGetSetValue::gsTcSource - Using irig natural (not converted) time code.

833 #define [GS_FRAMEDATA_UNKNOWN](#) 0x00000
No data, unknown data for cmdGetSetValue::gsFrameData, cmdType::ctSetValue/cmdTypectGetValue.

834 #define [GS_FRAMEDATA_ASCII](#) 0x00001
ASCII data (all printable) for cmdGetSetValue::gsFrameData, cmdType::ctSetValue/cmdTypectGetValue.

835 #define [GS_FRAMEDATA_HEX](#) 0x00002
Binary (hex) data for cmdGetSetValue::gsFrameData, cmdType::ctSetValue/cmdTypectGetValue.

836 #define [GS_FRAMEDATA_TELECINE](#) 0x10001

*Telecine RP-215 / DPX Data for cmdGetSetValue::gsFrameData,
 cmdType::ctSetValue/cmdTypectGetValue.*

837 #define [GS_FRAMEDATA_CC_TTEXT](#) 0x10002
*Close caption/teletext for cmdGetSetValue::gsFrameData,
 cmdType::ctSetValue/cmdTypectGetValue.*

838 #define [GS_FRAMEDATA_NAVY](#) 0x10003
*Navy telemetry data for cmdGetSetValue::gsFrameData,
 cmdType::ctSetValue/cmdTypectGetValue.*

839 #define [GS_CC_DISABLE](#) 0x0000
No CC cmdGetSetValue::gsCCSetup.

840 #define [GS_CC_CC1](#) 0x0001
CC1 cmdGetSetValue::gsCCSetup.

841 #define [GS_CC_CC2](#) 0x0004
CC2 cmdGetSetValue::gsCCSetup.

842 #define [GS_CC_CC3](#) 0x0008
CC3 cmdGetSetValue::gsCCSetup.

843 #define [GS_CC_CC4](#) 0x0010
CC4 cmdGetSetValue::gsCCSetup.

844 #define [GS_CC_TEXT1](#) 0x0020
Text1 cmdGetSetValue::gsCCSetup.

845 #define [GS_CC_TEXT2](#) 0x0040
Text2 cmdGetSetValue::gsCCSetup.

846 #define [GS_CC_TEXT3](#) 0x0080
Text3 cmdGetSetValue::gsCCSetup.

847 #define [GS_CC_TEXT4](#) 0x0100
Text4 cmdGetSetValue::gsCCSetup.

848 #define [GS_CC_XDS](#) 0x0200
XDS cmdGetSetValue::gsCCSetup.

849 #define [GS_CC_708](#) 0x1000
CEA 708 cmdGetSetValue::gsCCSetup.

850 #define [GS_SOURCEPRECEDENCE_RP188_V](#) 0x00000001
RP-188 Ancillary time code video.

851 #define [GS_SOURCEPRECEDENCE_RP188_L](#) 0x00000002
RP-188 Ancillary time code audio.

852 #define [GS_SOURCEPRECEDENCE_SMPTE](#) 0x00000004
SMPTE LTC audio time code.

853 #define [GS_SOURCEPRECEDENCE_TOD](#) 0x00000008
Time of day from computer clock (or GPS if available)

854 #define [GS_SOURCEPRECEDENCE_VITC](#) 0x00000010
Time of day from VITC encoded line, or D-VITC in HD.

855 #define [GS_SOURCEPRECEDENCE_IRIG](#) 0x00000020
Time of day from IRIG RP-215 encode.

856 #define [GS_SOURCEPRECEDENCE_RP215](#) 0x00000040
Time of day from ILM style RP-215 A->LTC, V-VITC.

857 #define [GS_SOURCEPRECEDENCE_FRAMECOUNT](#) 0x10000000

Use the record frame count.

858 #define [GS_EDL_EFFECT](#) 0x00000001

Returns from `cmdGetSetValue::gsTcClipInfo` effect.

859 #define [GS_EDL_EFFECT_DUR](#) 0x00000002

Returns from `cmdGetSetValue::gsTcClipInfo` effect duration.

860 #define [GS_EDL_COMMENT](#) 0x00000004

Returns from `cmdGetSetValue::gsTcClipInfo` comment.

861 #define [GS_EDL_EDITNO](#) 0x00000008

Returns from `cmdGetSetValue::gsTcClipInfo` edit number.

862 #define [METABASE_TYPE_UNKNOWN](#) 0

Set `cmdGetSetValue::gsMetaDataReadWrite`.

863 #define [METABASE_TYPE_CHAR](#) 1

Set `cmdGetSetValue::gsMetaDataReadWrite`.

864 #define [METABASE_TYPE_INT](#) 2

Set `cmdGetSetValue::gsMetaDataReadWrite`.

865 #define [GS_INSERT_EDIT](#) 0x01

Set `cmdGetSetValue::gsEditMode` for an insert edit.

866 #define [GS_ASSEMBLE_EDIT](#) 0x02

Set `cmdGetSetValue::gsEditMode` for an assemble edit.

867 #define [GS_AUDSELECT_UNBALANCED_10](#) 0x001

Audio in/out unbalanced (RCA connector) high impedance at -10db
(`cmdGetSetValue::gsAudInSelect` `cmdGetSetValue::gsAudOutSelect`)

868 #define [GS_AUDSELECT_UNBALANCED_4](#) 0x002

Audio in/out unbalanced (RCA connector) high impedance at -4db
(`cmdGetSetValue::gsAudInSelect` `cmdGetSetValue::gsAudOutSelect`)

869 #define [GS_AUDSELECT_BALANCED_10](#) 0x010

Audio in/out balanced (XLR connector) 600ohm impedance at -10db
(`cmdGetSetValue::gsAudInSelect` `cmdGetSetValue::gsAudOutSelect`)

870 #define [GS_AUDSELECT_BALANCED_4](#) 0x020

Audio in/out balanced (XLR connector) 600ohm impedance at +4db
(`cmdGetSetValue::gsAudInSelect` `cmdGetSetValue::gsAudOutSelect`)

871 #define [GS_AUDSELECT_SPDIF](#) 0x100

Audio in/out digital single wire (`cmdGetSetValue::gsAudInSelect`
`cmdGetSetValue::gsAudOutSelect`)

872 #define [GS_AUDSELECT_AES_EBU](#) 0x200

Audio in/out digital balanced with clock (`cmdGetSetValue::gsAudInSelect`
`cmdGetSetValue::gsAudOutSelect`)

873 #define [GS_AUDSELECT_EMBEDDED](#) 0x400

Audio in/out embedded in SDI or HD-SDI video signal (`cmdGetSetValue::gsAudInSelect`
`cmdGetSetValue::gsAudOutSelect`)

874 #define [GS_AUDSELECT_AES_EBU_PRO](#) 0x800

Audio in/out digital balanced with clock (`cmdGetSetValue::gsAudInSelect`
`cmdGetSetValue::gsAudOutSelect`)

875 #define [GS_AUDSELECT_HDMI](#) 0x1000

Use audio embedded in the HDMI signal.

876 #define [GS_AUDSELECT_NONE](#) 0

No audio in/out available, or cannot be configured (cmdGetSetValue::gsAudInSelect cmdGetSetValue::gsAudOutSelect)

877 #define [GS_AUDSELECT_SILENT](#) 0x040

No Audio Selected leave silent.

878 #define [GS_AUD_BIT_RATE_32000](#) 0x0001

879 #define [GS_AUD_BIT_RATE_41100](#) 0x0002

880 #define [GS_AUD_BIT_RATE_48000](#) 0x0004

881 #define [GS_AUD_BIT_RATE_56000](#) 0x0008

882 #define [GS_AUD_BIT_RATE_64000](#) 0x0010

883 #define [GS_AUD_BIT_RATE_80000](#) 0x0020

884 #define [GS_AUD_BIT_RATE_96000](#) 0x0040

885 #define [GS_AUD_BIT_RATE_112000](#) 0x0080

886 #define [GS_AUD_BIT_RATE_128000](#) 0x0100

887 #define [GS_AUD_BIT_RATE_160000](#) 0x0200

888 #define [GS_AUD_BIT_RATE_192000](#) 0x0400

889 #define [GS_AUD_BIT_RATE_224000](#) 0x0800

890 #define [GS_AUD_BIT_RATE_256000](#) 0x1000

891 #define [GS_AUD_BIT_RATE_320000](#) 0x2000

892 #define [GS_AUD_BIT_RATE_384000](#) 0x4000

893 #define [GS_AUD_STEREO](#) 0x001

894 #define [GS_AUD_JOINT_STEREO](#) 0x002

895 #define [GS_AUD_DUAL](#) 0x004

896 #define [GS_AUD_SINGLE](#) 0x008

897 #define [GS_AUD_MULTIPLE](#) 0x010

898 #define [GS_AUD_HEADROOM_18](#) 0x01

899 #define [GS_AUD_HEADROOM_20](#) 0x02

900 #define [GS_VIDFREEZE_NOT_FROZEN](#) 0

Freeze - no freeze (cmdGetSetValue::gsVidFreeze)

901 #define [GS_VIDFREEZE_FIELD0](#) 1

Freeze - first (0) field (cmdGetSetValue::gsVidFreeze)

902 #define [GS_VIDFREEZE_FIELD1](#) 2

Freeze - second (1) field (cmdGetSetValue::gsVidFreeze)

903 #define [GS_VIDFREEZE_FRAME](#) 3

Freeze - both fields (cmdGetSetValue::gsVidFreeze)

904 #define [GS_VIDSELECT_COMPOSITE](#) 0x001

Standard NTSC or PAL composite video (cmdGetSetValue::gsVidInSelect cmdGetSetValue::gsVidOutSelect)

905 #define [GS_VIDSELECT_SVIDEO](#) 0x002

SVHS/S-Video four wire NTSC or PAL video (cmdGetSetValue::gsVidInSelect cmdGetSetValue::gsVidOutSelect)

906 #define [GS_VIDSELECT_COMPOSITE_2](#) 0x004

Secondary NTSC or PAL video (often monitor selection) (cmdGetSetValue::gsVidInSelect cmdGetSetValue::gsVidOutSelect)

907 #define [GS_VIDSELECT_COMPOSITE_3](#) 0x008

third NTSC or PAL video (often monitor selection) (cmdGetSetValue::gsVidInSelect cmdGetSetValue::gsVidOutSelect)

908 #define [GS_VIDSELECT_COMPONENT_YUV](#) 0x010

BetaCam level YCrCb NTSC or PAL video (cmdGetSetValue::gsVidInSelect cmdGetSetValue::gsVidOutSelect)

909 #define [GS_VIDSELECT_COMPONENT_YUV_M2](#) 0x020

Panasonic M2 level YCrCb NTSC or PAL video (cmdGetSetValue::gsVidInSelect cmdGetSetValue::gsVidOutSelect)
 910 #define [GS_VIDSELECT_COMPONENT_YUV_SMPTE](#) 0x040
SMPTE standard level YCrCb NTSC or PAL video (cmdGetSetValue::gsVidInSelect cmdGetSetValue::gsVidOutSelect)
 911 #define [GS_VIDSELECT_COMPONENT_RGB](#) 0x080
RGB at video standard rate (cmdGetSetValue::gsVidInSelect cmdGetSetValue::gsVidOutSelect)
 912 #define [GS_VIDSELECT_D1_SERIAL](#) 0x100
D1 Serial Digital or HDSDI video (cmdGetSetValue::gsVidInSelect cmdGetSetValue::gsVidOutSelect)
 913 #define [GS_VIDSELECT_D1_PARALLEL](#) 0x200
D1 Serial Parallel video (cmdGetSetValue::gsVidInSelect cmdGetSetValue::gsVidOutSelect)
 914 #define [GS_VIDSELECT_SDTI](#) 0x400
SDTI/SDI including high speed transfer video (cmdGetSetValue::gsVidInSelect cmdGetSetValue::gsVidOutSelect)
 915 #define [GS_VIDSELECT_COMPOSITE_4](#) 0x800
 916 #define [GS_VIDSELECT_SVIDEO_2](#) 0x1000
Secondary NTSC or PAL video (often monitor selection) (cmdGetSetValue::gsVidInSelect cmdGetSetValue::gsVidOutSelect)
 917 #define [GS_VIDSELECT_COMPONENT_YUV_2](#) 0x2000
Secondary NTSC or PAL video (often monitor selection) (cmdGetSetValue::gsVidInSelect cmdGetSetValue::gsVidOutSelect)
 918 #define [GS_VIDSELECT_D1_SERIAL_2](#) 0x4000
Secondary NTSC or PAL video (often monitor selection) (cmdGetSetValue::gsVidInSelect cmdGetSetValue::gsVidOutSelect)
 919 #define [GS_VIDSELECT_COMPOSITE_JAPAN](#) 0x8000
Standard NTSC or PAL composite video (cmdGetSetValue::gsVidInSelect cmdGetSetValue::gsVidOutSelect)
 920 #define [GS_VIDSELECT_SVIDEO_JAPAN](#) 0x10000
SVHS/S-Video four wire NTSC or PAL video (cmdGetSetValue::gsVidInSelect cmdGetSetValue::gsVidOutSelect)
 921 #define [GS_VIDSELECT_COMPONENT_YUV_JAPAN](#) 0x20000
BetaCam level YCrCb NTSC or PAL video (cmdGetSetValue::gsVidInSelect cmdGetSetValue::gsVidOutSelect)
 922 #define [GS_VIDSELECT_COMPONENT_YUV_SMPTE_JAPAN](#) 0x40000
SMPTE standard level YCrCb NTSC or PAL video (cmdGetSetValue::gsVidInSelect cmdGetSetValue::gsVidOutSelect)
 923 #define [GS_VIDSELECT_XVID_RGB](#) 0x80000
xVGA compatible analog RGB
 924 #define [GS_VIDSELECT_HDMI](#) 0x100000
HDMI - Auto YCbCr/RGB.
 925 #define [GS_VIDSELECT_HDMI_RGB](#) 0x200000
HDMI - RGB I/O.
 926 #define [GS_VIDSELECT_HDMI_YCBCR](#) 0x400000
HDMI - YCBCR I/O.

927 #define [GS_VIDSELECT_DVI](#) 0x800000
DVI Protocol.

928 #define [GS_VIDSELECT_3G_DUAL_RATE](#) 0x1000000
2 HDSDI YCbCr signals at once

929 #define [GS_VIDSELECT_3G_DUAL_LINK](#) 0x2000000
Dual link 4:4:4 over 1 cable.

930 #define [GS_VIDSELECT_NONE](#) 0
No video available or no configurable settings (cmdGetSetValue::gsVidInSelect cmdGetSetValue::gsVidOutSelect)

931 #define [GS_VIDLOCKTYPE_VTR](#) 1
VTR (unruly hsync) lock for cmdGetSetValue::gsVidInLockType cmdGetSetValue::gsVidOutLockType cmdGetSetValue::gsVidOutLockType.

932 #define [GS_VIDLOCKTYPE_BROADCAST](#) 2
Perfect lock for cmdGetSetValue::gsVidInLockType cmdGetSetValue::gsVidOutLockType cmdGetSetValue::gsVidOutLockType.

933 #define [GS_VIDBAND_STANDARD](#) 0x01
Allow normal bandwidth (gsGetSetValue::gsVidInBandwidth gsGetSetValue::gsVidBandwidth)

934 #define [GS_VIDBAND_MEDIUM](#) 0x02
Allow medium bandwidth (gsGetSetValue::gsVidInBandwidth gsGetSetValue::gsVidBandwidth)

935 #define [GS_VIDBAND_HIGH](#) 0x04
Allow high bandwidth (gsGetSetValue::gsVidInBandwidth gsGetSetValue::gsVidBandwidth)

936 #define [GS_VIDBAND_NOTCH](#) 0x08
Impose notch filter on bandwidth (gsGetSetValue::gsVidInBandwidth gsGetSetValue::gsVidBandwidth)

937 #define [GS_VIDBLACK_SETUP](#) 0x01
Black at normal level (7.5 IRE NTSC, 0 IRE PAL) gsGetSetValue::gsVidBlackSetup gsGetSetValue::gsVidInBlack.

938 #define [GS_VIDBLACK_CRYSTAL](#) 0x02
Crystal black level (0 IRE NTSC, 0 IRE PAL) gsGetSetValue::gsVidBlackSetup gsGetSetValue::gsVidInBlack.

939 #define [GS_VIDBLACK_SUPER](#) 0x04
Super black level (0 > IRE NTSC/PAL) gsGetSetValue::gsVidBlackSetup gsGetSetValue::gsVidInBlack.

940 #define [GS_VIDWHITE_CLAMP](#) 0x01
Whites are clamped or 100 IRE (gsGetSetValue::gsVidInWhite)

941 #define [GS_VIDWHITE_SCALE](#) 0x02
Whites are scaled automatically from black level to 100 IRE (gsGetSetValue::gsVidInWhite)

942 #define [GS_VIDWHITE_FREE](#) 0x04
Whites are allowed to be greater than 100 IRE (gsGetSetValue::gsVidInWhite)

943 #define [GS_LOCKSRC_NONE](#) 0x0001
No external genlock source (free running on internal clock) (gsGetSetValue::gsVidOutGenlockSource)

944 #define [GS_LOCKSRC_EXTIN](#) 0x0002
External ref in is genlock source (gsGetSetValue::gsVidOutGenlockSource)

945 #define [GS_LOCKSRC_INPUT](#) 0x0004
Current input is genlock source (gsGetSetValue::gsVidOutGenlockSource)

946 #define [GS_LOCKSRC_CVBS](#) 0x0008
Composite (CVBS) input is genlock source (gsGetSetValue::gsVidOutGenlockSource)

947 #define [GS_LOCKSRC_SVIDEO](#) 0x0010
S-Video (SVHS) input is genlock source (gsGetSetValue::gsVidOutGenlockSource)

948 #define [GS_LOCKSRC_IN_Y](#) 0x0020
Component Y input is genlock source (gsGetSetValue::gsVidOutGenlockSource)

949 #define [GS_LOCKSRC_SDI](#) 0x0040
SDI serial digital input is genlock source (gsGetSetValue::gsVidOutGenlockSource)

950 #define [GS_LOCKSRC_HDMI](#) 0x0080
HDMI genlock.

951 #define [GS_ANALOGMONITORMETHOD_DIRECT](#) 0x0001
Keep analog monitor in line with digital (HD=HD, SD=SD)

952 #define [GS_ANALOGMONITORMETHOD_SD](#) 0x0002
Convert everything to the nearest SD type.

953 #define [GS_ANALOGMONITORMETHOD_HD720](#) 0x0004
Convert everything to the nearest 720 HD type.

954 #define [GS_ANALOGMONITORMETHOD_HD1080](#) 0x0008
Convert everything to the nearest 1080 HD type.

955 #define [GS_ANALOGMONITORMETHOD_FLIP720](#) 0x0010
SD->HD720 and HD->SD.

956 #define [GS_ANALOGMONITORMETHOD_FLIP1080](#) 0x0020
SD->HD720 and HD->SD.

957 #define [GS_ANALOGMONITORMETHOD_HSDL](#) 0x0040
HD / SD -> HDSL.

958 #define [GS_UPCONVERT_ANAMORPHIC](#) 0x0001
Upconvert to whole screen.

959 #define [GS_UPCONVERT_PILLARBOX](#) 0x0002
Upconvert with bars.

960 #define [GS_UPCONVERT_ZOOM14x9](#) 0x0004
Upconvert with some zoom.

961 #define [GS_UPCONVERT_LETTERBOX](#) 0x0008
Upconvert to letter box.

962 #define [GS_UPCONVERT_ZOOMWIDE](#) 0x0010
Upconvert to wide zoom.

963 #define [GS_DOWNCONVERT_LETTERBOX](#) 0x0001
Down convert with top/bottom black bars.

964 #define [GS_DOWNCONVERT_CROP](#) 0x0002
Down convert and crop image.

965 #define [GS_DOWNCONVERT_ANAMORPHIC](#) 0x0004
Down convert to whole screen.

966 #define [GS_DOWNCONVERT_14x9](#) 0x0008
Down convert to 14x9.

967 #define [GS_MPEG_RESOLUTION_120](#) 0x001

Standard MPEG resolution 120.
 968 #define [GS_MPEG_RESOLUTION_240](#) 0x002
Standard MPEG resolution 240.
 969 #define [GS_MPEG_RESOLUTION_288](#) 0x004
Standard MPEG resolution 288.
 970 #define [GS_MPEG_RESOLUTION_352](#) 0x008
Standard MPEG resolution 352.
 971 #define [GS_MPEG_RESOLUTION_480](#) 0x010
Standard MPEG resolution 480.
 972 #define [GS_MPEG_RESOLUTION_512](#) 0x020
Standard MPEG resolution 512.
 973 #define [GS_MPEG_RESOLUTION_544](#) 0x040
Standard MPEG resolution 522.
 974 #define [GS_MPEG_RESOLUTION_576](#) 0x080
Standard MPEG resolution 576.
 975 #define [GS_MPEG_RESOLUTION_608](#) 0x100
Standard MPEG resolution 608.
 976 #define [GS_MPEG_RESOLUTION_704](#) 0x200
Standard MPEG resolution 704.
 977 #define [GS_MPEG_RESOLUTION_720](#) 0x400
Standard MPEG resolution 720.
 978 #define [GS_CHROMA_FORMAT_420](#) 0x1
Chroma format 4:2:0.
 979 #define [GS_CHROMA_FORMAT_422](#) 0x2
Chroma format 4:2:2.
 980 #define [GS_CHROMA_FORMAT_444](#) 0x4
Chroma format 4:4:4.
 981 #define [GS_CHROMA_FORMAT_411](#) 0x8
Chroma format 4:1:1.
 982 #define [GS_MPEG_CHROMA_FORMAT_420](#) GS_CHROMA_FORMAT_420
MPEG chroma format 4:2:0 see [GS_CHROMA_FORMAT_420](#).
 983 #define [GS_MPEG_CHROMA_FORMAT_422](#) GS_CHROMA_FORMAT_422
MPEG chroma format 4:2:2 see [GS_CHROMA_FORMAT_422](#).
 984 #define [GS_MPEG_CHROMA_FORMAT_444](#) GS_CHROMA_FORMAT_444
MPEG chroma format 4:4:4 see [GS_CHROMA_FORMAT_444](#).
 985 #define [GS_MPEG_DC_PRECISION_8](#) 0x1
MPEG DCT Precision 8 bits.
 986 #define [GS_MPEG_DC_PRECISION_9](#) 0x2
MPEG DCT Precision 9 bits.
 987 #define [GS_MPEG_DC_PRECISION_10](#) 0x4
MPEG DCT Precision 10 bits.
 988 #define [GS_MPEG_DC_PRECISION_11](#) 0x8
MPEG DCT Precision 11 bits.
 989 #define [GS_ASPECT_RATIO_SQUARE](#) 0x1
Aspect ratio square.

990 #define [GS_ASPECT_RATIO_4x3](#) 0x2
Aspect ratio 4:3.

991 #define [GS_ASPECT_RATIO_16x9](#) 0x4
Aspect ratio 16:9.

992 #define [GS_ASPECT_RATIO_2_21x1](#) 0x8
Aspect ratio 2.21:1.

993 #define [GS_MPEG_ASPECT_RATIO_SQUARE](#) GS_ASPECT_RATIO_SQUARE
MPEG aspect ratio square see [GS_ASPECT_RATIO_SQUARE](#).

994 #define [GS_MPEG_ASPECT_RATIO_4x3](#) GS_ASPECT_RATIO_4x3
MPEG aspect ratio square see [GS_ASPECT_RATIO_4x3](#).

995 #define [GS_MPEG_ASPECT_RATIO_16x9](#) GS_ASPECT_RATIO_16x9
MPEG aspect ratio square see [GS_ASPECT_RATIO_16x9](#).

996 #define [GS_MPEG_ASPECT_RATIO_2_21x1](#) GS_ASPECT_RATIO_2_21x1
MPEG aspect ratio square see [GS_ASPECT_RATIO_2_21x1](#).

997 #define [GS_MPEG_STANDARD_SYSTEM](#) 0x1

998 #define [GS_MPEG_STANDARD_PROGRAM](#) 0x2

999 #define [GS_MPEG_STANDARD_TRANSPORT](#) 0x4

1000 #define [GS_MPEG_STANDARD_ELEMENTARY](#) 0x8

1001 #define [GS_MPEG_STANDARD_ELEMENTARY](#) GS_MPEG_STANDARD_ELEMENTARY

1002 #define [GS_MPEG_LANGUAGE_ENGLISH](#) 0x0001

1003 #define [GS_MPEG_LANGUAGE_SPANISH](#) 0x0002

1004 #define [GS_MPEG_LANGUAGE_FRENCH](#) 0x0004

1005 #define [GS_MPEG_LANGUAGE_GERMAN](#) 0x0008

1006 #define [GS_MPEG_LANGUAGE_JAPANESE](#) 0x0010

1007 #define [GS_MPEG_LANGUAGE_DUTCH](#) 0x0020

1008 #define [GS_MPEG_LANGUAGE_DANISH](#) 0x0040

1009 #define [GS_MPEG_LANGUAGE_FINNISH](#) 0x0080

1010 #define [GS_MPEG_LANGUAGE_ITALIAN](#) 0x0100

1011 #define [GS_MPEG_LANGUAGE_GREEK](#) 0x0200

1012 #define [GS_MPEG_LANGUAGE_PORTUGUESE](#) 0x0400

1013 #define [GS_MPEG_LANGUAGE_SWEDISH](#) 0x0800

1014 #define [GS_MPEG_LANGUAGE_RUSSIAN](#) 0x1000

1015 #define [GS_MPEG_LANGUAGE_CHINESE](#) 0x2000

1016 #define [GS_MPEG_CC_FORMAT_CCUBE](#) 0x1

1017 #define [GS_MPEG_CC_FORMAT_ATSC](#) 0x2

1018 #define [GS_MPEG_CC_FORMAT_CCUBE_REORDER](#) 0x4

1019 #define [GS_MPEG_CC_FORMAT_ATSC_REORDER](#) 0x8

1020 #define [GS_MPEG_ONE_FRAMES](#) 0x0001

1021 #define [GS_MPEG_TWO_FRAMES](#) 0x0002

1022 #define [GS_MPEG_THREE_FRAMES](#) 0x0004

1023 #define [GS_MPEG_FOUR_FRAMES](#) 0x0008

1024 #define [GS_MPEG_FIVE_FRAMES](#) 0x0010

1025 #define [GS_MPEG_SIX_FRAMES](#) 0x0020

1026 #define [GS_MPEG_SEVEN_FRAMES](#) 0x0040

1027 #define [GS_MPEG_EIGHT_FRAMES](#) 0x0080

1028 #define [GS_MPEG_NINE_FRAMES](#) 0x0100

1029 #define [GS_MPEG_TEN_FRAMES](#) 0x0200

1030 #define [GS_MPEG_ELEVEN_FRAMES](#) 0x0400

1031 #define [GS_MPEG_TWELVE_FRAMES](#) 0x0800

1032 #define [GS_MPEG_THIRTEEN_FRAMES](#) 0x1000

1033 #define [GS_MPEG_FOURTEEN_FRAMES](#) 0x2000

1034 #define [GS_MPEG_FIFTEEN_FRAMES](#) 0x4000

1035#define [GS_MPEG_SIXTEEN_FRAMES](#) 0x8000
1036#define [VIDEOWRITETYPE_AVI](#) 0x00000001
Video for windows avi (audio video interleave)
1037#define [VIDEOWRITETYPE_MOV](#) 0x00000002
QuickTime movie (apple)
1038#define [VIDEOWRITETYPE_WMV](#) 0x00000004
Windows Media Video (Microsoft)
1039#define [VIDEOWRITETYPE_GEN](#) 0x00000008
SoftImage/Avid uncompressed GEN.
1040#define [VIDEOWRITETYPE_SONY_HD_MXF](#) 0x00000010
Jaleo uncompressed format.
1041#define [VIDEOWRITETYPE_SONY_SR_MXF](#) 0x00000020
Sony HDCAM SR MXF.
1042#define [VIDEOWRITETYPE_HDR](#) 0x00000080
Iridas 8 bit RGB format.
1043#define [VIDEOWRITETYPE_YUV](#) 0x00000100
Stills - 8/10 bit YCbCr .yuv or .v210.
1044#define [VIDEOWRITETYPE_RAW](#) 0x00000200
Stills - Raw 24/32 bit RGB/RGBA.
1045#define [VIDEOWRITETYPE_TGA](#) 0x00000400
Stills - Targa 24/32 bit RGB.
1046#define [VIDEOWRITETYPE_BMP](#) 0x00000800
Stills - no longer supported.
1047#define [VIDEOWRITETYPE_TIFF](#) 0x00001000
Stills - Tiff 24/32 bit RGB/RGBA.
1048#define [VIDEOWRITETYPE_AVCI_MXF](#) 0x00002000
MXF - Panasonic AVCi - Different P2 plugin.
1049#define [VIDEOWRITETYPE_DPX](#) 0x00004000
Stills - DPX (SMPTE/Kodak) 10 bit RGB.
1050#define [VIDEOWRITETYPE_MPG](#) 0x00008000
MPEG program or transport stream - Note: VVW send YCbCr 8 unc to/from board, compression done in MediaFile/PlugIn.
1051#define [VIDEOWRITETYPE_4224](#) 0x00010000
Stills - 4224 individual frames of 8 or 10 bit YCbCr+A.
1052#define [VIDEOWRITETYPE_SONY_MXF](#) 0x00020000
MXF - Sony XDCam SD.
1053#define [VIDEOWRITETYPE_P2_MXF](#) 0x00040000
MXF - Panasonic P2 DV25/50/100, AVCi.
1054#define [VIDEOWRITETYPE_AVID_MXF](#) 0x00080000
MXF - Avid DNxHD, Uncompressed, JPEG.
1055#define [VIDEOWRITETYPE_ARRI](#) 0x00100000
ARI - Raw Arri frame format.
1056#define [VIDEOWRITETYPE_JP2K](#) 0x00200000
Jp2 - Jpeg2000 Still frames.
1057#define [VIDEOWRITETYPE_OP1a_MXF](#) 0x00400000

MXF - Omneon AVCi, DVxx, MPEG.

1058#define [VIDEOWRITETYPE_DCP_MXF](#) 0x00800000
MXF - DCP XYZ or RGB JPEG-2000.

1059#define [VIDEOWRITETYPE_TS](#) 0x01000000
MPEG transport stream - Note: VVW send YCbCr 8 unc to/from board, compression done in MediaFile/PlugIn.

1060#define [VIDEOWRITETYPE_MP4](#) 0x02000000
MPEG program or transport stream - Note: VVW send YCbCr 8 unc to/from board, compression done in MediaFile/PlugIn.

1061#define [VIDEOWRITETYPE_FLASH](#) 0x04000000
Flash video (264+mp3) - Note: VVW send YCbCr 8 unc to/from board, compression done in MediaFile/PlugIn.

1062#define [VIDEOWRITETYPE_DNG](#) 0x08000000
DNG - Cinema DNG format.

1063#define [AUDIOWRITETYPE_STEREO](#) 0x00000001
Audio in stereo channels.

1064#define [AUDIOWRITETYPE_MONO](#) 0x00000002
Audio in mono channels.

1065#define [AUDIOWRITETYPE_MULTI](#) 0x00000004
Audio in multichannel.

1066#define [AUDIOWRITETYPE_INTERNAL](#) 0x00000000
Audio write type internal.

1067#define [AUDIOWRITETYPE_AIFF](#) 0x00000010
Audio write type aiff.

1068#define [AUDIOWRITETYPE_WAVE](#) 0x00000020
Audio write type wave.

1069#define [AUDIOWRITETYPE_WAVE_INTERNAL](#) (AUDIOWRITETYPE_WAVE|
 AUDIOWRITETYPE_INTERNAL)

1070#define [AUDIOWRITETYPE_WAVE_STEREO](#) (AUDIOWRITETYPE_WAVE|
 AUDIOWRITETYPE_STEREO)

1071#define [AUDIOWRITETYPE_WAVE_MONO](#) (AUDIOWRITETYPE_WAVE|
 AUDIOWRITETYPE_MONO)

1072#define [AUDIOWRITETYPE_WAVE_MULTI](#) (AUDIOWRITETYPE_WAVE|
 AUDIOWRITETYPE_MULTI)

1073#define [AUDIOWRITETYPE_AIFF_INTERNAL](#) (AUDIOWRITETYPE_AIFF|
 AUDIOWRITETYPE_INTERNAL)

1074#define [AUDIOWRITETYPE_AIFF_STEREO](#) (AUDIOWRITETYPE_AIFF|
 AUDIOWRITETYPE_STEREO)

1075#define [GS_MONITORGRAB_NONE](#) 0x0000
Turn off monitor.

1076#define [GS_MONITORGRAB_TYPE_MASK](#) 0x00F0
Type mask (jpg, bmp)

1077#define [GS_MONITORGRAB_TYPE_BMP](#) 0x0000
Use BMP format for image.

1078#define [GS_MONITORGRAB_TYPE_JPG](#) 0x0010
Use JPEG format for image.

1079#define [GS_MONITORGRAB_SIZE_MASK](#) 0x000F

Size mask (full, half, quarter)

1080#define [GS_MONITORGRAB_SIZE_FULL](#) 0x0001

Full size image captured.

1081#define [GS_MONITORGRAB_SIZE_HALF](#) 0x0002

Half size image captured.

1082#define [GS_MONITORGRAB_SIZE_QUARTER](#) 0x0004

Quarter size image captured.

1083#define [GS_MONITORGRAB_TARGET_MASK](#) 0x0F00

Target/To mask.

1084#define [GS_MONITORGRAB_TO_MEMORY](#) 0x0100

Use the arbID area.

1085#define [GS_MONITORGRAB_TO_UNC_PATH](#) 0x0200

Save image to a UNC path.

1086#define [GS_MONITORGRAB_TO_HTTP](#) 0x0400

Save image to web server (use name sent in arbID)

1087#define [GS_MONITORGRAB_TO_NETWORK](#) 0x0800

Save image through 'to be announced' network transport.

1088#define [GS_SIGFORMMASK_FRAMERATE](#) 0x000001ff

Frame rate mask (portion of return for frame rate)

1089#define [GS_SIGFORMSHIFT_FRAMERATE](#) 0

Shift frame rate to 0.

1090#define [GS_SIGFORMMASK_HORIZONTAL](#) 0x000ffe00

Horizontal / 8 mask (portion of return for frame rate)

1091#define [GS_SIGFORMSHIFT_HORIZONTAL](#) 9

Horizontal / 8 shift to 0.

1092#define [GS_SIGFORMMASK_VERTICAL](#) 0x0ff00000

Vertical / 8 mask (portion of return for frame rate)

1093#define [GS_SIGFORMSHIFT_VERTICAL](#) 20

Vertical / 8 shift to 0.

1094#define [GS_SIGFORMMASK_FRAMETYPE](#) 0xF0000000UL

Frame type mask (portion of return for frame rate)

1095#define [GS_SIGFORMSHIFT_FRAMETYPE](#) 28

Frame type shift to 0.

1096#define [GS_SIGFORMFRAMERATE_5](#) 5

1097#define [GS_SIGFORMFRAMERATE_6](#) 6

1098#define [GS_SIGFORMFRAMERATE_7_5](#) 7

1099#define [GS_SIGFORMFRAMERATE_10](#) 10

1100#define [GS_SIGFORMFRAMERATE_14_98](#) 14

1101#define [GS_SIGFORMFRAMERATE_15](#) 15

1102#define [GS_SIGFORMFRAMERATE_23_98](#) 23

1103#define [GS_SIGFORMFRAMERATE_24](#) 24

1104#define [GS_SIGFORMFRAMERATE_25](#) 25

1105#define [GS_SIGFORMFRAMERATE_29_97](#) 29

1106#define [GS_SIGFORMFRAMERATE_30](#) 30

1107#define [GS_SIGFORMFRAMERATE_47_95](#) 47

1108#define [GS_SIGFORMFRAMERATE_48](#) 48

1109#define [GS_SIGFORMFRAMERATE_50](#) 50

1110#define [GS_SIGFORMFRAMERATE_59_94](#) 59
1111#define [GS_SIGFORMFRAMERATE_60](#) 60
1112#define [GS_SIGFORMFRAMERATE_71_93](#) 71
1113#define [GS_SIGFORMFRAMERATE_72](#) 72
1114#define [GS_SIGFORMFRAMERATE_100](#) 100
1115#define [GS_SIGFORMFRAMERATE_119_88](#) 119
1116#define [GS_SIGFORMFRAMERATE_CUSTOM](#) 0x100
1117#define [GS_SIGFORMSIZE_240](#) 0x01
1118#define [GS_SIGFORMSIZE_243](#) 0x02
1119#define [GS_SIGFORMSIZE_288](#) 0x03
1120#define [GS_SIGFORMSIZE_320](#) 0x08
1121#define [GS_SIGFORMSIZE_352](#) 0x09
1122#define [GS_SIGFORMSIZE_360](#) 0x0a
1123#define [GS_SIGFORMSIZE_480](#) 0x10
1124#define [GS_SIGFORMSIZE_483](#) 0x11
1125#define [GS_SIGFORMSIZE_486](#) 0x12
1126#define [GS_SIGFORMSIZE_496](#) 0x14
1127#define [GS_SIGFORMSIZE_504](#) 0x16
1128#define [GS_SIGFORMSIZE_512](#) 0x17
1129#define [GS_SIGFORMSIZE_576](#) 0x1a
1130#define [GS_SIGFORMSIZE_600](#) 0x1b
1131#define [GS_SIGFORMSIZE_608](#) 0x1c
1132#define [GS_SIGFORMSIZE_640](#) 0x20
1133#define [GS_SIGFORMSIZE_720](#) 0x21
1134#define [GS_SIGFORMSIZE_768](#) 0x22
1135#define [GS_SIGFORMSIZE_800](#) 0x23
1136#define [GS_SIGFORMSIZE_864](#) 0x24
1137#define [GS_SIGFORMSIZE_988](#) 0x25
1138#define [GS_SIGFORMSIZE_857](#) 0x26
1139#define [GS_SIGFORMSIZE_960](#) 0x28
1140#define [GS_SIGFORMSIZE_968](#) 0x29
1141#define [GS_SIGFORMSIZE_778](#) 0x2A
1142#define [GS_SIGFORMSIZE_872](#) 0x2B
1143#define [GS_SIGFORMSIZE_1024](#) 0x30
1144#define [GS_SIGFORMSIZE_1035](#) 0x31
1145#define [GS_SIGFORMSIZE_1044](#) 0x32
1146#define [GS_SIGFORMSIZE_1052](#) 0x33
1147#define [GS_SIGFORMSIZE_1050](#) 0x34
1148#define [GS_SIGFORMSIZE_1080](#) 0x38
1149#define [GS_SIGFORMSIZE_1088](#) 0x39
1150#define [GS_SIGFORMSIZE_1096](#) 0x3a
1151#define [GS_SIGFORMSIZE_1102](#) 0x3E
1152#define [GS_SIGFORMSIZE_1152](#) 0x40
1153#define [GS_SIGFORMSIZE_1200](#) 0x41
1154#define [GS_SIGFORMSIZE_1234](#) 0x43
1155#define [GS_SIGFORMSIZE_1280](#) 0x48
1156#define [GS_SIGFORMSIZE_1332](#) 0x49
1157#define [GS_SIGFORMSIZE_1400](#) 0x4B
1158#define [GS_SIGFORMSIZE_1440](#) 0x4C
1159#define [GS_SIGFORMSIZE_1536](#) 0x50
1160#define [GS_SIGFORMSIZE_1556](#) 0x51
1161#define [GS_SIGFORMSIZE_1588](#) 0x52
1162#define [GS_SIGFORMSIZE_1828](#) 0x56
1163#define [GS_SIGFORMSIZE_1714](#) 0x57
1164#define [GS_SIGFORMSIZE_1600](#) 0x58
1165#define [GS_SIGFORMSIZE_1920](#) 0x59

```

1166#define GS_SIGFORMSIZE_1782 0x5A
1167#define GS_SIGFORMSIZE_2048 0x60
1168#define GS_SIGFORMSIZE_2160 0x64
1169#define GS_SIGFORMSIZE_2650 0x68
1170#define GS_SIGFORMSIZE_2880 0x6A
1171#define GS_SIGFORMSIZE_3112 0x6b
1172#define GS_SIGFORMSIZE_3840 0x78
1173#define GS_SIGFORMSIZE_4096 0x80
1174#define GS_SIGFORMSIZE_640x480 ((GS_SIGFORMSIZE_640 <<
    GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_480 <<
    GS_SIGFORMSHIFT_VERTICAL))
1175#define GS_SIGFORMSIZE_640x576 ((GS_SIGFORMSIZE_640 <<
    GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_576 <<
    GS_SIGFORMSHIFT_VERTICAL))
1176#define GS_SIGFORMSIZE_720x480 ((GS_SIGFORMSIZE_720 <<
    GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_480 <<
    GS_SIGFORMSHIFT_VERTICAL))
1177#define GS_SIGFORMSIZE_720x483 ((GS_SIGFORMSIZE_720 <<
    GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_483 <<
    GS_SIGFORMSHIFT_VERTICAL))
1178#define GS_SIGFORMSIZE_720x486 ((GS_SIGFORMSIZE_720 <<
    GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_486 <<
    GS_SIGFORMSHIFT_VERTICAL))
1179#define GS_SIGFORMSIZE_720x512 ((GS_SIGFORMSIZE_720 <<
    GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_512 <<
    GS_SIGFORMSHIFT_VERTICAL))
1180#define GS_SIGFORMSIZE_720x576 ((GS_SIGFORMSIZE_720 <<
    GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_576 <<
    GS_SIGFORMSHIFT_VERTICAL))
1181#define GS_SIGFORMSIZE_720x608 ((GS_SIGFORMSIZE_720 <<
    GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_608 <<
    GS_SIGFORMSHIFT_VERTICAL))
1182#define GS_SIGFORMSIZE_720x504 ((GS_SIGFORMSIZE_720 <<
    GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_504 <<
    GS_SIGFORMSHIFT_VERTICAL))
1183#define GS_SIGFORMSIZE_800x600 ((GS_SIGFORMSIZE_800 <<
    GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_600 <<
    GS_SIGFORMSHIFT_VERTICAL))
1184#define GS_SIGFORMSIZE_960x486 ((GS_SIGFORMSIZE_960 <<
    GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_486 <<
    GS_SIGFORMSHIFT_VERTICAL))
1185#define GS_SIGFORMSIZE_960x576 ((GS_SIGFORMSIZE_960 <<
    GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_576 <<
    GS_SIGFORMSHIFT_VERTICAL))
1186#define GS_SIGFORMSIZE_960x504 ((GS_SIGFORMSIZE_960 <<
    GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_504 <<
    GS_SIGFORMSHIFT_VERTICAL))
1187#define GS_SIGFORMSIZE_1024x768 ((GS_SIGFORMSIZE_1024 <<
    GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_768 <<
    GS_SIGFORMSHIFT_VERTICAL))
1188#define GS_SIGFORMSIZE_1024x1024 ((GS_SIGFORMSIZE_1024 <<
    GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_1024 <<
    GS_SIGFORMSHIFT_VERTICAL))
1189#define GS_SIGFORMSIZE_1152x864 ((GS_SIGFORMSIZE_1152 <<
    GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_864 <<
    GS_SIGFORMSHIFT_VERTICAL))

```

```

1190#define GS\_SIGFORMSIZE\_1280x1024 ((GS_SIGFORMSIZE_1280 <<
    GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_1024 <<
    GS_SIGFORMSHIFT_VERTICAL))
1191#define GS\_SIGFORMSIZE\_1400x1050 ((GS_SIGFORMSIZE_1400 <<
    GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_1050 <<
    GS_SIGFORMSHIFT_VERTICAL))
1192#define GS\_SIGFORMSIZE\_1600x1200 ((GS_SIGFORMSIZE_1600 <<
    GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_1200 <<
    GS_SIGFORMSHIFT_VERTICAL))
1193#define GS\_SIGFORMSIZE\_1280x720 ((GS_SIGFORMSIZE_1280 <<
    GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_720 <<
    GS_SIGFORMSHIFT_VERTICAL))
1194#define GS\_SIGFORMSIZE\_1828x778 ((GS_SIGFORMSIZE_1828 <<
    GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_778 <<
    GS_SIGFORMSHIFT_VERTICAL))
1195#define GS\_SIGFORMSIZE\_1828x988 ((GS_SIGFORMSIZE_1828 <<
    GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_988 <<
    GS_SIGFORMSHIFT_VERTICAL))
1196#define GS\_SIGFORMSIZE\_1828x1102 ((GS_SIGFORMSIZE_1828 <<
    GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_1102 <<
    GS_SIGFORMSHIFT_VERTICAL))
1197#define GS\_SIGFORMSIZE\_1828x1332 ((GS_SIGFORMSIZE_1828 <<
    GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_1332 <<
    GS_SIGFORMSHIFT_VERTICAL))
1198#define GS\_SIGFORMSIZE\_1920x1035 ((GS_SIGFORMSIZE_1920 <<
    GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_1035 <<
    GS_SIGFORMSHIFT_VERTICAL))
1199#define GS\_SIGFORMSIZE\_1920x1080 ((GS_SIGFORMSIZE_1920 <<
    GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_1080 <<
    GS_SIGFORMSHIFT_VERTICAL))
1200#define GS\_SIGFORMSIZE\_1920x1088 ((GS_SIGFORMSIZE_1920 <<
    GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_1088 <<
    GS_SIGFORMSHIFT_VERTICAL))
1201#define GS\_SIGFORMSIZE\_2560x1080 ((GS_SIGFORMSIZE_2560 <<
    GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_1080 <<
    GS_SIGFORMSHIFT_VERTICAL))
1202#define GS\_SIGFORMSIZE\_2048x857 ((GS_SIGFORMSIZE_2048 <<
    GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_857 <<
    GS_SIGFORMSHIFT_VERTICAL))
1203#define GS\_SIGFORMSIZE\_2048x872 ((GS_SIGFORMSIZE_2048 <<
    GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_872 <<
    GS_SIGFORMSHIFT_VERTICAL))
1204#define GS\_SIGFORMSIZE\_2048x1102 ((GS_SIGFORMSIZE_2048 <<
    GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_1102 <<
    GS_SIGFORMSHIFT_VERTICAL))
1205#define GS\_SIGFORMSIZE\_2048x1234 ((GS_SIGFORMSIZE_2048 <<
    GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_1234 <<
    GS_SIGFORMSHIFT_VERTICAL))
1206#define GS\_SIGFORMSIZE\_2048x1080 ((GS_SIGFORMSIZE_2048 <<
    GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_1080 <<
    GS_SIGFORMSHIFT_VERTICAL))
1207#define GS\_SIGFORMSIZE\_2048x1536 ((GS_SIGFORMSIZE_2048 <<
    GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_1536 <<
    GS_SIGFORMSHIFT_VERTICAL))
1208#define GS\_SIGFORMSIZE\_2048x1556 ((GS_SIGFORMSIZE_2048 <<
    GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_1556 <<

```

```

    GS_SIGFORMSHIFT_VERTICAL))
1209#define GS\_SIGFORMSIZE\_4096x1714 ((GS_SIGFORMSIZE_4096 <<
    GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_1714 <<
    GS_SIGFORMSHIFT_VERTICAL))
1210#define GS\_SIGFORMSIZE\_3840x2880 ((GS_SIGFORMSIZE_3840 <<
    GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_2880 <<
    GS_SIGFORMSHIFT_VERTICAL))
1211#define GS\_SIGFORMSIZE\_4096x2880 ((GS_SIGFORMSIZE_4096 <<
    GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_2880 <<
    GS_SIGFORMSHIFT_VERTICAL))
1212#define GS\_SIGFORMSIZE\_4096x3112 ((GS_SIGFORMSIZE_4096 <<
    GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_3112 <<
    GS_SIGFORMSHIFT_VERTICAL))
1213#define GS\_SIGFORMSIZE\_2880x2160 ((GS_SIGFORMSIZE_2880 <<
    GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_2160 <<
    GS_SIGFORMSHIFT_VERTICAL))
1214#define GS\_SIGFORMSIZE\_2880x1782 ((GS_SIGFORMSIZE_2880 <<
    GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_1782 <<
    GS_SIGFORMSHIFT_VERTICAL))
1215#define GS\_SIGFORMTYPE\_UNKNOWN (0)
1216#define GS\_SIGFORMTYPE\_INTERLACED (1 << GS_SIGFORMSHIFT_FRAMETYPE)
1217#define GS\_SIGFORMTYPE\_PROGRESSIVE (2 << GS_SIGFORMSHIFT_FRAMETYPE)
1218#define GS\_SIGFORMTYPE\_SEGMENTEDFRAME (4 <<
    GS_SIGFORMSHIFT_FRAMETYPE)
1219#define GS\_SIGFORM\_NTSC (GS_SIGFORMSIZE_640x480 |
    GS_SIGFORMTYPE_INTERLACED | GS_SIGFORMFRAMERATE_29_97)
Signal format NTSC square pixel (320x240 or 640x480) @ 29.97 or 30 fps
gsGetSetValue::gsSignalFormat.
1220#define GS\_SIGFORM\_PAL (GS_SIGFORMSIZE_640x576 |
    GS_SIGFORMTYPE_INTERLACED | GS_SIGFORMFRAMERATE_25)
Signal format PAL square pixel (320x288 or 640x576) @ 25 fps
gsGetSetValue::gsSignalFormat.
1221#define GS\_SIGFORM\_CCIR\_NTSC (GS_SIGFORMSIZE_720x486 |
    GS_SIGFORMTYPE_INTERLACED | GS_SIGFORMFRAMERATE_29_97)
Signal format NTSC square pixel (360/352x243/240 or 720/704x486/480) @ 29.97 or 30 fps
gsGetSetValue::gsSignalFormat.
1222#define GS\_SIGFORM\_CCIR\_NTSC\_P483 (GS_SIGFORMSIZE_720x483 |
    GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_29_97)
Signal format NTSC square pixel (360/352x243/240 or 720/704x486/480) @ 29.97 or 30 fps
gsGetSetValue::gsSignalFormat.
1223#define GS\_SIGFORM\_CCIR\_PAL (GS_SIGFORMSIZE_720x576 |
    GS_SIGFORMTYPE_INTERLACED | GS_SIGFORMFRAMERATE_25)
Signal format PAL square pixel (360/352x288 or 720/704x576) @ 25 fps
gsGetSetValue::gsSignalFormat.
1224#define GS\_SIGFORM\_CCIR\_PNTSC\_30 (GS_SIGFORMSIZE_720x486 |
    GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_30)
Signal format NTSC at 30 hz Progressive.
1225#define GS\_SIGFORM\_CCIR\_PPAL\_25 (GS_SIGFORMSIZE_720x576 |
    GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_25)
Signal format PAL at 25 hz Progressive.
1226#define GS\_SIGFORM\_CCIR\_NTSC2398 (GS_SIGFORMSIZE_720x486 |
    GS_SIGFORMTYPE_INTERLACED | GS_SIGFORMFRAMERATE_23_98)

```


Signal format NTSC 23.98.

```
1227#define GS\_SIGFORM\_HD360 (GS_SIGFORMSIZE_960x504 |  
    GS_SIGFORMTYPE_INTERLACED | GS_SIGFORMFRAMERATE_29_97)
```

Signal format compressed HD 960x504 29.97.

```
1228#define GS\_SIGFORM\_ALT\_NTSC (GS_SIGFORMSIZE_960x486 |  
    GS_SIGFORMTYPE_INTERLACED | GS_SIGFORMFRAMERATE_29_97)
```

Signal format NTSC High Res (960x486)

```
1229#define GS\_SIGFORM\_ALT\_PAL (GS_SIGFORMSIZE_960x576 |  
    GS_SIGFORMTYPE_INTERLACED | GS_SIGFORMFRAMERATE_25)
```

Signal format PAL High Res (960x576)

```
1230#define GS\_SIGFORM\_1035i\_30\_260M (GS_SIGFORMSIZE_1920x1035 |  
    GS_SIGFORMTYPE_INTERLACED | GS_SIGFORMFRAMERATE_30)
```

*2200x1125 raster, 1920x1035 production aperture (1888x1017 clean) @ 30 fps
gsGetSetValue::gsSignalFormat*

```
1231#define GS\_SIGFORM\_1035i\_30X\_260M (GS_SIGFORMSIZE_1920x1035 |  
    GS_SIGFORMTYPE_INTERLACED | GS_SIGFORMFRAMERATE_29_97)
```

*2200x1125 raster, 1920x1035 production aperture (1888x1017 clean) @ 29.97 fp
gsGetSetValue::gsSignalFormats*

```
1232#define GS\_SIGFORM\_1080i\_30 (GS_SIGFORMSIZE_1920x1080 |  
    GS_SIGFORMTYPE_INTERLACED | GS_SIGFORMFRAMERATE_30)
```

1920x1080i (274M-1997 Table1 System 4) @ 29.97 gsGetSetValue::gsSignalFormat

```
1233#define GS\_SIGFORM\_1080sf\_30 (GS_SIGFORMSIZE_1920x1080 |  
    GS_SIGFORMTYPE_SEGMENTEDFRAME | GS_SIGFORMFRAMERATE_30)
```

```
1234#define GS\_SIGFORM\_1080i\_30X (GS_SIGFORMSIZE_1920x1080 |  
    GS_SIGFORMTYPE_INTERLACED | GS_SIGFORMFRAMERATE_29_97)
```

1920x1080i (274M-1997 Table1 System 4) @ 30 gsGetSetValue::gsSignalFormat

```
1235#define GS\_SIGFORM\_1080sf\_30X (GS_SIGFORMSIZE_1920x1080 |  
    GS_SIGFORMTYPE_SEGMENTEDFRAME | GS_SIGFORMFRAMERATE_29_97)
```

```
1236#define GS\_SIGFORM\_1080i\_25 (GS_SIGFORMSIZE_1920x1080 |  
    GS_SIGFORMTYPE_INTERLACED | GS_SIGFORMFRAMERATE_25)
```

1920x1080i (274M-1997 Table1 System 4) @ 25 gsGetSetValue::gsSignalFormat

```
1237#define GS\_SIGFORM\_1080sf\_25 (GS_SIGFORMSIZE_1920x1080 |  
    GS_SIGFORMTYPE_SEGMENTEDFRAME | GS_SIGFORMFRAMERATE_25)
```

```
1238#define GS\_SIGFORM\_1080i\_24 (GS_SIGFORMSIZE_1920x1080 |  
    GS_SIGFORMTYPE_INTERLACED | GS_SIGFORMFRAMERATE_24)
```

1920x1080sf (274M-1997 Table1 System 4) @ 24 gsGetSetValue::gsSignalFormat

```
1239#define GS\_SIGFORM\_1080sf\_24 (GS_SIGFORMSIZE_1920x1080 |  
    GS_SIGFORMTYPE_SEGMENTEDFRAME | GS_SIGFORMFRAMERATE_24)
```

```
1240#define GS\_SIGFORM\_1080i\_24X (GS_SIGFORMSIZE_1920x1080 |  
    GS_SIGFORMTYPE_INTERLACED | GS_SIGFORMFRAMERATE_23_98)
```

1920x1080sf (274M-1997 Table1 System 4) @ 23.98 gsGetSetValue::gsSignalFormat

```
1241#define GS\_SIGFORM\_1080sf\_24X (GS_SIGFORMSIZE_1920x1080 |  
    GS_SIGFORMTYPE_SEGMENTEDFRAME | GS_SIGFORMFRAMERATE_23_98)
```

```
1242#define GS\_SIGFORM\_1080\_30 (GS_SIGFORMSIZE_1920x1080 |  
    GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_30)
```

1920x1080P (274M-1997 Table1 System 4) @ 30 gsGetSetValue::gsSignalFormat

```
1243#define GS\_SIGFORM\_1080\_30X (GS_SIGFORMSIZE_1920x1080 |  
    GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_29_97)
```

1920x1080P (274M-1997 Table1 System 4) @ 29.97 gsGetSetValue::gsSignalFormat

```

1244#define GS\_SIGFORM\_1080\_25 (GS_SIGFORMSIZE_1920x1080 |
    GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_25)
1920x1080P (274M-1997 Table1 System 4) @ 25 gsGetSetValue::gsSignalFormat
1245#define GS\_SIGFORM\_1080\_24 (GS_SIGFORMSIZE_1920x1080 |
    GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_24)
1920x1080P (274M-1997 Table1 System 4) @ 24 gsGetSetValue::gsSignalFormat
1246#define GS\_SIGFORM\_1080\_24X (GS_SIGFORMSIZE_1920x1080 |
    GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_23_98)
1920x1080P (274M-1997 Table1 System 4) @ 23.98 gsGetSetValue::gsSignalFormat
1247#define GS\_SIGFORM\_1080\_60 (GS_SIGFORMSIZE_1920x1080 |
    GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_60)
1920x1080P 60 (Dual P30)
1248#define GS\_SIGFORM\_1080\_60X (GS_SIGFORMSIZE_1920x1080 |
    GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_59_94)
1920x1080P 59.94 (Dual P29.97)
1249#define GS\_SIGFORM\_1080\_50 (GS_SIGFORMSIZE_1920x1080 |
    GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_50)
1920x1080P 50 (Dual 25)
1250#define GS\_SIGFORM\_1080\_48 (GS_SIGFORMSIZE_1920x1080 |
    GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_48)
1920x1080P 48 (Dual 24)
1251#define GS\_SIGFORM\_1080\_48X (GS_SIGFORMSIZE_1920x1080 |
    GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_47_95)
1920x1080P 47.95 (Dual 23.98)
1252#define GS\_SIGFORM\_720\_60 (GS_SIGFORMSIZE_1280x720 |
    GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_60)
1650x750 raster, 1280x720 production aperture (1248x702 clean): @ 60
gsGetSetValue::gsSignalFormat
1253#define GS\_SIGFORM\_720\_60X (GS_SIGFORMSIZE_1280x720 |
    GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_59_94)
1650x750 raster, 1280x720 production aperture (1248x702 clean): @ 59.97
gsGetSetValue::gsSignalFormat
1254#define GS\_SIGFORM\_720\_50 (GS_SIGFORMSIZE_1280x720 |
    GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_50)
50 Hz DVS, IRT
1255#define GS\_SIGFORM\_720\_30 (GS_SIGFORMSIZE_1280x720 |
    GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_30)
Half frame rate 720/60.
1256#define GS\_SIGFORM\_720\_30X (GS_SIGFORMSIZE_1280x720 |
    GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_29_97)
Half frame rate 720/59.94.
1257#define GS\_SIGFORM\_720\_25 (GS_SIGFORMSIZE_1280x720 |
    GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_25)
Half 50 Hz DVS, IRT.
1258#define GS\_SIGFORM\_720\_24 (GS_SIGFORMSIZE_1280x720 |
    GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_24)
720x1280 true 24 (Varicam)
1259#define GS\_SIGFORM\_VESA\_640\_72 (GS_SIGFORMSIZE_640x480 |

```

```

    GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_72)
gsGetSetValue::gsSignalFormat Vesa 640x480@72
1260#define GS_SIGFORM_VESA_800_71X (GS_SIGFORMSIZE_800x600 |
    GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_71_93)
gsGetSetValue::gsSignalFormat Vesa 800x600@71.9
1261#define GS_SIGFORM_VESA_800_72 (GS_SIGFORMSIZE_800x600 |
    GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_72)
gsGetSetValue::gsSignalFormat Vesa 800x600@72
1262#define GS_SIGFORM_VESA_1024_71X (GS_SIGFORMSIZE_1024x768 |
    GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_71_93)
gsGetSetValue::gsSignalFormat Vesa 1024x768@71.9
1263#define GS_SIGFORM_VESA_1024_72 (GS_SIGFORMSIZE_1024x768 |
    GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_72)
gsGetSetValue::gsSignalFormat Vesa 1024x766@72
1264#define GS_SIGFORM_VESA_1280_24 (GS_SIGFORMSIZE_1280x1024 |
    GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_24)
gsGetSetValue::gsSignalFormat Vesa 1280x1024@24
1265#define GS_SIGFORM_VESA_1280i_30 (GS_SIGFORMSIZE_1280x1024 |
    GS_SIGFORMTYPE_INTERLACED | GS_SIGFORMFRAMERATE_30)
gsGetSetValue::gsSignalFormat Vesa 1280x1024@30
1266#define GS_SIGFORM_VESA_1280_71X (GS_SIGFORMSIZE_1280x1024 |
    GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_71_93)
gsGetSetValue::gsSignalFormat Vesa 1280x1024@71.9
1267#define GS_SIGFORM_VESA_1280_72 (GS_SIGFORMSIZE_1280x1024 |
    GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_72)
gsGetSetValue::gsSignalFormat Vesa 1280x1024@72
1268#define GS_SIGFORM_VESA_1600i_30 (GS_SIGFORMSIZE_1600x1200 |
    GS_SIGFORMTYPE_INTERLACED | GS_SIGFORMFRAMERATE_30)
gsGetSetValue::gsSignalFormat Vesa 1600x1200i@30
1269#define GS_SIGFORM_DVI_1400_1050_24 (GS_SIGFORMSIZE_1400x1050 |
    GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_24)
gsGetSetValue::gsSignalFormat Presentation
1270#define GS_SIGFORM_DVI_1400_1050_25 (GS_SIGFORMSIZE_1400x1050 |
    GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_25)
gsGetSetValue::gsSignalFormat Presentation
1271#define GS_SIGFORM_DCIN_2048_25 (GS_SIGFORMSIZE_2048x1080 |
    GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_25)
gsGetSetValue::gsSignalFormat Presentation
1272#define GS_SIGFORM_DCIN_2048sf_25 (GS_SIGFORMSIZE_2048x1080 |
    GS_SIGFORMTYPE_SEGMENTEDFRAME | GS_SIGFORMFRAMERATE_25)
1273#define GS_SIGFORM_DCIN_2048sf_24X (GS_SIGFORMSIZE_2048x1080 |
    GS_SIGFORMTYPE_SEGMENTEDFRAME | GS_SIGFORMFRAMERATE_23_98)
gsGetSetValue::gsSignalFormat Digital Cinema
1274#define GS_SIGFORM_DCIN_2048sf_24 (GS_SIGFORMSIZE_2048x1080 |
    GS_SIGFORMTYPE_SEGMENTEDFRAME | GS_SIGFORMFRAMERATE_24)
gsGetSetValue::gsSignalFormat Digital Cinema
1275#define GS_SIGFORM_DCIN_2048_24X (GS_SIGFORMSIZE_2048x1080 |
    GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_23_98)

```

gsGetSetValue::gsSignalFormat Digital Cinema

1276#define [GS_SIGFORM_DCIN_2048_24](#) (GS_SIGFORMSIZE_2048x1080 |
GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_24)

gsGetSetValue::gsSignalFormat Digital Cinema

1277#define [GS_SIGFORM_FILM_1828_778_24](#) (GS_SIGFORMSIZE_1828x778 |
GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_24)

gsGetSetValue::gsSignalFormat Film

1278#define [GS_SIGFORM_FILM_1828_778_25](#) (GS_SIGFORMSIZE_1828x778 |
GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_25)

gsGetSetValue::gsSignalFormat Film

1279#define [GS_SIGFORM_FILM_1828_988_24](#) (GS_SIGFORMSIZE_1828x988 |
GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_24)

gsGetSetValue::gsSignalFormat Film

1280#define [GS_SIGFORM_FILM_1828_988_25](#) (GS_SIGFORMSIZE_1828x988 |
GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_25)

gsGetSetValue::gsSignalFormat Film

1281#define [GS_SIGFORM_FILM_1828_1102_24](#) (GS_SIGFORMSIZE_1828x1102 |
GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_24)

gsGetSetValue::gsSignalFormat Film

1282#define [GS_SIGFORM_FILM_1828_1102_25](#) (GS_SIGFORMSIZE_1828x1102 |
GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_25)

gsGetSetValue::gsSignalFormat Film

1283#define [GS_SIGFORM_FILM_1828_1332_24](#) (GS_SIGFORMSIZE_1828x1332 |
GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_24)

gsGetSetValue::gsSignalFormat Film

1284#define [GS_SIGFORM_FILM_1828_1332_25](#) (GS_SIGFORMSIZE_1828x1332 |
GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_25)

gsGetSetValue::gsSignalFormat Film

1285#define [GS_SIGFORM_FILM_2048_857_24](#) (GS_SIGFORMSIZE_2048x857 |
GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_24)

gsGetSetValue::gsSignalFormat Film

1286#define [GS_SIGFORM_FILM_2048_857_25](#) (GS_SIGFORMSIZE_2048x857 |
GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_25)

gsGetSetValue::gsSignalFormat Film

1287#define [GS_SIGFORM_FILM_2048_872_24](#) (GS_SIGFORMSIZE_2048x872 |
GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_24)

gsGetSetValue::gsSignalFormat Film

1288#define [GS_SIGFORM_FILM_2048_872_25](#) (GS_SIGFORMSIZE_2048x872 |
GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_25)

gsGetSetValue::gsSignalFormat Film

1289#define [GS_SIGFORM_FILM_2048_1102_24](#) (GS_SIGFORMSIZE_2048x1102 |
GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_24)

gsGetSetValue::gsSignalFormat Film

1290#define [GS_SIGFORM_FILM_2048_1102_25](#) (GS_SIGFORMSIZE_2048x1102 |
GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_25)

gsGetSetValue::gsSignalFormat Film

1291#define [GS_SIGFORM_FILM_2048_1234_24](#) (GS_SIGFORMSIZE_2048x1234 |
GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_24)

gsGetSetValue::gsSignalFormat Film

1292#define [GS_SIGFORM_FILM_2048_1234_25](#) (GS_SIGFORMSIZE_2048x1234 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_25)

gsGetSetValue::gsSignalFormat Film

1293#define [GS_SIGFORM_FILM_2048_15X](#) (GS_SIGFORMSIZE_2048x1556 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_14_98)

gsGetSetValue::gsSignalFormat Film 2K

1294#define [GS_SIGFORM_FILM_2048_14](#) GS_SIGFORM_FILM_2048_15X

gsGetSetValue::gsSignalFormat Film 2K

1295#define [GS_SIGFORM_FILM_2048_15](#) (GS_SIGFORMSIZE_2048x1556 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_15)

gsGetSetValue::gsSignalFormat Film 2K

1296#define [GS_SIGFORM_FILM_2048sf_15X](#) (GS_SIGFORMSIZE_2048x1556 | GS_SIGFORMTYPE_SEGMENTEDFRAME | GS_SIGFORMFRAMERATE_14_98)

gsGetSetValue::gsSignalFormat Film 2K

1297#define [GS_SIGFORM_FILM_2048sf_15](#) (GS_SIGFORMSIZE_2048x1556 | GS_SIGFORMTYPE_SEGMENTEDFRAME | GS_SIGFORMFRAMERATE_15)

gsGetSetValue::gsSignalFormat Film 2K

1298#define [GS_SIGFORM_FILM_2048_24X](#) (GS_SIGFORMSIZE_2048x1556 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_23_98)

gsGetSetValue::gsSignalFormat Film 2K

1299#define [GS_SIGFORM_FILM_2048_24](#) (GS_SIGFORMSIZE_2048x1556 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_24)

gsGetSetValue::gsSignalFormat Film 2K

1300#define [GS_SIGFORM_FILM_2048sf_24X](#) (GS_SIGFORMSIZE_2048x1556 | GS_SIGFORMTYPE_SEGMENTEDFRAME | GS_SIGFORMFRAMERATE_23_98)

gsGetSetValue::gsSignalFormat Film 2K

1301#define [GS_SIGFORM_FILM_2048sf_24](#) (GS_SIGFORMSIZE_2048x1556 | GS_SIGFORMTYPE_SEGMENTEDFRAME | GS_SIGFORMFRAMERATE_24)

gsGetSetValue::gsSignalFormat Film 2K

1302#define [GS_SIGFORM_FILM_2048_48](#) (GS_SIGFORMSIZE_2048x1556 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_48)

gsGetSetValue::gsSignalFormat Film 2K

1303#define [GS_SIGFORM_FILM_2048_1536_25](#) (GS_SIGFORMSIZE_2048x1536 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_25)

gsGetSetValue::gsSignalFormat Film 2K(1536)

1304#define [GS_SIGFORM_FILM_2048_1536sf_25](#) (GS_SIGFORMSIZE_2048x1536 | GS_SIGFORMTYPE_SEGMENTEDFRAME | GS_SIGFORMFRAMERATE_25)

gsGetSetValue::gsSignalFormat Film 2K(1536)

1305#define [GS_SIGFORM_FILM_2048_25](#) (GS_SIGFORMSIZE_2048x1556 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_25)

gsGetSetValue::gsSignalFormat Film 2K(1536)

1306#define [GS_SIGFORM_FILM_2048sf_25](#) (GS_SIGFORMSIZE_2048x1556 | GS_SIGFORMTYPE_SEGMENTEDFRAME | GS_SIGFORMFRAMERATE_25)

gsGetSetValue::gsSignalFormat Film 2K(1536)

1307#define [GS_SIGFORM_FILM_2048_1536_15X](#) (GS_SIGFORMSIZE_2048x1536 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_14_98)

gsGetSetValue::gsSignalFormat Film 2K(1536)

1308#define [GS_SIGFORM_FILM_2048_1536_15](#) (GS_SIGFORMSIZE_2048x1536 |
GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_15)
gsGetSetValue::gsSignalFormat Film 2K(1536)

1309#define [GS_SIGFORM_FILM_2048_1536sf_15X](#) (GS_SIGFORMSIZE_2048x1536 |
GS_SIGFORMTYPE_SEGMENTEDFRAME | GS_SIGFORMFRAMERATE_14_98)
gsGetSetValue::gsSignalFormat Film 2K(1536)

1310#define [GS_SIGFORM_FILM_2048_1536sf_15](#) (GS_SIGFORMSIZE_2048x1536 |
GS_SIGFORMTYPE_SEGMENTEDFRAME | GS_SIGFORMFRAMERATE_15)
gsGetSetValue::gsSignalFormat Film 2K(1536)

1311#define [GS_SIGFORM_FILM_2048_1536_24X](#) (GS_SIGFORMSIZE_2048x1536 |
GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_23_98)
gsGetSetValue::gsSignalFormat Film 2K(1536)

1312#define [GS_SIGFORM_FILM_2048_1536_24](#) (GS_SIGFORMSIZE_2048x1536 |
GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_24)
gsGetSetValue::gsSignalFormat Film 2K(1536)

1313#define [GS_SIGFORM_FILM_2048_1536sf_24X](#) (GS_SIGFORMSIZE_2048x1536 |
GS_SIGFORMTYPE_SEGMENTEDFRAME | GS_SIGFORMFRAMERATE_23_98)
gsGetSetValue::gsSignalFormat Film 2K(1536)

1314#define [GS_SIGFORM_FILM_2048_1536sf_24](#) (GS_SIGFORMSIZE_2048x1536 |
GS_SIGFORMTYPE_SEGMENTEDFRAME | GS_SIGFORMFRAMERATE_24)
gsGetSetValue::gsSignalFormat Film 2K(1536)

1315#define [GS_SIGFORM_FILM_2048_1536_48X](#) (GS_SIGFORMSIZE_2048x1536 |
GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_47_95)
gsGetSetValue::gsSignalFormat Film 2K(1536)

1316#define [GS_SIGFORM_FILM_2048_1536_48](#) (GS_SIGFORMSIZE_2048x1536 |
GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_48)
gsGetSetValue::gsSignalFormat Film 2K(1536)

1317#define [GS_SIGFORM_QUADHD_24X](#) (GS_SIGFORMSIZE_3840x2880 |
GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_23_98)
gsGetSetValue::gsSignalFormat Quad HD 23.98 fps (4 x 1920x1080)

1318#define [GS_SIGFORM_QUADHDsf_24X](#) (GS_SIGFORMSIZE_3840x2880 |
GS_SIGFORMTYPE_SEGMENTEDFRAME | GS_SIGFORMFRAMERATE_23_98)
gsGetSetValue::gsSignalFormat Quad HD 23.98 sf fps (4 x 1920x1080)

1319#define [GS_SIGFORM_QUADHD_24](#) (GS_SIGFORMSIZE_3840x2880 |
GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_24)
gsGetSetValue::gsSignalFormat Quad HD 24 fps (4 x 1920x1080)

1320#define [GS_SIGFORM_QUADHDsf_24](#) (GS_SIGFORMSIZE_3840x2880 |
GS_SIGFORMTYPE_SEGMENTEDFRAME | GS_SIGFORMFRAMERATE_24)
gsGetSetValue::gsSignalFormat Quad HD 24 sf fps (4 x 1920x1080)

1321#define [GS_SIGFORM_QUADHD_25](#) (GS_SIGFORMSIZE_3840x2880 |
GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_25)

1322#define [GS_SIGFORM_QUADHDsf_25](#) (GS_SIGFORMSIZE_3840x2880 |
GS_SIGFORMTYPE_SEGMENTEDFRAME | GS_SIGFORMFRAMERATE_25)

1323#define [GS_SIGFORM_4K_QUAD_24X](#) (GS_SIGFORMSIZE_4096x2880 |
GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_23_98)
gsGetSetValue::gsSignalFormat Quad 2K 23.98 fps (4 x 2048x1080)

1324#define [GS_SIGFORM_4K_QUADsf_24X](#) (GS_SIGFORMSIZE_4096x2880 |
GS_SIGFORMTYPE_SEGMENTEDFRAME | GS_SIGFORMFRAMERATE_23_98)
gsGetSetValue::gsSignalFormat Quad 2K sf 23.98 fps (4 x 2048x1080)

```

1325#define GS\_SIGFORM\_4K\_QUAD\_24 (GS_SIGFORMSIZE_4096x2880 |
    GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_24)
gsGetSetValue::gsSignalFormat Quad 2K 24 fps (4 x 2048x1080)
1326#define GS\_SIGFORM\_4K\_QUADsf\_24 (GS_SIGFORMSIZE_4096x2880 |
    GS_SIGFORMTYPE_SEGMENTEDFRAME | GS_SIGFORMFRAMERATE_24)
gsGetSetValue::gsSignalFormat Quad 2K sf 24 fps (4 x 2048x1080)
1327#define GS\_SIGFORM\_4K\_QUAD\_25 (GS_SIGFORMSIZE_4096x2880 |
    GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_25)
1328#define GS\_SIGFORM\_4K\_QUADsf\_25 (GS_SIGFORMSIZE_4096x2880 |
    GS_SIGFORMTYPE_SEGMENTEDFRAME | GS_SIGFORMFRAMERATE_25)
1329#define GS\_SIGFORM\_FILM\_4096\_1714\_24 (GS_SIGFORMSIZE_4096x1714 |
    GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_24)
gsGetSetValue::gsSignalFormat Film 4K Half
1330#define GS\_SIGFORM\_FILM\_4096\_1714\_24X (GS_SIGFORMSIZE_4096x1714 |
    GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_23_98)
gsGetSetValue::gsSignalFormat Film 4K Half
1331#define GS\_SIGFORM\_FILM\_4096\_3112sf\_5 (GS_SIGFORMSIZE_4096x3112 |
    GS_SIGFORMTYPE_SEGMENTEDFRAME | GS_SIGFORMFRAMERATE_5)
gsGetSetValue::gsSignalFormat Film 4K
1332#define GS\_SIGFORM\_FILM\_4096\_3112\_24 (GS_SIGFORMSIZE_4096x3112 |
    GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_24)
gsGetSetValue::gsSignalFormat Film 4K
1333#define GS\_SIGFORM\_FILM\_4096\_3112\_24X (GS_SIGFORMSIZE_4096x3112 |
    GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_23_98)
gsGetSetValue::gsSignalFormat Film 4K
1334#define GS\_SIGFORM\_ARRI\_D21 (GS_SIGFORMSIZE_2880x2160 |
    GS_SIGFORMTYPE_PROGRESSIVE)
1335#define GS\_SIGFORM\_ARRI\_ALEXA (GS_SIGFORMSIZE_2880x1782 |
    GS_SIGFORMTYPE_PROGRESSIVE)
1336#define GS\_SIGFORM\_CUSTOM 0xF0000000UL
All non video rate types (e.g.. 15fps, 10fps, 37fps) gsGetSetValue::gsSignalFormat.
1337#define GS\_SIGFORM\_NOT\_PRESENT 0
For input and genlock status returns.
1338#define GS\_SIGFORM\_SUPPORTS\_NTSC 0x00000001
NTSC (CCIR or sqp) 720x480/486/508/512.
1339#define GS\_SIGFORM\_SUPPORTS\_PAL 0x00000002
PAL (CCIR or sqp) 720x576/608.
1340#define GS\_SIGFORM\_SUPPORTS\_HR 0x00000004
960 width SD, not used
1341#define GS\_SIGFORM\_SUPPORTS\_360 0x00000008
360 compressed, not used
1342#define GS\_SIGFORM\_SUPPORTS\_720 0x00000010
720p Rasters (59/60 and sometimes 50)
1343#define GS\_SIGFORM\_SUPPORTS\_1035 0x00000020
1035x1080 Production rasters
1344#define GS\_SIGFORM\_SUPPORTS\_1080 0x00000040
1080/1088/1092/1112x1920 HD rasters
1345#define GS\_SIGFORM\_SUPPORTS\_EXTRA8 0x00000080

```

1088 HD rasters

1346#define [GS_SIGFORM_SUPPORTS_1536](#) 0x00000100
Film 2K 1536 lines.

1347#define [GS_SIGFORM_SUPPORTS_1556](#) 0x00000200
Film 2K 1556 lines.

1348#define [GS_SIGFORM_SUPPORTS_DCIN](#) 0x00000400
Digital Cinema 2048x1080.

1349#define [GS_SIGFORM_SUPPORTS_1400](#) 0x00000800
Presentation 1440x1050.

1350#define [GS_SIGFORM_SUPPORTS_QUADSDI](#) 0x00001000
Quad HD-SDI 3840 and 4096.

1351#define [GS_SIGFORM_SUPPORTS_V480](#) 0x00010000
Vesa 640x480.

1352#define [GS_SIGFORM_SUPPORTS_V600](#) 0x00020000
Vesa 800x600.

1353#define [GS_SIGFORM_SUPPORTS_V768](#) 0x00040000
Vesa 1024x768.

1354#define [GS_SIGFORM_SUPPORTS_V1024](#) 0x00080000
Vesa 1280x1024.

1355#define [GS_SIGFORM_SUPPORTS_V1200](#) 0x00100000
Vesa 1600x1200.

1356#define [GS_SIGFORM_SUPPORTS_V1600](#) 0x00200000
Vesa 1600.

1357#define [GS_SIGFORM_SUPPORTS_X2](#) 0x20000000
Modifier times 2.

1358#define [GS_SIGFORM_SUPPORTS_X3](#) 0x40000000
Modifier times 3.

1359#define [GS_SIGFORM_SUPPORTS_X4](#) 0x80000000
Modifier times 4.

1360#define [GS_SIGFORM_SUPPORTS_1080_X2](#) (GS_SIGFORM_SUPPORTS_1080|
GS_SIGFORM_SUPPORTS_X2)
Supports 1080p 50/59/60.

1361#define [GS_COMPTYPE_SOFTWARE](#) 0x00000001
Software passed codec on main processor (gsGetSetValue::gsCompType)

1362#define [GS_COMPTYPE_BAYER](#) 0x00000002
Motion JPEG hardware codec (LSI, Zoran, C-Cube, etc) (gsGetSetValue::gsCompType)

1363#define [GS_COMPTYPE_H264](#) 0x00000004
MPEG-4 h.264.

1364#define [GS_COMPTYPE_JPEG2000](#) 0x00000008

1365#define [GS_COMPTYPE_CINEFORM_3D](#) 0x00000010
MPEG 1 hardware compatible codec (gsGetSetValue::gsCompType)

1366#define [GS_COMPTYPE_BGR](#) 0x00000020

1367#define [GS_COMPTYPE_HDCAM](#) 0x00000040
Editable MPEG 2 I Frame Only compatible codec (gsGetSetValue::gsCompType)

1368#define [GS_COMPTYPE_MPEG](#) 0x00000080

MPEG 2 long GOP or IFrame hardware compatible codec (gsGetSetValue::gsCompType)
 1369#define [GS_COMPTYPE_DV25](#) 0x00000100
 Hardware DV25, DVCPRO, DVCPRO25 (gsGetSetValue::gsCompType)
 1370#define [GS_COMPTYPE_DV50](#) 0x00000200
 Hardware DV50, DVCPRO50 (gsGetSetValue::gsCompType)
 1371#define [GS_COMPTYPE_DVSD](#) 0x00000400
 Hardware Standard DV Bluebook, DVPRO, DVSD (gsGetSetValue::gsCompType)
 1372#define [GS_COMPTYPE_DV100](#) 0x00000800
 High Def DV codec (gsGetSetValue::gsCompType)
 1373#define [GS_COMPTYPE_CINEFORM](#) 0x00001000
 8Bit Y'CrCb uncompressed video (gsGetSetValue::gsCompType)
 1374#define [GS_COMPTYPE_YCRCB_V210](#) 0x00002000
 10Bit Y'CrCb uncompressed video (gsGetSetValue::gsCompType)
 1375#define [GS_COMPTYPE_RGB](#) 0x00004000
 Uncompressed RGB 24 Bit.
 1376#define [GS_COMPTYPE_DNxHD](#) 0x00008000
 Avid DNxHD.
 1377#define [GS_COMPTYPE_AVCi](#) 0x00010000
 Panasonic AVCi (gsGetSetValue::gsCompType)
 1378#define [GS_COMPTYPE_PRORES](#) 0x00020000
 Apple ProRes (gsGetSetValue::gsCompType)
 1379#define [GS_COMPTYPE_BGRA_INVERT](#) 0x00040000
 Inverted 32 bit TGA.
 1380#define [GS_COMPTYPE_DPX_YCBCR10](#) 0x00080000
 DPX 10 bit YCbCr.
 1381#define [GS_COMPTYPE_ARGB](#) 0x00100000
 Uncompressed RGB (DVS)
 1382#define [GS_COMPTYPE_RGBA](#) 0x00200000
 Uncompressed RGBA (DVS)
 1383#define [GS_COMPTYPE_ABGR](#) 0x00400000
 Uncompressed A BGR - TIFF.
 1384#define [GS_COMPTYPE_BGRA](#) 0x00800000
 Uncompressed BGR A - BMP/TGA.
 1385#define [GS_COMPTYPE_YCRCB_422](#) 0x01000000
 Uncompressed Y'CrCb 4:2:2 (DVS, VG)
 1386#define [GS_COMPTYPE_YCRCB_422A](#) 0x02000000
 Uncompressed Y'CrCb 4:2:2A (DVS, Dual VG)
 1387#define [GS_COMPTYPE_YCRCB_444](#) 0x04000000
 Uncompressed Y'CrCb 4:4:4 (DVS, Dual VG)
 1388#define [GS_COMPTYPE_YCRCB_444A](#) 0x08000000
 Uncompressed Y'CrCb 4:4:4A (DVS, Dual VG)
 1389#define [GS_COMPTYPE_STEREO](#) 0x10000000
 Uncompressed Y'CrCb 4:4:4A (DVS, Dual VG) or 3D 8, 10, 30 or 32 bit.
 1390#define [GS_COMPTYPE_YCRCB_420](#) 0x20000000
 Uncompressed Y'CrCb 4:2:0.

1391#define [GS_COMPTYPE_DPX_RGB10](#) 0x40000000
DPX 10 bit rgb.

1392#define [GS_COMPTYPE_ALT](#) 0x80000000
Use as generic alternative for use through AVCodec.

1393#define [GS_HDSDBAYER_DUALBIT](#) 0x10000000
Flag bit for dual rate capture.

1394#define [GS_HDSDBAYER_DUALLINKBIT](#) 0x20000000
Flag bit for dual pipe capture.

1395#define [GS_HDSDBAYER_ARRI_D21](#) 0x00000001
Raw bayer HD-SDI: Arri D21 T-Link.

1396#define [GS_HDSDBAYER_ARRI_ALEXA](#) 0x00000002
Raw bayer HD-SDI: Arri Alexa.

1397#define [GS_HDSDBAYER_WIESS_ONEFRAME](#) 0x00000100
Raw bayer HD-SDI: Weisscam 1:1 - up to 30 in 30.

1398#define [GS_HDSDBAYER_WIESS_2K1536](#) 0x00000400
Raw bayer HD-SDI: Weisscam Film2K at 25p.

1399#define [GS_HDSDBAYER_WIESS_TWOFRAME](#)
 (GS_HDSDBAYER_WIESS_ONEFRAME | GS_HDSDBAYER_DUALBIT)
Raw bayer HD-SDI: Weisscam 720p at 100p, 1080p up to 60.

1400#define [GS_HDSDBAYER_WIESS_QUADFRAME](#)
 (GS_HDSDBAYER_WIESS_ONEFRAME | GS_HDSDBAYER_DUALBIT |
 GS_HDSDBAYER_DUALLINKBIT)
Raw bayer HD-SDI: Weisscam 720p at 200p, 1080p up to 120.

1401#define [GS_HDSDBAYER_WIESS_TWO2K1536](#) (GS_HDSDBAYER_WIESS_2K1536 |
 GS_HDSDBAYER_DUALBIT)
Raw dual 2K film.

1402#define [GS_HSDTI_HDCAM_SR](#) (GS_HDSDBAYER_DUALLINKBIT | 0x00100000)
HDCamSR SDTI.

1403#define [GS_PBEE_AUTO](#) 0x00000000
Allow playback or edit to edit output as necessary (gsGetSetValue::gsPBEE)

1404#define [GS_PBEE_PB](#) 0x00000001
Allow playback only output - no passthrough (gsGetSetValue::gsPBEE)

1405#define [GS_PBEE_EE](#) 0x00000002
Allow passthrough only output - no playback (gsGetSetValue::gsPBEE)

1406#define [GS_PBEE_DEFAULT](#) 0x000000FF
Device dependent output (gsGetSetValue::gsPBEE)

1407#define [GS_SERVOREF_AUTO](#) 0x00000000
Video servo reference auto (gsGetSetValue::gsServoRefSelect)

1408#define [GS_SERVOREF_EXT](#) 0x00000001
Video servo reference external only (gsGetSetValue::gsServoRefSelect)

1409#define [GS_SERVOREF_DEFAULT](#) 0x000000FF
Video servo reference device default (gsGetSetValue::gsServoRefSelect)

1410#define [GS_HEADSEL_RECPLAY](#) 0x00000000
Use record/play head (gsGetSetValue::gsHeadSelect)

1411#define [GS_HEADSEL_PLAY](#) 0x00000001
Use play head (gsGetSetValue::gsHeadSelect)

1412#define [GS_HEADSEL_DEFAULT](#) 0x000000FF
(gsGetSetValue::gsHeadSelect)

1413#define [GS_CLRFRM_2FLD](#) 0x00000000
Edit color frame 2 field (gsGetSetValue::gsColorFrame)

1414#define [GS_CLRFRM_4FLD](#) 0x00000001
Edit color frame 4 field (gsGetSetValue::gsColorFrame)

1415#define [GS_CLRFRM_8FLD](#) 0x00000002
Edit color frame 8 field (gsGetSetValue::gsColorFrame)

1416#define [GS_CLRFRM_DEFAULT](#) 0x000000FF
Edit color frame device default (gsGetSetValue::gsColorFrame)

1417#define [GS_VIDREF_DISABLE](#) 0x00000000
Disable video reference (gsGetSetValue::gsVidRefDisable)

1418#define [GS_VIDREF_ENABLE](#) 0x00000001
Enable video reference (gsGetSetValue::gsVidRefDisable)

1419#define [GS_CHANCAP_PLAY](#) 0x00000001
Channel can play (video or audio or both) (gsGetSetValue::gsChanCapabilities)

1420#define [GS_CHANCAP_REVPLAY](#) 0x00000002
Channel can reverse play (video or audio or both) (gsGetSetValue::gsChanCapabilities)

1421#define [GS_CHANCAP_PAUSE](#) 0x00000004
Channel can pause and display frame (video or audio or both) (gsGetSetValue::gsChanCapabilities)

1422#define [GS_CHANCAP_JOG](#) 0x00000008
Channel can jog below play speed (video or audio or both) (gsGetSetValue::gsChanCapabilities)

1423#define [GS_CHANCAP_SHUTTLE](#) 0x00000010
Channel can shuttle above play speed (video or audio or both) (gsGetSetValue::gsChanCapabilities)

1424#define [GS_CHANCAP_SEEK](#) 0x00000020
Channel can seek to any point (video or audio or both) (gsGetSetValue::gsChanCapabilities)

1425#define [GS_CHANCAP_PREVIEW](#) 0x00000040
Channel can preview from in to out (video or audio or both) (gsGetSetValue::gsChanCapabilities)

1426#define [GS_CHANCAP_STOP](#) 0x00001000
Channel has a stop mode (video or audio or both) (gsGetSetValue::gsChanCapabilities)

1427#define [GS_CHANCAP_ETOE](#) 0x00002000
Channel can pass through video (in stop) (video or audio or both) (gsGetSetValue::gsChanCapabilities)

1428#define [GS_CHANCAP_RECORD](#) 0x00004000
Channel can record (video or audio or both) (gsGetSetValue::gsChanCapabilities)

1429#define [GS_CHANCAP_EDIT](#) 0x00008000
Channel can edit from in to out (video or audio or both) (gsGetSetValue::gsChanCapabilities)

1430#define [GS_CHANCAP_RECSTOP](#) 0x00010000
Channel can set clip name and prep record (video or audio or both) (gsGetSetValue::gsChanCapabilities)

1431#define [GS_CHANCAP_SELECTPRESET](#) 0x00020000

Channel can select recording channels (video or audio or both)
(gsGetSetValue::gsChanCapabilities)

1432#define [GS_CHANCAP_EJECT](#) 0x00040000
Channel can eject the media (video or audio or both) *(gsGetSetValue::gsChanCapabilities)*

1433#define [GS_CHANCAP_LOOP](#) 0x00100000
Channel can play in a loop (video or audio or both) *(gsGetSetValue::gsChanCapabilities)*

1434#define [GS_CHANCAP_VGAPREVIEW](#) 0x00200000
Channel can display a VGA preview (video or audio or both)
(gsGetSetValue::gsChanCapabilities)

1435#define [GS_CHANCAP_AUDPREVIEW](#) 0x00200000
Channel can preview audio on a multi media card (video or audio or both)
(gsGetSetValue::gsChanCapabilities)

1436#define [GS_CHANCAP_FILE](#) 0x01000000
Channel can play from a file (video or audio or both) *(gsGetSetValue::gsChanCapabilities)*

1437#define [GS_CHANCAP_NET](#) 0x02000000
Channel can play from a network feed (video or audio or both)
(gsGetSetValue::gsChanCapabilities)

1438#define [GS_CHANCAP_CLIPSPACE](#) 0x10000000
Channel can act like a clip space (video or audio or both)
(gsGetSetValue::gsChanCapabilities)

1439#define [GS_CHANCAP_TCSPACE](#) 0x20000000
Channel can act like a VTR time code space (video or audio or both)
(gsGetSetValue::gsChanCapabilities)

1440#define [GS_CHANCAP_MPEG_ENC](#) 0x40000000
1441#define [GS_CHANCAP_ERROR](#) 0x80000000
Do not use this bit - indicates error.

1442#define [GS_CHANCAP_ALL](#) 0x3FFFFFFF
Channel can do anything except MPEG_ENC (by default this should not be)

1443#define [GS_PRODUCTION_MODE_PLAY](#) 0x01
Stop if frames dropped in playback.

1444#define [GS_PRODUCTION_MODE_RECORD](#) 0x02
Stop if frames dropped in record.

1445#define [GS_SERIALEDITMODE_NONE](#) 0
Normal editing mode, no special speed compensation *cmdGetSetValue::gsSerialEditMode*.

1446#define [GS_SERIALEDITMODE_IGNORE](#) 1
Ignore all off speed play commands (CBS TimeLogic Mode)
cmdGetSetValue::gsSerialEditMode.

1447#define [GS_SERIALEDITMODE_FAKE](#) 2
Pause at each speed change, call time play when real play comes (CTV mode)
cmdGetSetValue::gsSerialEditMode.

1448#define [GS_SERIALPROTOCOLS_SONY422](#) 1
Enable Sony VTR 422 *(gsSerialProtocols)*

1449#define [GS_SERIALPROTOCOLS_ODETICS](#) 2
Enable Odetics extensions *(gsSerialProtocols)*

1450#define [GS_SERIALPROTOCOLS_VDCP](#) 4
Enable VDCP Louth *(gsSerialProtocols)*

```

1451#define GS\_PROXYMODE\_NOTHING 0x000000
cmdGetSetValue::gsProxyMode Do not automatically proxy anything
1452#define GS\_PROXYMODE EVERYTHING 0x000002
cmdGetSetValue::gsProxyMode Proxy any file that is opened (for read/write/check)
1453#define GS\_PROXYMODE\_RECORD 0x000001
cmdGetSetValue::gsProxyMode Proxy files while they are recording (with supported source
types)
1454#define GS\_PROXYMODE\_AFTERRECORD 0x000010
cmdGetSetValue::gsProxyMode Proxy files once they have finished recording
1455#define GS\_PROXYMODE\_ABORTALL 0x0ffff0
cmdGetSetValue::gsProxyMode Abort all active proxies
1456#define GS\_WAVEVECTOR\_PICTURE 0x00000001
cmdGetSetValue::gsWaveVectorSetup standard picture
1457#define GS\_WAVEVECTOR\_VECTORSCOPE 0x00000002
cmdGetSetValue::gsWaveVectorSetup standard vectorscope
1458#define GS\_WAVEVECTOR\_WAVEFORM 0x00000004
cmdGetSetValue::gsWaveVectorSetup standard waveform
1459#define GS\_WAVEVECTOR\_WAVEFORM\_RGB 0x00000008
cmdGetSetValue::gsWaveVectorSetup parade RGB waveform
1460#define GS\_WAVEVECTOR\_WAVEFORM\_YCBCR 0x00000010
cmdGetSetValue::gsWaveVectorSetup parade Y CB CR waveform
1461#define GS\_WAVEVECTOR\_HISTOGRAM 0x00000020
cmdGetSetValue::gsWaveVectorSetup histogram
1462#define GS\_WAVEVECTOR\_HISTOGRAM\_SEP 0x00000040
cmdGetSetValue::gsWaveVectorSetup parade histogram
1463#define GS\_WAVEVECTOR\_DATA 0x00000080
cmdGetSetValue::gsWaveVectorSetup illegal colors
1464#define GS\_WAVEVECTOR\_ILLEGAL 0x00000100
cmdGetSetValue::gsWaveVectorSetup illegal colors
1465#define GS\_WAVEVECTOR\_MASK 0x000001FF
Video types mask.
1466#define GS\_WAVEAUDIO\_WAVE 0x00000200
Audio wave form.
1467#define GS\_WAVEAUDIO\_METERS 0x00000400
Audio meters.
1468#define GS\_WAVEAUDIO\_LISSAJOUSXY 0x00000800
Audio Lissajous X-Y.
1469#define GS\_WAVEAUDIO\_SURROUND 0x00001000
Surround monitor.
1470#define GS\_WAVEAUDIO\_2 0x00002000
1471#define GS\_WAVEAUDIO\_4 0x00004000
1472#define GS\_WAVEAUDIO\_8 0x00008000
1473#define GS\_WAVEAUDIO\_MASK 0x0000FE00
Audio types mask.
1474#define GS\_WAVEVECTOR\_FLAG\_MASK 0xFFFF0000
1475#define GS\_WAVEVECTOR\_ALT\_GRATICULE 0x10000000

```

```

1476#define GS\_WAVEVECTOR\_HIDE75VECTOR 0x01000000
1477#define GS\_WAVEVECTOR\_HIDE100VECTOR 0x02000000
1478#define GS\_WAVEVECTOR\_HIDEFLESHVECTOR 0x04000000
1479#define GS\_WAVEVECTOR\_HIDEANGLES 0x08000000
1480#define GS\_WAVEVECTOR\_USESMPTESCALE 0x00000000
1481#define GS\_WAVEVECTOR\_USEFULLSCALE 0x01000000
1482#define GS\_WAVEVECTOR\_PICT\_CLEAN 0x00000000
1483#define GS\_WAVEVECTOR\_PICT\_SAFE 0x01000000
1484#define GS\_WAVEVECTOR\_PICT\_TITLE\_SAFE 0x02000000
1485#define GS\_WAVEVECTOR\_CHANNEL\_MASK 0x00FF0000
1486#define GS\_WAVEVECTOR\_CHANNEL\_R 0x00010000
cmdGetSetValue::gsWaveVectorSetup dwStart color channel RED
1487#define GS\_WAVEVECTOR\_CHANNEL\_G 0x00020000
cmdGetSetValue::gsWaveVectorSetup dwStart color channel GREEN
1488#define GS\_WAVEVECTOR\_CHANNEL\_B 0x00040000
cmdGetSetValue::gsWaveVectorSetup dwStart color channel BLUE
1489#define GS\_WAVEVECTOR\_CHANNEL\_A 0x00080000
cmdGetSetValue::gsWaveVectorSetup dwStart color channel ALPHA
1490#define GS\_WAVEVECTOR\_CHANNEL\_Y 0x00100000
cmdGetSetValue::gsWaveVectorSetup dwStart color channel Y (Luma)
1491#define GS\_WAVEVECTOR\_CHANNEL\_CB 0x00200000
cmdGetSetValue::gsWaveVectorSetup dwStart color channel CB
1492#define GS\_WAVEVECTOR\_CHANNEL\_CR 0x00400000
cmdGetSetValue::gsWaveVectorSetup dwStart color channel CR
1493#define GS\_TCSRC\_DISABLE\_EXTERNAL 0x8000
cmdGetSetValue::gsTimecodeSources Don't use any external time code of any kind
1494#define GS\_TCSRC\_FORCE\_VTR\_TC 0x0002
cmdGetSetValue::gsTimecodeSources Use the RS-422 VTR time code
1495#define GS\_TCSRC\_USE\_TIMEOFDAY 0x0080
cmdGetSetValue::gsTimecodeSources Use the time of day as time code
1496#define GS\_DXRGB\_DIRECT 0x001
DirectX allow direct RGB plane (for cmdGetSetValue::gsVgaDirectXConfig)
1497#define GS\_DXRGB\_OVERLAY 0x002
DirectX allow overlay RGB plane (for cmdGetSetValue::gsVgaDirectXConfig)
1498#define GS\_DXYUV\_OVERLAY 0x004
DirectX allow overlay YUV plane (for cmdGetSetValue::gsVgaDirectXConfig)
1499#define GS\_DXYUV\_DIRECT 0x008
DirectX allow direct YUV plane (for cmdGetSetValue::gsVgaDirectXConfig)
1500#define GS\_3DVGA\_TYPE\_MASK 0x00FFFFFF
3D VGA viewing mode MASK for all modes (for cmdGetSetValue::gsVga3DConfig)
1501#define GS\_3DVGA\_FLAGS\_MASK 0xFF000000
3D VGA viewing mode MASK for flags (for cmdGetSetValue::gsVga3DConfig)
1502#define GS\_3DVGA\_LEFTEYE 0x00000001
3D VGA view left eye only (top picture) (for cmdGetSetValue::gsVga3DConfig)
1503#define GS\_3DVGA\_RIGHTEYE 0x00000002
3D VGA view right eye only (bottom picture) (for cmdGetSetValue::gsVga3DConfig)
1504#define GS\_3DVGA\_ANAGLYPH\_REDBLUE 0x00000004

```

3D VGA view comics red/blue (for cmdGetSetValue::gsVga3DConfig)

1505#define [GS_3DVGA_ANAGLYPH_REDCYAN](#) 0x00000008
3D VGA view 50s movie red/cyan (for cmdGetSetValue::gsVga3DConfig)

1506#define [GS_3DVGA_ANAGLYPH_AMBERBLUE](#) 0x00000010
3D VGA view 50s movie amber/blue (for cmdGetSetValue::gsVga3DConfig)

1507#define [GS_3DVGA_ANAGLYPH_GREENMAGENTA](#) 0x00000020
3D VGA view 50s movie green/magenta (for cmdGetSetValue::gsVga3DConfig)

1508#define [GS_3DVGA_INTERLACED](#) 0x00000040
3D VGA view interlaced (Zalman, real 3D, IMAX) (for cmdGetSetValue::gsVga3DConfig)

1509#define [GS_3DVGA_ONIONSKIN](#) 0x00000080
3D VGA view onion skin like 2D animation programs (for cmdGetSetValue::gsVga3DConfig)

1510#define [GS_3DVGA_DIFFERENCE](#) 0x00000100
3D VGA view absolute difference (for cmdGetSetValue::gsVga3DConfig)

1511#define [GS_3DVGA_OVERUNDER](#) 0x00000200
3D VGA view images on top and bottom (squeeze vert) (for cmdGetSetValue::gsVga3DConfig)

1512#define [GS_3DVGA_SIDEBYSIDE](#) 0x00000400
3D VGA view image next to each other (squeeze horiz) (for cmdGetSetValue::gsVga3DConfig)

1513#define [GS_3DVGA_SPLIT](#) 0x00000800
3D VGA view arbitrary split (for cmdGetSetValue::gsVga3DConfig)

1514#define [GS_3DVGA_MIRROR](#) 0x00001000
3D VGA view squeeze and mirror (for cmdGetSetValue::gsVga3DConfig)

1515#define [GS_3DVGA_BUTTERFLY](#) 0x00002000
3D VGA view invert right and split (for cmdGetSetValue::gsVga3DConfig)

1516#define [GS_3DVGA_AMINUSB_THRESHOLD](#) 0x00004000
3D VGA view difference above a certain threshold (for cmdGetSetValue::gsVga3DConfig)

1517#define [GS_3DVGA DISSOLVE](#) 0x00008000
3D VGA dissolve between (for cmdGetSetValue::gsVga3DConfig)

1518#define [GS_3DVGA_WIPE](#) 0x00010000
3D VGA wipe (smpte +) (for cmdGetSetValue::gsVga3DConfig)

1519#define [GS_3DVGA_LUMA_DIFF](#) 0x00020000
3D VGA luma invert diff (invert luma of second frame, then diff)

1520#define [GS_3DVGA_2DSHOWCOMPONENT](#) 0x00800000
3D VGA 2D show component (0xFF000000 Alpha, 0x00FF0000 Red, 0x0000FF00 Green, 0x000000FF Blue, 0xFFFFFFFF Gray)

1521#define [GS_3DVGA_FLAG_ADDGRID](#) 0x01000000
3D VGA view flag to add a grid (for cmdGetSetValue::gsVga3DConfig)

1522#define [GS_3DVGA_FLAG_INVERT](#) 0x80000000
3D VGA view flag to invert left/right (for cmdGetSetValue::gsVga3DConfig)

1523#define [GS_3DVGA_FLAG_FLIPLEFTVERT](#) 0x40000000
3D VGA view flag to flip left vertically (for cmdGetSetValue::gsVga3DConfig)

1524#define [GS_3DVGA_FLAG_FLIPRIGHTVERT](#) 0x04000000
3D VGA view flag to flip right vertically (for cmdGetSetValue::gsVga3DConfig)

```

1525#define GS\_3DVGA\_FLAG\_FLIPLEFTHORIZ 0x20000000
3D VGA view flag to flip left horizontally (for cmdGetSetValue::gsVga3DConfig)
1526#define GS\_3DVGA\_FLAG\_FLIPRIGHTHORIZ 0x02000000
3D VGA view flag to flip left horizontally (for cmdGetSetValue::gsVga3DConfig)
1527#define GS\_3DVGA\_FLAG\_SPLITVERT 0x10000000
3D VGA view split flags make it a vertical split/mirror/butterfly
1528#define GS\_3DVGA\_FLAG\_LENTICULAR GS\_3DVGA\_FLAG\_SPLITVERT
1529#define GS\_XML\_FILENAME 1
cmdGetSetValue::gsDTProjectToXml
1530#define GS\_XML\_CLIPFILE 2
cmdGetSetValue::gsDTProjectToXml
1531#define GS\_XML\_EDL 3
cmdGetSetValue::gsDTProjectToXml
1532#define GS\_XML\_STRING 4
cmdGetSetValue::gsDTProjectToXml
1533#define GS\_XML\_DWORD 5
cmdGetSetValue::gsDTProjectToXml
1534#define GS\_XML\_SAVE 6
cmdGetSetValue::gsDTProjectToXml
1535#define GS\_XML\_CHECK\_OPEN 7
cmdGetSetValue::gsDTProjectToXml
1536#define GS\_XML\_OPEN\_DELETE\_FILES 8
cmdGetSetValue::gsDTProjectToXml
1537#define GS\_XML\_OPEN\_IGNORE\_FILES 9
cmdGetSetValue::gsDTProjectToXml
1538#define GS\_ERROR\_FILE\_EXISTS -2
cmdGetSetValue::gsDTProjectToXml
1539#define GS\_APP\_NONE 0x00000000
For cmdGetSetValue::gsApplicationID set to no specific application.
1540#define GS\_APP\_QUICKCLIP 0x00000001
For cmdGetSetValue::gsApplicationID set to no specific application.
1541#define GS\_APP\_QUICKCLIPXO 0x00000002
For cmdGetSetValue::gsApplicationID set to no specific application.
1542#define GS\_APP\_VTRID 0x00000004
For cmdGetSetValue::gsApplicationID set to no specific application.
1543#define GS\_APP\_MEDIANXS 0x00000008
For cmdGetSetValue::gsApplicationID set to no specific application.
1544#define GS\_APP\_DTREPLAYLIVE 0x00000010
For cmdGetSetValue::gsApplicationID set to no specific application.
1545#define GS\_APP\_DTOUCH 0x00000020
For cmdGetSetValue::gsApplicationID set to no specific application.
1546#define GS\_APP\_BBREPLAY 0x00000040
For cmdGetSetValue::gsApplicationID set to no specific application.
1547#define GS\_SHUTDOWNAPPLICATION\_POSITION 0x01010101
1548#define GS\_SHUTDOWNAPPLICATION\_START 0xA5A5A5A5

```


1549#define [GS_SHUTDOWNAPPLICATION_END](#) 0x5F5F5F5F
1550#define [GS_SHUTDOWNSYSTEM_RESTART](#) 0xA5A5A5A4
1551#define [GS_SHUTDOWNSYSTEM_POSITION](#) 0x11111111
1552#define [GS_SHUTDOWNSYSTEM_START](#) 0x5F5F5F5F
1553#define [GS_SHUTDOWNSYSTEM_END](#) 0xA5A5A5A5
1554#define [GS_CLEANRECORDWIPE_ROOTDIR](#) 0x0
1555#define [GS_CLEANRECORDWIPE_WHOLEDRIIVE](#) 0x1
1556#define [GS_CLEANRECORDWIPE_START](#) 0x5F5F5F5F
1557#define [GS_CLEANRECORDWIPE_END](#) 0xA5A5A5A5
1558#define [GS_INSTALLSYSTEM_POSITION](#) 0x2B2B2B2B
1559#define [GS_INSTALLSYSTEM_START](#) 0x11111111
1560#define [GS_INSTALLSYSTEM_END](#) 0x4E4E4E4E
1561#define [GS_NOT_SUPPORTED](#) 0xFFFFFFFF
Command is not supported see cmdType::ctGetValue, cmdType::ctSetValue, cmdType::ctValueSupported.
1562#define [GS_BAD_PARAM](#) 0xFFFFFFFFE
1563#define [GS_FALSE](#) 0x00
1564#define [GS_TRUE](#) 0x01
1565#define [GS_DISABLE](#) 0x00
1566#define [GS_ENABLE](#) 0x01
1567#define [GS_DEFAULT](#) 0xFF
1568#define [GS_FIELD](#) 0x00
Use field cmdType::ctGetValue, cmdType::ctSetValue (for pause/freeze as opposed to frame)
1569#define [GS_FIELD1](#) 0x01
Use field 1 cmdType::ctGetValue, cmdType::ctSetValue (for record/playback starts and edits)
1570#define [GS_FIELD2](#) 0x02
Use field 2 cmdType::ctGetValue, cmdType::ctSetValue (for record/playback starts and edits)
1571#define [GS_FRAME](#) 0x03
Use frame cmdType::ctGetValue, cmdType::ctSetValue (for pause/freeze as opposed to field)
1572#define [GS_UNITY](#) 0xFFFFFFFF
Set value to unity (levels, tbc) or default (compression type, amount)
1573#define [GS_ALPHACHROMA_SINGLE](#) 0x01
AvHAL input set normal SDI or Analog single link.
1574#define [GS_ALPHACHROMA_ALPHA](#) 0x02
AvHAL input set normal SDI plus a Y only SDI alpha plane.
1575#define [GS_ALPHACHROMA_DUAL](#) 0x04
Dual Link or HSDL input setup (2 HD-SDI 4:4:4 combined)
1576#define [GS_BITCOUNT_8](#) 0x01
Supports 8 bits per pixel component (normally YCbCr, for RGB see below)
1577#define [GS_BITCOUNT_10](#) 0x02
Supports 10 bits per pixel component (normally YCbCr, for RGB see below)
1578#define [GS_BITCOUNT_24](#) 0x04
Supports 3 (RGB) 8 bit components per pixel.
1579#define [GS_BITCOUNT_30](#) 0x08
Supports 3 (RGB) 10 bit components per pixel (e.g. standard DPX)
1580#define [GS_BITCOUNT_32](#) 0x10
Supports 4 (RGBA) 8 bit components per pixel (e.g. standard TGA)

1581#define [GS_BITCOUNT_12](#) 0x20
Supports YCbCr 4:2:0 (YUV) 8 bit components (e.g. i420, yv12) as well as bayer types.

1582#define [GS_BITCOUNT_14](#) 0x40
Supports Bayer types.

1583#define [GS_BITCOUNT_16](#) 0x80
Supports Bayer types.

1584#define [GS_FRAMEDROPMODE_NONE](#) 0x000000

1585#define [GS_FRAMEDROPMODE_VARICAM_MASK_FPS](#) 0x0000FF

1586#define [GS_FRAMEDROPMODE_VARICAM_2398](#) 0x000023

1587#define [GS_FRAMEDROPMODE_VARICAM_24](#) 0x000024

1588#define [GS_FRAMEDROPMODE_VARICAM_25](#) 0x000025

1589#define [GS_FRAMEDROPMODE_VARICAM_2997](#) 0x000029

1590#define [GS_FRAMEDROPMODE_VARICAM_30](#) 0x000030

1591#define [GS_FRAMEDROPMODE_VARICAM_50](#) 0x000050

1592#define [GS_FRAMEDROPMODE_VARICAM_5994](#) 0x000059

1593#define [GS_FRAMEDROPMODE_VARICAM_60](#) 0x000060

1594#define [GS_FRAMEDROPMODE_VARICAM_VARI](#) 0x000001

1595#define [GS_FRAMEDROPMODE_VARICAM_ILLEGAL](#) 0x0000FF

1596#define [GS_FRAMEDROPMODE_VARICAM_UB_INVERT](#) 0x000100

1597#define [GS_FRAMEDROPMODE_HALF](#) 0x010000

1598#define [GS_FILE_HAS_CHANGED_REMOTELY](#) 0xFFFFFFFF

1599#define [GS_ONTRAK_NONE](#) 0x00000000

1600#define [GS_ONTRAK_K0](#) 0x00000001

1601#define [GS_ONTRAK_K1](#) 0x00000002

1602#define [GS_ONTRAK_K2](#) 0x00000004

1603#define [GS_ONTRAK_K3](#) 0x00000008

1604#define [GS_ONTRAK_ILLEGAL](#) 0xFFFFFFFF

1605#define [GS_ONTRAK_PA0](#) 0x00000001

1606#define [GS_ONTRAK_PA1](#) 0x00000002

1607#define [GS_ONTRAK_PA2](#) 0x00000004

1608#define [GS_ONTRAK_PA3](#) 0x00000008

1609#define [COMPILE_ASSERT](#)(x) extern int __dummy[(int)x]

1610#define [CmdQueueElem](#) [MEDIACMD](#)
Old name, use [MEDIACMD](#) instead.

1611#define [pCmdQueueElem](#) [PMEDIACMD](#)
Old name, use [PMEDIACMD](#) instead.

1612#define [DECLARE_MEDIACMD](#)(_mCmd_)
Declare a media cmd structure to all pause.

1613#define [INIT_MEDIACMD](#)(_mCmd_)
Initialize a media cmd structure to all illegal (no command)

1614#define [INIT_PMEDIACMD](#)(_mCmd_)
Initialize a media cmd pointer to all illegal (no command)

1615#define [CMD_QUEUE_ELEMSIZE](#) ((size_t)&((pCmdQueueElem)(0))->arbID[0])
SizeOf a command queue without the arbID.

1616#define [SIZEOF_MEDIACMD_BASE](#) [CMD_QUEUE_ELEMSIZE](#)
SizeOf basic mediacmd structure without any clip id.

1617#define [SIZEOF_MEDIACMD_CLIPID](#) ([CMD_QUEUE_ELEMSIZE](#) + 9)
SizeOf mediacmd structure with a 8 unsigned char clip id and terminating 0.

1618#define [SIZEOF_MEDIACMD](#) sizeof([MEDIACMD](#))
SizeOf a complete mediacmd structure.

```

1619#define GS\_USERRIGHTS\_NONE 0x0000
1620#define GS\_USERRIGHTS\_READ 0x0001
1621#define GS\_USERRIGHTS\_MODIFY 0x0002
1622#define GS\_USERRIGHTS\_WRITE 0x0004
1623#define GS\_USERRIGHTS\_SETUP 0x0008
1624#define GS\_USERRIGHTS\_PLAY 0x0010
1625#define GS\_USERRIGHTS\_RECORD 0x0020
1626#define GS\_USERRIGHTS\_ADD 0x0100
1627#define GS\_USERRIGHTS\_DELETE 0x0200
1628#define GS\_USERRIGHTS\_FULL 0x7FFF
1629#define GS\_USERRIGHTS\_ADMIN 0x8000

```

```

1630#define GS\_SUPFILE\_AVI 0x00000001
1631#define GS\_SUPFILE\_ODML 0x00000002
1632#define GS\_SUPFILE\_OT 0x00000004
1633#define GS\_SUPFILE\_OMFI 0x00000008
1634#define GS\_SUPFILE\_FIX 0x00000100
1635#define GS\_SUPFILE\_AUDONLY 0x00010000
1636#define GS\_SUPFILE\_STILLS 0x00100000
1637#define GS\_SUPFILE\_UNK 0x40000000
1638#define GS\_SUPFILE\_ANY 0x80000000

```

Typedefs

```

1639typedef struct MEDIACMD * PMEDIACMD

```

Enumerations

```

1640enum cmdType { ctStop, ctPause, ctPlay, ctRecord, ctRecStop, ctEject, ctTransfer, ctInsert,
ctBlank, ctDelete, ctTrim, ctChanSelect, ctGetState, ctSetState, ctGetValue, ctSetValue,
ctValueSupported, ctError, ctTerminate, ctAbort, ctEdit, ctSwitch }
1641enum cmdFlags { cfDeferred = 1, cfOverrideDeferred = 1 << 30, cfTimeMs = 1 << 1,
cfTimeTarget = 1 << 2, cfTimeHouseClock = 1 << 3, cfUseSpeed = 1 << 4, cfUsePresets = 1
<< 5, cfUsePosition = 1 << 6, cfUsePositionOffset = 1 << 7, cfUseStart = 1 << 8,
cfUseStartOffset = 1 << 9, cfUseEnd = 1 << 10, cfUseEndOffset = 1 << 11, cfUseAllIDs = 1
<< 12, cfUseClipID = 1 << 13, cfCopy = 1 << 14, cfNoClipFiles = cfCopy, cfConvert = 1 <<
15, cfNoTCSpaces = cfConvert, cfUseCmdAlt = 1 << 16, cflsShuttle = 1 << 17, cfUsingCurrent
= 1 << 18, cfUseFrameCount = 1 << 19, cfFields = 1 << 20, cfRipple = 1 << 21, cfLoop = 1
<< 22, cfTrigger = 1 << 23, cfPreview = 1 << 24, cfRemoteCommand = 1 << 25, cfSecondField
= 1 << 27, cfUseNextField = cfSecondField, cfInvert = 1 << 28, cfTest = 1 << 29, cfNoReturn
= 1UL << 31UL }
1642enum cmdinf { infLtc = 1, infVite = 1 << 1, infSrcCtl = 1 << 2, infSrcLtc = 1 << 3, infSrcVite
= 1 << 4, infRecTime = 1 << 5, infRecDate = 1 << 6, infCC = 1 << 7, infAuth = 1 << 8,
infCopyright = 1 << 9, infOwner = 1 << 10, infSourceName = 1 << 11, infProxyName = 1 <<
12, inf13 = 1 << 13, inf14 = 1 << 14, inf15 = 1 << 15, inf16 = 1 << 16, inf17 = 1 << 17, inf18
= 1 << 18, inf19 = 1 << 19, inf20 = 1 << 20, inf21 = 1 << 21, infVB0 = 1 << 22, infVB1 = 1
<< 23, infVB2 = 1 << 24, infVB3 = 1 << 25, infVB4 = 1 << 26, infVB5 = 1 << 27, infVB6 = 1
<< 28, infVB7 = 1 << 29, infVB8 = 1 << 30, infVB9 = 1UL << 31UL }

1643enum cmdVidChan { vidChan0 = 1, vidChan1 = 1 << 1, vidChan2 = 1 << 2, vidChan3 = 1 <<
3, vidChan4 = 1 << 4, vidChan5 = 1 << 5, vidChan6 = 1 << 6, vidChan7 = 1 << 7, vidChan8 =
1 << 8, vidChan9 = 1 << 9, vidChan10 = 1 << 10, vidChan11 = 1 << 11, vidChan12 = 1 << 12,
vidChan13 = 1 << 13, vidChan14 = 1 << 14, vidChan15 = 1 << 15, vidChan16 = 1 << 16,
vidChan17 = 1 << 17, vidChan18 = 1 << 18, vidChan19 = 1 << 19, vidChan20 = 1 << 20,

```

[vidChan21](#) = 1 << 21, [vidChan22](#) = 1 << 22, [vidChan23](#) = 1 << 23, [vidChan24](#) = 1 << 24,
[vidChan25](#) = 1 << 25, [vidChan26](#) = 1 << 26, [vidChan27](#) = 1 << 27, [vidChan28](#) = 1 << 28,
[vidChan29](#) = 1 << 29, [vidChan30](#) = 1 << 30, [vidChan31](#) = 1UL << 31UL }

1644enum [cmdAudChan](#) { [audChan0](#) = 1, [audChan1](#) = 1 << 1, [audChan2](#) = 1 << 2, [audChan3](#) = 1
<< 3, [audChan4](#) = 1 << 4, [audChan5](#) = 1 << 5, [audChan6](#) = 1 << 6, [audChan7](#) = 1 << 7,
[audChan8](#) = 1 << 8, [audChan9](#) = 1 << 9, [audChan10](#) = 1 << 10, [audChan11](#) = 1 << 11,
[audChan12](#) = 1 << 12, [audChan13](#) = 1 << 13, [audChan14](#) = 1 << 14, [audChan15](#) = 1 << 15,
[audChan16](#) = 1 << 16, [audChan17](#) = 1 << 17, [audChan18](#) = 1 << 18, [audChan19](#) = 1 << 19,
[audChan20](#) = 1 << 20, [audChan21](#) = 1 << 21, [audChan22](#) = 1 << 22, [audChan23](#) = 1 << 23,
[audChan24](#) = 1 << 24, [audChan25](#) = 1 << 25, [audChan26](#) = 1 << 26, [audChan27](#) = 1 << 27,
[audChan28](#) = 1 << 28, [audChan29](#) = 1 << 29, [audChan30](#) = 1 << 30, [audChan31](#) = 1UL <<
31UL }

1645#define [gsAllowIndependantChanConfig](#) [gsAllowIndependentChanConfig](#)

Spelling error.

1646#define [GS_CLIPMODE_ILLEGAL](#) 0xFFFFFFFF

Known file types.

1647#define [GS_CLIPMODE_CLIPSPACE](#) 0

1648#define [GS_CLIPMODE_TCSPACE](#) 1

1649#define [GS_CLIPMODE_SINGLE](#) 2

1650#define [GS_CLIPMODE_FILM](#) 3

1651enum [cmdGetSetValue](#) { [gsTc](#) = 1, [gsUb](#), [gsLtcTc](#), [gsLtcUb](#), [gsVtcTc](#), [gsVtcUb](#), [gsTcSource](#),
[gsTcType](#), [gsStart](#), [gsEnd](#), [gsIn](#), [gsLastIn](#), [gsOut](#), [gsLastOut](#), [gsEditOn](#), [gsEditOff](#), [gsPreroll](#),
[gsPostroll](#), [gsAutoMode](#), [gsPlayDelay](#), [gsLtcTcPreset](#), [gsLtcUbPreset](#), [gsVtcTcPreset](#),
[gsVtcUbPreset](#), [gsFrameData](#), [gsKeyCode](#), [gsInkCode](#), [gs215Code](#), [gsHeadsAndTails](#),
[gsTimecodeSources](#), [gsCCSetup](#), [gsDisableTimecode](#), [gsVITCSourcePrecedence](#),
[gsLTCSOURCEPrecedence](#), [gsGetNextClip](#) = 90, [gsFirstClip](#), [gsNextClip](#), [gsTCSGetTLClipState](#),
[gsTCSGetTLClipInfo](#), [gsClipInfo](#), [gsClipCopy](#), [gsTcClipInfo](#), [gsAudChan](#) = 100, [gsVidChan](#),
[gsInfChan](#), [gsAudSelect](#), [gsVidSelect](#), [gsInfSelect](#), [gsAudEdit](#), [gsVidEdit](#), [gsInfEdit](#), [gsEditMode](#),
[gsMetaData](#) = 150, [gsMetaDataDirectory](#), [gsMetaDataVolume](#), [gsMetaDataCurrentUser](#),
[gsMetaDataLocalMachine](#), [gsMetaDataGlobal](#), [gsMetaDataReadWrite](#), [gsMetaBaseOpen](#),
[gsMetaBaseCreate](#), [gsMetaBaseClose](#), [gsMetaBaseFileCount](#), [gsMetaBaseFileName](#),
[gsMetaBaseFileRemove](#), [gsMetaBaseFileAdd](#), [gsMetaBaseTagIndex](#), [gsMetaBaseGetTag](#),
[gsMetaBaseSetTag](#), [gsMetaBaseDefaultTags](#), [gsMetaBaseFileRename](#), [gsMetaBaseReplayMark](#),
[gsMetaBaseGetName](#), [gsMetaBaseGetTable](#), [gsMetaBaseResetAndRecord](#), [gsAudInSelect](#) = 200,
[gsAudOutSelect](#), [gsAudInputLevel](#), [gsAudOutputLevel](#), [gsAudAdvanceLevel](#), [gsAudOutPhase](#),
[gsAudOutAdvancePhase](#), [gsAudCrossFadeTime](#), [gsAudLtcEnable](#), [gsAudInLtcChannel](#),
[gsAudOutLtcChannel](#), [gsAudDtmfEnable](#), [gsAudInDtmfChannel](#), [gsAudOutDtmfChannel](#),
[gsAudWavePeakRMS](#), [gsAudInputBitRate](#), [gsAudInputSampleRate](#), [gsAudInputMode](#),
[gsAudInputHeadRoom](#), [gsAudInputOriginal](#), [gsAudInputErrorProtect](#), [gsAudInputCopyright](#),
[gsAudInputSlave](#), [gsAudInputBass](#), [gsAudInputTreble](#), [gsAudInputStatus](#), [gsAudioMappingInput](#),
[gsAudioMappingOutput](#), [gsAudMonitorSelect](#), [gsAudChannelsEncoded](#), [gsAudAudioScrub](#),
[gsVidFreeze](#) = 300, [gsVidPreReadMode](#), [gsVidEditField](#), [gsVidRecFrame](#), [gsVidPlayFrame](#),
[gsVidNoEE](#), [gsVidSuperimpose](#), [gsVidAnalogMonitorSDType](#), [gsVidAnalogMonitorHDType](#),
[gsVidAnalogMonitorMethod](#), [gsVidAnalogMonitorUpMode](#), [gsVidAnalogMonitorDownMode](#),
[gsVidPanScanZoom](#), [gsVidSlowMotionMode](#), [gsVidVariCamMode](#), [gsVidCustomSuperimpose](#),
[gsVidInSelect](#) = 400, [gsVidInLockType](#), [gsVidInSetup](#), [gsVidInVideo](#), [gsVidInHue](#),
[gsVidInChroma](#), [gsVidInUChroma](#), [gsVidInVChroma](#), [gsVidInColorKiller](#), [gsVidInAGC](#),
[gsVidInBandwidth](#), [gsVidInBlack](#), [gsVidInWhite](#), [gsVidInCoring](#), [gsVidInPeaking](#),
[gsVidInSharpness](#), [gsVidInGamma](#), [gsVidInSignalFormat](#), [gsVidInQuality](#), [gsVidSetup](#) = 500,
[gsVidVideo](#), [gsVidHue](#), [gsVidChroma](#), [gsVidUChroma](#), [gsVidVChroma](#), [gsVidBandwidth](#),
[gsVidBlackSetup](#), [gsVidColor](#), [gsVidInputRouting](#), [gsVidOutSelect](#) = 600, [gsVidOutGenlock](#),
[gsVidOutGenlockSource](#), [gsVidOutLockType](#), [gsVidOutHPhase](#), [gsVidOutSubCarrier](#),
[gsVidOutCoring](#), [gsVidOutPeaking](#), [gsVidOutAdjust1](#), [gsVidOutAdjust2](#),

[gsVidOutGenlockDelay](#), [gsVidOutLockSignalFormat](#), [gsVidOutDisableDualLink](#),
[gsVidOutReferenceWipeMix](#), [gsDisableGenlockForInfiniteLoop](#), [gsCompChVerticalRes](#) = 700,
[gsMpegVerticalRes](#) = [gsCompChVerticalRes](#), [gsCompChHorizontalRes](#), [gsMpegHorizontalRes](#) =
[gsCompChHorizontalRes](#), [gsCompChChromaFormat](#), [gsMpegChromaFormat](#) =
[gsCompChChromaFormat](#), [gsCompChDCPrecision](#), [gsMpegDCPrecision](#) =
[gsCompChDCPrecision](#), [gsCompChAspectRatio](#), [gsMpegAspectRatio](#) = [gsCompChAspectRatio](#),
[gsCompChStandard](#), [gsMpegStandard](#) = [gsCompChStandard](#), [gsCompChLanguageCode](#),
[gsMpegLanguageCode](#) = [gsCompChLanguageCode](#), [gsCompChCCFormat](#), [gsMpegCCFormat](#) =
[gsCompChCCFormat](#), [gsCompChConcealmentVector](#), [gsMpegConcealmentVector](#) =
[gsCompChConcealmentVector](#), [gsCompChClosedGop](#), [gsMpegClosedGop](#) =
[gsCompChClosedGop](#), [gsCompChAdjustGopTC](#), [gsMpegAdjustGopTC](#) =
[gsCompChAdjustGopTC](#), [gsCompChAltCoEffTable](#), [gsMpegAltCoEffTable](#) =
[gsCompChAltCoEffTable](#), [gsCompChNonLinearQuant](#), [gsMpegNonLinearQuant](#) =
[gsCompChNonLinearQuant](#), [gsCompChMuxRate](#), [gsMpegMuxRate](#) = [gsCompChMuxRate](#),
[gsCompChAudPacketSize](#), [gsMpegAudPacketSize](#) = [gsCompChAudPacketSize](#),
[gsCompChVidPacketSize](#), [gsMpegVidPacketSize](#) = [gsCompChVidPacketSize](#),
[gsCompChAudioStreamID](#), [gsMpegAudioStreamID](#) = [gsCompChAudioStreamID](#),
[gsCompChVideoStreamID](#), [gsMpegVideoStreamID](#) = [gsCompChVideoStreamID](#),
[gsCompChAudioStreamPID](#), [gsMpegAudioStreamPID](#) = [gsCompChAudioStreamPID](#),
[gsCompChVideoStreamPID](#), [gsMpegVideoStreamPID](#) = [gsCompChVideoStreamPID](#),
[gsCompChAllowSettings](#), [gsMpegAllowSettings](#) = [gsCompChAllowSettings](#),
[gsCompChFourCC](#), [gsCompChBitCount](#), [gsCompChSizeImage](#), [gsCompChRate](#),
[gsCompChScale](#), [gsCompChPitch](#), [gsVideoEncodeFormat](#), [gsAudioEncodeFormat](#),
[gsCompChannelChangeMs](#), [gsAlphaChromaSource](#), [gsCompressionType](#), [gsVideoStandard](#),
[gsResetChannel](#), [gsEnableHDSDFormat](#), [gsVBlankEnable](#), [gsLUTEnable](#), [gsAudioFileType](#),
[gsAudioBitSize](#), [gsAudioFrequency](#), [gsEnableOverlappedWrites](#), [gsMatchOutputToClip](#),
[gsAllowIndependentChanConfig](#), [gsSignalFormat](#) = 900, [gsCompType](#), [gsCompRateSize](#),
[gsCompRateRatio](#), [gsCompRatePercent](#), [gsCompGOPSize](#), [gsCompIFactor](#), [gsCompBFactor](#),
[gsCompPFactor](#), [gsCompRefPeriod](#), [gsTotalStorageAvail](#), [gsTotalStorageFree](#), [gsTotalTimeAvail](#),
[gsTotalTimeFree](#), [gsVtrType](#), [gsHSDITransferType](#), [gsLocal](#) = 1000, [gsSupportedFileTypes](#),
[gsIgnoreFileTypes](#), [gsRecInhibit](#), [gsRecDrive](#), [gsRecFileName](#), [gsRecRate](#), [gsRecFileFormat](#),
[gsRecAudFileFormat](#), [gsDelInhibit](#), [gsInsInhibit](#), [gsConvertFileFormat](#), [gsConvertAudFileFormat](#),
[gsDefStillLen](#), [gsSysTime](#), [gsDSyncMs](#), [gsHwPort](#), [gsPBEE](#), [gsServoRefSelect](#), [gsHeadSelect](#),
[gsColorFrame](#), [gsVidRefDisable](#), [gsPlayCountDelay](#), [gsEmulateEditBumping](#),
[cmdaltNearestKeyFrame](#), [cmdaltNextKeyFrame](#), [cmdaltPrevKeyFrame](#), [cmdaltStartOfMessage](#),
[gsVidInputValid](#), [gsVidGenlockValid](#), [gsSerialEditMode](#), [gsSerialProtocols](#), [gsPauseBeforeStop](#),
[gsPauseDelay](#), [gsFrontPanel](#), [gsFrontPanelComPort](#), [gsVDCPPreroll](#), [gsChannelAdd](#) = 2000,
[gsChannelDel](#), [gsChannelEnable](#), [gsChannelAddress](#), [gsChannelPort](#), [gsChannelComPort](#),
[gsChannelTarget](#), [gsChannelPath](#), [gsChannelType](#), [gsChannelUserName](#), [gsChannelPassWord](#),
[gsUserData0](#) = 3000, [gsUserData1](#), [gsUserData2](#), [gsUserData3](#), [gsUserData4](#), [gsUserData5](#),
[gsUserData6](#), [gsUserData7](#), [gsUserData8](#), [gsUserData9](#), [gsErrorLog](#) = 10000, [gsErrorLogName](#),
[gsErrorLogStartMs](#), [gsErrorLogCurrentMs](#), [gsErrorLogLastChange](#), [gsErrorLogMessage](#),
[gsSysBufferLevel](#) = 20000, [gsSysMemoryUsage](#), [gsSysCPUUsage](#), [gsDroppedFrames](#),
[gsProxyMode](#) = 50000, [gsProxyStatus](#), [gsGetNextProxy](#), [gsAddProxy](#), [gsPromoteProxy](#),
[gsRemoveProxy](#), [gsProxyCPUUsage](#), [gsTransferToArchive](#), [gsTransferFromArchive](#),
[gsGetNextArchiveClip](#), [gsGetNextTransferToArchiveClip](#), [gsGetNextTransferFromArchiveClip](#),
[gsAddProxyAndOutputName](#), [gsVWVVersion](#) = 60000, [gsMEVersion](#), [gsVWVType](#),
[gsVWVChannelType](#), [gsVWVChannelName](#), [gsVWVLicense](#), [gsMonitor](#) = 64000,
[gsMonitorHwnds](#), [gsHwnds](#) = [gsMonitorHwnds](#), [gsMonitorDisplay](#), [gsMonitorGrab](#),
[gsUtilityMonitorDraw](#), [gsUtilityMonitorDrawSetup](#), [gsWaveVectorSetup](#), [gsWaveVectorType](#),
[gsWaveVectorArea](#), [gsWaveVectorLastChangeMs](#), [gsMonitorLoadBuffers](#), [gsDirGetList](#) =
64250, [gsDirGetInfo](#), [gsDirGetFileInfo](#), [gsDirGetFileGrab](#), [gsVgaDisplayEnable](#),
[gsVgaDirectXConfig](#), [gsVga3DConfig](#), [gsVga3DWipeType](#), [gsVga3DMix](#), [gsVga3DThreshold](#),
[gsVga3DSplitHorizontal](#), [gsVga3DSplitVertical](#), [gsVga3DGridSize](#), [gsVga3DGridType](#),
[gsVgaFullscreenEnable](#), [gsLimitAvailableChannels](#), [gsVGAZoomPan](#), [gsChannelsExist](#) = 65536,
[gsClipMode](#), [gsRecOffset](#), [gsChanCapabilities](#), [gsLastChangeMs](#), [gsGPIIn](#), [gsGPIOut](#),

[gsCurrentMs](#), [gsClipModePreroll](#), [gsClipModeBackup](#), [gsSaveCurrent](#) = 10000,
[gsLoadClipSpace](#), [gsLoadTCSpace](#), [gsLoadFilmSpace](#), [gsLoadEditSpace](#),
[gsSaveClipSpaceToDisk](#), [gsSaveTCSpaceToDisk](#), [gsUserLogIn](#) = 900000, [gsUserLastChangeMs](#),
[gsUserList](#), [gsUserAdd](#), [gsUserDel](#), [gsUserRights](#), [gsUserPasswd](#), [gsPiconFrame](#) = 1000000,
[gsJpegFrame](#), [gsImageDirectory](#), [gsFrameInfo](#), [gsRawFrame](#), [gsPreallocateEditFile](#),
[gsCreateEditFile](#), [gsPreviewFrame](#), [gsVWVService](#) = 1100000, [gsInsertQueue](#), [gsXlatQueue](#),
[gsXMLRateScale](#), [gsXMLFileProperties](#), [gsDTProjectToXml](#), [gsApplicationID](#), [gsInlay](#),
[gsInlayFile](#), [gsInlaySourceArea](#), [gsInlayDestinationArea](#), [gsInlayOffset](#), [gsInlayTcType](#),
[gsExportClip](#), [gsExportClipDirectory](#), [gsExportFileName](#), [gsImportClipDirectory](#),
[gsFileSegmentSize](#), [gsRecordFileUpdateFrames](#), [gsCheckFrameTimeStamp](#),
[gsShutdownApplication](#) = 2147418112, [gsShutdownSystem](#), [gsCleanRecordWipeDrive](#),
[gsInstallSystem](#), [gsInternalGetImageOffset](#) = 0xFFFFFFFFDUL }

Define Documentation

#define audChanAll 0xFFFFFFFFUL

Definition at line 464 of file `mediacmd.h`.

#define AUDIOWRITETYPE_AIFF 0x00000010

Audio write type aiff.

Definition at line 5625 of file `mediacmd.h`.

**#define AUDIOWRITETYPE_AIFF_INTERNAL (AUDIOWRITETYPE_AIFF|
AUDIOWRITETYPE_INTERNAL)**

Definition at line 5633 of file `mediacmd.h`.

**#define AUDIOWRITETYPE_AIFF_STEREO (AUDIOWRITETYPE_AIFF|
AUDIOWRITETYPE_STEREO)**

Definition at line 5634 of file `mediacmd.h`.

#define AUDIOWRITETYPE_INTERNAL 0x00000000

Audio write type internal.

Definition at line 5623 of file `mediacmd.h`.

#define AUDIOWRITETYPE_MONO 0x00000002

Audio in mono channels.

Definition at line 5619 of file `mediacmd.h`.

#define AUDIOWRITETYPE_MULTI 0x00000004

Audio in multichannel.

Definition at line 5621 of file mediacmd.h.

#define AUDIOWRITETYPE_STEREO 0x00000001

Audio in stereo channels.

Definition at line 5617 of file mediacmd.h.

#define AUDIOWRITETYPE_WAVE 0x00000020

Audio write type wave.

Definition at line 5627 of file mediacmd.h.

**#define AUDIOWRITETYPE_WAVE_INTERNAL (AUDIOWRITETYPE_WAVE|
AUDIOWRITETYPE_INTERNAL)**

Definition at line 5629 of file mediacmd.h.

**#define AUDIOWRITETYPE_WAVE_MONO (AUDIOWRITETYPE_WAVE|
AUDIOWRITETYPE_MONO)**

Definition at line 5631 of file mediacmd.h.

**#define AUDIOWRITETYPE_WAVE_MULTI (AUDIOWRITETYPE_WAVE|
AUDIOWRITETYPE_MULTI)**

Definition at line 5632 of file mediacmd.h.

**#define AUDIOWRITETYPE_WAVE_STEREO (AUDIOWRITETYPE_WAVE|
AUDIOWRITETYPE_STEREO)**

Definition at line 5630 of file mediacmd.h.

#define cfOverrideDeferred cfOverrideDeferred

Spelling.

Definition at line 435 of file mediacmd.h.

#define CHAN_ILLEGAL 0xFFFFFFFF

Illegal channel, or All Channels. Set [MEDIACMD::dwAudioChannels](#), [MEDIACMD::dwVideoChannels](#), [MEDIACMD::dwInfoChannels](#) to this if not used.

3D VGA view absolute difference (for cmdGetSetValue::gsVga3DConfig)
Definition at line 6602 of file mediacmd.h.

#define GS_3DVGA_DISSOLVE 0x00008000

3D VGA dissolve between (for cmdGetSetValue::gsVga3DConfig)
Definition at line 6616 of file mediacmd.h.

#define GS_3DVGA_FLAG_ADDGRID 0x01000000

3D VGA view flag to add a grid (for cmdGetSetValue::gsVga3DConfig)
Definition at line 6626 of file mediacmd.h.

#define GS_3DVGA_FLAG_FLIPLEFTHORIZ 0x20000000

3D VGA view flag to flip left horizontally (for cmdGetSetValue::gsVga3DConfig)
Definition at line 6634 of file mediacmd.h.

#define GS_3DVGA_FLAG_FLIPLEFTVERT 0x40000000

3D VGA view flag to flip left vertically (for cmdGetSetValue::gsVga3DConfig)
Definition at line 6630 of file mediacmd.h.

#define GS_3DVGA_FLAG_FLIPRIGHTHORIZ 0x02000000

3D VGA view flag to flip left horizontally (for cmdGetSetValue::gsVga3DConfig)
Definition at line 6636 of file mediacmd.h.

#define GS_3DVGA_FLAG_FLIPRIGHTVERT 0x04000000

3D VGA view flag to flip right vertically (for cmdGetSetValue::gsVga3DConfig)
Definition at line 6632 of file mediacmd.h.

#define GS_3DVGA_FLAG_INVERT 0x80000000

3D VGA view flag to invert left/right (for cmdGetSetValue::gsVga3DConfig)
Definition at line 6628 of file mediacmd.h.

#define GS_3DVGA_FLAG_LENTICULAR GS_3DVGA_FLAG_SPLITVERT

Definition at line 6640 of file mediacmd.h.

#define GS_3DVGA_FLAG_SPLITVERT 0x10000000

3D VGA view split flags make it a vertical split/mirror/butterfly
Definition at line 6638 of file mediacmd.h.

#define GS_3DVGA_FLAGS_MASK 0xFF000000

3D VGA viewing mode MASK for flags (for cmdGetSetValue::gsVga3DConfig)
Definition at line 6584 of file mediacmd.h.

#define GS_3DVGA_INTERLACED 0x00000040

3D VGA view interlaced (Zalman, real 3D, IMAX) (for cmdGetSetValue::gsVga3DConfig)
Definition at line 6598 of file mediacmd.h.

#define GS_3DVGA_LEFTEYE 0x00000001

3D VGA view left eye only (top picture) (for cmdGetSetValue::gsVga3DConfig)
Definition at line 6586 of file mediacmd.h.

#define GS_3DVGA_LUMA_DIFF 0x00020000

3D VGA luma invert diff (invert luma of second frame, then diff)
Definition at line 6620 of file mediacmd.h.

#define GS_3DVGA_MIRROR 0x00001000

3D VGA view squeeze and mirror (for cmdGetSetValue::gsVga3DConfig)
Definition at line 6610 of file mediacmd.h.

#define GS_3DVGA_ONIONSKIN 0x00000080

3D VGA view onion skin like 2D animation programs (for
cmdGetSetValue::gsVga3DConfig)
Definition at line 6600 of file mediacmd.h.

#define GS_3DVGA_OVERUNDER 0x00000200

3D VGA view images on top and bottom (squeeze vert) (for
cmdGetSetValue::gsVga3DConfig)
Definition at line 6604 of file mediacmd.h.

#define GS_3DVGA_RIGHTEYE 0x00000002

3D VGA view right eye only (bottom picture) (for cmdGetSetValue::gsVga3DConfig)
Definition at line 6588 of file mediacmd.h.

#define GS_3DVGA_SIDEBYSIDE 0x00000400

3D VGA view image next to each other (squeeze horiz) (for cmdGetSetValue::gsVga3DConfig)
Definition at line 6606 of file mediacmd.h.

#define GS_3DVGA_SPLIT 0x00000800

3D VGA view arbitrary split (for cmdGetSetValue::gsVga3DConfig)
Definition at line 6608 of file mediacmd.h.

#define GS_3DVGA_TYPE_MASK 0x00FFFFFF

3D VGA viewing mode MASK for all modes (for cmdGetSetValue::gsVga3DConfig)
Definition at line 6582 of file mediacmd.h.

#define GS_3DVGA_WIPE 0x00010000

3D VGA wipe (smpte +) (for cmdGetSetValue::gsVga3DConfig)
Definition at line 6618 of file mediacmd.h.

#define GS_ALPHACHROMA_ALPHA 0x02

AvHAL input set normal SDI plus a Y only SDI alpha plane.
Definition at line 6739 of file mediacmd.h.

#define GS_ALPHACHROMA_DUAL 0x04

Dual Link or HSDL input setup (2 HD-SDI 4:4:4 combined)
Definition at line 6741 of file mediacmd.h.

#define GS_ALPHACHROMA_SINGLE 0x01

AvHAL input set normal SDI or Analog single link.
Definition at line 6737 of file mediacmd.h.

#define GS_ANALOGMONITORMETHOD_DIRECT 0x0001

Keep analog monitor in line with digital (HD=HD, SD=SD)
Definition at line 5410 of file mediacmd.h.

#define GS_ANALOGMONITORMETHOD_FLIP1080 0x0020

SD->HD720 and HD->SD.
Definition at line 5420 of file mediacmd.h.

#define GS_ANALOGMONITORMETHOD_FLIP720 0x0010

SD->HD720 and HD->SD.
Definition at line 5418 of file mediacmd.h.

#define GS_ANALOGMONITORMETHOD_HD1080 0x0008

Convert everything to the nearest 1080 HD type.
Definition at line 5416 of file mediacmd.h.

#define GS_ANALOGMONITORMETHOD_HD720 0x0004

Convert everything to the nearest 720 HD type.
Definition at line 5414 of file mediacmd.h.

#define GS_ANALOGMONITORMETHOD_HSDL 0x0040

HD / SD -> HDSL.
Definition at line 5422 of file mediacmd.h.

#define GS_ANALOGMONITORMETHOD_SD 0x0002

Convert everything to the nearest SD type.
Definition at line 5412 of file mediacmd.h.

#define GS_APP_BBREPLAY 0x00000040

For cmdGetSetValue::gsApplicationID set to no specific application.
Definition at line 6678 of file mediacmd.h.

#define GS_APP_DTOUCH 0x00000020

For cmdGetSetValue::gsApplicationID set to no specific application.
Definition at line 6676 of file mediacmd.h.

#define GS_APP_DTREPLAYLIVE 0x00000010

For cmdGetSetValue::gsApplicationID set to no specific application.
Definition at line 6674 of file mediacmd.h.

#define GS_APP_MEDIANXS 0x00000008

For cmdGetSetValue::gsApplicationID set to no specific application.
Definition at line 6672 of file mediacmd.h.

#define GS_APP_NONE 0x00000000

For cmdGetSetValue::gsApplicationID set to no specific application.
Definition at line 6664 of file mediacmd.h.

#define GS_APP_QUICKCLIP 0x00000001

For cmdGetSetValue::gsApplicationID set to no specific application.
Definition at line 6666 of file mediacmd.h.

#define GS_APP_QUICKCLIPXO 0x00000002

For cmdGetSetValue::gsApplicationID set to no specific application.
Definition at line 6668 of file mediacmd.h.

#define GS_APP_VTRID 0x00000004

For cmdGetSetValue::gsApplicationID set to no specific application.
Definition at line 6670 of file mediacmd.h.

#define GS_ASPECT_RATIO_16x9 0x4

Aspect ratio 16:9.
Definition at line 5497 of file mediacmd.h.

#define GS_ASPECT_RATIO_2_21x1 0x8

Aspect ratio 2.21:1.
Definition at line 5499 of file mediacmd.h.

#define GS_ASPECT_RATIO_4x3 0x2

Aspect ratio 4:3.

Definition at line 5495 of file mediacmd.h.

#define GS_ASPECT_RATIO_SQUARE 0x1

Aspect ratio square.

Definition at line 5493 of file mediacmd.h.

#define GS_ASSEMBLE_EDIT 0x02

Set cmdGetSetValue::gsEditMode for an assemble edit.

Definition at line 5249 of file mediacmd.h.

#define GS_AUD_BIT_RATE_112000 0x0080

Definition at line 5282 of file mediacmd.h.

#define GS_AUD_BIT_RATE_128000 0x0100

Definition at line 5283 of file mediacmd.h.

#define GS_AUD_BIT_RATE_160000 0x0200

Definition at line 5284 of file mediacmd.h.

#define GS_AUD_BIT_RATE_192000 0x0400

Definition at line 5285 of file mediacmd.h.

#define GS_AUD_BIT_RATE_224000 0x0800

Definition at line 5286 of file mediacmd.h.

#define GS_AUD_BIT_RATE_256000 0x1000

Definition at line 5287 of file mediacmd.h.

#define GS_AUD_BIT_RATE_32000 0x0001

Definition at line 5275 of file mediacmd.h.

#define GS_AUD_BIT_RATE_320000 0x2000

Definition at line 5288 of file mediacmd.h.

#define GS_AUD_BIT_RATE_384000 0x4000

Definition at line 5289 of file mediacmd.h.

#define GS_AUD_BIT_RATE_41100 0x0002

Definition at line 5276 of file mediacmd.h.

#define GS_AUD_BIT_RATE_48000 0x0004

Definition at line 5277 of file mediacmd.h.

#define GS_AUD_BIT_RATE_56000 0x0008

Definition at line 5278 of file mediacmd.h.

#define GS_AUD_BIT_RATE_64000 0x0010

Definition at line 5279 of file mediacmd.h.

#define GS_AUD_BIT_RATE_80000 0x0020

Definition at line 5280 of file mediacmd.h.

#define GS_AUD_BIT_RATE_96000 0x0040

Definition at line 5281 of file mediacmd.h.

#define GS_AUD_DUAL 0x004

Definition at line 5293 of file mediacmd.h.

#define GS_AUD_HEADROOM_18 0x01

Definition at line 5296 of file mediacmd.h.

#define GS_AUD_HEADROOM_20 0x02

Definition at line 5297 of file mediacmd.h.

#define GS_AUD_JOINT_STEREO 0x002

Definition at line 5292 of file mediacmd.h.

#define GS_AUD_MULTIPLE 0x010

Definition at line 5295 of file mediacmd.h.

#define GS_AUD_SINGLE 0x008

Definition at line 5294 of file mediacmd.h.

#define GS_AUD_STEREO 0x001

Definition at line 5291 of file mediacmd.h.

#define GS_AUDSELECT_AES_EBU 0x200

Audio in/out digital balanced with clock (cmdGetSetValue::gsAudInSelect
cmdGetSetValue::gsAudOutSelect)

Definition at line 5262 of file mediacmd.h.

#define GS_AUDSELECT_AES_EBU_PRO 0x800

Audio in/out digital balanced with clock (cmdGetSetValue::gsAudInSelect
cmdGetSetValue::gsAudOutSelect)

Definition at line 5266 of file mediacmd.h.

#define GS_AUDSELECT_BALANCED_10 0x010

Audio in/out balanced (XLR connector) 600ohm impedance at -10db
(cmdGetSetValue::gsAudInSelect cmdGetSetValue::gsAudOutSelect)

Definition at line 5256 of file mediacmd.h.

#define GS_AUDSELECT_BALANCED_4 0x020

Audio in/out balanced (XLR connector) 600ohm impedance at +4db
(cmdGetSetValue::gsAudInSelect cmdGetSetValue::gsAudOutSelect)

Definition at line 5258 of file mediacmd.h.

#define GS_AUDSELECT_EMBEDDED 0x400

Audio in/out embedded in SDI or HD-SDI video signal (cmdGetSetValue::gsAudInSelect
cmdGetSetValue::gsAudOutSelect)

Definition at line 5264 of file mediacmd.h.

#define GS_AUDSELECT_HDMI 0x1000

Use audio embedded in the HDMI signal.

Definition at line 5268 of file mediacmd.h.

#define GS_AUDSELECT_NONE 0

No audio in/out available, or cannot be configured (cmdGetSetValue::gsAudInSelect cmdGetSetValue::gsAudOutSelect)

Definition at line 5270 of file mediacmd.h.

#define GS_AUDSELECT_SILENT 0x040

No Audio Selected leave silent.

Definition at line 5272 of file mediacmd.h.

#define GS_AUDSELECT_SPDIF 0x100

Audio in/out digital single wire (cmdGetSetValue::gsAudInSelect cmdGetSetValue::gsAudOutSelect)

Definition at line 5260 of file mediacmd.h.

#define GS_AUDSELECT_UNBALANCED_10 0x001

Audio in/out unbalanced (RCA connector) high impedance at -10db (cmdGetSetValue::gsAudInSelect cmdGetSetValue::gsAudOutSelect)

Definition at line 5252 of file mediacmd.h.

#define GS_AUDSELECT_UNBALANCED_4 0x002

Audio in/out unbalanced (RCA connector) high impedance at -4db (cmdGetSetValue::gsAudInSelect cmdGetSetValue::gsAudOutSelect)

Definition at line 5254 of file mediacmd.h.

#define GS_BAD_PARAM 0xFFFFFFFF

Parameter is bad see cmdType::ctGetValue, cmdType::ctSetValue, cmdType::ctValueSupported and [MEDIACMD::dwPosition](#), [MEDIACMD::dwStart](#) and [MEDIACMD::dwEnd](#)

Definition at line 6704 of file mediacmd.h.

#define GS_BITCOUNT_10 0x02

Supports 10 bits per pixel component (normally YCbCr, for RGB see below)

Definition at line 6746 of file mediacmd.h.

#define GS_BITCOUNT_12 0x20

Supports YCbCr 4:2:0 (YUV) 8 bit components (e.g. i420, yv12) as well as bayer types.

Definition at line 6754 of file mediacmd.h.

#define GS_BITCOUNT_14 0x40

Supports Bayer types.

Definition at line 6756 of file mediacmd.h.

#define GS_BITCOUNT_16 0x80

Supports Bayer types.

Definition at line 6758 of file mediacmd.h.

#define GS_BITCOUNT_24 0x04

Supports 3 (RGB) 8 bit components per pixel.

Definition at line 6748 of file mediacmd.h.

#define GS_BITCOUNT_30 0x08

Supports 3 (RGB) 10 bit components per pixel (e.g. standard DPX)

Definition at line 6750 of file mediacmd.h.

#define GS_BITCOUNT_32 0x10

Supports 4 (RGBA) 8 bit components per pixel (e.g. standard TGA)

Definition at line 6752 of file mediacmd.h.

#define GS_BITCOUNT_8 0x01

Supports 8 bits per pixel component (normally YCbCr, for RGB see below)

Definition at line 6744 of file mediacmd.h.

#define GS_CC_708 0x1000

CEA 708 cmdGetSetValue::gsCCSetup.

Definition at line 5211 of file mediacmd.h.

#define GS_CC_CC1 0x0001

CC1 cmdGetSetValue::gsCCSetup.
Definition at line 5193 of file mediacmd.h.

#define GS_CC_CC2 0x0004

CC2 cmdGetSetValue::gsCCSetup.
Definition at line 5195 of file mediacmd.h.

#define GS_CC_CC3 0x0008

CC3 cmdGetSetValue::gsCCSetup.
Definition at line 5197 of file mediacmd.h.

#define GS_CC_CC4 0x0010

CC4 cmdGetSetValue::gsCCSetup.
Definition at line 5199 of file mediacmd.h.

#define GS_CC_DISABLE 0x0000

No CC cmdGetSetValue::gsCCSetup.
Definition at line 5191 of file mediacmd.h.

#define GS_CC_TEXT1 0x0020

Text1 cmdGetSetValue::gsCCSetup.
Definition at line 5201 of file mediacmd.h.

#define GS_CC_TEXT2 0x0040

Text2 cmdGetSetValue::gsCCSetup.
Definition at line 5203 of file mediacmd.h.

#define GS_CC_TEXT3 0x0080

Text3 cmdGetSetValue::gsCCSetup.
Definition at line 5205 of file mediacmd.h.

#define GS_CC_TEXT4 0x0100

Text4 cmdGetSetValue::gsCCSetup.
Definition at line 5207 of file mediacmd.h.

#define GS_CC_XDS 0x0200

XDS cmdGetSetValue::gsCCSetup.
Definition at line 5209 of file mediacmd.h.

#define GS_CHANCAP_ALL 0x3FFFFFFF

Channel can do anything except MPEG_ENC (by default this should not be)
Definition at line 6466 of file mediacmd.h.

#define GS_CHANCAP_AUDPREVIEW 0x00200000

Channel can preview audio on a multi media card (video or audio or both)
(gsGetSetValue::gsChanCapabilities)
Definition at line 6451 of file mediacmd.h.

#define GS_CHANCAP_CLIPSPACE 0x10000000

Channel can act like a clip space (video or audio or both)
(gsGetSetValue::gsChanCapabilities)
Definition at line 6457 of file mediacmd.h.

#define GS_CHANCAP_EDIT 0x00008000

Channel can edit from in to out (video or audio or both) (gsGetSetValue::gsChanCapabilities)
Definition at line 6439 of file mediacmd.h.

#define GS_CHANCAP_EJECT 0x00040000

Channel can eject the media (video or audio or both) (gsGetSetValue::gsChanCapabilities)
Definition at line 6445 of file mediacmd.h.

#define GS_CHANCAP_ERROR 0x80000000

Do not use this bit - indicates error.
Definition at line 6464 of file mediacmd.h.

#define GS_CHANCAP_ETOE 0x00002000

Channel can pass through video (in stop) (video or audio or both)

(gsGetSetValue::gsChanCapabilities)
Definition at line 6435 of file mediacmd.h.

#define GS_CHANCAP_FILE 0x01000000

Channel can play from a file (video or audio or both) (gsGetSetValue::gsChanCapabilities)
Definition at line 6453 of file mediacmd.h.

#define GS_CHANCAP_JOG 0x00000008

Channel can jog below play speed (video or audio or both)
(gsGetSetValue::gsChanCapabilities)
Definition at line 6425 of file mediacmd.h.

#define GS_CHANCAP_LOOP 0x00100000

Channel can play in a loop (video or audio or both) (gsGetSetValue::gsChanCapabilities)
Definition at line 6447 of file mediacmd.h.

#define GS_CHANCAP_MPEG_ENC 0x40000000

Channel can be configured as MPEG -- opens a whole bunch of settings on the options
(specifically for Argus board right now).
Definition at line 6462 of file mediacmd.h.

#define GS_CHANCAP_NET 0x02000000

Channel can play from a network feed (video or audio or both)
(gsGetSetValue::gsChanCapabilities)
Definition at line 6455 of file mediacmd.h.

#define GS_CHANCAP_PAUSE 0x00000004

Channel can pause and display frame (video or audio or both)
(gsGetSetValue::gsChanCapabilities)
Definition at line 6423 of file mediacmd.h.

#define GS_CHANCAP_PLAY 0x00000001

Channel can play (video or audio or both) (gsGetSetValue::gsChanCapabilities)
Definition at line 6419 of file mediacmd.h.

#define GS_CHANCAP_PREVIEW 0x00000040

Channel can preview from in to out (video or audio or both)
(gsGetSetValue::gsChanCapabilities)

Definition at line 6431 of file mediacmd.h.

#define GS_CHANCAP_RECORD 0x00004000

Channel can record (video or audio or both) (gsGetSetValue::gsChanCapabilities)

Definition at line 6437 of file mediacmd.h.

#define GS_CHANCAP_RECSTOP 0x00010000

Channel can set clip name and prep record (video or audio or both)
(gsGetSetValue::gsChanCapabilities)

Definition at line 6441 of file mediacmd.h.

#define GS_CHANCAP_REVPLAY 0x00000002

Channel can reverse play (video or audio or both) (gsGetSetValue::gsChanCapabilities)

Definition at line 6421 of file mediacmd.h.

#define GS_CHANCAP_SEEK 0x00000020

Channel can seek to any point (video or audio or both) (gsGetSetValue::gsChanCapabilities)

Definition at line 6429 of file mediacmd.h.

#define GS_CHANCAP_SELECTPRESET 0x00020000

Channel can select recording channels (video or audio or both)
(gsGetSetValue::gsChanCapabilities)

Definition at line 6443 of file mediacmd.h.

#define GS_CHANCAP_SHUTTLE 0x00000010

Channel can shuttle above play speed (video or audio or both)
(gsGetSetValue::gsChanCapabilities)

Definition at line 6427 of file mediacmd.h.

#define GS_CHANCAP_STOP 0x00001000

Channel has a stop mode (video or audio or both) (gsGetSetValue::gsChanCapabilities)

Definition at line 6433 of file mediacmd.h.

#define GS_CHANCAP_TCSPACE 0x2000000

Channel can act like a VTR time code space (video or audio or both)
(gsGetSetValue::gsChanCapabilities)

Definition at line 6459 of file mediacmd.h.

#define GS_CHANCAP_VGAPREVIEW 0x0020000

Channel can display a VGA preview (video or audio or both)
(gsGetSetValue::gsChanCapabilities)

Definition at line 6449 of file mediacmd.h.

#define GS_CHROMA_FORMAT_411 0x8

Chroma format 4:1:1.

Definition at line 5474 of file mediacmd.h.

#define GS_CHROMA_FORMAT_420 0x1

Chroma format 4:2:0.

Definition at line 5468 of file mediacmd.h.

#define GS_CHROMA_FORMAT_422 0x2

Chroma format 4:2:2.

Definition at line 5470 of file mediacmd.h.

#define GS_CHROMA_FORMAT_444 0x4

Chroma format 4:4:4.

Definition at line 5472 of file mediacmd.h.

#define GS_CLEANRECORDWIPE_END 0xA5A5A5A5

Definition at line 6692 of file mediacmd.h.

#define GS_CLEANRECORDWIPE_ROOTDIR 0x0

Definition at line 6689 of file mediacmd.h.

#define GS_CLEANRECORDWIPE_START 0x5F5F5F5F

Definition at line 6691 of file mediacmd.h.

#define GS_CLEANRECORDWIPE_WHOLEDRIVE 0x1

Definition at line 6690 of file mediacmd.h.

#define GS_CLIPMODE_CLIPSPACE 0

Definition at line 5158 of file mediacmd.h.

#define GS_CLIPMODE_FILM 3

Definition at line 5161 of file mediacmd.h.

#define GS_CLIPMODE_ILLEGAL 0xFFFFFFFF

Known file types.

Definition at line 5157 of file mediacmd.h.

#define GS_CLIPMODE_SINGLE 2

Definition at line 5160 of file mediacmd.h.

#define GS_CLIPMODE_TCSPACE 1

Definition at line 5159 of file mediacmd.h.

#define GS_CLRFRM_2FLD 0x00000000

Edit color frame 2 field (gsGetSetValue::gsColorFrame)

Definition at line 6405 of file mediacmd.h.

#define GS_CLRFRM_4FLD 0x00000001

Edit color frame 4 field (gsGetSetValue::gsColorFrame)

Definition at line 6407 of file mediacmd.h.

#define GS_CLRFRM_8FLD 0x00000002

Edit color frame 8 field (gsGetSetValue::gsColorFrame)

Definition at line 6409 of file mediacmd.h.

#define GS_CLRFRM_DEFAULT 0x000000FF

Edit color frame device default (gsGetSetValue::gsColorFrame)

Definition at line 6411 of file mediacmd.h.

#define GS_COMPTYPE_ABGR 0x00400000

Uncompressed A BGR - TIFF.

Definition at line 6310 of file mediacmd.h.

#define GS_COMPTYPE_ALT 0x80000000

Use as generic alternative for use through AVCodec.

Definition at line 6328 of file mediacmd.h.

#define GS_COMPTYPE_ARGB 0x00100000

Uncompressed RGB (DVS)

Definition at line 6306 of file mediacmd.h.

#define GS_COMPTYPE_AVCi 0x00010000

Panasonic AVCi (gsGetSetValue::gsCompType)

Definition at line 6297 of file mediacmd.h.

#define GS_COMPTYPE_BAYER 0x00000002

Motion JPEG hardware codec (LSI, Zoran, C-Cube, etc) (gsGetSetValue::gsCompType)

Bayer

Definition at line 6257 of file mediacmd.h.

#define GS_COMPTYPE_BGR 0x00000020

define GS_COMPTYPE_MPEG2 0x00000020 Uncompressed BGR 24 Bit

Definition at line 6270 of file mediacmd.h.

#define GS_COMPTYPE_BGRA 0x00800000

Uncompressed BGR A - BMP/TGA.

Definition at line 6312 of file mediacmd.h.

#define GS_COMPTYPE_BGRA_INVERT 0x00040000

Inverted 32 bit TGA.

Definition at line 6302 of file mediacmd.h.

#define GS_COMPTYPE_CINEFORM 0x00001000

8Bit Y'CrCb uncompressed video (gsGetSetValue::gsCompType)
Definition at line 6288 of file mediacmd.h.

#define GS_COMPTYPE_CINEFORM_3D 0x00000010

MPEG 1 hardware compatible codec (gsGetSetValue::gsCompType)
Definition at line 6266 of file mediacmd.h.

#define GS_COMPTYPE_DNxHD 0x00008000

Avid DNxHD.
Definition at line 6294 of file mediacmd.h.

#define GS_COMPTYPE_DPX_RGB10 0x40000000

DPX 10 bit rgb.
Definition at line 6326 of file mediacmd.h.

#define GS_COMPTYPE_DPX_YCBCR10 0x00080000

DPX 10 bit YCbCr.
Definition at line 6304 of file mediacmd.h.

#define GS_COMPTYPE_DV100 0x00000800

High Def DV codec (gsGetSetValue::gsCompType)
Definition at line 6284 of file mediacmd.h.

#define GS_COMPTYPE_DV25 0x00000100

Hardware DV25, DVCPRO. DVCPRO25 (gsGetSetValue::gsCompType)
Definition at line 6278 of file mediacmd.h.

#define GS_COMPTYPE_DV50 0x00000200

Hardware DV50, DVCPRO50 (gsGetSetValue::gsCompType)
Definition at line 6280 of file mediacmd.h.

#define GS_COMPTYPE_DVSD 0x00000400

Hardware Standard DV Bluebook, DVPRO, DVSD (gsGetSetValue::gsCompType)
Definition at line 6282 of file mediacmd.h.

#define GS_COMPTYPE_H264 0x00000004

MPEG-4 h.264.
Definition at line 6259 of file mediacmd.h.

#define GS_COMPTYPE_HDCAM 0x00000040

Editable MPEG 2 I Frame Only compatible codec (gsGetSetValue::gsCompType)
Definition at line 6273 of file mediacmd.h.

#define GS_COMPTYPE_JPEG2000 0x00000008

Definition at line 6262 of file mediacmd.h.

#define GS_COMPTYPE_MPEG 0x00000080

MPEG 2 long GOP or IFrame hardware compatible codec (gsGetSetValue::gsCompType)
Definition at line 6276 of file mediacmd.h.

#define GS_COMPTYPE_PRORES 0x00020000

Apple ProRes (gsGetSetValue::gsCompType)
Definition at line 6300 of file mediacmd.h.

#define GS_COMPTYPE_RGB 0x00004000

Uncompressed RGB 24 Bit.
Definition at line 6292 of file mediacmd.h.

#define GS_COMPTYPE_RGBA 0x00200000

Uncompressed RGBA (DVS)
Definition at line 6308 of file mediacmd.h.

#define GS_COMPTYPE_SOFTWARE 0x00000001

Software passed codec on main processor (gsGetSetValue::gsCompType)
Definition at line 6253 of file mediacmd.h.

#define GS_COMPTYPE_STEREO 0x10000000

Uncompressed Y'CrCb 4:4:4A (DVS, Dual VG) or 3D 8, 10, 30 or 32 bit.
Definition at line 6322 of file mediacmd.h.

#define GS_COMPTYPE_YCRCB_420 0x20000000

Uncompressed Y'CrCb 4:2:0.
Definition at line 6324 of file mediacmd.h.

#define GS_COMPTYPE_YCRCB_422 0x01000000

Uncompressed Y'CrCb 4:2:2 (DVS, VG)
Definition at line 6314 of file mediacmd.h.

#define GS_COMPTYPE_YCRCB_422A 0x02000000

Uncompressed Y'CrCb 4:2:2A (DVS, Dual VG)
Definition at line 6316 of file mediacmd.h.

#define GS_COMPTYPE_YCRCB_444 0x04000000

Uncompressed Y'CrCb 4:4:4 (DVS, Dual VG)
Definition at line 6318 of file mediacmd.h.

#define GS_COMPTYPE_YCRCB_444A 0x08000000

Uncompressed Y'CrCb 4:4:4A (DVS, Dual VG)
Definition at line 6320 of file mediacmd.h.

#define GS_COMPTYPE_YCRCB_V210 0x00002000

10Bit Y'CrCb uncompressed video (gsGetSetValue::gsCompType)
Definition at line 6290 of file mediacmd.h.

#define GS_DEFAULT 0xFF

Default for cmdType::ctGetValue, cmdType::ctSetValue (usually in relation to VTR setup)
Definition at line 6724 of file mediacmd.h.

#define GS_DISABLE 0x00

Disable a feature or command
Definition at line 6716 of file mediacmd.h.

#define GS_DOWNCONVERT_14x9 0x0008

Down convert to 14x9.

Definition at line 5442 of file mediacmd.h.

#define GS_DOWNCONVERT_ANAMORPHIC 0x0004

Down convert to whole screen.

Definition at line 5440 of file mediacmd.h.

#define GS_DOWNCONVERT_CROP 0x0002

Down convert and crop image.

Definition at line 5438 of file mediacmd.h.

#define GS_DOWNCONVERT_LETTERBOX 0x0001

Down convert with top/bottom black bars.

Definition at line 5436 of file mediacmd.h.

#define GS_DXRGB_DIRECT 0x0001

DirectX allow direct RGB plane (for cmdGetSetValue::gsVgaDirectXConfig)

Definition at line 6573 of file mediacmd.h.

#define GS_DXRGB_OVERLAY 0x0002

DirectX allow overlay RGB plane (for cmdGetSetValue::gsVgaDirectXConfig)

Definition at line 6575 of file mediacmd.h.

#define GS_DXYUV_DIRECT 0x0008

DirectX allow direct YUV plane (for cmdGetSetValue::gsVgaDirectXConfig)

Definition at line 6579 of file mediacmd.h.

#define GS_DXYUV_OVERLAY 0x0004

DirectX allow overlay YUV plane (for cmdGetSetValue::gsVgaDirectXConfig)

Definition at line 6577 of file mediacmd.h.

#define GS_EDL_COMMENT 0x00000004

Returns from cmdGetSetValue::gsTcClipInfo comment.
Definition at line 5235 of file mediacmd.h.

#define GS_EDL_EDITNO 0x00000008

Returns from cmdGetSetValue::gsTcClipInfo edit number.
Definition at line 5237 of file mediacmd.h.

#define GS_EDL_EFFECT 0x00000001

Returns from cmdGetSetValue::gsTcClipInfo effect.
Definition at line 5231 of file mediacmd.h.

#define GS_EDL_EFFECT_DUR 0x00000002

Returns from cmdGetSetValue::gsTcClipInfo effect duration.
Definition at line 5233 of file mediacmd.h.

#define GS_ENABLE 0x01

Enable a feature or command
Definition at line 6720 of file mediacmd.h.

#define GS_ERROR_FILE_EXISTS -2

cmdGetSetValue::gsDTProjectToXml
Definition at line 6661 of file mediacmd.h.

#define GS_FALSE 0x00

False for boolean cmdType::ctGetValue, cmdType::ctSetValue
Definition at line 6708 of file mediacmd.h.

#define GS_FIELD 0x00

Use field cmdType::ctGetValue, cmdType::ctSetValue (for pause/freeze as opposed to frame)
Definition at line 6726 of file mediacmd.h.

#define GS_FIELD1 0x01

Use field 1 cmdType::ctGetValue, cmdType::ctSetValue (for record/playback starts and edits)
Definition at line 6728 of file mediacmd.h.

#define GS_FIELD2 0x02

Use field 2 cmdType::ctGetValue, cmdType::ctSetValue (for record/playback starts and edits)

Definition at line 6730 of file mediacmd.h.

#define GS_FILE_HAS_CHANGED_REMOTELY 0xFFFFFFFF

Definition at line 6778 of file mediacmd.h.

#define GS_FRAME 0x03

Use frame cmdType::ctGetValue, cmdType::ctSetValue (for pause/freeze as opposed to field)

Definition at line 6732 of file mediacmd.h.

#define GS_FRAMEDATA_ASCII 0x00001

ASCII data (all printable) for cmdGetSetValue::gsFrameData, cmdType::ctSetValue/cmdType[ctGetValue](#).

Definition at line 5180 of file mediacmd.h.

#define GS_FRAMEDATA_CC_TTEXT 0x10002

Close caption/teletext for cmdGetSetValue::gsFrameData, cmdType::ctSetValue/cmdType[ctGetValue](#).

Definition at line 5186 of file mediacmd.h.

#define GS_FRAMEDATA_HEX 0x00002

Binary (hex) data for cmdGetSetValue::gsFrameData, cmdType::ctSetValue/cmdType[ctGetValue](#).

Definition at line 5182 of file mediacmd.h.

#define GS_FRAMEDATA_NAVY 0x10003

Navy telemetry data for cmdGetSetValue::gsFrameData, cmdType::ctSetValue/cmdType[ctGetValue](#).

Definition at line 5188 of file mediacmd.h.

#define GS_FRAMEDATA_TELECINE 0x10001

Telecine RP-215 / DPX Data for cmdGetSetValue::gsFrameData, cmdType::ctSetValue/cmdType[ctGetValue](#).

Definition at line 5184 of file mediacmd.h.

#define GS_FRAMEDATA_UNKNOWN 0x00000

No data, unknown data for cmdGetSetValue::gsFrameData,
cmdType::ctSetValue/cmdTypectGetValue.

Definition at line 5178 of file mediacmd.h.

#define GS_FRAMEDROPMODE_HALF 0x010000

Definition at line 6773 of file mediacmd.h.

#define GS_FRAMEDROPMODE_NONE 0x000000

Definition at line 6760 of file mediacmd.h.

#define GS_FRAMEDROPMODE_VARICAM_2398 0x000023

Definition at line 6762 of file mediacmd.h.

#define GS_FRAMEDROPMODE_VARICAM_24 0x000024

Definition at line 6763 of file mediacmd.h.

#define GS_FRAMEDROPMODE_VARICAM_25 0x000025

Definition at line 6764 of file mediacmd.h.

#define GS_FRAMEDROPMODE_VARICAM_2997 0x000029

Definition at line 6765 of file mediacmd.h.

#define GS_FRAMEDROPMODE_VARICAM_30 0x000030

Definition at line 6766 of file mediacmd.h.

#define GS_FRAMEDROPMODE_VARICAM_50 0x000050

Definition at line 6767 of file mediacmd.h.

#define GS_FRAMEDROPMODE_VARICAM_5994 0x000059

Definition at line 6768 of file mediacmd.h.

#define GS_FRAMEDROPMODE_VARICAM_60 0x000060

Definition at line 6769 of file mediacmd.h.

#define GS_FRAMEDROPMODE_VARICAM_ILLEGAL 0x0000FF

Definition at line 6771 of file mediacmd.h.

#define GS_FRAMEDROPMODE_VARICAM_MASK_FPS 0x0000FF

Definition at line 6761 of file mediacmd.h.

#define GS_FRAMEDROPMODE_VARICAM_UB_INVERT 0x000100

Definition at line 6772 of file mediacmd.h.

#define GS_FRAMEDROPMODE_VARICAM_VARI 0x000001

Definition at line 6770 of file mediacmd.h.

#define GS_HSDIBAYER_ARRI_ALEXA 0x00000002

Raw bayer HD-SDI: Arri Alexa.

Definition at line 6337 of file mediacmd.h.

#define GS_HSDIBAYER_ARRI_D21 0x00000001

Raw bayer HD-SDI: Arri D21 T-Link.

Definition at line 6335 of file mediacmd.h.

#define GS_HSDIBAYER_DUALBIT 0x10000000

Flag bit for dual rate capture.

Definition at line 6331 of file mediacmd.h.

#define GS_HSDIBAYER_DUALLINKBIT 0x20000000

Flag bit for dual pipe capture.

Definition at line 6333 of file mediacmd.h.

#define GS_HSDIBAYER_WIESS_2K1536 0x00000400

Raw bayer HD-SDI: Weisscam Film2K at 25p.

Definition at line 6341 of file mediacmd.h.

#define GS_HSDIBAYER_WIESS_ONEFRAME 0x00000100

Raw bayer HD-SDI: Weisscam 1:1 - up to 30 in 30.

Definition at line 6339 of file mediacmd.h.

#define GS_HSDIBAYER_WIESS_QUADFRAME (GS_HSDIBAYER_WIESS_ONEFRAME | GS_HSDIBAYER_DUALBIT | GS_HSDIBAYER_DUALLINKBIT)

Raw bayer HD-SDI: Weisscam 720p at 200p, 1080p up to 120.

Definition at line 6345 of file mediacmd.h.

#define GS_HSDIBAYER_WIESS_TWO2K1536 (GS_HSDIBAYER_WIESS_2K1536 | GS_HSDIBAYER_DUALBIT)

Raw dual 2K film.

Definition at line 6347 of file mediacmd.h.

#define GS_HSDIBAYER_WIESS_TWOFRAME (GS_HSDIBAYER_WIESS_ONEFRAME | GS_HSDIBAYER_DUALBIT)

Raw bayer HD-SDI: Weisscam 720p at 100p, 1080p up to 60.

Definition at line 6343 of file mediacmd.h.

#define GS_HSDTI_HDCAM_SR (GS_HSDIBAYER_DUALLINKBIT | 0x00100000)

HDCamSR SDTI.

Definition at line 6349 of file mediacmd.h.

#define GS_HEADSEL_DEFAULT 0x000000FF

(gsGetSetValue::gsHeadSelect)

Definition at line 6402 of file mediacmd.h.

#define GS_HEADSEL_PLAY 0x00000001

Use play head (gsGetSetValue::gsHeadSelect)

Definition at line 6400 of file mediacmd.h.

#define GS_HEADSEL_RECPLAY 0x00000000

Use record/play head (gsGetSetValue::gsHeadSelect)

Definition at line 6398 of file mediacmd.h.

#define GS_INSERT_EDIT 0x01

Set cmdGetSetValue::gsEditMode for an insert edit.

Definition at line 5247 of file mediacmd.h.

#define GS_INSTALLSYSTEM_END 0x4E4E4E4E

Definition at line 6696 of file mediacmd.h.

#define GS_INSTALLSYSTEM_POSITION 0x2B2B2B2B

Definition at line 6694 of file mediacmd.h.

#define GS_INSTALLSYSTEM_START 0x11111111

Definition at line 6695 of file mediacmd.h.

#define GS_LOCKSRC_CVBS 0x0008

Composite (CVBS) input is genlock source (gsGetSetValue::gsVidOutGenlockSource)

Definition at line 5399 of file mediacmd.h.

#define GS_LOCKSRC_EXTIN 0x0002

External ref in is genlock source (gsGetSetValue::gsVidOutGenlockSource)

Definition at line 5395 of file mediacmd.h.

#define GS_LOCKSRC_HDMI 0x0080

HDMI genlock.

Definition at line 5407 of file mediacmd.h.

#define GS_LOCKSRC_IN_Y 0x0020

Component Y input is genlock source (gsGetSetValue::gsVidOutGenlockSource)

Definition at line 5403 of file mediacmd.h.

#define GS_LOCKSRC_INPUT 0x0004

Current input is genlock source (gsGetSetValue::gsVidOutGenlockSource)

Definition at line 5397 of file mediacmd.h.

#define GS_LOCKSRC_NONE 0x0001

No external genlock source (free running on internal clock)
(gsGetSetValue::gsVidOutGenlockSource)

Definition at line 5393 of file mediacmd.h.

#define GS_LOCKSRC_SDI 0x0040

SDI serial digital input is genlock source (gsGetSetValue::gsVidOutGenlockSource)

Definition at line 5405 of file mediacmd.h.

#define GS_LOCKSRC_SVIDEO 0x0010

S-Video (SVHS) input is genlock source (gsGetSetValue::gsVidOutGenlockSource)

Definition at line 5401 of file mediacmd.h.

#define GS_MONITORGRAB_NONE 0x0000

Turn off monitor.

Definition at line 5637 of file mediacmd.h.

#define GS_MONITORGRAB_SIZE_FULL 0x0001

Full size image captured.

Definition at line 5647 of file mediacmd.h.

#define GS_MONITORGRAB_SIZE_HALF 0x0002

Half size image captured.

Definition at line 5649 of file mediacmd.h.

#define GS_MONITORGRAB_SIZE_MASK 0x000F

Size mask (full, half, quarter)

Definition at line 5645 of file mediacmd.h.

#define GS_MONITORGRAB_SIZE_QUARTER 0x0004

Quarter size image captured.

Definition at line 5651 of file mediacmd.h.

#define GS_MONITORGRAB_TARGET_MASK 0x0F00

Target/To mask.

Definition at line 5653 of file mediacmd.h.

#define GS_MONITORGRAB_TO_HTTP 0x0400

Save image to web server (use name sent in arbID)

Definition at line 5659 of file mediacmd.h.

#define GS_MONITORGRAB_TO_MEMORY 0x0100

Use the arbID area.

Definition at line 5655 of file mediacmd.h.

#define GS_MONITORGRAB_TO_NETWORK 0x0800

Save image through 'to be announced' network transport.

Definition at line 5661 of file mediacmd.h.

#define GS_MONITORGRAB_TO_UNC_PATH 0x0200

Save image to a UNC path.

Definition at line 5657 of file mediacmd.h.

#define GS_MONITORGRAB_TYPE_BMP 0x0000

Use BMP format for image.

Definition at line 5641 of file mediacmd.h.

#define GS_MONITORGRAB_TYPE_JPG 0x0010

Use JPEG format for image.

Definition at line 5643 of file mediacmd.h.

#define GS_MONITORGRAB_TYPE_MASK 0x00F0

Type mask (jpg, bmp)

Definition at line 5639 of file mediacmd.h.

#define GS_MPEG_ASPECT_RATIO_16x9 GS_ASPECT_RATIO_16x9

MPEG aspect ratio square see [GS_ASPECT_RATIO_16x9](#).

Definition at line 5505 of file mediacmd.h.

#define GS_MPEG_ASPECT_RATIO_2_21x1 GS_ASPECT_RATIO_2_21x1

MPEG aspect ratio square see [GS_ASPECT_RATIO_2_21x1](#).

Definition at line 5507 of file mediacmd.h.

#define GS_MPEG_ASPECT_RATIO_4x3 GS_ASPECT_RATIO_4x3

MPEG aspect ratio square see [GS_ASPECT_RATIO_4x3](#).

Definition at line 5503 of file mediacmd.h.

#define GS_MPEG_ASPECT_RATIO_SQUARE GS_ASPECT_RATIO_SQUARE

MPEG aspect ratio square see [GS_ASPECT_RATIO_SQUARE](#).

Definition at line 5501 of file mediacmd.h.

#define GS_MPEG_CC_FORMAT_ATSC 0x2

Definition at line 5531 of file mediacmd.h.

#define GS_MPEG_CC_FORMAT_ATSC_REORDER 0x8

Definition at line 5533 of file mediacmd.h.

#define GS_MPEG_CC_FORMAT_CCUBE 0x1

Definition at line 5530 of file mediacmd.h.

#define GS_MPEG_CC_FORMAT_CCUBE_REORDER 0x4

Definition at line 5532 of file mediacmd.h.

#define GS_MPEG_CHROMA_FORMAT_420 GS_CHROMA_FORMAT_420

MPEG chroma format 4:2:0 see [GS_CHROMA_FORMAT_420](#).

Definition at line 5477 of file mediacmd.h.

#define GS_MPEG_CHROMA_FORMAT_422 GS_CHROMA_FORMAT_422

MPEG chroma format 4:2:2 see [GS_CHROMA_FORMAT_422](#).

Definition at line 5479 of file mediacmd.h.

#define GS_MPEG_CHROMA_FORMAT_444 GS_CHROMA_FORMAT_444

MPEG chroma format 4:4:4 see [GS_CHROMA_FORMAT_444](#).

Definition at line 5481 of file mediacmd.h.

#define GS_MPEG_DC_PRECISION_10 0x4

MPEG DCT Precision 10 bits.

Definition at line 5488 of file mediacmd.h.

#define GS_MPEG_DC_PRECISION_11 0x8

MPEG DCT Precision 11 bits.

Definition at line 5490 of file mediacmd.h.

#define GS_MPEG_DC_PRECISION_8 0x1

MPEG DCT Precision 8 bits.

Definition at line 5484 of file mediacmd.h.

#define GS_MPEG_DC_PRECISION_9 0x2

MPEG DCT Precision 9 bits.

Definition at line 5486 of file mediacmd.h.

#define GS_MPEG_EIGHT_FRAMES 0x0080

Definition at line 5542 of file mediacmd.h.

#define GS_MPEG_ELEVEN_FRAMES 0x0400

Definition at line 5545 of file mediacmd.h.

#define GS_MPEG_FIFTEEN_FRAMES 0x4000

Definition at line 5549 of file mediacmd.h.

#define GS_MPEG_FIVE_FRAMES 0x0010

Definition at line 5539 of file mediacmd.h.

#define GS_MPEG_FOUR_FRAMES 0x0008

Definition at line 5538 of file mediacmd.h.

#define GS_MPEG_FOURTEEN_FRAMES 0x2000

Definition at line 5548 of file mediacmd.h.

#define GS_MPEG_LANGUAGE_CHINESE 0x2000

Definition at line 5528 of file mediacmd.h.

#define GS_MPEG_LANGUAGE_DANISH 0x0040

Definition at line 5521 of file mediacmd.h.

#define GS_MPEG_LANGUAGE_DUTCH 0x0020

Definition at line 5520 of file mediacmd.h.

#define GS_MPEG_LANGUAGE_ENGLISH 0x0001

Definition at line 5515 of file mediacmd.h.

#define GS_MPEG_LANGUAGE_FINNISH 0x0080

Definition at line 5522 of file mediacmd.h.

#define GS_MPEG_LANGUAGE_FRENCH 0x0004

Definition at line 5517 of file mediacmd.h.

#define GS_MPEG_LANGUAGE_GERMAN 0x0008

Definition at line 5518 of file mediacmd.h.

#define GS_MPEG_LANGUAGE_GREEK 0x0200

Definition at line 5524 of file mediacmd.h.

#define GS_MPEG_LANGUAGE_ITALIAN 0x0100

Definition at line 5523 of file mediacmd.h.

#define GS_MPEG_LANGUAGE_JAPANESE 0x0010

Definition at line 5519 of file mediacmd.h.

#define GS_MPEG_LANGUAGE_PORTUGUESE 0x0400

Definition at line 5525 of file mediacmd.h.

#define GS_MPEG_LANGUAGE_RUSSIAN 0x1000

Definition at line 5527 of file mediacmd.h.

#define GS_MPEG_LANGUAGE_SPANISH 0x0002

Definition at line 5516 of file mediacmd.h.

#define GS_MPEG_LANGUAGE_SWEDISH 0x0800

Definition at line 5526 of file mediacmd.h.

#define GS_MPEG_NINE_FRAMES 0x0100

Definition at line 5543 of file mediacmd.h.

#define GS_MPEG_ONE_FRAMES 0x0001

Definition at line 5535 of file mediacmd.h.

#define GS_MPEG_RESOLUTION_120 0x001

Standard MPEG resolution 120.

Definition at line 5445 of file mediacmd.h.

#define GS_MPEG_RESOLUTION_240 0x002

Standard MPEG resolution 240.

Definition at line 5447 of file mediacmd.h.

#define GS_MPEG_RESOLUTION_288 0x004

Standard MPEG resolution 288.

Definition at line 5449 of file mediacmd.h.

#define GS_MPEG_RESOLUTION_352 0x008

Standard MPEG resolution 352.

Definition at line 5451 of file mediacmd.h.

#define GS_MPEG_RESOLUTION_480 0x010

Standard MPEG resolution 480.

Definition at line 5453 of file mediacmd.h.

#define GS_MPEG_RESOLUTION_512 0x020

Standard MPEG resolution 512.

Definition at line 5455 of file mediacmd.h.

#define GS_MPEG_RESOLUTION_544 0x040

Standard MPEG resolution 522.

Definition at line 5457 of file mediacmd.h.

#define GS_MPEG_RESOLUTION_576 0x080

Standard MPEG resolution 576.

Definition at line 5459 of file mediacmd.h.

#define GS_MPEG_RESOLUTION_608 0x100

Standard MPEG resolution 608.

Definition at line 5461 of file mediacmd.h.

#define GS_MPEG_RESOLUTION_704 0x200

Standard MPEG resolution 704.

Definition at line 5463 of file mediacmd.h.

#define GS_MPEG_RESOLUTION_720 0x400

Standard MPEG resolution 720.

Definition at line 5465 of file mediacmd.h.

#define GS_MPEG_SEVEN_FRAMES 0x0040

Definition at line 5541 of file mediacmd.h.

#define GS_MPEG_SIX_FRAMES 0x0020

Definition at line 5540 of file mediacmd.h.

#define GS_MPEG_SIXTEEN_FRAMES 0x8000

Definition at line 5550 of file mediacmd.h.

#define GS_MPEG_STANDARD_ELEMENTARY 0x8

Definition at line 5512 of file mediacmd.h.

#define GS_MPEG_STANDARD_ELEMENTRY GS_MPEG_STANDARD_ELEMENTARY

Definition at line 5513 of file mediacmd.h.

#define GS_MPEG_STANDARD_PROGRAM 0x2

Definition at line 5510 of file mediacmd.h.

#define GS_MPEG_STANDARD_SYSTEM 0x1

Definition at line 5509 of file mediacmd.h.

#define GS_MPEG_STANDARD_TRANSPORT 0x4

Definition at line 5511 of file mediacmd.h.

#define GS_MPEG_TEN_FRAMES 0x0200

Definition at line 5544 of file mediacmd.h.

#define GS_MPEG_THIRTEEN_FRAMES 0x1000

Definition at line 5547 of file mediacmd.h.

#define GS_MPEG_THREE_FRAMES 0x0004

Definition at line 5537 of file mediacmd.h.

#define GS_MPEG_TWELVE_FRAMES 0x0800

Definition at line 5546 of file mediacmd.h.

#define GS_MPEG_TWO_FRAMES 0x0002

Definition at line 5536 of file mediacmd.h.

#define GS_NOT_SUPPORTED 0xFFFFFFFF

Command is not supported see cmdType::ctGetValue, cmdType::ctSetValue,
cmdType::ctValueSupported.

Definition at line 6699 of file mediacmd.h.

#define GS_ONTRAK_ILLEGAL 0xFFFFFFFF

Definition at line 6785 of file mediacmd.h.

#define GS_ONTRAK_K0 0x00000001

Definition at line 6781 of file mediacmd.h.

#define GS_ONTRAK_K1 0x00000002

Definition at line 6782 of file mediacmd.h.

#define GS_ONTRAK_K2 0x00000004

Definition at line 6783 of file mediacmd.h.

#define GS_ONTRAK_K3 0x00000008

Definition at line 6784 of file mediacmd.h.

#define GS_ONTRAK_NONE 0x00000000

Definition at line 6780 of file mediacmd.h.

#define GS_ONTRAK_PA0 0x00000001

Definition at line 6786 of file mediacmd.h.

#define GS_ONTRAK_PA1 0x00000002

Definition at line 6787 of file mediacmd.h.

#define GS_ONTRAK_PA2 0x00000004

Definition at line 6788 of file mediacmd.h.

#define GS_ONTRAK_PA3 0x00000008

Definition at line 6789 of file mediacmd.h.

#define GS_PBEE_AUTO 0x00000000

Allow playback or edit to edit output as necessary (gsGetSetValue::gsPBEE)

Definition at line 6382 of file mediacmd.h.

#define GS_PBEE_DEFAULT 0x000000FF

Device dependent output (gsGetSetValue::gsPBEE)

Definition at line 6388 of file mediacmd.h.

#define GS_PBEE_EE 0x00000002

Allow passthrough only output - no playback (gsGetSetValue::gsPBEE)

Definition at line 6386 of file mediacmd.h.

#define GS_PBEE_PB 0x00000001

Allow playback only output - no passthrough (gsGetSetValue::gsPBEE)

Definition at line 6384 of file mediacmd.h.

#define GS_PRODUCTION_MODE_PLAY 0x01

Stop if frames dropped in playback.

Definition at line 6469 of file mediacmd.h.

#define GS_PRODUCTION_MODE_RECORD 0x02

Stop if frames dropped in record.

Definition at line 6471 of file mediacmd.h.

#define GS_PROXYMODE_ABORTALL 0x0ffff0

cmdGetSetValue::gsProxyMode Abort all active proxies

Definition at line 6496 of file mediacmd.h.

#define GS_PROXYMODE_AFTERRECORD 0x000010

cmdGetSetValue::gsProxyMode Proxy files once they have finished recording

Definition at line 6494 of file mediacmd.h.

#define GS_PROXYMODE_EVERYTHING 0x000002

cmdGetSetValue::gsProxyMode Proxy any file that is opened (for read/write/check)
Definition at line 6490 of file mediacmd.h.

#define GS_PROXYMODE_NOTHING 0x000000

cmdGetSetValue::gsProxyMode Do not automatically proxy anything
Definition at line 6488 of file mediacmd.h.

#define GS_PROXYMODE_RECORD 0x000001

cmdGetSetValue::gsProxyMode Proxy files while they are recording (with supported source types)
Definition at line 6492 of file mediacmd.h.

#define GS_SERIALEDITMODE_FAKE 2

Pause at each speed change, call time play when real play comes (CTV mode)
cmdGetSetValue::gsSerialEditMode.
Definition at line 6478 of file mediacmd.h.

#define GS_SERIALEDITMODE_IGNORE 1

Ignore all off speed play commands (CBS TimeLogic Mode)
cmdGetSetValue::gsSerialEditMode.
Definition at line 6476 of file mediacmd.h.

#define GS_SERIALEDITMODE_NONE 0

Normal editing mode, no special speed compensation cmdGetSetValue::gsSerialEditMode.
Definition at line 6474 of file mediacmd.h.

#define GS_SERIALPROTOCOLS_ODETICS 2

Enable Odetics extensions (gsSerialProtocols)
Definition at line 6483 of file mediacmd.h.

#define GS_SERIALPROTOCOLS_SONY422 1

Enable Sony VTR 422 (gsSerialProtocols)
Definition at line 6481 of file mediacmd.h.

#define GS_SERIALPROTOCOLS_VDCP 4

Enable VDCP Louth (gsSerialProtocols)

Definition at line 6485 of file mediacmd.h.

#define GS_SERVOREF_AUTO 0x00000000

Video servo reference auto (gsGetSetValue::gsServoRefSelect)

Definition at line 6391 of file mediacmd.h.

#define GS_SERVOREF_DEFAULT 0x000000FF

Video servo reference device default (gsGetSetValue::gsServoRefSelect)

Definition at line 6395 of file mediacmd.h.

#define GS_SERVOREF_EXT 0x00000001

Video servo reference external only (gsGetSetValue::gsServoRefSelect)

Definition at line 6393 of file mediacmd.h.

#define GS_SHUTDOWNAPPLICATION_END 0x5F5F5F5F

Definition at line 6682 of file mediacmd.h.

#define GS_SHUTDOWNAPPLICATION_POSITION 0x01010101

Definition at line 6680 of file mediacmd.h.

#define GS_SHUTDOWNAPPLICATION_START 0xA5A5A5A5

Definition at line 6681 of file mediacmd.h.

#define GS_SHUTDOWNSYSTEM_END 0xA5A5A5A5

Definition at line 6687 of file mediacmd.h.

#define GS_SHUTDOWNSYSTEM_POSITION 0x11111111

Definition at line 6685 of file mediacmd.h.

#define GS_SHUTDOWNSYSTEM_RESTART 0xA5A5A5A4

Definition at line 6684 of file mediacmd.h.

#define GS_SHUTDOWNSYSTEM_START 0x5F5F5F5F

Definition at line 6686 of file mediacmd.h.

#define GS_SIGFORM_1035i_30_260M (GS_SIGFORMSIZE_1920x1035 | GS_SIGFORMTYPE_INTERLACED | GS_SIGFORMFRAMERATE_30)

2200x1125 raster, 1920x1035 production aperture (1888x1017 clean) @ 30 fps
gsGetSetValue::gsSignalFormat

Definition at line 5864 of file mediacmd.h.

#define GS_SIGFORM_1035i_30X_260M (GS_SIGFORMSIZE_1920x1035 | GS_SIGFORMTYPE_INTERLACED | GS_SIGFORMFRAMERATE_29_97)

2200x1125 raster, 1920x1035 production aperture (1888x1017 clean) @ 29.97 fp
gsGetSetValue::gsSignalFormats

Definition at line 5866 of file mediacmd.h.

#define GS_SIGFORM_1080_24 (GS_SIGFORMSIZE_1920x1080 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_24)

1920x1080P (274M-1997 Table1 System 4) @ 24 gsGetSetValue::gsSignalFormat

Definition at line 5889 of file mediacmd.h.

#define GS_SIGFORM_1080_24X (GS_SIGFORMSIZE_1920x1080 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_23_98)

1920x1080P (274M-1997 Table1 System 4) @ 23.98 gsGetSetValue::gsSignalFormat

Definition at line 5891 of file mediacmd.h.

#define GS_SIGFORM_1080_25 (GS_SIGFORMSIZE_1920x1080 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_25)

1920x1080P (274M-1997 Table1 System 4) @ 25 gsGetSetValue::gsSignalFormat

Definition at line 5887 of file mediacmd.h.

#define GS_SIGFORM_1080_30 (GS_SIGFORMSIZE_1920x1080 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_30)

1920x1080P (274M-1997 Table1 System 4) @ 30 gsGetSetValue::gsSignalFormat

Definition at line 5883 of file mediacmd.h.

#define GS_SIGFORM_1080_30X (GS_SIGFORMSIZE_1920x1080 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_29_97)

1920x1080P (274M-1997 Table1 System 4) @ 29.97 gsGetSetValue::gsSignalFormat
Definition at line 5885 of file mediacmd.h.

**#define GS_SIGFORM_1080_48 (GS_SIGFORMSIZE_1920x1080 |
GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_48)**

1920x1080P 48 (Dual 24)
Definition at line 5901 of file mediacmd.h.

**#define GS_SIGFORM_1080_48X (GS_SIGFORMSIZE_1920x1080 |
GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_47_95)**

1920x1080P 47.95 (Dual 23.98)
Definition at line 5903 of file mediacmd.h.

**#define GS_SIGFORM_1080_50 (GS_SIGFORMSIZE_1920x1080 |
GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_50)**

1920x1080P 50 (Dual 25)
Definition at line 5899 of file mediacmd.h.

**#define GS_SIGFORM_1080_60 (GS_SIGFORMSIZE_1920x1080 |
GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_60)**

1920x1080P 60 (Dual P30)
Definition at line 5895 of file mediacmd.h.

**#define GS_SIGFORM_1080_60X (GS_SIGFORMSIZE_1920x1080 |
GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_59_94)**

1920x1080P 59.94 (Dual P29.97)
Definition at line 5897 of file mediacmd.h.

**#define GS_SIGFORM_1080i_24 (GS_SIGFORMSIZE_1920x1080 |
GS_SIGFORMTYPE_INTERLACED | GS_SIGFORMFRAMERATE_24)**

1920x1080sf (274M-1997 Table1 System 4) @ 24 gsGetSetValue::gsSignalFormat
Definition at line 5877 of file mediacmd.h.

**#define GS_SIGFORM_1080i_24X (GS_SIGFORMSIZE_1920x1080 |
GS_SIGFORMTYPE_INTERLACED | GS_SIGFORMFRAMERATE_23_98)**

1920x1080sf (274M-1997 Table1 System 4) @ 23.98 gsGetSetValue::gsSignalFormat
Definition at line 5880 of file mediacmd.h.

**#define GS_SIGFORM_1080i_25 (GS_SIGFORMSIZE_1920x1080 |
GS_SIGFORMTYPE_INTERLACED | GS_SIGFORMFRAMERATE_25)**

1920x1080i (274M-1997 Table1 System 4) @ 25 gsGetSetValue::gsSignalFormat
Definition at line 5874 of file mediacmd.h.

**#define GS_SIGFORM_1080i_30 (GS_SIGFORMSIZE_1920x1080 |
GS_SIGFORMTYPE_INTERLACED | GS_SIGFORMFRAMERATE_30)**

1920x1080i (274M-1997 Table1 System 4) @ 29.97 gsGetSetValue::gsSignalFormat
Definition at line 5868 of file mediacmd.h.

**#define GS_SIGFORM_1080i_30X (GS_SIGFORMSIZE_1920x1080 |
GS_SIGFORMTYPE_INTERLACED | GS_SIGFORMFRAMERATE_29_97)**

1920x1080i (274M-1997 Table1 System 4) @ 30 gsGetSetValue::gsSignalFormat
Definition at line 5871 of file mediacmd.h.

**#define GS_SIGFORM_1080sf_24 (GS_SIGFORMSIZE_1920x1080 |
GS_SIGFORMTYPE_SEGMENTEDFRAME | GS_SIGFORMFRAMERATE_24)**

Definition at line 5878 of file mediacmd.h.

**#define GS_SIGFORM_1080sf_24X (GS_SIGFORMSIZE_1920x1080 |
GS_SIGFORMTYPE_SEGMENTEDFRAME | GS_SIGFORMFRAMERATE_23_98)**

Definition at line 5881 of file mediacmd.h.

**#define GS_SIGFORM_1080sf_25 (GS_SIGFORMSIZE_1920x1080 |
GS_SIGFORMTYPE_SEGMENTEDFRAME | GS_SIGFORMFRAMERATE_25)**

Definition at line 5875 of file mediacmd.h.

**#define GS_SIGFORM_1080sf_30 (GS_SIGFORMSIZE_1920x1080 |
GS_SIGFORMTYPE_SEGMENTEDFRAME | GS_SIGFORMFRAMERATE_30)**

Definition at line 5869 of file mediacmd.h.

**#define GS_SIGFORM_1080sf_30X (GS_SIGFORMSIZE_1920x1080 |
GS_SIGFORMTYPE_SEGMENTEDFRAME | GS_SIGFORMFRAMERATE_29_97)**

Definition at line 5872 of file mediacmd.h.

#define GS_SIGFORM_4K_QUAD_24 (GS_SIGFORMSIZE_4096x2880 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_24)

gsGetSetValue::gsSignalFormat Quad 2K 24 fps (4 x 2048x1080)

Definition at line 6069 of file mediacmd.h.

#define GS_SIGFORM_4K_QUAD_24X (GS_SIGFORMSIZE_4096x2880 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_23_98)

gsGetSetValue::gsSignalFormat Quad 2K 23.98 fps (4 x 2048x1080)

Definition at line 6065 of file mediacmd.h.

#define GS_SIGFORM_4K_QUAD_25 (GS_SIGFORMSIZE_4096x2880 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_25)

Definition at line 6072 of file mediacmd.h.

#define GS_SIGFORM_4K_QUADsf_24 (GS_SIGFORMSIZE_4096x2880 | GS_SIGFORMTYPE_SEGMENTEDFRAME | GS_SIGFORMFRAMERATE_24)

gsGetSetValue::gsSignalFormat Quad 2K sf 24 fps (4 x 2048x1080)

Definition at line 6071 of file mediacmd.h.

#define GS_SIGFORM_4K_QUADsf_24X (GS_SIGFORMSIZE_4096x2880 | GS_SIGFORMTYPE_SEGMENTEDFRAME | GS_SIGFORMFRAMERATE_23_98)

gsGetSetValue::gsSignalFormat Quad 2K sf 23.98 fps (4 x 2048x1080)

Definition at line 6067 of file mediacmd.h.

#define GS_SIGFORM_4K_QUADsf_25 (GS_SIGFORMSIZE_4096x2880 | GS_SIGFORMTYPE_SEGMENTEDFRAME | GS_SIGFORMFRAMERATE_25)

Definition at line 6073 of file mediacmd.h.

#define GS_SIGFORM_720_24 (GS_SIGFORMSIZE_1280x720 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_24)

720x1280 true 24 (Varicam)

Definition at line 5920 of file mediacmd.h.

#define GS_SIGFORM_720_25 (GS_SIGFORMSIZE_1280x720 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_25)

Half 50 Hz DVS, IRT.

Definition at line 5918 of file mediacmd.h.

```
#define GS_SIGFORM_720_30 (GS_SIGFORMSIZE_1280x720 |  
GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_30)
```

Half frame rate 720/60.

Definition at line 5914 of file mediacmd.h.

```
#define GS_SIGFORM_720_30X (GS_SIGFORMSIZE_1280x720 |  
GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_29_97)
```

Half frame rate 720/59.94.

Definition at line 5916 of file mediacmd.h.

```
#define GS_SIGFORM_720_50 (GS_SIGFORMSIZE_1280x720 |  
GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_50)
```

50 Hz DVS, IRT

Definition at line 5911 of file mediacmd.h.

```
#define GS_SIGFORM_720_60 (GS_SIGFORMSIZE_1280x720 |  
GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_60)
```

1650x750 raster, 1280x720 production aperture (1248x702 clean): @ 60
gsGetSetValue::gsSignalFormat

Definition at line 5907 of file mediacmd.h.

```
#define GS_SIGFORM_720_60X (GS_SIGFORMSIZE_1280x720 |  
GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_59_94)
```

1650x750 raster, 1280x720 production aperture (1248x702 clean): @ 59.97
gsGetSetValue::gsSignalFormat

Definition at line 5909 of file mediacmd.h.

```
#define GS_SIGFORM_ALT_NTSC (GS_SIGFORMSIZE_960x486 |  
GS_SIGFORMTYPE_INTERLACED | GS_SIGFORMFRAMERATE_29_97)
```

Signal format NTSC High Res (960x486)

Definition at line 5859 of file mediacmd.h.

```
#define GS_SIGFORM_ALT_PAL (GS_SIGFORMSIZE_960x576 |  
GS_SIGFORMTYPE_INTERLACED | GS_SIGFORMFRAMERATE_25)
```

Signal format PAL High Res (960x576)

Definition at line 5861 of file mediacmd.h.

```
#define GS_SIGFORM_ARRI_ALEXA (GS_SIGFORMSIZE_2880x1782 |  
GS_SIGFORMTYPE_PROGRESSIVE)
```

Definition at line 6088 of file mediacmd.h.

```
#define GS_SIGFORM_ARRI_D21 (GS_SIGFORMSIZE_2880x2160 |  
GS_SIGFORMTYPE_PROGRESSIVE)
```

Definition at line 6087 of file mediacmd.h.

```
#define GS_SIGFORM_CCIR_NTSC (GS_SIGFORMSIZE_720x486 |  
GS_SIGFORMTYPE_INTERLACED | GS_SIGFORMFRAMERATE_29_97)
```

Signal format NTSC square pixel (360/352x243/240 or 720/704x486/480) @ 29.97 or 30 fps
gsGetSetValue::gsSignalFormat.

Definition at line 5845 of file mediacmd.h.

```
#define GS_SIGFORM_CCIR_NTSC2398 (GS_SIGFORMSIZE_720x486 |  
GS_SIGFORMTYPE_INTERLACED | GS_SIGFORMFRAMERATE_23_98)
```

Signal format NTSC 23.98.

Definition at line 5855 of file mediacmd.h.

```
#define GS_SIGFORM_CCIR_NTSC_P483 (GS_SIGFORMSIZE_720x483 |  
GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_29_97)
```

Signal format NTSC square pixel (360/352x243/240 or 720/704x486/480) @ 29.97 or 30 fps
gsGetSetValue::gsSignalFormat.

Definition at line 5847 of file mediacmd.h.

```
#define GS_SIGFORM_CCIR_PAL (GS_SIGFORMSIZE_720x576 |  
GS_SIGFORMTYPE_INTERLACED | GS_SIGFORMFRAMERATE_25)
```

Signal format PAL square pixel (360/352x288 or 720/704x576) @ 25 fps
gsGetSetValue::gsSignalFormat.

Definition at line 5849 of file mediacmd.h.

```
#define GS_SIGFORM_CCIR_PNTSC_30 (GS_SIGFORMSIZE_720x486 |  
GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_30)
```

Signal format NTSC at 30 hz Progressive.

Definition at line 5851 of file mediacmd.h.

```
#define GS_SIGFORM_CCIR_PPAL_25 (GS_SIGFORMSIZE_720x576 |  
GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_25)
```

Signal format PAL at 25 hz Progressive.

Definition at line 5853 of file mediacmd.h.

```
#define GS_SIGFORM_CUSTOM 0xF0000000UL
```

All non video rate types (e.g.. 15fps, 10fps, 37fps) gsGetSetValue::gsSignalFormat.

Dump from June 24 2010, signal format values

```
var GS_SIGFORM_NTSC = 285229085; //0x1100401D var GS_SIGFORM_PAL =  
295714841; //0x11A04019 var GS_SIGFORM_CCIR_NTSC = 287326749; //0x1120421D  
var GS_SIGFORM_CCIR_NTSC_P483 = 554713629; //0x2110421D var  
GS_SIGFORM_CCIR_PAL = 295715353; //0x11A04219 var  
GS_SIGFORM_CCIR_PNTSC_30 = 555762206; //0x2120421E var  
GS_SIGFORM_CCIR_PPAL_25 = 564150809; //0x21A04219 var  
GS_SIGFORM_CCIR_NTSC2398 = 287326743; //0x11204217 var GS_SIGFORM_HD360  
= 291524637; //0x1160501D var GS_SIGFORM_ALT_NTSC = 287330333; //0x1120501D  
var GS_SIGFORM_ALT_PAL = 295718937; //0x11A05019 var  
GS_SIGFORM_1035i_30_260M = 319861278; //0x1310B21E var  
GS_SIGFORM_1035i_30X_260M = 319861277; //0x1310B21D var  
GS_SIGFORM_1080i_30 = 327201310; //0x1380B21E var GS_SIGFORM_1080sf_30 =  
1132507678; //0x4380B21E var GS_SIGFORM_1080i_30X = 327201309; //0x1380B21D  
var GS_SIGFORM_1080sf_30X = 1132507677; //0x4380B21D var  
GS_SIGFORM_1080i_25 = 327201305; //0x1380B219 var GS_SIGFORM_1080sf_25 =  
1132507673; //0x4380B219 var GS_SIGFORM_1080i_24 = 327201304; //0x1380B218 var  
GS_SIGFORM_1080sf_24 = 1132507672; //0x4380B218 var GS_SIGFORM_1080i_24X =  
327201303; //0x1380B217 var GS_SIGFORM_1080sf_24X = 1132507671; //0x4380B217  
var GS_SIGFORM_1080_30 = 595636766; //0x2380B21E var GS_SIGFORM_1080_30X =  
595636765; //0x2380B21D var GS_SIGFORM_1080_25 = 595636761; //0x2380B219 var  
GS_SIGFORM_1080_24 = 595636760; //0x2380B218 var GS_SIGFORM_1080_24X =  
595636759; //0x2380B217 var GS_SIGFORM_1080_60 = 595636796; //0x2380B23C var  
GS_SIGFORM_1080_60X = 595636795; //0x2380B23B var GS_SIGFORM_1080_50 =  
595636786; //0x2380B232 var GS_SIGFORM_1080_48 = ; var GS_SIGFORM_1080_48X =  
; var GS_SIGFORM_720_60 = 571510844; //0x2210903C var GS_SIGFORM_720_60X =  
571510843; //0x2210903B var GS_SIGFORM_720_50 = 571510834; //0x22109032 var  
GS_SIGFORM_720_30 = 571510814; //0x2210901E var GS_SIGFORM_720_30X =  
571510813; //0x2210901D var GS_SIGFORM_720_25 = 571510809; //0x22109019 var  
GS_SIGFORM_720_24 = 571510808; //0x22109018 var GS_SIGFORM_VESA_640_72 =  
553664584; //0x21004048 var GS_SIGFORM_VESA_800_71X =  
565200455; //0x21B04647 var GS_SIGFORM_VESA_800_72 = 565200456; //0x21B04648  
var GS_SIGFORM_VESA_1024_71X = 572547143; //0x22206047 var  
GS_SIGFORM_VESA_1024_72 = 572547144; //0x22206048 var  
GS_SIGFORM_VESA_1280_24 = 587239448; //0x23009018 var  
GS_SIGFORM_VESA_1280i_30 = 318803998; //0x1300901E var  
GS_SIGFORM_VESA_1280_71X = 587239495; //0x23009047 var  
GS_SIGFORM_VESA_1280_72 = 587239496; //0x23009048 var  
GS_SIGFORM_VESA_1600i_30 = 336637982; //0x1410B01E var  
GS_SIGFORM_DVI_1400_1050_24 = 591435288; //0x23409618 var
```

GS_SIGFORM_DVI_1400_1050_25	=	591435289;	//0x23409619	var
GS_SIGFORM_DCIN_2048_25	=	595640345;	//0x2380C019	var
GS_SIGFORM_DCIN_2048sf_25	=	1132511257;	//0x4380C019	var
GS_SIGFORM_DCIN_2048sf_24X	=	1132511255;	//0x4380C017	var
GS_SIGFORM_DCIN_2048sf_24	=	1132511256;	//0x4380C018	var
GS_SIGFORM_DCIN_2048_24X	=	595640343;	//0x2380C017	var
GS_SIGFORM_DCIN_2048_24	=	595640344;	//0x2380C018	var
GS_SIGFORM_FILM_1828_778_24	=	580955160;	//0x22A0AC18	var
GS_SIGFORM_FILM_1828_778_25	=	580955161;	//0x22A0AC19	var
GS_SIGFORM_FILM_1828_988_24	=	575712280;	//0x2250AC18	var
GS_SIGFORM_FILM_1828_988_25	=	575712281;	//0x2250AC19	var
GS_SIGFORM_FILM_1828_1102_24	=	601926680;	//0x23E0AC18	var
GS_SIGFORM_FILM_1828_1102_25	=	601926681;	//0x23E0AC19	var
GS_SIGFORM_FILM_1828_1332_24	=	613461016;	//0x2490AC18	var
GS_SIGFORM_FILM_1828_1332_25	=	613461017;	//0x2490AC19	var
GS_SIGFORM_FILM_2048_857_24	=	576765976;	//0x2260C018	var
GS_SIGFORM_FILM_2048_857_25	=	576765977;	//0x2260C019	var
GS_SIGFORM_FILM_2048_872_24	=	582008856;	//0x22B0C018	var
GS_SIGFORM_FILM_2048_872_25	=	582008857;	//0x22B0C019	var
GS_SIGFORM_FILM_2048_1102_24	=	601931800;	//0x23E0C018	var
GS_SIGFORM_FILM_2048_1102_25	=	601931801;	//0x23E0C019	var
GS_SIGFORM_FILM_2048_1234_24	=	607174680;	//0x2430C018	var
GS_SIGFORM_FILM_2048_1234_25	=	607174681;	//0x2430C019	var
GS_SIGFORM_FILM_2048_15X	=	621854734;	//0x2510C00E	var
GS_SIGFORM_FILM_2048_14	=	621854734;	//0x2510C00E	var
GS_SIGFORM_FILM_2048_15	=	621854735;	//0x2510C00F	var
GS_SIGFORM_FILM_2048sf_15X	=	1158725646;	//0x4510C00E	var
GS_SIGFORM_FILM_2048sf_15	=	1158725647;	//0x4510C00F	var
GS_SIGFORM_FILM_2048_24X	=	621854743;	//0x2510C017	var
GS_SIGFORM_FILM_2048_24	=	621854744;	//0x2510C018	var
GS_SIGFORM_FILM_2048sf_24X	=	1158725655;	//0x4510C017	var
GS_SIGFORM_FILM_2048sf_24	=	1158725656;	//0x4510C018	var
GS_SIGFORM_FILM_2048_48	=	621854768;	//0x2510C030	var
GS_SIGFORM_FILM_2048_1536_25	=	620806169;	//0x2500C019	var
GS_SIGFORM_FILM_2048_1536sf_25	=	1157677081;	//0x4500C019	var
GS_SIGFORM_FILM_2048_25	=	621854745;	//0x2510C019	var
GS_SIGFORM_FILM_2048sf_25	=	1158725657;	//0x4510C019	var
GS_SIGFORM_FILM_2048_1536_15X	=	620806158;	//0x2500C00E	var
GS_SIGFORM_FILM_2048_1536_15	=	620806159;	//0x2500C00F	var
GS_SIGFORM_FILM_2048_1536sf_15X	=	1157677070;	//0x4500C00E	var
GS_SIGFORM_FILM_2048_1536sf_15	=	1157677071;	//0x4500C00F	var
GS_SIGFORM_FILM_2048_1536_24X	=	620806167;	//0x2500C017	var
GS_SIGFORM_FILM_2048_1536_24	=	620806168;	//0x2500C018	var
GS_SIGFORM_FILM_2048_1536sf_24X	=	1157677079;	//0x4500C017	var
GS_SIGFORM_FILM_2048_1536sf_24	=	1157677080;	//0x4500C018	var
GS_SIGFORM_FILM_2048_1536_48X	=	620806191;	//0x2500C02F	var
GS_SIGFORM_FILM_2048_1536_48	=	620806192;	//0x2500C030	var
GS_SIGFORM_FILM_4096_1714_24	=	628162584;	//0x25710018	var
GS_SIGFORM_FILM_4096_1714_24X	=	628162583;	//0x25710017	var
GS_SIGFORM_FILM_4096_3112sf_5	=	1186004997;	//0x46B10005	var
GS_SIGFORM_FILM_4096_3112_24	=	649134104;	//0x26B10018	var

GS_SIGFORM_FILM_4096_3112_24X = 649134103; //0x26B10017

Definition at line 6197 of file mediacmd.h.

#define GS_SIGFORM_DCIN_2048_24 (GS_SIGFORMSIZE_2048x1080 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_24)

gsGetSetValue::gsSignalFormat Digital Cinema

Definition at line 5964 of file mediacmd.h.

#define GS_SIGFORM_DCIN_2048_24X (GS_SIGFORMSIZE_2048x1080 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_23_98)

gsGetSetValue::gsSignalFormat Digital Cinema

Definition at line 5962 of file mediacmd.h.

#define GS_SIGFORM_DCIN_2048_25 (GS_SIGFORMSIZE_2048x1080 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_25)

gsGetSetValue::gsSignalFormat Presentation

Definition at line 5952 of file mediacmd.h.

#define GS_SIGFORM_DCIN_2048sf_24 (GS_SIGFORMSIZE_2048x1080 | GS_SIGFORMTYPE_SEGMENTEDFRAME | GS_SIGFORMFRAMERATE_24)

gsGetSetValue::gsSignalFormat Digital Cinema

Definition at line 5960 of file mediacmd.h.

#define GS_SIGFORM_DCIN_2048sf_24X (GS_SIGFORMSIZE_2048x1080 | GS_SIGFORMTYPE_SEGMENTEDFRAME | GS_SIGFORMFRAMERATE_23_98)

gsGetSetValue::gsSignalFormat Digital Cinema

Digital cinema 2048x1080

Definition at line 5958 of file mediacmd.h.

#define GS_SIGFORM_DCIN_2048sf_25 (GS_SIGFORMSIZE_2048x1080 | GS_SIGFORMTYPE_SEGMENTEDFRAME | GS_SIGFORMFRAMERATE_25)

Definition at line 5953 of file mediacmd.h.

#define GS_SIGFORM_DVI_1400_1050_24 (GS_SIGFORMSIZE_1400x1050 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_24)

gsGetSetValue::gsSignalFormat Presentation

Presentation res

Definition at line 5948 of file mediacmd.h.

```
#define GS_SIGFORM_DVI_1400_1050_25 (GS_SIGFORMSIZE_1400x1050 |  
GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_25)
```

gsGetSetValue::gsSignalFormat Presentation

Definition at line 5950 of file mediacmd.h.

```
#define GS_SIGFORM_FILM_1828_1102_24 (GS_SIGFORMSIZE_1828x1102 |  
GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_24)
```

gsGetSetValue::gsSignalFormat Film

Definition at line 5975 of file mediacmd.h.

```
#define GS_SIGFORM_FILM_1828_1102_25 (GS_SIGFORMSIZE_1828x1102 |  
GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_25)
```

gsGetSetValue::gsSignalFormat Film

Definition at line 5977 of file mediacmd.h.

```
#define GS_SIGFORM_FILM_1828_1332_24 (GS_SIGFORMSIZE_1828x1332 |  
GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_24)
```

gsGetSetValue::gsSignalFormat Film

Definition at line 5979 of file mediacmd.h.

```
#define GS_SIGFORM_FILM_1828_1332_25 (GS_SIGFORMSIZE_1828x1332 |  
GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_25)
```

gsGetSetValue::gsSignalFormat Film

Definition at line 5981 of file mediacmd.h.

```
#define GS_SIGFORM_FILM_1828_778_24 (GS_SIGFORMSIZE_1828x778 |  
GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_24)
```

gsGetSetValue::gsSignalFormat Film

Definition at line 5967 of file mediacmd.h.

```
#define GS_SIGFORM_FILM_1828_778_25 (GS_SIGFORMSIZE_1828x778 |  
GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_25)
```

gsGetSetValue::gsSignalFormat Film

Definition at line 5969 of file mediacmd.h.

#define GS_SIGFORM_FILM_1828_988_24 (GS_SIGFORMSIZE_1828x988 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_24)

gsGetSetValue::gsSignalFormat Film
Definition at line 5971 of file mediacmd.h.

#define GS_SIGFORM_FILM_1828_988_25 (GS_SIGFORMSIZE_1828x988 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_25)

gsGetSetValue::gsSignalFormat Film
Definition at line 5973 of file mediacmd.h.

#define GS_SIGFORM_FILM_2048_1102_24 (GS_SIGFORMSIZE_2048x1102 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_24)

gsGetSetValue::gsSignalFormat Film
Definition at line 5993 of file mediacmd.h.

#define GS_SIGFORM_FILM_2048_1102_25 (GS_SIGFORMSIZE_2048x1102 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_25)

gsGetSetValue::gsSignalFormat Film
Definition at line 5995 of file mediacmd.h.

#define GS_SIGFORM_FILM_2048_1234_24 (GS_SIGFORMSIZE_2048x1234 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_24)

gsGetSetValue::gsSignalFormat Film
Definition at line 5997 of file mediacmd.h.

#define GS_SIGFORM_FILM_2048_1234_25 (GS_SIGFORMSIZE_2048x1234 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_25)

gsGetSetValue::gsSignalFormat Film
Definition at line 5999 of file mediacmd.h.

#define GS_SIGFORM_FILM_2048_14 GS_SIGFORM_FILM_2048_15X

gsGetSetValue::gsSignalFormat Film 2K
Definition at line 6006 of file mediacmd.h.

#define GS_SIGFORM_FILM_2048_15 (GS_SIGFORMSIZE_2048x1556 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_15)

gsGetSetValue::gsSignalFormat Film 2K

Definition at line 6008 of file mediacmd.h.

#define GS_SIGFORM_FILM_2048_1536_15 (GS_SIGFORMSIZE_2048x1536 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_15)

gsGetSetValue::gsSignalFormat Film 2K(1536)

Definition at line 6036 of file mediacmd.h.

#define GS_SIGFORM_FILM_2048_1536_15X (GS_SIGFORMSIZE_2048x1536 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_14_98)

gsGetSetValue::gsSignalFormat Film 2K(1536)

Definition at line 6034 of file mediacmd.h.

#define GS_SIGFORM_FILM_2048_1536_24 (GS_SIGFORMSIZE_2048x1536 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_24)

gsGetSetValue::gsSignalFormat Film 2K(1536)

Definition at line 6044 of file mediacmd.h.

#define GS_SIGFORM_FILM_2048_1536_24X (GS_SIGFORMSIZE_2048x1536 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_23_98)

gsGetSetValue::gsSignalFormat Film 2K(1536)

Definition at line 6042 of file mediacmd.h.

#define GS_SIGFORM_FILM_2048_1536_25 (GS_SIGFORMSIZE_2048x1536 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_25)

gsGetSetValue::gsSignalFormat Film 2K(1536)

Definition at line 6025 of file mediacmd.h.

#define GS_SIGFORM_FILM_2048_1536_48 (GS_SIGFORMSIZE_2048x1536 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_48)

gsGetSetValue::gsSignalFormat Film 2K(1536)

Definition at line 6052 of file mediacmd.h.

#define GS_SIGFORM_FILM_2048_1536_48X (GS_SIGFORMSIZE_2048x1536 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_47_95)

gsGetSetValue::gsSignalFormat Film 2K(1536)

Definition at line 6050 of file mediacmd.h.

#define GS_SIGFORM_FILM_2048_1536sf_15 (GS_SIGFORMSIZE_2048x1536 | GS_SIGFORMTYPE_SEGMENTEDFRAME | GS_SIGFORMFRAMERATE_15)

gsGetSetValue::gsSignalFormat Film 2K(1536)

Definition at line 6040 of file mediacmd.h.

#define GS_SIGFORM_FILM_2048_1536sf_15X (GS_SIGFORMSIZE_2048x1536 | GS_SIGFORMTYPE_SEGMENTEDFRAME | GS_SIGFORMFRAMERATE_14_98)

gsGetSetValue::gsSignalFormat Film 2K(1536)

Definition at line 6038 of file mediacmd.h.

#define GS_SIGFORM_FILM_2048_1536sf_24 (GS_SIGFORMSIZE_2048x1536 | GS_SIGFORMTYPE_SEGMENTEDFRAME | GS_SIGFORMFRAMERATE_24)

gsGetSetValue::gsSignalFormat Film 2K(1536)

Definition at line 6048 of file mediacmd.h.

#define GS_SIGFORM_FILM_2048_1536sf_24X (GS_SIGFORMSIZE_2048x1536 | GS_SIGFORMTYPE_SEGMENTEDFRAME | GS_SIGFORMFRAMERATE_23_98)

gsGetSetValue::gsSignalFormat Film 2K(1536)

Definition at line 6046 of file mediacmd.h.

#define GS_SIGFORM_FILM_2048_1536sf_25 (GS_SIGFORMSIZE_2048x1536 | GS_SIGFORMTYPE_SEGMENTEDFRAME | GS_SIGFORMFRAMERATE_25)

gsGetSetValue::gsSignalFormat Film 2K(1536)

Definition at line 6027 of file mediacmd.h.

#define GS_SIGFORM_FILM_2048_15X (GS_SIGFORMSIZE_2048x1556 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_14_98)

gsGetSetValue::gsSignalFormat Film 2K

Film transfer

Definition at line 6004 of file mediacmd.h.

#define GS_SIGFORM_FILM_2048_24 (GS_SIGFORMSIZE_2048x1556 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_24)

gsGetSetValue::gsSignalFormat Film 2K

Definition at line 6016 of file mediacmd.h.

#define GS_SIGFORM_FILM_2048_24X (GS_SIGFORMSIZE_2048x1556 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_23_98)

gsGetSetValue::gsSignalFormat Film 2K

Definition at line 6014 of file mediacmd.h.

#define GS_SIGFORM_FILM_2048_25 (GS_SIGFORMSIZE_2048x1556 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_25)

gsGetSetValue::gsSignalFormat Film 2K(1536)

Definition at line 6029 of file mediacmd.h.

#define GS_SIGFORM_FILM_2048_48 (GS_SIGFORMSIZE_2048x1556 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_48)

gsGetSetValue::gsSignalFormat Film 2K

Definition at line 6022 of file mediacmd.h.

#define GS_SIGFORM_FILM_2048_857_24 (GS_SIGFORMSIZE_2048x857 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_24)

gsGetSetValue::gsSignalFormat Film

Definition at line 5984 of file mediacmd.h.

#define GS_SIGFORM_FILM_2048_857_25 (GS_SIGFORMSIZE_2048x857 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_25)

gsGetSetValue::gsSignalFormat Film

Definition at line 5986 of file mediacmd.h.

#define GS_SIGFORM_FILM_2048_872_24 (GS_SIGFORMSIZE_2048x872 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_24)

gsGetSetValue::gsSignalFormat Film

Definition at line 5989 of file mediacmd.h.

#define GS_SIGFORM_FILM_2048_872_25 (GS_SIGFORMSIZE_2048x872 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_25)

gsGetSetValue::gsSignalFormat Film

Definition at line 5991 of file mediacmd.h.

#define GS_SIGFORM_FILM_2048sf_15 (GS_SIGFORMSIZE_2048x1556 | GS_SIGFORMTYPE_SEGMENTEDFRAME | GS_SIGFORMFRAMERATE_15)

gsGetSetValue::gsSignalFormat Film 2K

Definition at line 6012 of file mediacmd.h.

#define GS_SIGFORM_FILM_2048sf_15X (GS_SIGFORMSIZE_2048x1556 | GS_SIGFORMTYPE_SEGMENTEDFRAME | GS_SIGFORMFRAMERATE_14_98)

gsGetSetValue::gsSignalFormat Film 2K

Definition at line 6010 of file mediacmd.h.

#define GS_SIGFORM_FILM_2048sf_24 (GS_SIGFORMSIZE_2048x1556 | GS_SIGFORMTYPE_SEGMENTEDFRAME | GS_SIGFORMFRAMERATE_24)

gsGetSetValue::gsSignalFormat Film 2K

Definition at line 6020 of file mediacmd.h.

#define GS_SIGFORM_FILM_2048sf_24X (GS_SIGFORMSIZE_2048x1556 | GS_SIGFORMTYPE_SEGMENTEDFRAME | GS_SIGFORMFRAMERATE_23_98)

gsGetSetValue::gsSignalFormat Film 2K

Definition at line 6018 of file mediacmd.h.

#define GS_SIGFORM_FILM_2048sf_25 (GS_SIGFORMSIZE_2048x1556 | GS_SIGFORMTYPE_SEGMENTEDFRAME | GS_SIGFORMFRAMERATE_25)

gsGetSetValue::gsSignalFormat Film 2K(1536)

Definition at line 6031 of file mediacmd.h.

#define GS_SIGFORM_FILM_4096_1714_24 (GS_SIGFORMSIZE_4096x1714 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_24)

gsGetSetValue::gsSignalFormat Film 4K Half

Definition at line 6076 of file mediacmd.h.

#define GS_SIGFORM_FILM_4096_1714_24X (GS_SIGFORMSIZE_4096x1714 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_23_98)

gsGetSetValue::gsSignalFormat Film 4K Half

Definition at line 6078 of file mediacmd.h.

#define GS_SIGFORM_FILM_4096_3112_24 (GS_SIGFORMSIZE_4096x3112 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_24)

gsGetSetValue::gsSignalFormat Film 4K

Definition at line 6083 of file mediacmd.h.

#define GS_SIGFORM_FILM_4096_3112_24X (GS_SIGFORMSIZE_4096x3112 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_23_98)

gsGetSetValue::gsSignalFormat Film 4K

Definition at line 6085 of file mediacmd.h.

#define GS_SIGFORM_FILM_4096_3112sf_5 (GS_SIGFORMSIZE_4096x3112 | GS_SIGFORMTYPE_SEGMENTEDFRAME | GS_SIGFORMFRAMERATE_5)

gsGetSetValue::gsSignalFormat Film 4K

Definition at line 6081 of file mediacmd.h.

#define GS_SIGFORM_HD360 (GS_SIGFORMSIZE_960x504 | GS_SIGFORMTYPE_INTERLACED | GS_SIGFORMFRAMERATE_29_97)

Signal format compressed HD 960x504 29.97.

Definition at line 5857 of file mediacmd.h.

#define GS_SIGFORM_NOT_PRESENT 0

For input and genlock status returns.

Definition at line 6199 of file mediacmd.h.

#define GS_SIGFORM_NTSC (GS_SIGFORMSIZE_640x480 | GS_SIGFORMTYPE_INTERLACED | GS_SIGFORMFRAMERATE_29_97)

Signal format NTSC square pixel (320x240 or 640x480) @ 29.97 or 30 fps
gsGetSetValue::gsSignalFormat.

Definition at line 5841 of file mediacmd.h.

#define GS_SIGFORM_PAL (GS_SIGFORMSIZE_640x576 | GS_SIGFORMTYPE_INTERLACED | GS_SIGFORMFRAMERATE_25)

Signal format PAL square pixel (320x288 or 640x576) @ 25 fps
gsGetSetValue::gsSignalFormat.

Definition at line 5843 of file mediacmd.h.

#define GS_SIGFORM_QUADHD_24 (GS_SIGFORMSIZE_3840x2880 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_24)

gsGetSetValue::gsSignalFormat Quad HD 24 fps (4 x 1920x1080)

Definition at line 6059 of file mediacmd.h.

#define GS_SIGFORM_QUADHD_24X (GS_SIGFORMSIZE_3840x2880 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_23_98)

gsGetSetValue::gsSignalFormat Quad HD 23.98 fps (4 x 1920x1080)

Definition at line 6055 of file mediacmd.h.

#define GS_SIGFORM_QUADHD_25 (GS_SIGFORMSIZE_3840x2880 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_25)

Definition at line 6062 of file mediacmd.h.

#define GS_SIGFORM_QUADHDsf_24 (GS_SIGFORMSIZE_3840x2880 | GS_SIGFORMTYPE_SEGMENTEDFRAME | GS_SIGFORMFRAMERATE_24)

gsGetSetValue::gsSignalFormat Quad HD 24 sf fps (4 x 1920x1080)

Definition at line 6061 of file mediacmd.h.

#define GS_SIGFORM_QUADHDsf_24X (GS_SIGFORMSIZE_3840x2880 | GS_SIGFORMTYPE_SEGMENTEDFRAME | GS_SIGFORMFRAMERATE_23_98)

gsGetSetValue::gsSignalFormat Quad HD 23.98 sf fps (4 x 1920x1080)

Definition at line 6057 of file mediacmd.h.

#define GS_SIGFORM_QUADHDsf_25 (GS_SIGFORMSIZE_3840x2880 | GS_SIGFORMTYPE_SEGMENTEDFRAME | GS_SIGFORMFRAMERATE_25)

Definition at line 6063 of file mediacmd.h.

#define GS_SIGFORM_SUPPORTS_1035 0x00000020

1035x1080 Production rasters

Definition at line 6215 of file mediacmd.h.

#define GS_SIGFORM_SUPPORTS_1080 0x00000040

1080/1088/1092/1112x1920 HD rasters

Definition at line 6217 of file mediacmd.h.

#define GS_SIGFORM_SUPPORTS_1080_X2 (GS_SIGFORM_SUPPORTS_1080|GS_SIGFORM_SUPPORTS_X2)

Supports 1080p 50/59/60.

Definition at line 6249 of file mediacmd.h.

#define GS_SIGFORM_SUPPORTS_1400 0x00000800

Presentation 1440x1050.

Definition at line 6227 of file mediacmd.h.

#define GS_SIGFORM_SUPPORTS_1536 0x00000100

Film 2K 1536 lines.

Definition at line 6221 of file mediacmd.h.

#define GS_SIGFORM_SUPPORTS_1556 0x00000200

Film 2K 1556 lines.

Definition at line 6223 of file mediacmd.h.

#define GS_SIGFORM_SUPPORTS_360 0x00000008

360 compressed, not used

Definition at line 6211 of file mediacmd.h.

#define GS_SIGFORM_SUPPORTS_720 0x00000010

720p Rasters (59/60 and sometimes 50)

Definition at line 6213 of file mediacmd.h.

#define GS_SIGFORM_SUPPORTS_DCIN 0x00000400

Digital Cinema 2048x1080.

Definition at line 6225 of file mediacmd.h.

#define GS_SIGFORM_SUPPORTS_EXTRA8 0x00000080

1088 HD rasters

Definition at line 6219 of file mediacmd.h.

#define GS_SIGFORM_SUPPORTS_HR 0x00000004

960 width SD, not used
Definition at line 6209 of file mediacmd.h.

#define GS_SIGFORM_SUPPORTS_NTSC 0x00000001

NTSC (CCIR or sqp) 720x480/486/508/512.
Supported formats
Definition at line 6205 of file mediacmd.h.

#define GS_SIGFORM_SUPPORTS_PAL 0x00000002

PAL (CCIR or sqp) 720x576/608.
Definition at line 6207 of file mediacmd.h.

#define GS_SIGFORM_SUPPORTS_QUADSDI 0x00001000

Quad HD-SDI 3840 and 4096.
Definition at line 6229 of file mediacmd.h.

#define GS_SIGFORM_SUPPORTS_V1024 0x00080000

Vesa 1280x1024.
Definition at line 6237 of file mediacmd.h.

#define GS_SIGFORM_SUPPORTS_V1200 0x00100000

Vesa 1600x1200.
Definition at line 6239 of file mediacmd.h.

#define GS_SIGFORM_SUPPORTS_V1600 0x00200000

Vesa 1600.
Definition at line 6241 of file mediacmd.h.

#define GS_SIGFORM_SUPPORTS_V480 0x00010000

Vesa 640x480.
Definition at line 6231 of file mediacmd.h.

#define GS_SIGFORM_SUPPORTS_V600 0x00020000

Vesa 800x600.

Definition at line 6233 of file mediacmd.h.

#define GS_SIGFORM_SUPPORTS_V768 0x00040000

Vesa 1024x768.

Definition at line 6235 of file mediacmd.h.

#define GS_SIGFORM_SUPPORTS_X2 0x20000000

Modifier times 2.

Definition at line 6243 of file mediacmd.h.

#define GS_SIGFORM_SUPPORTS_X3 0x40000000

Modifier times 3.

Definition at line 6245 of file mediacmd.h.

#define GS_SIGFORM_SUPPORTS_X4 0x80000000

Modifier times 4.

Definition at line 6247 of file mediacmd.h.

#define GS_SIGFORM_VESA_1024_71X (GS_SIGFORMSIZE_1024x768 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_71_93)

gsGetSetValue::gsSignalFormat Vesa 1024x768@71.9

Definition at line 5931 of file mediacmd.h.

#define GS_SIGFORM_VESA_1024_72 (GS_SIGFORMSIZE_1024x768 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_72)

gsGetSetValue::gsSignalFormat Vesa 1024x766@72

Definition at line 5933 of file mediacmd.h.

#define GS_SIGFORM_VESA_1280_24 (GS_SIGFORMSIZE_1280x1024 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_24)

gsGetSetValue::gsSignalFormat Vesa 1280x1024@24

Definition at line 5935 of file mediacmd.h.

#define GS_SIGFORM_VESA_1280_71X (GS_SIGFORMSIZE_1280x1024 | GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_71_93)

gsGetSetValue::gsSignalFormat Vesa 1280x1024@71.9

Definition at line 5939 of file mediacmd.h.

```
#define GS_SIGFORM_VESA_1280_72 (GS_SIGFORMSIZE_1280x1024 |  
GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_72)
```

```
gsGetSetValue::gsSignalFormat Vesa 1280x1024@72
```

Definition at line 5941 of file mediacmd.h.

```
#define GS_SIGFORM_VESA_1280i_30 (GS_SIGFORMSIZE_1280x1024 |  
GS_SIGFORMTYPE_INTERLACED | GS_SIGFORMFRAMERATE_30)
```

```
gsGetSetValue::gsSignalFormat Vesa 1280x1024@30
```

Definition at line 5937 of file mediacmd.h.

```
#define GS_SIGFORM_VESA_1600i_30 (GS_SIGFORMSIZE_1600x1200 |  
GS_SIGFORMTYPE_INTERLACED | GS_SIGFORMFRAMERATE_30)
```

```
gsGetSetValue::gsSignalFormat Vesa 1600x1200i@30
```

Definition at line 5943 of file mediacmd.h.

```
#define GS_SIGFORM_VESA_640_72 (GS_SIGFORMSIZE_640x480 |  
GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_72)
```

```
gsGetSetValue::gsSignalFormat Vesa 640x480@72
```

VGA res

Definition at line 5925 of file mediacmd.h.

```
#define GS_SIGFORM_VESA_800_71X (GS_SIGFORMSIZE_800x600 |  
GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_71_93)
```

```
gsGetSetValue::gsSignalFormat Vesa 800x600@71.9
```

Definition at line 5927 of file mediacmd.h.

```
#define GS_SIGFORM_VESA_800_72 (GS_SIGFORMSIZE_800x600 |  
GS_SIGFORMTYPE_PROGRESSIVE | GS_SIGFORMFRAMERATE_72)
```

```
gsGetSetValue::gsSignalFormat Vesa 800x600@72
```

Definition at line 5929 of file mediacmd.h.

```
#define GS_SIGFORMFRAMERATE_10 10
```

Definition at line 5703 of file mediacmd.h.

#define GS_SIGFORMFRAMERATE_100 100

Definition at line 5718 of file mediacmd.h.

#define GS_SIGFORMFRAMERATE_119_88 119

Definition at line 5719 of file mediacmd.h.

#define GS_SIGFORMFRAMERATE_14_98 14

Definition at line 5704 of file mediacmd.h.

#define GS_SIGFORMFRAMERATE_15 15

Definition at line 5705 of file mediacmd.h.

#define GS_SIGFORMFRAMERATE_23_98 23

Definition at line 5706 of file mediacmd.h.

#define GS_SIGFORMFRAMERATE_24 24

Definition at line 5707 of file mediacmd.h.

#define GS_SIGFORMFRAMERATE_25 25

Definition at line 5708 of file mediacmd.h.

#define GS_SIGFORMFRAMERATE_29_97 29

Definition at line 5709 of file mediacmd.h.

#define GS_SIGFORMFRAMERATE_30 30

Definition at line 5710 of file mediacmd.h.

#define GS_SIGFORMFRAMERATE_47_95 47

Definition at line 5711 of file mediacmd.h.

#define GS_SIGFORMFRAMERATE_48 48

Definition at line 5712 of file mediacmd.h.

#define GS_SIGFORMFRAMERATE_5 5

Definition at line 5700 of file mediacmd.h.

#define GS_SIGFORMFRAMERATE_50 50

Definition at line 5713 of file mediacmd.h.

#define GS_SIGFORMFRAMERATE_59_94 59

Definition at line 5714 of file mediacmd.h.

#define GS_SIGFORMFRAMERATE_6 6

Definition at line 5701 of file mediacmd.h.

#define GS_SIGFORMFRAMERATE_60 60

Definition at line 5715 of file mediacmd.h.

#define GS_SIGFORMFRAMERATE_71_93 71

Definition at line 5716 of file mediacmd.h.

#define GS_SIGFORMFRAMERATE_72 72

Definition at line 5717 of file mediacmd.h.

#define GS_SIGFORMFRAMERATE_7_5 7

Definition at line 5702 of file mediacmd.h.

#define GS_SIGFORMFRAMERATE_CUSTOM 0x100

Definition at line 5720 of file mediacmd.h.

#define GS_SIGFORMMASK_FRAMERATE 0x000001ff

Frame rate mask (portion of return for frame rate)

Definition at line 5681 of file mediacmd.h.

#define GS_SIGFORMMASK_FRAMETYPE 0xF0000000UL

Frame type mask (portion of return for frame rate)

Definition at line 5693 of file mediacmd.h.

#define GS_SIGFORMMASK_HORIZONTAL 0x000ffe00

Horizontal / 8 mask (portion of return for frame rate)

Definition at line 5685 of file mediacmd.h.

#define GS_SIGFORMMASK_VERTICAL 0x0ff00000

Vertical / 8 mask (portion of return for frame rate)

Definition at line 5689 of file mediacmd.h.

#define GS_SIGFORMSHIFT_FRAMERATE 0

Shift frame rate to 0.

Definition at line 5683 of file mediacmd.h.

#define GS_SIGFORMSHIFT_FRAMETYPE 28

Frame type shift to 0.

Definition at line 5695 of file mediacmd.h.

#define GS_SIGFORMSHIFT_HORIZONTAL 9

Horizontal / 8 shift to 0.

Definition at line 5687 of file mediacmd.h.

#define GS_SIGFORMSHIFT_VERTICAL 20

Vertical / 8 shift to 0.

Definition at line 5691 of file mediacmd.h.

#define GS_SIGFORMSIZE_1024 0x30

Definition at line 5751 of file mediacmd.h.

**#define GS_SIGFORMSIZE_1024x1024 ((GS_SIGFORMSIZE_1024 <<
GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_1024 <<
GS_SIGFORMSHIFT_VERTICAL))**

Definition at line 5800 of file mediacmd.h.


```
#define GS_SIGFORMSIZE_1024x768 ((GS_SIGFORMSIZE_1024 <<  
GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_768 <<  
GS_SIGFORMSHIFT_VERTICAL))
```

Definition at line 5799 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_1035 0x31
```

Definition at line 5752 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_1044 0x32
```

Definition at line 5753 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_1050 0x34
```

Definition at line 5755 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_1052 0x33
```

Definition at line 5754 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_1080 0x38
```

Definition at line 5756 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_1088 0x39
```

Definition at line 5757 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_1096 0x3a
```

Definition at line 5758 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_1102 0x3E
```

Definition at line 5759 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_1152 0x40
```

Definition at line 5760 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_1152x864 ((GS_SIGFORMSIZE_1152 <<  
GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_864 <<  
GS_SIGFORMSHIFT_VERTICAL))
```

Definition at line 5801 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_1200 0x41
```

Definition at line 5761 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_1234 0x43
```

Definition at line 5762 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_1280 0x48
```

Definition at line 5763 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_1280x1024 ((GS_SIGFORMSIZE_1280 <<  
GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_1024 <<  
GS_SIGFORMSHIFT_VERTICAL))
```

Definition at line 5802 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_1280x720 ((GS_SIGFORMSIZE_1280 <<  
GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_720 <<  
GS_SIGFORMSHIFT_VERTICAL))
```

Definition at line 5805 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_1332 0x49
```

Definition at line 5764 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_1400 0x4B
```

Definition at line 5765 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_1400x1050 ((GS_SIGFORMSIZE_1400 <<  
GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_1050 <<  
GS_SIGFORMSHIFT_VERTICAL))
```

Definition at line 5803 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_1440 0x4C
```

Definition at line 5766 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_1536 0x50
```

Definition at line 5767 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_1556 0x51
```

Definition at line 5768 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_1588 0x52
```

Definition at line 5769 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_1600 0x58
```

Definition at line 5772 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_1600x1200 ((GS_SIGFORMSIZE_1600 <<  
GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_1200 <<  
GS_SIGFORMSHIFT_VERTICAL))
```

Definition at line 5804 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_1714 0x57
```

Definition at line 5771 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_1782 0x5A
```

Definition at line 5774 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_1828 0x56
```

Definition at line 5770 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_1828x1102 ((GS_SIGFORMSIZE_1828 <<  
GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_1102 <<  
GS_SIGFORMSHIFT_VERTICAL))
```

Definition at line 5808 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_1828x1332 ((GS_SIGFORMSIZE_1828 <<  
GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_1332 <<  
GS_SIGFORMSHIFT_VERTICAL))
```

Definition at line 5809 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_1828x778 ((GS_SIGFORMSIZE_1828 <<  
GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_778 <<  
GS_SIGFORMSHIFT_VERTICAL))
```

Definition at line 5806 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_1828x988 ((GS_SIGFORMSIZE_1828 <<  
GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_988 <<  
GS_SIGFORMSHIFT_VERTICAL))
```

Definition at line 5807 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_1920 0x59
```

Definition at line 5773 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_1920x1035 ((GS_SIGFORMSIZE_1920 <<  
GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_1035 <<  
GS_SIGFORMSHIFT_VERTICAL))
```

Definition at line 5810 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_1920x1080 ((GS_SIGFORMSIZE_1920 <<  
GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_1080 <<  
GS_SIGFORMSHIFT_VERTICAL))
```

Definition at line 5811 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_1920x1088 ((GS_SIGFORMSIZE_1920 <<  
GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_1088 <<  
GS_SIGFORMSHIFT_VERTICAL))
```

Definition at line 5812 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_2048 0x60
```

Definition at line 5775 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_2048x1080 ((GS_SIGFORMSIZE_2048 <<  
GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_1080 <<  
GS_SIGFORMSHIFT_VERTICAL))
```

Definition at line 5818 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_2048x1102 ((GS_SIGFORMSIZE_2048 <<  
GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_1102 <<  
GS_SIGFORMSHIFT_VERTICAL))
```

Definition at line 5816 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_2048x1234 ((GS_SIGFORMSIZE_2048 <<  
GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_1234 <<  
GS_SIGFORMSHIFT_VERTICAL))
```

Definition at line 5817 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_2048x1536 ((GS_SIGFORMSIZE_2048 <<  
GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_1536 <<  
GS_SIGFORMSHIFT_VERTICAL))
```

Definition at line 5819 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_2048x1556 ((GS_SIGFORMSIZE_2048 <<  
GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_1556 <<  
GS_SIGFORMSHIFT_VERTICAL))
```

Definition at line 5820 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_2048x857 ((GS_SIGFORMSIZE_2048 <<  
GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_857 <<  
GS_SIGFORMSHIFT_VERTICAL))
```

Definition at line 5814 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_2048x872 ((GS_SIGFORMSIZE_2048 <<  
GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_872 <<  
GS_SIGFORMSHIFT_VERTICAL))
```

Definition at line 5815 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_2160 0x64
```

Definition at line 5776 of file mediacmd.h.

#define GS_SIGFORMSIZE_240 0x01

Definition at line 5725 of file mediacmd.h.

#define GS_SIGFORMSIZE_243 0x02

Definition at line 5726 of file mediacmd.h.

#define GS_SIGFORMSIZE_2560x1080 ((GS_SIGFORMSIZE_2650 << GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_1080 << GS_SIGFORMSHIFT_VERTICAL))

Definition at line 5813 of file mediacmd.h.

#define GS_SIGFORMSIZE_2650 0x68

Definition at line 5777 of file mediacmd.h.

#define GS_SIGFORMSIZE_288 0x03

Definition at line 5727 of file mediacmd.h.

#define GS_SIGFORMSIZE_2880 0x6A

Definition at line 5778 of file mediacmd.h.

#define GS_SIGFORMSIZE_2880x1782 ((GS_SIGFORMSIZE_2880 << GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_1782 << GS_SIGFORMSHIFT_VERTICAL))

Definition at line 5830 of file mediacmd.h.

#define GS_SIGFORMSIZE_2880x2160 ((GS_SIGFORMSIZE_2880 << GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_2160 << GS_SIGFORMSHIFT_VERTICAL))

Definition at line 5828 of file mediacmd.h.

#define GS_SIGFORMSIZE_3112 0x6b

Definition at line 5779 of file mediacmd.h.

#define GS_SIGFORMSIZE_320 0x08

Definition at line 5728 of file mediacmd.h.

#define GS_SIGFORMSIZE_352 0x09

Definition at line 5729 of file mediacmd.h.

#define GS_SIGFORMSIZE_360 0x0a

Definition at line 5730 of file mediacmd.h.

#define GS_SIGFORMSIZE_3840 0x78

Definition at line 5780 of file mediacmd.h.

#define GS_SIGFORMSIZE_3840x2880 ((GS_SIGFORMSIZE_3840 << GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_2880 << GS_SIGFORMSHIFT_VERTICAL))

Definition at line 5823 of file mediacmd.h.

#define GS_SIGFORMSIZE_4096 0x80

Definition at line 5781 of file mediacmd.h.

#define GS_SIGFORMSIZE_4096x1714 ((GS_SIGFORMSIZE_4096 << GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_1714 << GS_SIGFORMSHIFT_VERTICAL))

Definition at line 5821 of file mediacmd.h.

#define GS_SIGFORMSIZE_4096x2880 ((GS_SIGFORMSIZE_4096 << GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_2880 << GS_SIGFORMSHIFT_VERTICAL))

Definition at line 5825 of file mediacmd.h.

#define GS_SIGFORMSIZE_4096x3112 ((GS_SIGFORMSIZE_4096 << GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_3112 << GS_SIGFORMSHIFT_VERTICAL))

Definition at line 5826 of file mediacmd.h.

#define GS_SIGFORMSIZE_480 0x10

Definition at line 5731 of file mediacmd.h.

#define GS_SIGFORMSIZE_483 0x11

Definition at line 5732 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_486 0x12
```

Definition at line 5733 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_496 0x14
```

Definition at line 5734 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_504 0x16
```

Definition at line 5735 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_512 0x17
```

Definition at line 5736 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_576 0x1a
```

Definition at line 5737 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_600 0x1b
```

Definition at line 5738 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_608 0x1c
```

Definition at line 5739 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_640 0x20
```

Definition at line 5740 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_640x480 ((GS_SIGFORMSIZE_640 <<  
GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_480 <<  
GS_SIGFORMSHIFT_VERTICAL))
```

Definition at line 5786 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_640x576 ((GS_SIGFORMSIZE_640 <<  
GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_576 <<  
GS_SIGFORMSHIFT_VERTICAL))
```

Definition at line 5787 of file mediacmd.h.

#define GS_SIGFORMSIZE_720 0x21

Definition at line 5741 of file mediacmd.h.

#define GS_SIGFORMSIZE_720x480 ((GS_SIGFORMSIZE_720 << GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_480 << GS_SIGFORMSHIFT_VERTICAL))

Definition at line 5788 of file mediacmd.h.

#define GS_SIGFORMSIZE_720x483 ((GS_SIGFORMSIZE_720 << GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_483 << GS_SIGFORMSHIFT_VERTICAL))

Definition at line 5789 of file mediacmd.h.

#define GS_SIGFORMSIZE_720x486 ((GS_SIGFORMSIZE_720 << GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_486 << GS_SIGFORMSHIFT_VERTICAL))

Definition at line 5790 of file mediacmd.h.

#define GS_SIGFORMSIZE_720x504 ((GS_SIGFORMSIZE_720 << GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_504 << GS_SIGFORMSHIFT_VERTICAL))

Definition at line 5794 of file mediacmd.h.

#define GS_SIGFORMSIZE_720x512 ((GS_SIGFORMSIZE_720 << GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_512 << GS_SIGFORMSHIFT_VERTICAL))

Definition at line 5791 of file mediacmd.h.

#define GS_SIGFORMSIZE_720x576 ((GS_SIGFORMSIZE_720 << GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_576 << GS_SIGFORMSHIFT_VERTICAL))

Definition at line 5792 of file mediacmd.h.

#define GS_SIGFORMSIZE_720x608 ((GS_SIGFORMSIZE_720 << GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_608 << GS_SIGFORMSHIFT_VERTICAL))

Definition at line 5793 of file mediacmd.h.

#define GS_SIGFORMSIZE_768 0x22

Definition at line 5742 of file mediacmd.h.

#define GS_SIGFORMSIZE_778 0x2A

Definition at line 5749 of file mediacmd.h.

#define GS_SIGFORMSIZE_800 0x23

Definition at line 5743 of file mediacmd.h.

#define GS_SIGFORMSIZE_800x600 ((GS_SIGFORMSIZE_800 << GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_600 << GS_SIGFORMSHIFT_VERTICAL))

Definition at line 5795 of file mediacmd.h.

#define GS_SIGFORMSIZE_857 0x26

Definition at line 5746 of file mediacmd.h.

#define GS_SIGFORMSIZE_864 0x24

Definition at line 5744 of file mediacmd.h.

#define GS_SIGFORMSIZE_872 0x2B

Definition at line 5750 of file mediacmd.h.

#define GS_SIGFORMSIZE_960 0x28

Definition at line 5747 of file mediacmd.h.

#define GS_SIGFORMSIZE_960x486 ((GS_SIGFORMSIZE_960 << GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_486 << GS_SIGFORMSHIFT_VERTICAL))

Definition at line 5796 of file mediacmd.h.

#define GS_SIGFORMSIZE_960x504 ((GS_SIGFORMSIZE_960 << GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_504 << GS_SIGFORMSHIFT_VERTICAL))

Definition at line 5798 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_960x576 ((GS_SIGFORMSIZE_960 <<  
GS_SIGFORMSHIFT_HORIZONTAL) | (GS_SIGFORMSIZE_576 <<  
GS_SIGFORMSHIFT_VERTICAL))
```

Definition at line 5797 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_968 0x29
```

Definition at line 5748 of file mediacmd.h.

```
#define GS_SIGFORMSIZE_988 0x25
```

Definition at line 5745 of file mediacmd.h.

```
#define GS_SIGFORMTYPE_INTERLACED (1 << GS_SIGFORMSHIFT_FRAMETYPE)
```

Definition at line 5836 of file mediacmd.h.

```
#define GS_SIGFORMTYPE_PROGRESSIVE (2 << GS_SIGFORMSHIFT_FRAMETYPE)
```

Definition at line 5837 of file mediacmd.h.

```
#define GS_SIGFORMTYPE_SEGMENTEDFRAME (4 <<  
GS_SIGFORMSHIFT_FRAMETYPE)
```

Definition at line 5838 of file mediacmd.h.

```
#define GS_SIGFORMTYPE_UNKNOWN (0)
```

Definition at line 5835 of file mediacmd.h.

```
#define GS_SOURCEPRECEDENCE_FRAMECOUNT 0x10000000
```

Use the record frame count.

Definition at line 5228 of file mediacmd.h.

```
#define GS_SOURCEPRECEDENCE_IRIG 0x00000020
```

Time of day from IRIG RP-215 encode.

Definition at line 5224 of file mediacmd.h.

```
#define GS_SOURCEPRECEDENCE_RP188_L 0x00000002
```

RP-188 Ancillary time code audio.

Definition at line 5216 of file mediacmd.h.

#define GS_SOURCEPRECEDENCE_RP188_V 0x00000001

RP-188 Ancillary time code video.

Definition at line 5214 of file mediacmd.h.

#define GS_SOURCEPRECEDENCE_RP215 0x00000040

Time of day from ILM style RP-215 A->LTC, V-VITC.

Definition at line 5226 of file mediacmd.h.

#define GS_SOURCEPRECEDENCE_SMPTE 0x00000004

SMPTE LTC audio time code.

Definition at line 5218 of file mediacmd.h.

#define GS_SOURCEPRECEDENCE_TOD 0x00000008

Time of day from computer clock (or GPS if available)

Definition at line 5220 of file mediacmd.h.

#define GS_SOURCEPRECEDENCE_VITC 0x00000010

Time of day from VITC encoded line, or D-VITC in HD.

Definition at line 5222 of file mediacmd.h.

#define GS_SUPFILE_ANY 0x80000000

Any supported MediaReactor format (gsGetSetValue::gsSupportedFileTypes
gsGetSetValue::gsIgnoreFileTypes) gsGetSetValue::gsRecFileFormat
gsGetSetValue::gsRecAudFileFormat gsGetSetValue::gsConvertFileFormat
gsGetSetValue::gsConvertAudFileFormat

Definition at line 6378 of file mediacmd.h.

#define GS_SUPFILE_AUDONLY 0x00010000

Audio only or separate audio formats (gsGetSetValue::gsSupportedFileTypes
gsGetSetValue::gsIgnoreFileTypes) gsGetSetValue::gsRecFileFormat
gsGetSetValue::gsRecAudFileFormat gsGetSetValue::gsConvertFileFormat
gsGetSetValue::gsConvertAudFileFormat

Definition at line 6369 of file mediacmd.h.

#define GS_SUPFILE_AVI 0x00000001

Standard Windows AVI container (gsGetSetValue::gsSupportedFileTypes
gsGetSetValue::gsIgnoreFileTypes) gsGetSetValue::gsRecFileFormat

gsGetSetValue::gsRecAudFileFormat
gsGetSetValue::gsConvertAudFileFormat
Definition at line 6354 of file mediacmd.h.

gsGetSetValue::gsConvertFileFormat

#define GS_SUPFILE_FIX 0x00000100

Drastic Fixed Frame container
gsGetSetValue::gsIgnoreFileTypes)
gsGetSetValue::gsRecAudFileFormat
gsGetSetValue::gsConvertAudFileFormat
Definition at line 6366 of file mediacmd.h.

(gsGetSetValue::gsSupportedFileTypes
gsGetSetValue::gsRecFileFormat
gsGetSetValue::gsConvertFileFormat

#define GS_SUPFILE_ODML 0x00000002

OpenDML AVI container
gsGetSetValue::gsIgnoreFileTypes)
gsGetSetValue::gsRecAudFileFormat
gsGetSetValue::gsConvertAudFileFormat
Definition at line 6357 of file mediacmd.h.

(gsGetSetValue::gsSupportedFileTypes
gsGetSetValue::gsRecFileFormat
gsGetSetValue::gsConvertFileFormat

#define GS_SUPFILE_OMFI 0x00000008

Avid Open Media Format container
gsGetSetValue::gsIgnoreFileTypes)
gsGetSetValue::gsRecAudFileFormat
gsGetSetValue::gsConvertAudFileFormat
Definition at line 6363 of file mediacmd.h.

(gsGetSetValue::gsSupportedFileTypes
gsGetSetValue::gsRecFileFormat
gsGetSetValue::gsConvertFileFormat

#define GS_SUPFILE_QT 0x00000004

QuickTime Mov/MooV container
gsGetSetValue::gsIgnoreFileTypes)
gsGetSetValue::gsRecAudFileFormat
gsGetSetValue::gsConvertAudFileFormat
Definition at line 6360 of file mediacmd.h.

(gsGetSetValue::gsSupportedFileTypes
gsGetSetValue::gsRecFileFormat
gsGetSetValue::gsConvertFileFormat

#define GS_SUPFILE_STILLS 0x00100000

Series of still file formats
gsGetSetValue::gsIgnoreFileTypes)
gsGetSetValue::gsRecAudFileFormat
gsGetSetValue::gsConvertAudFileFormat
Definition at line 6372 of file mediacmd.h.

(gsGetSetValue::gsSupportedFileTypes
gsGetSetValue::gsRecFileFormat
gsGetSetValue::gsConvertFileFormat

#define GS_SUPFILE_UNK 0x40000000

Other unspecified formats
gsGetSetValue::gsIgnoreFileTypes)
gsGetSetValue::gsRecAudFileFormat
gsGetSetValue::gsConvertAudFileFormat
Definition at line 6375 of file mediacmd.h.

(gsGetSetValue::gsSupportedFileTypes
gsGetSetValue::gsRecFileFormat
gsGetSetValue::gsConvertFileFormat

#define GS_TCSOURCE_CLIP 7

For cmdGetSetValue::gsTcSource - Using absolute clip.
Definition at line 5173 of file mediacmd.h.

#define GS_TCSOURCE_CTL 4

For cmdGetSetValue::gsTcSource - Using CTL.
Definition at line 5171 of file mediacmd.h.

#define GS_TCSOURCE_IRIG 8

For cmdGetSetValue::gsTcSource - Using irig natural (not converted) time code.
Definition at line 5175 of file mediacmd.h.

#define GS_TCSOURCE_LTC 1

For cmdGetSetValue::gsTcSource - Using LTC.
Definition at line 5167 of file mediacmd.h.

#define GS_TCSOURCE_VITC 2

For cmdGetSetValue::gsTcSource - Using VITC.
Definition at line 5169 of file mediacmd.h.

#define GS_TCSRC_DISABLE_EXTERNAL 0x8000

cmdGetSetValue::gsTimecodeSources Don't use any external time code of any kind
Definition at line 6566 of file mediacmd.h.

#define GS_TCSRC_FORCE_VTR_TC 0x0002

cmdGetSetValue::gsTimecodeSources Use the RS-422 VTR time code
Definition at line 6568 of file mediacmd.h.

#define GS_TCSRC_USE_TIMEOFDAY 0x0080

cmdGetSetValue::gsTimecodeSources Use the time of day as time code
Definition at line 6570 of file mediacmd.h.

#define GS_TRUE 0x01

True for boolean cmdType::ctGetValue, cmdType::ctSetValue

Definition at line 6712 of file mediacmd.h.

#define GS_UNITY 0xFFFFFFFF

Set value to unity (levels, tbc) or default (compression type, amount)

Definition at line 6734 of file mediacmd.h.

#define GS_UPCONVERT_ANAMORPHIC 0x0001

Upconvert to whole screen.

Definition at line 5425 of file mediacmd.h.

#define GS_UPCONVERT_LETTERBOX 0x0008

Upconvert to letter box.

Definition at line 5431 of file mediacmd.h.

#define GS_UPCONVERT_PILLARBOX 0x0002

Upconvert with bars.

Definition at line 5427 of file mediacmd.h.

#define GS_UPCONVERT_ZOOM14x9 0x0004

Upconvert with some zoom.

Definition at line 5429 of file mediacmd.h.

#define GS_UPCONVERT_ZOOMWIDE 0x0010

Upconvert to wide zoom.

Definition at line 5433 of file mediacmd.h.

#define GS_USERRIGHTS_ADD 0x0100

Definition at line 5671 of file mediacmd.h.

#define GS_USERRIGHTS_ADMIN 0x8000

Definition at line 5674 of file mediacmd.h.

#define GS_USERRIGHTS_DELETE 0x0200

Definition at line 5672 of file mediacmd.h.

#define GS_USERRIGHTS_FULL 0x7FFF

Definition at line 5673 of file mediacmd.h.

#define GS_USERRIGHTS_MODIFY 0x0002

Definition at line 5666 of file mediacmd.h.

#define GS_USERRIGHTS_NONE 0x0000

Definition at line 5664 of file mediacmd.h.

#define GS_USERRIGHTS_PLAY 0x0010

Definition at line 5669 of file mediacmd.h.

#define GS_USERRIGHTS_READ 0x0001

Definition at line 5665 of file mediacmd.h.

#define GS_USERRIGHTS_RECORD 0x0020

Definition at line 5670 of file mediacmd.h.

#define GS_USERRIGHTS_SETUP 0x0008

Definition at line 5668 of file mediacmd.h.

#define GS_USERRIGHTS_WRITE 0x0004

Definition at line 5667 of file mediacmd.h.

#define GS_VIDBAND_HIGH 0x04

Allow high bandwidth (gsGetSetValue::gsVidInBandwidth
gsGetSetValue::gsVidBandwidth)

Definition at line 5374 of file mediacmd.h.

#define GS_VIDBAND_MEDIUM 0x02

Allow medium bandwidth (gsGetSetValue::gsVidInBandwidth
gsGetSetValue::gsVidBandwidth)

Definition at line 5372 of file mediacmd.h.

#define GS_VIDBAND_NOTCH 0x08

Impose notch filter on bandwidth (gsGetSetValue::gsVidInBandwidth
gsGetSetValue::gsVidBandwidth)

Definition at line 5376 of file mediacmd.h.

#define GS_VIDBAND_STANDARD 0x01

Allow normal bandwidth (gsGetSetValue::gsVidInBandwidth
gsGetSetValue::gsVidBandwidth)

Definition at line 5370 of file mediacmd.h.

#define GS_VIDBLACK_CRYSTAL 0x02

Crystal black level (0 IRE NTSC, 0 IRE PAL) gsGetSetValue::gsVidBlackSetup
gsGetSetValue::gsVidInBlack.

Definition at line 5381 of file mediacmd.h.

#define GS_VIDBLACK_SETUP 0x01

Black at normal level (7.5 IRE NTSC, 0 IRE PAL) gsGetSetValue::gsVidBlackSetup
gsGetSetValue::gsVidInBlack.

Definition at line 5379 of file mediacmd.h.

#define GS_VIDBLACK_SUPER 0x04

Super black level (0 > IRE NTSC/PAL) gsGetSetValue::gsVidBlackSetup
gsGetSetValue::gsVidInBlack.

Definition at line 5383 of file mediacmd.h.

#define GS_VIDFREEZE_FIELD0 1

Freeze - first (0) field (cmdGetSetValue::gsVidFreeze)

Definition at line 5302 of file mediacmd.h.

#define GS_VIDFREEZE_FIELD1 2

Freeze - second (1) field (cmdGetSetValue::gsVidFreeze)

Definition at line 5304 of file mediacmd.h.

#define GS_VIDFREEZE_FRAME 3

Freeze - both fields (cmdGetSetValue::gsVidFreeze)

Definition at line 5306 of file mediacmd.h.

#define GS_VIDFREEZE_NOT_FROZEN 0

Freeze - no freeze (cmdGetSetValue::gsVidFreeze)

Definition at line 5300 of file mediacmd.h.

#define GS_VIDLOCKTYPE_BROADCAST 2

Perfect lock for cmdGetSetValue::gsVidInLockType cmdGetSetValue::gsVidOutLockType
cmdGetSetValue::gsVidOutLockType.

Definition at line 5367 of file mediacmd.h.

#define GS_VIDLOCKTYPE_VTR 1

VTR (unruly hsync) lock for cmdGetSetValue::gsVidInLockType
cmdGetSetValue::gsVidOutLockType cmdGetSetValue::gsVidOutLockType.

Definition at line 5365 of file mediacmd.h.

#define GS_VIDREF_DISABLE 0x00000000

Disable video reference (gsGetSetValue::gsVidRefDisable)

Definition at line 6414 of file mediacmd.h.

#define GS_VIDREF_ENABLE 0x00000001

Enable video reference (gsGetSetValue::gsVidRefDisable)

Definition at line 6416 of file mediacmd.h.

#define GS_VIDSELECT_3G_DUAL_LINK 0x2000000

Dual link 4:4:4 over 1 cable.

Definition at line 5360 of file mediacmd.h.

#define GS_VIDSELECT_3G_DUAL_RATE 0x1000000

2 HDSDI YCbCr signals at once

Definition at line 5358 of file mediacmd.h.

#define GS_VIDSELECT_COMPONENT_RGB 0x080

RGB at video standard rate (cmdGetSetValue::gsVidInSelect
cmdGetSetValue::gsVidOutSelect)

Definition at line 5323 of file mediacmd.h.

#define GS_VIDSELECT_COMPONENT_YUV 0x010

BetaCam level YCrCb NTSC or PAL video (cmdGetSetValue::gsVidInSelect
cmdGetSetValue::gsVidOutSelect)

Definition at line 5317 of file mediacmd.h.

#define GS_VIDSELECT_COMPONENT_YUV_2 0x2000

Secondary NTSC or PAL video (often monitor selection) (cmdGetSetValue::gsVidInSelect
cmdGetSetValue::gsVidOutSelect)

Definition at line 5336 of file mediacmd.h.

#define GS_VIDSELECT_COMPONENT_YUV_JAPAN 0x20000

BetaCam level YCrCb NTSC or PAL video (cmdGetSetValue::gsVidInSelect
cmdGetSetValue::gsVidOutSelect)

Definition at line 5344 of file mediacmd.h.

#define GS_VIDSELECT_COMPONENT_YUV_M2 0x020

Panasonic M2 level YCrCb NTSC or PAL video (cmdGetSetValue::gsVidInSelect
cmdGetSetValue::gsVidOutSelect)

Definition at line 5319 of file mediacmd.h.

#define GS_VIDSELECT_COMPONENT_YUV_SMPTE 0x040

SMPTE standard level YCrCb NTSC or PAL video (cmdGetSetValue::gsVidInSelect
cmdGetSetValue::gsVidOutSelect)

Definition at line 5321 of file mediacmd.h.

#define GS_VIDSELECT_COMPONENT_YUV_SMPTE_JAPAN 0x40000

SMPTE standard level YCrCb NTSC or PAL video (cmdGetSetValue::gsVidInSelect
cmdGetSetValue::gsVidOutSelect)

Definition at line 5346 of file mediacmd.h.

#define GS_VIDSELECT_COMPOSITE 0x001

Standard NTSC or PAL composite video (cmdGetSetValue::gsVidInSelect
cmdGetSetValue::gsVidOutSelect)

Definition at line 5309 of file mediacmd.h.

#define GS_VIDSELECT_COMPOSITE_2 0x004

Secondary NTSC or PAL video (often monitor selection) (cmdGetSetValue::gsVidInSelect cmdGetSetValue::gsVidOutSelect)

Definition at line 5313 of file mediacmd.h.

#define GS_VIDSELECT_COMPOSITE_3 0x008

third NTSC or PAL video (often monitor selection) (cmdGetSetValue::gsVidInSelect cmdGetSetValue::gsVidOutSelect)

Definition at line 5315 of file mediacmd.h.

#define GS_VIDSELECT_COMPOSITE_4 0x800

Extra for Digital Rapids--order is screwed up but as long as it works I guess Secondary NTSC or PAL video (often monitor selection) (cmdGetSetValue::gsVidInSelect cmdGetSetValue::gsVidOutSelect)

Definition at line 5332 of file mediacmd.h.

#define GS_VIDSELECT_COMPOSITE_JAPAN 0x8000

Standard NTSC or PAL composite video (cmdGetSetValue::gsVidInSelect cmdGetSetValue::gsVidOutSelect)

Definition at line 5340 of file mediacmd.h.

#define GS_VIDSELECT_D1_PARALLEL 0x200

D1 Serial Parallel video (cmdGetSetValue::gsVidInSelect cmdGetSetValue::gsVidOutSelect)

Definition at line 5327 of file mediacmd.h.

#define GS_VIDSELECT_D1_SERIAL 0x100

D1 Serial Digital or HDS DI video (cmdGetSetValue::gsVidInSelect cmdGetSetValue::gsVidOutSelect)

Definition at line 5325 of file mediacmd.h.

#define GS_VIDSELECT_D1_SERIAL_2 0x4000

Secondary NTSC or PAL video (often monitor selection) (cmdGetSetValue::gsVidInSelect cmdGetSetValue::gsVidOutSelect)

Definition at line 5338 of file mediacmd.h.

#define GS_VIDSELECT_DVI 0x800000

DVI Protocol.

Definition at line 5356 of file mediacmd.h.

#define GS_VIDSELECT_HDMI 0x100000

HDMI - Auto YCbCr/RGB.

Definition at line 5350 of file mediacmd.h.

#define GS_VIDSELECT_HDMI_RGB 0x200000

HDMI - RGB I/O.

Definition at line 5352 of file mediacmd.h.

#define GS_VIDSELECT_HDMI_YCBCR 0x400000

HDMI - YCBCR I/O.

Definition at line 5354 of file mediacmd.h.

#define GS_VIDSELECT_NONE 0

No video available or no configurable settings (cmdGetSetValue::gsVidInSelect
cmdGetSetValue::gsVidOutSelect)

Definition at line 5362 of file mediacmd.h.

#define GS_VIDSELECT_SDTI 0x400

SDTI/SDI including high speed transfer video (cmdGetSetValue::gsVidInSelect
cmdGetSetValue::gsVidOutSelect)

Definition at line 5329 of file mediacmd.h.

#define GS_VIDSELECT_SVIDEO 0x002

SVHS/S-Video four wire NTSC or PAL video (cmdGetSetValue::gsVidInSelect
cmdGetSetValue::gsVidOutSelect)

Definition at line 5311 of file mediacmd.h.

#define GS_VIDSELECT_SVIDEO_2 0x1000

Secondary NTSC or PAL video (often monitor selection) (cmdGetSetValue::gsVidInSelect
cmdGetSetValue::gsVidOutSelect)

Definition at line 5334 of file mediacmd.h.

#define GS_VIDSELECT_SVIDEO_JAPAN 0x10000

SVHS/S-Video four wire NTSC or PAL video (cmdGetSetValue::gsVidInSelect
cmdGetSetValue::gsVidOutSelect)

Definition at line 5342 of file mediacmd.h.

#define GS_VIDSELECT_XVID_RGB 0x80000

xVGA compatible analog RGB

Definition at line 5348 of file mediacmd.h.

#define GS_VIDWHITE_CLAMP 0x01

Whites are clamped or 100 IRE (gsGetSetValue::gsVidInWhite)

Definition at line 5386 of file mediacmd.h.

#define GS_VIDWHITE_FREE 0x04

Whites are allowed to be greater than 100 IRE (gsGetSetValue::gsVidInWhite)

Definition at line 5390 of file mediacmd.h.

#define GS_VIDWHITE_SCALE 0x02

Whites are scaled automatically from black level to 100 IRE (gsGetSetValue::gsVidInWhite)

Definition at line 5388 of file mediacmd.h.

#define GS_WAVEAUDIO_2 0x00002000

Definition at line 6528 of file mediacmd.h.

#define GS_WAVEAUDIO_4 0x00004000

Definition at line 6529 of file mediacmd.h.

#define GS_WAVEAUDIO_8 0x00008000

Definition at line 6530 of file mediacmd.h.

#define GS_WAVEAUDIO_LISSAJOUSXY 0x00000800

Audio Lissajous X-Y.

Definition at line 6525 of file mediacmd.h.

#define GS_WAVEAUDIO_MASK 0x0000FE00

Audio types mask.

Definition at line 6532 of file mediacmd.h.

#define GS_WAVEAUDIO_SURROUND 0x00001000

Surround monitor.

Definition at line 6527 of file mediacmd.h.

#define GS_WAVEAUDIO_WAVE 0x00000200

Audio wave form.

Definition at line 6521 of file mediacmd.h.

#define GS_WAVEVAUDIO_METERS 0x00000400

Audio meters.

Definition at line 6523 of file mediacmd.h.

#define GS_WAVEVECTOR_ALT_GRATICULE 0x10000000

Definition at line 6535 of file mediacmd.h.

#define GS_WAVEVECTOR_CHANNEL_A 0x00080000

cmdGetSetValue::gsWaveVectorSetup dwStart color channel ALPHA

Definition at line 6557 of file mediacmd.h.

#define GS_WAVEVECTOR_CHANNEL_B 0x00040000

cmdGetSetValue::gsWaveVectorSetup dwStart color channel BLUE

Definition at line 6555 of file mediacmd.h.

#define GS_WAVEVECTOR_CHANNEL_CB 0x00200000

cmdGetSetValue::gsWaveVectorSetup dwStart color channel CB

Definition at line 6561 of file mediacmd.h.

#define GS_WAVEVECTOR_CHANNEL_CR 0x00400000

cmdGetSetValue::gsWaveVectorSetup dwStart color channel CR

Definition at line 6563 of file mediacmd.h.

#define GS_WAVEVECTOR_CHANNEL_G 0x00020000

cmdGetSetValue::gsWaveVectorSetup dwStart color channel GREEN
Definition at line 6553 of file mediacmd.h.

#define GS_WAVEVECTOR_CHANNEL_MASK 0x00FF0000

Definition at line 6549 of file mediacmd.h.

#define GS_WAVEVECTOR_CHANNEL_R 0x00010000

cmdGetSetValue::gsWaveVectorSetup dwStart color channel RED
Definition at line 6551 of file mediacmd.h.

#define GS_WAVEVECTOR_CHANNEL_Y 0x00100000

cmdGetSetValue::gsWaveVectorSetup dwStart color channel Y (Luma)
Definition at line 6559 of file mediacmd.h.

#define GS_WAVEVECTOR_DATA 0x00000080

cmdGetSetValue::gsWaveVectorSetup illegal colors
Definition at line 6515 of file mediacmd.h.

#define GS_WAVEVECTOR_FLAG_MASK 0xFFFF0000

Definition at line 6534 of file mediacmd.h.

#define GS_WAVEVECTOR_HIDE100VECTOR 0x02000000

Definition at line 6538 of file mediacmd.h.

#define GS_WAVEVECTOR_HIDE75VECTOR 0x01000000

Definition at line 6537 of file mediacmd.h.

#define GS_WAVEVECTOR_HIDEANGLES 0x08000000

Definition at line 6540 of file mediacmd.h.

#define GS_WAVEVECTOR_HIDEFLESHVECTOR 0x04000000

Definition at line 6539 of file mediacmd.h.

#define GS_WAVEVECTOR_HISTOGRAM 0x00000020

cmdGetSetValue::gsWaveVectorSetup histogram
Definition at line 6511 of file mediacmd.h.

#define GS_WAVEVECTOR_HISTOGRAM_SEP 0x00000040

cmdGetSetValue::gsWaveVectorSetup parade histogram
Definition at line 6513 of file mediacmd.h.

#define GS_WAVEVECTOR_ILLEGAL 0x00000100

cmdGetSetValue::gsWaveVectorSetup illegal colors
Definition at line 6517 of file mediacmd.h.

#define GS_WAVEVECTOR_MASK 0x000001FF

Video types mask.
Definition at line 6519 of file mediacmd.h.

#define GS_WAVEVECTOR_PICT_CLEAN 0x00000000

Definition at line 6545 of file mediacmd.h.

#define GS_WAVEVECTOR_PICT_SAFE 0x01000000

Definition at line 6546 of file mediacmd.h.

#define GS_WAVEVECTOR_PICT_TITLE_SAFE 0x02000000

Definition at line 6547 of file mediacmd.h.

#define GS_WAVEVECTOR_PICTURE 0x00000001

cmdGetSetValue::gsWaveVectorSetup standard picture
Definition at line 6501 of file mediacmd.h.

#define GS_WAVEVECTOR_USEFULLSCALE 0x01000000

Definition at line 6543 of file mediacmd.h.

#define GS_WAVEVECTOR_USESMPTESCALE 0x00000000

Definition at line 6542 of file mediacmd.h.

#define GS_WAVEVECTOR_VECTORSCOPE 0x00000002

cmdGetSetValue::gsWaveVectorSetup standard vectorscope

Definition at line 6503 of file mediacmd.h.

#define GS_WAVEVECTOR_WAVEFORM 0x00000004

cmdGetSetValue::gsWaveVectorSetup standard waveform

Definition at line 6505 of file mediacmd.h.

#define GS_WAVEVECTOR_WAVEFORM_RGB 0x00000008

cmdGetSetValue::gsWaveVectorSetup parade RGB waveform

Definition at line 6507 of file mediacmd.h.

#define GS_WAVEVECTOR_WAVEFORM_YCBCR 0x00000010

cmdGetSetValue::gsWaveVectorSetup parade Y CB CR waveform

Definition at line 6509 of file mediacmd.h.

#define GS_XML_CHECK_OPEN 7

cmdGetSetValue::gsDTProjectToXml

Definition at line 6655 of file mediacmd.h.

#define GS_XML_CLIPFILE 2

cmdGetSetValue::gsDTProjectToXml

Definition at line 6645 of file mediacmd.h.

#define GS_XML_DWORD 5

cmdGetSetValue::gsDTProjectToXml

Definition at line 6651 of file mediacmd.h.

#define GS_XML_EDL 3

cmdGetSetValue::gsDTProjectToXml

Definition at line 6647 of file mediacmd.h.

#define GS_XML_FILENAME 1

cmdGetSetValue::gsDTProjectToXml
Definition at line 6643 of file mediacmd.h.

#define GS_XML_OPEN_DELETE_FILES 8

cmdGetSetValue::gsDTProjectToXml
Definition at line 6657 of file mediacmd.h.

#define GS_XML_OPEN_IGNORE_FILES 9

cmdGetSetValue::gsDTProjectToXml
Definition at line 6659 of file mediacmd.h.

#define GS_XML_SAVE 6

cmdGetSetValue::gsDTProjectToXml
Definition at line 6653 of file mediacmd.h.

#define GS_XML_STRING 4

cmdGetSetValue::gsDTProjectToXml
Definition at line 6649 of file mediacmd.h.

#define gsAllowIndependantChanConfig gsAllowIndependentChanConfig

Spelling error.
Definition at line 5153 of file mediacmd.h.

#define infChanAll 0xFFFFFFFFFUL

Definition at line 503 of file mediacmd.h.

#define INIT_MEDIACMD(__mCmd_)

```
Value: {
    memset(&__mCmd_, 0, sizeof(MEDIACMD));
    __mCmd_.dwCmdID = MEDIACMD_CURRENT;
    __mCmd_.dwStructSize = sizeof(MEDIACMD);
    __mCmd_.dwChannel = CHAN_ILLEGAL;
    __mCmd_.lSpeed = SPD_ILLEGAL;
    __mCmd_.dwVideoChannels = (DWORD)vidChanAll;
    __mCmd_.dwAudioChannels = (DWORD)audChanAll;
    __mCmd_.dwInfoChannels = (DWORD)infChanAll;
    __mCmd_.dwCmdAlt = 0;

    __mCmd_.dwPosition = TC_ILLEGAL;
    __mCmd_.dwStart = TC_ILLEGAL;
}
```

```
    __mCmd_.dwEnd = TC_ILLEGAL;    }
```

Initialize a media cmd structure to all illegal (no command)

Definition at line 7012 of file mediacmd.h.

```
#define INIT_PMEDIACMD( __mCmd_)
```

```
Value: {
    memset( __mCmd_, 0, sizeof(MEDIACMD) );
    ((PMEDIACMD) __mCmd_)->dwCmdID = MEDIACMD_CURRENT;
    ((PMEDIACMD) __mCmd_)->dwStructSize = sizeof(MEDIACMD);
    ((PMEDIACMD) __mCmd_)->dwChannel = CHAN_ILLEGAL;
    ((PMEDIACMD) __mCmd_)->lSpeed = SPD_ILLEGAL;
    ((PMEDIACMD) __mCmd_)->dwVideoChannels = (DWORD)vidChanAll;
    ((PMEDIACMD) __mCmd_)->dwAudioChannels = (DWORD)audChanAll;
    ((PMEDIACMD) __mCmd_)->dwInfoChannels = (DWORD)infChanAll;
    ((PMEDIACMD) __mCmd_)->dwCmdAlt = CHAN_ILLEGAL;
    ((PMEDIACMD) __mCmd_)->dwPosition = TC_ILLEGAL;
    ((PMEDIACMD) __mCmd_)->dwStart = TC_ILLEGAL;
    ((PMEDIACMD) __mCmd_)->dwEnd = TC_ILLEGAL;
}
```

Initialize a media cmd pointer to all illegal (no command)

Definition at line 7027 of file mediacmd.h.

```
#define MEDIACMD_CHECK_MASK 0xFFFF0000UL
```

Mask for permanent magic number of command id. See [MEDIACMD::dwCmdID](#).

Definition at line 55 of file mediacmd.h.

```
#define MEDIACMD_CHECK_VER 0xFA250000UL
```

Permanent magic number of command id. See [MEDIACMD::dwCmdID](#).

Definition at line 53 of file mediacmd.h.

```
#define MEDIACMD_CURRENT (MEDIACMD_VERSION_MAJOR |  
MEDIACMD_VERSION_MINOR | MEDIACMD_CHECK_VER)
```

Current version and magic number. Place in [MEDIACMD::dwCmdID](#).

Definition at line 57 of file mediacmd.h.

```
#define MEDIACMD_VERSION_MAJOR 0x0200UL
```

Major command versioning for upgrades to the command set. See [MEDIACMD::dwCmdID](#).

Definition at line 47 of file mediacmd.h.

```
#define MEDIACMD_VERSION_MAJOR_X32 0x0101UL
```

Major command versioning for upgrades to the command set. See [MEDIACMD::dwCmdID](#).

Definition at line 40 of file mediacmd.h.

#define MEDIACMD_VERSION_MASK 0xFFFFUL

Mask for checking the command set version. See [MEDIACMD::dwCmdID](#).
Definition at line 51 of file mediacmd.h.

#define MEDIACMD_VERSION_MINOR 0x0000UL

Minor command versioning for upgrades to the command set. See [MEDIACMD::dwCmdID](#).
Definition at line 49 of file mediacmd.h.

#define MEDIACMD_VERSION_MINOR_X32 0x0003UL

Minor command versioning for upgrades to the command set. See [MEDIACMD::dwCmdID](#).
Definition at line 42 of file mediacmd.h.

#define METABASE_TYPE_CHAR 1

Set cmdGetSetValue::gsMetaDataReadWrite.
Definition at line 5242 of file mediacmd.h.

#define METABASE_TYPE_INT 2

Set cmdGetSetValue::gsMetaDataReadWrite.
Definition at line 5244 of file mediacmd.h.

#define METABASE_TYPE_UNKNOWN 0

Set cmdGetSetValue::gsMetaDataReadWrite.
Definition at line 5240 of file mediacmd.h.

#define pCmdQueueElem [PMEDIACMD](#)

Old name, use PMEDIACMD instead.
Definition at line 6953 of file mediacmd.h.

#define SIZEOF_MEDIACMD sizeof([MEDIACMD](#))

SizeOf a complete mediacmd structure.
Definition at line 7050 of file mediacmd.h.

#define SIZEOF_MEDIACMD_BASE CMD_QUEUE_ELEMSIZE

SizeOf basic mediacmd structure without any clip id.

Definition at line 7046 of file mediacmd.h.

#define SIZEOF_MEDIACMD_CLIPID (CMD_QUEUE_ELEMSIZE + 9)

SizeOf mediacmd structure with a 8 unsigned char clip id and terminating 0.

Definition at line 7048 of file mediacmd.h.

#define SPD_FAST_BUMP 114660L

Max speed for bumping.

Definition at line 71 of file mediacmd.h.

#define SPD_FWD_MAX 5896800

Maximum possible play speed in VVW. See [MEDIACMD::lSpeed](#).

Definition at line 67 of file mediacmd.h.

#define SPD_FWD_PLAY 65520L

Forward play speed (normal) in VVW (65520) see [MEDIACMD::lSpeed](#).

Definition at line 61 of file mediacmd.h.

#define SPD_ILLEGAL 2147483647L

Illegal speed, set [MEDIACMD::lSpeed](#) to this value if not used.

Definition at line 75 of file mediacmd.h.

#define SPD_PAUSE 0L

Pause speed (0%) in VVW (0) see [MEDIACMD::lSpeed](#).

Definition at line 63 of file mediacmd.h.

#define SPD_REV_MAX (-SPD_FWD_MAX)

Minimum possible play speed in VVW. See [MEDIACMD::lSpeed](#).

Definition at line 69 of file mediacmd.h.

#define SPD_REV_PLAY (-SPD_FWD_PLAY)

Reverse play speed (-100%) in VVW (-65520) see [MEDIACMD::lSpeed](#).

Definition at line 65 of file mediacmd.h.

#define SPD_SLOW_BUMP 32760I

Min Sped for bumping.

Definition at line 73 of file mediacmd.h.

#define TC_ILLEGAL 0xFFFFFFFF

Illegal time code reference, set [MEDIACMD::dwPosition](#), [MEDIACMD::dwStart](#), [MEDIACMD::dwEnd](#) to this if not used.

Definition at line 77 of file mediacmd.h.

#define vidChanAll 0xFFFFFFFFUL

Definition at line 449 of file mediacmd.h.

#define VIDEOWRITETYPE_4224 0x00010000

Stills - 4224 individual frames of 8 or 10 bit YCbCr+A.

Definition at line 5587 of file mediacmd.h.

#define VIDEOWRITETYPE_ARRI 0x00100000

ARI - Raw Arri frame format.

Definition at line 5595 of file mediacmd.h.

#define VIDEOWRITETYPE_AVCI_MXF 0x00002000

MXF - Pansonic AVCi - Different P2 plugin.

Definition at line 5581 of file mediacmd.h.

#define VIDEOWRITETYPE_AVI 0x00000001

Video for windows avi (audio video interleave)

Definition at line 5553 of file mediacmd.h.

#define VIDEOWRITETYPE_AVID_MXF 0x00080000

MXF - Avid DNxHD, Uncompressed, JPEG.

Definition at line 5593 of file mediacmd.h.

#define VIDEOWRITETYPE_BMP 0x00000800

Stills - no longer supported.

Definition at line 5577 of file mediacmd.h.

#define VIDEOWRITETYPE_DCP_MXF 0x00800000

MXF - DCP XYZ or RGB JPEG-2000.

Definition at line 5601 of file mediacmd.h.

#define VIDEOWRITETYPE_DNG 0x08000000

DNG - Cinema DNG format.

Definition at line 5609 of file mediacmd.h.

#define VIDEOWRITETYPE_DPX 0x00004000

Stills - DPX (SMPTE/Kodak) 10 bit RGB.

Definition at line 5583 of file mediacmd.h.

#define VIDEOWRITETYPE_FLASH 0x04000000

Flash video (264+mp3) - Note: VVW send YCbCr 8 unc to/from board, compression done in MediaFile/PlugIn.

Definition at line 5607 of file mediacmd.h.

#define VIDEOWRITETYPE_GEN 0x00000008

SoftImage/Avid uncompressed GEN.

Definition at line 5559 of file mediacmd.h.

#define VIDEOWRITETYPE_HDR 0x00000080

Iridas 8 bit RGB format.

HDR+Raw descriptor for raw streams

Definition at line 5569 of file mediacmd.h.

#define VIDEOWRITETYPE_JP2K 0x00200000

Jp2 - Jpeg2000 Still frames.

Definition at line 5597 of file mediacmd.h.

#define VIDEOWRITETYPE_MOV 0x00000002

QuickTime movie (apple)

Definition at line 5555 of file mediacmd.h.

#define VIDEOWRITETYPE_MP4 0x02000000

MPEG program or transport stream - Note: VVW send YCbCr 8 unc to/from board, compression done in MediaFile/PlugIn.

Definition at line 5605 of file mediacmd.h.

#define VIDEOWRITETYPE_MPG 0x00008000

MPEG program or transport stream - Note: VVW send YCbCr 8 unc to/from board, compression done in MediaFile/PlugIn.

Definition at line 5585 of file mediacmd.h.

#define VIDEOWRITETYPE_OP1a_MXF 0x00400000

MXF - Omneon AVCi, DVxx, MPEG.

Definition at line 5599 of file mediacmd.h.

#define VIDEOWRITETYPE_P2_MXF 0x00040000

MXF - Panasonic P2 DV25/50/100, AVCi.

Definition at line 5591 of file mediacmd.h.

#define VIDEOWRITETYPE_RAW 0x00000200

Stills - Raw 24/32 bit RGB/RGBA.

Definition at line 5573 of file mediacmd.h.

#define VIDEOWRITETYPE_SONY_HD_MXF 0x00000010

Jaleo uncompressed format.

MXF - Sony HDCam 4:2:0/4:2:2 MPEG

Definition at line 5563 of file mediacmd.h.

#define VIDEOWRITETYPE_SONY_MXF 0x00020000

MXF - Sony XDCam SD.

Definition at line 5589 of file mediacmd.h.

#define VIDEOWRITETYPE_SONY_SR_MXF 0x00000020

Sony HDCAM SR MXF.

Definition at line 5565 of file mediacmd.h.

#define VIDEOWRITETYPE_TGA 0x00000400

Stills - Targa 24/32 bit RGB.

Definition at line 5575 of file mediacmd.h.

#define VIDEOWRITETYPE_TIFF 0x00001000

Stills - Tiff 24/32 bit RGB/RGBA.

Definition at line 5579 of file mediacmd.h.

#define VIDEOWRITETYPE_TS 0x01000000

MPEG transport stream - Note: VVW send YCbCr 8 unc to/from board, compression done in MediaFile/PlugIn.

Definition at line 5603 of file mediacmd.h.

#define VIDEOWRITETYPE_WMV 0x00000004

Windows Media Video (Microsoft)

Definition at line 5557 of file mediacmd.h.

#define VIDEOWRITETYPE_YUV 0x00000100

Stills - 8/10 bit YCbCr .yuv or .v210.

Definition at line 5571 of file mediacmd.h.

Typedef Documentation

typedef struct [MEDIACMD](#) * [PMEDIACMD](#)

Enumeration Type Documentation

enum [cmdAudChan](#)

Audio channel bit array for [MEDIACMD::dwAudioChannels](#)

Enumerator:

audChan0
audChan1
audChan2
audChan3
audChan4
audChan5
audChan6
audChan7

audChan8
audChan9
audChan10
audChan11
audChan12
audChan13
audChan14
audChan15
audChan16
audChan17
audChan18
audChan19
audChan20
audChan21
audChan22
audChan23
audChan24
audChan25
audChan26
audChan27
audChan28
audChan29
audChan30
audChan31

Definition at line 455 of file mediacmd.h.

enum [cmdFlags](#)

Flags that modify [cmdType](#) in the [MEDIACMD](#) structure. Mostly used to specify which fields in the structure are valid.

Enumerator:

cfDeferred Delay this command until the end of the previous one. This is the method for playing back clips non-linearly. Send one clip to play, then send each clip after it with this flag set and they will play seamlessly back to back.

In the case of `ctInsert`, the deferred indicates that the clip to be inserted will be translated from its current location to the current record directory and then added to the `bin/tcspace`.

cfOverrideDeferred Delay the command, as in [cfDeferred](#), but kill any other waiting commands and use this command as soon as the current command completes.

cfTimeMs Time is in milliseconds. Applies only to the [MEDIACMD::dwCmdAlt](#) member. The millisecond reference is derived from the performance counter (or on extremely old machines `timeGetTime()`) via `vsynceGetCurMs()` which is implemented in `DSync.DLL` for user and kernel modes. The default timing without this flag set is in video frames.

cfTimeTarget Time is set for event occurrence. This means the command will occur when the time specified is reached. If this flag is not set and [cfTimeMs](#) is set, then the time indicates the time the command was received and may be used for a deterministic offset. May be in frames (default) or milliseconds [cfTimeMs](#), requires [cfUseCmdAlt](#).

cfTimeHouseClock Time reference is the system clock (time of day) not the performance clock. This is used to sync network or serial based communication where there is no relationship between performance clocks. For proper operation, the two devices must be genlocked to the same video source, which VVW will interpolate with the correct system clock to keep everything together. Note: This is only as accurate as the genlock readers and LTC or Network time transport connected to BOTH machines. In general, a pair of VVWs are accurate to 1 field, which is ample for editing and broadcast insertion

cfUseSpeed Means the [MEDIACMD::ISpeed](#) member is valid.

cfUsePresets Means the [MEDIACMD::dwVideoChannels](#), [MEDIACMD::dwAudioChannels](#) and [MEDIACMD::dwInfoChannels](#) members are valid.

cfUsePosition Means the [MEDIACMD::dwPosition](#) member is valid.

cfUsePositionOffset Means the [MEDIACMD::dwPosition](#) member is valid and should be used as a long (signed) against the current channel position counter.

cfUseStart Means the [MEDIACMD::dwStart](#) member is valid.

cfUseStartOffset Means the [MEDIACMD::dwStart](#) member is valid and should be used as a long (signed) against the current channel position counter.

cfUseEnd Means the [MEDIACMD::dwEnd](#) member is valid.

cfUseEndOffset Means the [MEDIACMD::dwEnd](#) member is valid and should be used as a long (signed) against the current channel position counter.

cfUseAllIDs Causes the command to act on all IDs in the system. Used for clipspace to delete all IDs quickly.

cfUseClipID Means the [MEDIACMD::arbID](#) member is valid.

cfCopy Copy the media to the current record folder when inserting

cfNoClipFiles Means the command should not be used on any clip or clip spaces

cfConvert Convert the media to the current record folder when inserting

cfNoTCSpaces Means the command should not be used on any clip within or the TCSpace itself

cfUseCmdAlt Means the [MEDIACMD::dwCmdAlt](#) is valid

cfIsShuttle Sent by shuttle/jog/var controllers for drivers that require a special play state that

takes too much time to get into. If this flag is true, the command is a shuttle and true play does not need to be used

cfUsingCurrent If set then elements that are not illegal are current at the reception of the command [cfUsePosition](#) and [cfUsePositionOffset](#) are NOT set and [MEDIACMD::dwPosition](#) is not [TC_ILLEGAL](#), then it is the current position when the command was received [cfUseStart](#) and [cfUseStartOffset](#) are NOT set and [MEDIACMD::dwStart](#) is not [TC_ILLEGAL](#), then it is the current start location when the command was received [cfUseEnd](#) and [cfUseEndOffset](#) are NOT set and [MEDIACMD::dwEnd](#) is not [TC_ILLEGAL](#), then it is the current end time when the command was received [cfUseSpeed](#) is set are NOT set and [MEDIACMD::lSpeed](#) is not [SPD_ILLEGAL](#), then it is the current speed when the command was received [cfUsePresets](#) is NOT set and [MEDIACMD::dwAudioChannels](#), [MEDIACMD::dwVideoChannels](#) and [MEDIACMD::dwInfoChannels](#) are not 0xFFFFFFFF, then they are the current presets when the command was received [cfUseClipID](#) is NOT set and [MEDIACMD::arbID](#)[0] is not equal to NULL (""), then it is the current clip ID when the command was received. If [MEDIACMD::arbID](#)[9] is not equal to NULL ("") then it is the current file name when the command was received.

cfUseFrameCount If set then [MEDIACMD::dwPosition](#), [MEDIACMD::dwStart](#) and [MEDIACMD::dwEnd](#) are absolute (0 based) frame counts, else they are the current type (CTL/CLIP(frame count) or LTC/VITC(time code offset)).

cfFields If set then [MEDIACMD::dwPosition](#), [MEDIACMD::dwStart](#) and [MEDIACMD::dwEnd](#) should be interpreted as fields, not frames, if they are valid

cfRipple Close up any holes created by this command. Most importantly [cmdType::ctDelete](#), [cmdType::ctBlank](#), [cmdType::ctInsert](#) and [cmdType::ctTrim](#).

cfLoop Command should be looped. Mostly used for loop playback where a start and end are specified. The play will begin at the start, proceed to the end, and once reached, loop back to the start again.

cfTrigger INTERNAL - Allows one channel to setup a DSync trigger with another. Use [cmdType::ctTransfer](#) instead as this is very inefficient for non local command transports.

cfPreview This command is part of a preview. Either it notes a channel change (passthrough to emulate an edit) or that the playback does not have to be consistent and frame accurate. Also returned if the channel can only produce preview quality playback (as in VGA playback of HDTV media without hardware assist).

cfRemoteCommand This tells the DDR that the command originated from a remote machine before being accepted from vvwNet. This is the only way to tell if we have full system access. If this flag is set, Windows commands (HANDLES) will be ignored at the avHal Level. This is for all commands not originating from localhost

cfSecondField When returned in a status, it means the second field in time (the later field) is the current one being displayed. When sent, it indicates which field is to be displayed, if only one field is going to be displayed, or which field to start the edit on (edit start is NOT supported in the 3.0 version of VVW).

cfUseNextField When used in pause command it will advance to the next

cfInvert For cmdType::ctTransfer, invert the source and target. Use to allow an external device (such as a VTR) to always master the transfer procedure. Because of the high latency and poor ballistics of VTRs, the internal transfer slaves to it regardless of whether it is the source or target of the transfer.

cfTest Means do not act on this command, but return [GS_NOT_SUPPORTED](#) in dwPosition if you cannot handle it. Used to determine basic capabilities of the channel. For instance, if it's an MPEG-2 playback channel, it can't record but if it has a passthrough, it may be able to stop. Using cfTest with cmdType::ctRecord, cmdType::ctStop will tell the caller this so the interface may be adjusted accordingly.

Caution: This flag has not been tested with all transport types. Avoid for now.

cfNoReturn Instructs the channel that no return is required. The channel then has the option of remembering the command and acting on it within a reasonable time. This means the caller does not know if the command completed successfully at return time, but the status should be monitored anyways to figure that out. Especially when long time functions like a VTR seek will return that the command was successfully initiated, but not wait for the completion of the seek, regardless of this flag.

Definition at line 246 of file mediacmd.h.

enum [cmdGetSetValue](#)

Enum sent in [MEDIACMD::dwCmdAlt](#) for the commands cmdType::ctGetValue, cmdType::ctSetValue, cmdType::ctValueSupported.

ctGetValue will return information in the mediacmd per this document

ctSetValue will change the state of the channel using the members of mediacmd per this document

ctValueSupport will return GS_NOT_SUPPORTED in [MEDIACMD::dwPosition](#) if it is NOT supported. If it is supported, [MEDIACMD::dwPosition](#) will be set to some other value

NOTE:

1652'nada' is Spanish for nothing¹ and is used here to indicate that the command is not supported. Time Code - There are three main time code types, each with their own user bit information. The 0 based absolute time code is referred to by 'Tc' and 'Ub'. The LTC (longitudinal time code or SMPTE time code often sent via audio) is referred to by 'LtcTc' and 'LtcUb'. The VITC (vertical interval time code, usually encoded in the vertical blank area of the video signal) is referred to by 'ViteTc' and 'ViteUb'. Not all devices will support all types or the user bits values for some types. Use value supported to determine support

1653

Enumerator:

gsTc Current internal time - control or clip absolute zero based time code (0..total frames exclusive)

1654cmdType::ctSetValue
1655nada
1656cmdType::ctGetValue
1657[MEDIACMD::dwPosition](#) - current tc
1658[MEDIACMD::dwVideoChannels](#) - Always VITC time code frame value
1659[MEDIACMD::dwStart](#) - Always VITC user bits
1660[MEDIACMD::dwAudioChannels](#) - Always LTC time code frame value
1661[MEDIACMD::dwEnd](#) - Always VITC user bits
1662[MEDIACMD::dwInfoChannels](#) - Always absolute frame number
[MEDIACMD::lSpeed](#) - Always VITC Aux setting (dwVitcAux)

1663

gsUb Current internal user bits

1664cmdType::ctSetValue
1665nada
1666cmdType::ctGetValue
1667[MEDIACMD::dwPosition](#) - current ub
1668[MEDIACMD::dwVideoChannels](#) - Always VITC time code frame value
1669[MEDIACMD::dwStart](#) - Always VITC user bits
1670[MEDIACMD::dwAudioChannels](#) - Always LTC time code frame value
1671[MEDIACMD::dwEnd](#) - Always VITC user bits
1672[MEDIACMD::dwInfoChannels](#) - Always absolute frame number
[MEDIACMD::lSpeed](#) - Always VITC Aux setting (dwVitcAux)

1673

gsLtcTc Current LTC time

1674cmdType::ctSetValue
1675[MEDIACMD::dwPosition](#) - to set VITC generator for next gen (record) if in preset mod
1676cmdType::ctGetValue
1677[MEDIACMD::dwPosition](#) - current Ltc
1678[MEDIACMD::dwVideoChannels](#) - Always VITC time code frame value
1679[MEDIACMD::dwStart](#) - Always VITC user bits
1680[MEDIACMD::dwAudioChannels](#) - Always LTC time code frame value
1681[MEDIACMD::dwEnd](#) - Always VITC user bits
1682[MEDIACMD::dwInfoChannels](#) - Always absolute frame number
[MEDIACMD::lSpeed](#) - Always VITC Aux setting (dwVitcAux)

1683

gsLtcUb Current LTC user bits

1684cmdType::ctSetValue
1685[MEDIACMD::dwPosition](#) - to set LTC generator
1686cmdType::ctGetValue
1687[MEDIACMD::dwPosition](#) - current Lub
1688[MEDIACMD::dwVideoChannels](#) - Always VITC time code frame value
1689[MEDIACMD::dwStart](#) - Always VITC user bits
1690[MEDIACMD::dwAudioChannels](#) - Always LTC time code frame value
1691[MEDIACMD::dwEnd](#) - Always VITC user bits
1692[MEDIACMD::dwInfoChannels](#) - Always absolute frame number

[MEDIACMD::lSpeed](#) - Always VITC Aux setting (dwVitcAux)

1693

gsVitcTc Current VITC time

1694cmdType::ctSetValue

1695[MEDIACMD::dwPosition](#) - to set LTC generator

1696cmdType::ctGetValue

1697[MEDIACMD::dwPosition](#) - current Vitc

1698[MEDIACMD::dwVideoChannels](#) - Always VITC time code frame value

1699[MEDIACMD::dwStart](#) - Always VITC user bits

1700[MEDIACMD::dwAudioChannels](#) - Always LTC time code frame value

1701[MEDIACMD::dwEnd](#) - Always VITC user bits

1702[MEDIACMD::dwInfoChannels](#) - Always absolute frame number

[MEDIACMD::lSpeed](#) - Always VITC Aux setting (dwVitcAux)

1703

gsVitcUb Current VITC user bits

1704cmdType::ctSetValue

1705[MEDIACMD::dwPosition](#) - to set VITC generator

1706cmdType::ctGetValue

1707[MEDIACMD::dwPosition](#) - current Viub

1708[MEDIACMD::dwVideoChannels](#) - Always VITC time code frame value

1709[MEDIACMD::dwStart](#) - Always VITC user bits

1710[MEDIACMD::dwAudioChannels](#) - Always LTC time code frame value

1711[MEDIACMD::dwEnd](#) - Always VITC user bits

1712[MEDIACMD::dwInfoChannels](#) - Always absolute frame number

[MEDIACMD::lSpeed](#) - Always VITC Aux setting (dwVitcAux)

1713

gsTcSource Current time code source

1714cmdType::ctSetValue

1715[MEDIACMD::dwPosition](#) - [GS_TCSOURCE_LTC](#), [GS_TCSOURCE_VITC](#),
[GS_TCSOURCE_CTL](#), [GS_TCSOURCE_CLIP](#), [GS_TCSOURCE_IRIG](#)

1716cmdType::ctGetValue

1717[MEDIACMD::dwPosition](#) - [GS_TCSOURCE_LTC](#), [GS_TCSOURCE_VITC](#),
[GS_TCSOURCE_CTL](#), [GS_TCSOURCE_CLIP](#), [GS_TCSOURCE_IRIG](#)

[MEDIACMD::dwStart](#) - supported types using bit array of above

1718

gsTcType Current time code type

1719cmdType::ctSetValue

1720[MEDIACMD::dwPosition](#) - [TC2_TCTYPE_FILM](#), [TC2_TCTYPE_NDF](#), [TC2_TCTYPE_DF](#),
[TC2_TCTYPE_PAL](#), [TC2_TCTYPE_50](#), [TC2_TCTYPE_5994](#), [TC2_TCTYPE_60](#),
[TC2_TCTYPE_NTSCFILM](#), [TC2_TCTYPE_2398](#), [TC2_TCTYPE_100](#)

1721cmdType::ctGetValue

1722[MEDIACMD::dwPosition](#) - [TC2_TCTYPE_FILM](#), [TC2_TCTYPE_NDF](#), [TC2_TCTYPE_DF](#),
[TC2_TCTYPE_PAL](#), [TC2_TCTYPE_50](#), [TC2_TCTYPE_5994](#), [TC2_TCTYPE_60](#),
[TC2_TCTYPE_NTSCFILM](#), [TC2_TCTYPE_2398](#), [TC2_TCTYPE_100](#)

[MEDIACMD::dwStart](#) - supported types using bit array of above

1723

gsStart Lowest possible time code frame

1724cmdType::ctSetValue
 1725[MEDIACMD::dwPosition](#) - New minimum value
 1726cmdType::ctGetValue
 1727[MEDIACMD::dwPosition](#) - Current minimum value
[MEDIACMD::dwStart](#) - Absolute minimum possible value (usually 0)

 1728
 gsEnd Highest possible time code frame plus 1 (out is never included)

1729cmdType::ctSetValue
 1730[MEDIACMD::dwPosition](#) - New maximum value + 1
 1731cmdType::ctGetValue
 1732[MEDIACMD::dwPosition](#) - Current maximum value + 1
[MEDIACMD::dwStart](#) - Absolute maximum possible value (usually clip end + 1)

 1733
 gsIn Current mark in time set by any caller

1734cmdType::ctSetValue
 1735[MEDIACMD::dwPosition](#) - New in time
 1736cmdType::ctGetValue
[MEDIACMD::dwPosition](#) - Current in time

 1737
 gsLastIn Previous (currently non active) mark in time set by RS-422 protocol

1738cmdType::ctSetValue
 1739- not supported (internal)
 1740cmdType::ctGetValue
[MEDIACMD::dwPosition](#) - last known in time

 1741
 gsOut Current mark out time set by any caller

1742cmdType::ctSetValue
 1743[MEDIACMD::dwPosition](#) - New out time
 1744cmdType::ctGetValue
[MEDIACMD::dwPosition](#) - Current out time

 1745
 gsLastOut Previous (currently non active) mark out time set by RS-422 protocol

1746cmdType::ctSetValue
 1747- not supported (internal)
 1748cmdType::ctGetValue
[MEDIACMD::dwPosition](#) - last known out time

 1749
 gsEditOn Number of frames from Edit On command to start of Record (usually 4~7)

1750cmdType::ctSetValue
 1751[MEDIACMD::dwPosition](#) - New number of frames
 1752cmdType::ctGetValue
[MEDIACMD::dwPosition](#) - Current number of frames

1753

gsEditOff Number of frames from Edit Off command to end of Record (usually 4~7) should match [gsEditOn](#) in most cases

1754cmdType::ctSetValue

1755[MEDIACMD::dwPosition](#) - New number of frames

1756cmdType::ctGetValue

[MEDIACMD::dwPosition](#) - Current number of frames

1757

gsPreroll Number of frames to preroll before in point for an edit

1758cmdType::ctSetValue

1759[MEDIACMD::dwPosition](#) - New number of frames

1760cmdType::ctGetValue

[MEDIACMD::dwPosition](#) - Current number of frames

1761

gsPostroll Number of frames to postroll after an out point for an edit

1762cmdType::ctSetValue

1763[MEDIACMD::dwPosition](#) - New number of frames

1764cmdType::ctGetValue

[MEDIACMD::dwPosition](#) - Current number of frames

1765

gsAutoMode Switch from normal mode to auto mode. For Sony VTR emulation it sets up Pioneer dual head emulation. For Louth and Odetics, enables preview play look ahead for seamless clip playback

1766cmdType::ctSetValue

1767[MEDIACMD::dwPosition](#) - [GS_TRUE](#) or [GS_FALSE](#)

1768cmdType::ctGetValue

[MEDIACMD::dwPosition](#) - Current auto mode state as above

1769

gsPlayDelay Number of frames from receiving Play command to actual Play

1770cmdType::ctSetValue

1771[MEDIACMD::dwPosition](#) - New number of frames

1772cmdType::ctGetValue

[MEDIACMD::dwPosition](#) - Current number of frames

1773

gsLtcTcPreset LTC time code preset (generator preset)

1774cmdType::ctSetValue

1775[MEDIACMD::dwPosition](#) - to set generator for the next record. Will be used not in regen mode.

1776cmdType::ctGetValue

[MEDIACMD::dwPosition](#) - returns the current generator preset.

1777

gsLtcUbPreset LTC user bit preset (generator preset)

1778cmdType::ctSetValue
1779[MEDIACMD::dwPosition](#) - to set generator for the next record. Will be used not in regen mode.
1780cmdType::ctGetValue
[MEDIACMD::dwPosition](#) - returns the current generator preset.

1781
gsVtCtPreset VITC time code preset (generator preset)

1782cmdType::ctSetValue
1783[MEDIACMD::dwPosition](#) - to set generator for the next record. Will be used not in regen mode.
1784cmdType::ctGetValue
[MEDIACMD::dwPosition](#) - returns the current generator preset.

1785
gsVtCtUpPreset VITC time code preset (generator preset)

1786cmdType::ctSetValue
1787[MEDIACMD::dwPosition](#) - to set generator for the next record. Will be used not in regen mode.
1788cmdType::ctGetValue
[MEDIACMD::dwPosition](#) - returns the current generator preset.

1789
gsFrameData Returns the block of data for a frame

1790cmdType::ctSetValue
1791Not really used at this point
1792cmdType::ctGetValue
1793[MEDIACMD::arbID](#) - The data block
1794[MEDIACMD::dwEnd](#) - The size of the data block (0..n)
[MEDIACMD::dwPosition](#) - The expected type of data

1795
gsKeyCode Current Key Code

1796cmdType::ctSetValue
1797[MEDIACMD::dwStart](#) - Key Code prefix (4 unsigned chars)
1798[MEDIACMD::dwPosition](#) - Key Code (4 unsigned chars)
1799cmdType::ctGetValue
1800[MEDIACMD::dwStart](#) - Key Code prefix (4 unsigned chars)
[MEDIACMD::dwPosition](#) - Key Code (4 unsigned chars)

1801
gsInkCode Current Ink Code

1802cmdType::ctSetValue
1803[MEDIACMD::dwStart](#) - Ink Code prefix (4 unsigned chars)
1804[MEDIACMD::dwPosition](#) - Ink Code (3 unsigned chars)
1805cmdType::ctGetValue
1806[MEDIACMD::dwStart](#) - Ink Code prefix (4 unsigned chars)
[MEDIACMD::dwPosition](#) - Ink Code (3 unsigned chars)

1807
gs215Code Current 215 Code Code

1808cmdType::ctSetValue

1809 [MEDIACMD::dwStart](#) - Audio Phase [msb], Audio Modulus, pull down, sequence [lsb]

1810 [MEDIACMD::dwPosition](#) - Absolute Frame

1811 cmdType::ctGetValue

1812 [MEDIACMD::dwStart](#) - Audio Phase [msb], Audio Modulus, pull down, sequence [lsb]

[MEDIACMD::dwPosition](#) - Absolute Frame

1813

gsHeadsAndTails Set the heads and tails to be used for the next record. Automatically set to zero after record/add

1814 cmdType::ctSetValue

1815 [MEDIACMD::dwPosition](#) - Heads and Tails size in frames

1816 cmdType::ctGetValue - not supported

1817

gsTimecodeSources Set timecode sources limitations

1818 cmdType::ctSetValue

1819 [MEDIACMD::dwPosition](#) - [GS_TCSRC_DISABLE_EXTERNAL](#),

[GS_TCSRC_FORCE_VTR_TC](#), [GS_TCSRC_USE_TIMEOFDAY](#)

1820 cmdType::ctGetValue - not supported

1821

gsCCSetup Get/Set closed captioning

1822 cmdType::ctSetValue

1823 [MEDIACMD::dwPosition](#) - Set as primary: [GS_CC_DISABLE](#) [GS_CC_CC1](#) [GS_CC_CC2](#) [GS_CC_CC3](#) [GS_CC_CC4](#) [GS_CC_TEXT1](#) [GS_CC_TEXT2](#) [GS_CC_TEXT3](#) [GS_CC_TEXT4](#) [GS_CC_XDS](#) [GS_CC_708](#)

1824 [MEDIACMD::dwStart](#) - Bitwise 'use these if available': [GS_CC_DISABLE](#) [GS_CC_CC1](#) [GS_CC_CC2](#) [GS_CC_CC3](#) [GS_CC_CC4](#) [GS_CC_TEXT1](#) [GS_CC_TEXT2](#) [GS_CC_TEXT3](#) [GS_CC_TEXT4](#) [GS_CC_XDS](#) [GS_CC_708](#)

1825 [MEDIACMD::dwEnd](#) - If 1, display captions on VGA/Waveform Vector

1826 cmdType::ctGetValue - not supported

1827 [MEDIACMD::dwPosition](#) - Current primary: [GS_CC_DISABLE](#) [GS_CC_CC1](#) [GS_CC_CC2](#) [GS_CC_CC3](#) [GS_CC_CC4](#) [GS_CC_TEXT1](#) [GS_CC_TEXT2](#) [GS_CC_TEXT3](#) [GS_CC_TEXT4](#) [GS_CC_XDS](#) [GS_CC_708](#)

1828 [MEDIACMD::dwStart](#) - Bitwise available types: [GS_CC_DISABLE](#) [GS_CC_CC1](#) [GS_CC_CC2](#) [GS_CC_CC3](#) [GS_CC_CC4](#) [GS_CC_TEXT1](#) [GS_CC_TEXT2](#) [GS_CC_TEXT3](#) [GS_CC_TEXT4](#) [GS_CC_XDS](#) [GS_CC_708](#)

1829 [MEDIACMD::dwEnd](#) - If 1, displaying captions on VGA/Waveform Vector

[MEDIACMD::arbID](#) - String of last know characters

1830

gsDisableTimecode Enable or disable all time code, or force time of day

1831 cmdType::ctSetValue

1832 [MEDIACMD::dwPosition](#) - [GS_TCSRC_DISABLE_EXTERNAL](#),

[GS_TCSRC_FORCE_VTR_TC](#), [GS_TCSRC_USE_TIMEOFDAY](#)

1833 cmdType::ctGetValue - not supported

1834

gsVITCSourcesPrecedence Set the order of precedence for time code sources for the

dwVtcFrame/Ub in fiInfo

1835cmdType::ctSetValue

1836MEDIACMD::dwPosition - Most favoured type: [GS_SOURCEPRECEDENCE_RP188_V](#),
[GS_SOURCEPRECEDENCE_RP188_L](#), [GS_SOURCEPRECEDENCE_SMPTE](#),
[GS_SOURCEPRECEDENCE_TOD](#), [GS_SOURCEPRECEDENCE_VITC](#),
[GS_SOURCEPRECEDENCE_IRIG](#), [GS_SOURCEPRECEDENCE_RP215](#),
[GS_SOURCEPRECEDENCE_FRAMECOUNT](#)

1837MEDIACMD::dwStart - 2nd most favoured type: see MEDIACMD::dwPostion for possible types

1838MEDIACMD::dwEnd - 3nd most favoured type: see MEDIACMD::dwPostion for possible types

1839MEDIACMD::dwVideoChannels - 4th most favoured type: see MEDIACMD::dwPostion for possible types

1840MEDIACMD::dwAudioChannels - 5th most favoured type: see MEDIACMD::dwPostion for possible types

1841MEDIACMD::dwInfoChannels - 6th most favoured type: see MEDIACMD::dwPostion for possible types

1842MEDIACMD::lSpeed - 7th most favoured type: see MEDIACMD::dwPostion for possible types

1843cmdType::ctGetValue - Returns the current order of precedence

1844MEDIACMD::dwPosition - Most favoured type: [GS_SOURCEPRECEDENCE_RP188_V](#),
[GS_SOURCEPRECEDENCE_RP188_L](#), [GS_SOURCEPRECEDENCE_SMPTE](#),
[GS_SOURCEPRECEDENCE_TOD](#), [GS_SOURCEPRECEDENCE_VITC](#),
[GS_SOURCEPRECEDENCE_IRIG](#), [GS_SOURCEPRECEDENCE_RP215](#),
[GS_SOURCEPRECEDENCE_FRAMECOUNT](#)

1845MEDIACMD::dwStart - 2nd most favoured type: see MEDIACMD::dwPostion for possible types

1846MEDIACMD::dwEnd - 3nd most favoured type: see MEDIACMD::dwPostion for possible types

1847MEDIACMD::dwVideoChannels - 4th most favoured type: see MEDIACMD::dwPostion for possible types

1848MEDIACMD::dwAudioChannels - 5th most favoured type: see MEDIACMD::dwPostion for possible types

1849MEDIACMD::dwInfoChannels - 6th most favoured type: see MEDIACMD::dwPostion for possible types

[MEDIACMD::lSpeed](#) - 7th most favoured type: see MEDIACMD::dwPostion for possible types

1850

gsLTCSourcePrecedence Set the order of precedence for time code sources for the dwLtcFrame/Ub in fiInfo

1851cmdType::ctSetValue

1852MEDIACMD::dwPosition - Most favoured type: [GS_SOURCEPRECEDENCE_RP188_V](#),
[GS_SOURCEPRECEDENCE_RP188_L](#), [GS_SOURCEPRECEDENCE_SMPTE](#),
[GS_SOURCEPRECEDENCE_TOD](#), [GS_SOURCEPRECEDENCE_VITC](#),
[GS_SOURCEPRECEDENCE_IRIG](#), [GS_SOURCEPRECEDENCE_RP215](#),
[GS_SOURCEPRECEDENCE_FRAMECOUNT](#)

1853MEDIACMD::dwStart - 2nd most favoured type: see MEDIACMD::dwPostion for possible types

1854MEDIACMD::dwEnd - 3nd most favoured type: see MEDIACMD::dwPostion for possible types

1855MEDIACMD::dwVideoChannels - 4th most favoured type: see MEDIACMD::dwPostion for possible types

1856MEDIACMD::dwAudioChannels - 5th most favoured type: see MEDIACMD::dwPostion for possible types

1857MEDIACMD::dwInfoChannels - 6th most favoured type: see MEDIACMD::dwPostion for possible types

1858MEDIACMD::lSpeed - 7th most favoured type: see MEDIACMD::dwPostion for possible types

1859cmdType::ctGetValue - Returns the current order of precedence

1860MEDIACMD::dwPosition - Most favoured type: [GS_SOURCEPRECEDENCE_RP188_V](#),
[GS_SOURCEPRECEDENCE_RP188_L](#), [GS_SOURCEPRECEDENCE_SMPTE](#),

[GS_SOURCEPRECEDENCE_TOD](#), [GS_SOURCEPRECEDENCE_VITC](#),
[GS_SOURCEPRECEDENCE_IRIG](#), [GS_SOURCEPRECEDENCE_RP215](#),
[GS_SOURCEPRECEDENCE_FRAMECOUNT](#)

1861 [MEDIACMD::dwStart](#) - 2nd most favoured type: see [MEDIACMD::dwPosition](#) for possible types

1862 [MEDIACMD::dwEnd](#) - 3rd most favoured type: see [MEDIACMD::dwPosition](#) for possible types

1863 [MEDIACMD::dwVideoChannels](#) - 4th most favoured type: see [MEDIACMD::dwPosition](#) for possible types

1864 [MEDIACMD::dwAudioChannels](#) - 5th most favoured type: see [MEDIACMD::dwPosition](#) for possible types

1865 [MEDIACMD::dwInfoChannels](#) - 6th most favoured type: see [MEDIACMD::dwPosition](#) for possible types

[MEDIACMD::!Speed](#) - 7th most favoured type: see [MEDIACMD::dwPosition](#) for possible types

1866

gsGetNextClip Get the next clip value. In clip mode, this returns a series of clips. To get the first clip, pass a NULL or all spaces string as the 8 character clip. To get each subsequent clip, send back the 8 character clip returned previously. The full clip name will be in the [MEDIACMD::arbID](#) starting at position [9] (right after the 8 character clip name)

1867cmdType::ctSetValue

1868- not supported

1869cmdType::ctGetValue - return the next clip name

1870 [MEDIACMD::dwPosition](#) - not != [GS_NOT_SUPPORTED](#) if clip is value, else end of list

1871 [MEDIACMD::dwStart](#) - starting frame of clip

1872 [MEDIACMD::dwEnd](#) - ending frame of clip (exclusive)

[MEDIACMD::arbID](#) - 8 char clip name, terminating 0, long name (starting at 9), terminating 0

1873

gsFirstClip Obsolete - use [gsGetNextClip](#)

1874cmdType::ctSetValue

1875- not supported in new drivers

1876cmdType::ctGetValue

- not supported in new drivers

1877

gsNextClip Obsolete - use [gsGetNextClip](#)

1878cmdType::ctSetValue

1879- not supported in new drivers

1880cmdType::ctGetValue

- not supported in new drivers

1881

gsTCSGetTLClipState Return the next state info when working through a time code space time line to retrieve all the edits in order. The state uses [MEDIACMD::dwPosition](#), [MEDIACMD::dwVideoChannels](#), [MEDIACMD::dwAudioChannels](#), [MEDIACMD::dwInfoChannels](#) to maintain the state (Please note that [MEDIACMD::arbID](#) is reserved and must be maintained between calls). The [dwPosition](#) describes the current position in the timeline and the channel bits are set for channels already returned. See [gsTCSGetTLNextClip](#) for more info

1882cmdType::ctSetValue

1883- not supported
1884cmdType::ctGetValue
1885[MEDIACMD::dwPosition](#) - Current time line position
1886[MEDIACMD::dwVideoChannels](#) - Video channels used so far
1887[MEDIACMD::dwAudioChannels](#) - Audio channels used so far
[MEDIACMD::dwInfoChannels](#) - Info channels used so far

1888

gsTCSGetTLClipInfo CALL Pos Start End V A I arbID gsTCSGetTLClipInfo 0 x x 0 0
0 x - Restart at 0 Rtn 0 0 300 1 2 0 file1 - 10 sec VA2 from file1 gsTCSGetTLNextState
0 0 0 0 - First state 0 Rtn 0 1 2 0 - First clip channels (Copy prev gsTCSGetTLNextState
into gsTCSGetTLClipInfo before sending) gsTCSGetTLClipInfo 0 1 2 0 - Last get state
Rtn 0 0 150 0 1 0 file2 - 5 sec A1 from file2 (Use last gsTCSGetTLNextState for this
call) gsTCSGetTLNextState 0 1 2 0 - Use last state to get next Rtn 0 1 3 0 - Channels
used so far (Copy prev gsTCSGetTLNextState into gsTCSGetTLClipInfo before sending
) gsTCSGetTLClipInfo 0 1 3 0 - Last get state Rtn 150 150 210 0 1 0 file3 - 2 sec A1
from file3 (Use last gsTCSGetTLNextState for this call) gsTCSGetTLNextState 0 1 3 0
- Use last state to get next Rtn 150 0 1 0 - Channels used so far Take the [MEDIACMD](#)
struct returned from gsTCSGetTLClipState and find the next active clip. For the first clip
in time line, send all zeroes. Other then the first call, all calls should include the
position/channel bits from the previous gsTCSGetTLNextState call (other than the first
call) and gsTCSGetTLNextState should be called immediately before
gsTCSGetTLClipInfo.

1889cmdType::ctSetValue

1890- not supported

1891cmdType::ctGetValue

1892[MEDIACMD::arbID](#) - Clip ID

1893for MCMD2 -out- [MEDIACMD::arbID](#) - Next 8 character ID and unc file path separated by
NULL or 8 NULLs if clip list complete

1894[MEDIACMD::cfFlags](#) - Set cfUsePosition|cfUseStart|cfUseEnd to search next clip, set
cfUsePosition & [MEDIACMD::dwPosition](#) for info at specified position

1895[MEDIACMD::dwPosition](#) - Reference time code for time line

1896[MEDIACMD::dwStart](#) - First frame of clip

1897[MEDIACMD::dwEnd](#) - Last frame of clip

1898[MEDIACMD::dwVideoChannels](#) - Channels this clip exists in for the dwStart/dwEnd range

1899[MEDIACMD::dwAudioChannels](#) - Channels this clip exists in for the dwStart/dwEnd range

[MEDIACMD::dwInfoChannels](#) - Channels this clip exists in for the dwStart/dwEnd range

1900

gsClipInfo Get or change the information on a clip (currently for clip space only)

1901cmdType::ctSetValue

1902- not supported

1903cmdType::ctGetValue

1904[MEDIACMD::arbID](#) - Last returned clip ID or 8 NULLs for first clip

1905for MCMD2 -out- [MEDIACMD::arbID](#) - Next 8 character ID and unc file path separated by
NULL or 8 NULLs if clip list complete

1906[MEDIACMD::dwPosition](#) - Starting timecode if known, else First frame of clip

1907[MEDIACMD::dwStart](#) - First frame of clip

1908[MEDIACMD::dwEnd](#) - Last frame of clip

1909[MEDIACMD::dwVideoChannels](#) - Channels this clip exists in for the dwStart/dwEnd range

1910[MEDIACMD::dwAudioChannels](#) - Channels this clip exists in for the dwStart/dwEnd range

[MEDIACMD::dwInfoChannels](#) - Channels this clip exists in for the dwStart/dwEnd range

1911

gsClipCopy Create a virtual copy of a clip from a current clip. Must change at least name to succeed. To affect the source clip, use [gsClipInfo](#)

1912cmdType::ctSetValue

1913Requires [MEDIACMD::cfFlags](#) set to affect stored clip info for each member

1914[MEDIACMD::arbID](#) - Source ClipID [8 unsigned chars], NULL, New ClipID [8 unsigned chars]
- min size 17 unsigned chars.

1915[MEDIACMD::dwStart](#) - First frame of new clip (referenced from source clip)

1916[MEDIACMD::dwEnd](#) - Last frame of new clip (referenced from source clip)

1917[MEDIACMD::dwVideoChannels](#) - Channels this clip exists in for the dwStart/dwEnd range

1918[MEDIACMD::dwAudioChannels](#) - Channels this clip exists in for the dwStart/dwEnd range

1919[MEDIACMD::dwInfoChannels](#) - Channels this clip exists in for the dwStart/dwEnd range

1920cmdType::ctGetValue

- no supported

1921

gsTcClipInfo Get set timecode edl comment info, later other things as well such as effects (cut wipe fade)

1922cmdType::ctSetValue

1923Requires [MEDIACMD::cfFlags](#) set to affect stored clip info for each member

1924[MEDIACMD::arbID](#) - String to set

1925[MEDIACMD::dwPosition](#) - Location on edl

1926[MEDIACMD::dwStart](#) - type to set (Comment / Effect)

1927cmdType::ctGetValue

1928[MEDIACMD::arbID](#) - Comment

1929[MEDIACMD::dwPosition](#) - Location on edl

[MEDIACMD::dwStart](#) - type set (Comment / Effect)

1930

gsAudChan Returns the available audio channels (read only)

1931cmdType::ctSetValue

1932- not supported

1933cmdType::ctGetValue

[MEDIACMD::dwPosition](#) - Available channels

1934

gsVidChan Returns the available video channels (read only)

1935cmdType::ctSetValue

1936- not supported

1937cmdType::ctGetValue

[MEDIACMD::dwPosition](#) - Available channels

1938

gsInfChan Returns the available information channels (read only)

1939cmdType::ctSetValue

1940- not supported

1941cmdType::ctGetValue

[MEDIACMD::dwPosition](#) - Available channels

1942

gsAudSelect Return or set the selected audio channels

1943cmdType::ctSetValue

1944[MEDIACMD::dwPosition](#) - New channel selection

1945cmdType::ctGetValue

1946[MEDIACMD::dwPosition](#) - Currently selected channels

[MEDIACMD::dwStart](#) - Available channels for selection

1947

gsVidSelect Return or set the selected video channels

1948cmdType::ctSetValue

1949[MEDIACMD::dwPosition](#) - New channel selection

1950cmdType::ctGetValue

1951[MEDIACMD::dwPosition](#) - Currently selected channels

[MEDIACMD::dwStart](#) - Available channels for selection

1952

gsInfSelect Return or set the selected information channels

1953cmdType::ctSetValue

1954[MEDIACMD::dwPosition](#) - New channel selection

1955cmdType::ctGetValue

1956[MEDIACMD::dwPosition](#) - Currently selected channels

[MEDIACMD::dwStart](#) - Available channels for selection

1957

gsAudEdit Return or set the audio channels for the next edit

1958cmdType::ctSetValue

1959[MEDIACMD::dwPosition](#) - New channel edit selection

1960cmdType::ctGetValue

1961[MEDIACMD::dwPosition](#) - Currently selected edit channels

[MEDIACMD::dwStart](#) - Available channels for edit

1962

gsVidEdit Return or set the video channels for the next edit

1963cmdType::ctSetValue

1964[MEDIACMD::dwPosition](#) - New channel edit selection

1965cmdType::ctGetValue

1966[MEDIACMD::dwPosition](#) - Currently selected edit channels

[MEDIACMD::dwStart](#) - Available channels for edit

1967

gsInfEdit Return or set the information channels for the next edit

1968cmdType::ctSetValue

1969[MEDIACMD::dwPosition](#) - New channel edit selection

1970cmdType::ctGetValue

1971[MEDIACMD::dwPosition](#) - Currently selected edit channels

[MEDIACMD::dwStart](#) - Available channels for edit

1972

gsEditMode Return or set the information channels for the next edit

1973cmdType::ctSetValue
1974[MEDIACMD::dwPosition](#) - New channel edit selection
1975cmdType::ctGetValue
1976[MEDIACMD::dwPosition](#) - Currently selected edit channels
[MEDIACMD::dwStart](#) - Available channels for edit

1977

gsMetaData Access one metadata element for the current media.

See the enum [vvwInfoMetaTypes](#) in vvwTypes.h

1978cmdType::ctSetValue

1979[MEDIACMD::dwPosition](#) - ID = #vwvInfoMetaTypes::vwvwiFileName ..

#vwvInfoMetaTypes::vwvwiGamma1000

1980[MEDIACMD::dwStart](#) - value for #vwvInfoMetaTypes::vwvwiTimeCode ..

#vwvInfoMetaTypes::vwvwiGamma1000

1981[MEDIACMD::dwEnd](#) - GS_TRUE if the element exists, GS_NOT_SUPPORTED if not

1982[MEDIACMD::arbID](#) - value for #vwvInfoMetaTypes::vwvwiFileName ..

#vwvInfoMetaTypes::vwvwiFrameAttribute

1983cmdType::ctGetValue

1984[MEDIACMD::dwPosition](#) - ID = #vwvInfoMetaTypes::vwvwiFileName ..

#vwvInfoMetaTypes::vwvwiGamma1000

1985[MEDIACMD::dwStart](#) - value for #vwvInfoMetaTypes::vwvwiTimeCode ..

#vwvInfoMetaTypes::vwvwiGamma1000

1986[MEDIACMD::dwEnd](#) - GS_TRUE if the element exists, GS_NOT_SUPPORTED if not

[MEDIACMD::arbID](#) - value for #vwvInfoMetaTypes::vwvwiFileName ..

#vwvInfoMetaTypes::vwvwiFrameAttribute

1987

gsMetaDataDirectory Access one metadata element for the directory of the current media (./Default.xml).

See the enum [vvwInfoMetaTypes](#) in vvwTypes.h

1988cmdType::ctSetValue

1989[MEDIACMD::dwPosition](#) - ID = #vwvInfoMetaTypes::vwvwiFileName ..

#vwvInfoMetaTypes::vwvwiGamma1000

1990[MEDIACMD::dwStart](#) - value for #vwvInfoMetaTypes::vwvwiTimeCode ..

#vwvInfoMetaTypes::vwvwiGamma1000

1991[MEDIACMD::arbID](#) - value for #vwvInfoMetaTypes::vwvwiFileName ..

#vwvInfoMetaTypes::vwvwiFrameAttribute

1992cmdType::ctGetValue

1993[MEDIACMD::dwPosition](#) - ID = #vwvInfoMetaTypes::vwvwiFileName ..

#vwvInfoMetaTypes::vwvwiGamma1000

1994[MEDIACMD::dwStart](#) - value for #vwvInfoMetaTypes::vwvwiTimeCode ..

#vwvInfoMetaTypes::vwvwiGamma1000

[MEDIACMD::arbID](#) - value for #vwvInfoMetaTypes::vwvwiFileName ..

#vwvInfoMetaTypes::vwvwiFrameAttribute

1995

gsMetaDataVolume Access one metadata element for the drive/volume of the current media (/Default.xml).

See the enum [vvwInfoMetaTypes](#) in vvwTypes.h

1996cmdType::ctSetValue
1997[MEDIACMD::dwPosition](#) - ID = #vwwInfoMetaTypes::vwwiFileName ..
#vwwInfoMetaTypes::vwwiGamma1000
1998[MEDIACMD::dwStart](#) - value for #vwwInfoMetaTypes::vwwiTimeCode ..
#vwwInfoMetaTypes::vwwiGamma1000
1999[MEDIACMD::arbID](#) - value for #vwwInfoMetaTypes::vwwiFileName ..
#vwwInfoMetaTypes::vwwiFrameAttribute
2000cmdType::ctGetValue
2001[MEDIACMD::dwPosition](#) - ID = #vwwInfoMetaTypes::vwwiFileName ..
#vwwInfoMetaTypes::vwwiGamma1000
2002[MEDIACMD::dwStart](#) - value for #vwwInfoMetaTypes::vwwiTimeCode ..
#vwwInfoMetaTypes::vwwiGamma1000
[MEDIACMD::arbID](#) - value for #vwwInfoMetaTypes::vwwiFileName ..
#vwwInfoMetaTypes::vwwiFrameAttribute

2003

gsMetaDataCurrentUser Access one metadata element for the default metadata of the current user (HKEY_CURRENT_USER windows, /home/user/default.xml unix).

See the enum [vwwInfoMetaTypes](#) in vwwTypes.h

2004cmdType::ctSetValue
2005[MEDIACMD::dwPosition](#) - ID = #vwwInfoMetaTypes::vwwiFileName ..
#vwwInfoMetaTypes::vwwiGamma1000
2006[MEDIACMD::dwStart](#) - value for #vwwInfoMetaTypes::vwwiTimeCode ..
#vwwInfoMetaTypes::vwwiGamma1000
2007[MEDIACMD::arbID](#) - value for #vwwInfoMetaTypes::vwwiFileName ..
#vwwInfoMetaTypes::vwwiFrameAttribute
2008cmdType::ctGetValue
2009[MEDIACMD::dwPosition](#) - ID = #vwwInfoMetaTypes::vwwiFileName ..
#vwwInfoMetaTypes::vwwiGamma1000
2010[MEDIACMD::dwStart](#) - value for #vwwInfoMetaTypes::vwwiTimeCode ..
#vwwInfoMetaTypes::vwwiGamma1000
[MEDIACMD::arbID](#) - value for #vwwInfoMetaTypes::vwwiFileName ..
#vwwInfoMetaTypes::vwwiFrameAttribute

2011

gsMetaDataLocalMachine Access one metadata element for the default metadata of the current user (HKEY_LOCAL_MACHINE windows, /var/metadata/default.xml unix).

See the enum [vwwInfoMetaTypes](#) in vwwTypes.h

2012cmdType::ctSetValue
2013[MEDIACMD::dwPosition](#) - ID = #vwwInfoMetaTypes::vwwiFileName ..
#vwwInfoMetaTypes::vwwiGamma1000
2014[MEDIACMD::dwStart](#) - value for #vwwInfoMetaTypes::vwwiTimeCode ..
#vwwInfoMetaTypes::vwwiGamma1000
2015[MEDIACMD::arbID](#) - value for #vwwInfoMetaTypes::vwwiFileName ..
#vwwInfoMetaTypes::vwwiFrameAttribute
2016cmdType::ctGetValue
2017[MEDIACMD::dwPosition](#) - ID = #vwwInfoMetaTypes::vwwiFileName ..
#vwwInfoMetaTypes::vwwiGamma1000
2018[MEDIACMD::dwStart](#) - value for #vwwInfoMetaTypes::vwwiTimeCode ..
#vwwInfoMetaTypes::vwwiGamma1000
[MEDIACMD::arbID](#) - value for #vwwInfoMetaTypes::vwwiFileName ..
#vwwInfoMetaTypes::vwwiFrameAttribute

2019

gsMetaDataGlobal Access one meta data element for the default metadata for the facility (Requires group or facility media proxy and database).

See the enum [vvwInfoMetaTypes](#) in `vwTypes.h`

2020cmdType::ctSetValue

2021[MEDIACMD::dwPosition](#) - ID = #vwInfoMetaTypes::vwFileName ..
#vwInfoMetaTypes::vwGamma1000

2022[MEDIACMD::dwStart](#) - value for #vwInfoMetaTypes::vwTimeCode ..
#vwInfoMetaTypes::vwGamma1000

2023[MEDIACMD::arbID](#) - value for #vwInfoMetaTypes::vwFileName ..
#vwInfoMetaTypes::vwFrameAttribute

2024cmdType::ctGetValue

2025[MEDIACMD::dwPosition](#) - ID = #vwInfoMetaTypes::vwFileName ..
#vwInfoMetaTypes::vwGamma1000

2026[MEDIACMD::dwStart](#) - value for #vwInfoMetaTypes::vwTimeCode ..
#vwInfoMetaTypes::vwGamma1000

[MEDIACMD::arbID](#) - value for #vwInfoMetaTypes::vwFileName ..
#vwInfoMetaTypes::vwFrameAttribute

2027

gsMetaDataReadWrite Write or read the current metadata structure from the XML on the disk

2028cmdType::ctSetValue

2029[MEDIACMD::dwPosition](#) - to 0 = media file, 1 = directory, 2 = volume, 3 = current user, 4 = current machine, 5 = master server

2030cmdType::ctGetValue

[MEDIACMD::dwPosition](#) - from 0 = media file, 1 = directory, 2 = volume, 3 = current user, 4 = current machine, 5 = master server

2031

gsMetaBaseOpen Open a metadata database file

2032cmdType::ctSetValue

2033[MEDIACMD::arbID](#) - DataBase File Name (IN)

2034[MEDIACMD::dwPosition](#) - Success/Failure (OUT) NOTE: return Success/Failure (-1 if it doesn't exist)

gsMetaBaseCreate Create a new metadata database file

2035cmdType::ctSetValue

2036[MEDIACMD::arbID](#) - DataBase File Name (IN)

2037[MEDIACMD::dwPosition](#) - Success/Failure (OUT)

gsMetaBaseClose

2038cmdType::ctSetValue Close previously opened database

gsMetaBaseFileCount

2039cmdType::ctGetValue

2040[MEDIACMD::dwPosition](#) - Count number of entries in database (OUT)

gsMetaBaseFileName Get each file item in database

2041cmdType::ctGetValue

2042[MEDIACMD::dwPosition](#) - Index (IN)

2043[MEDIACMD::arbID](#) - File Name (OUT)

2044[MEDIACMD::dwPosition](#) - Success/Index (OUT)

gsMetaBaseFileRemove Removing a file from the database

2045cmdType::ctSetValue
2046[MEDIACMD::arbID](#) - File Name (IN)
2047[MEDIACMD::dwPosition](#) - return -1 if failed (OUT)
gsMetaBaseFileAdd Add a new table entry for a file

2048cmdType::ctSetValue
2049[MEDIACMD::arbID](#) - FileName (IN)
2050[MEDIACMD::dwPosition](#) - Success/Index (-1 if exist) (OUT)
gsMetaBaseTagIndex Get metadata item and it's value for specified file item

2051cmdType::ctGetValue
2052[MEDIACMD::arbID](#) - FileName + MetaDataTag (IN)
2053[MEDIACMD::dwPosition](#) - Index of meta item, or -1 if !exist (OUT)
gsMetaBaseGetTag Get metadata item and it's value for specified file item

2054cmdType::ctGetValue
2055[MEDIACMD::arbID](#) - File Name (Table in DataBase) (IN)
2056[MEDIACMD::dwPosition](#) - Index of meta item (IN)
2057[MEDIACMD::arbID](#) - Name/Value (Name is first starting at [0]) (OUT)
2058[MEDIACMD::dwStart](#) - Where in arbID the value starts (OUT)
2059[MEDIACMD::dwPosition](#) - Success/Index (OUT)
gsMetaBaseSetTag Set/Insert metadata item and it's value for specified file

2060cmdType::ctSetValue
2061[MEDIACMD::arbID](#) - Name/Tag/Value (Name is first starting at [0]) (IN)
2062Tag starts at [MEDIACMD::dwStart](#), Value starts at [MEDIACMD::dwEnd](#)
gsMetaBaseDefaultTags Insert all default metadata tags, (no values)

2063cmdType::ctSetValue
2064[MEDIACMD::arbID](#) - Name (IN)
gsMetaBaseFileRename Reset table (filename)

2065cmdType::ctSetValue
2066[MEDIACMD::arbID](#) - Name (IN)
gsMetaBaseReplayMark Set replay mark 15 seconds after tc value sent

2067cmdType::ctSetValue
2068[MEDIACMD::arbID](#) -
gsMetaBaseGetName Get the current database name

2069cmdType::ctGetValue
2070[MEDIACMD::arbID](#) -
gsMetaBaseGetTable Get the current database name

2071cmdType::ctGetValue
2072[MEDIACMD::arbID](#) -
gsMetaBaseResetAndRecord Reset and start the record session

2073cmdType::ctSetValue
2074[MEDIACMD::dwPosition](#) - Record Start (IN)
2075[MEDIACMD::dwChannel](#) - Record Channel (IN)
2076[MEDIACMD::arbID](#) - ClipName (IN)
gsAudInSelect Audio input select

2077cmdType::ctSetValue
2078[MEDIACMD::dwPosition](#) - New audio input [GS_AUDSELECT_UNBALANCED_10](#)
[GS_AUDSELECT_UNBALANCED_4](#) [GS_AUDSELECT_BALANCED_10](#)
[GS_AUDSELECT_BALANCED_4](#) [GS_AUDSELECT_SPDIF](#) [GS_AUDSELECT_AES_EBU](#)
[GS_AUDSELECT_EMBEDDED](#)
2079[MEDIACMD::dwAudioChannels](#) - Channels affected

2080cmdType::ctGetValue

2081 [MEDIACMD::dwPosition](#) - Current input [GS_AUDSELECT_UNBALANCED_10](#)
[GS_AUDSELECT_UNBALANCED_4](#) [GS_AUDSELECT_BALANCED_10](#)
[GS_AUDSELECT_BALANCED_4](#) [GS_AUDSELECT_SPDIF](#) [GS_AUDSELECT_AES_EBU](#)
[GS_AUDSELECT_EMBEDDED](#)

2082 [MEDIACMD::dwStart](#) - Bit array of available inputs, see above or [GS_AUDSELECT_NONE](#)
[MEDIACMD::dwAudioChannels](#) - Channels requested

2083

gsAudOutSelect Audio output select (in general all outputs are active)

2084cmdType::ctSetValue

2085 [MEDIACMD::dwPosition](#) - New audio output [GS_AUDSELECT_UNBALANCED_10](#)
[GS_AUDSELECT_UNBALANCED_4](#) [GS_AUDSELECT_BALANCED_10](#)
[GS_AUDSELECT_BALANCED_4](#) [GS_AUDSELECT_SPDIF](#) [GS_AUDSELECT_AES_EBU](#)
[GS_AUDSELECT_EMBEDDED](#)

2086 [MEDIACMD::dwAudioChannels](#) - Channels affected

2087cmdType::ctGetValue

2088 [MEDIACMD::dwPosition](#) - Current output [GS_AUDSELECT_UNBALANCED_10](#)
[GS_AUDSELECT_UNBALANCED_4](#) [GS_AUDSELECT_BALANCED_10](#)
[GS_AUDSELECT_BALANCED_4](#) [GS_AUDSELECT_SPDIF](#) [GS_AUDSELECT_AES_EBU](#)
[GS_AUDSELECT_EMBEDDED](#)

2089 [MEDIACMD::dwStart](#) - Bit array of available outputs, see above or [GS_AUDSELECT_NONE](#)
[MEDIACMD::dwAudioChannels](#) - Channels requested

2090

gsAudInputLevel Audio input level (gain)

2091cmdType::ctSetValue

2092 [MEDIACMD::dwPosition](#) - Level (0-65535)

2093 [MEDIACMD::dwAudioChannels](#) - Channels affected

2094cmdType::ctGetValue

2095 [MEDIACMD::dwPosition](#) - Level (0-65535)

2096 [MEDIACMD::dwStart](#) - Minimum level (usually 0)

2097 [MEDIACMD::dwEnd](#) - Maximum level (usually 65535)

[MEDIACMD::dwAudioChannels](#) - Channels affected

2098

gsAudOutputLevel Audio output level (master)

2099cmdType::ctSetValue

2100 [MEDIACMD::dwPosition](#) - Level (0-65535)

2101 [MEDIACMD::dwAudioChannels](#) - Channels affected

2102cmdType::ctGetValue

2103 [MEDIACMD::dwPosition](#) - Level (0-65535)

2104 [MEDIACMD::dwStart](#) - Minimum level (usually 0)

2105 [MEDIACMD::dwEnd](#) - Maximum level (usually 65535)

[MEDIACMD::dwAudioChannels](#) - Channels affected

2106

gsAudAdvanceLevel Audio advanced level (advanced cue head master) - Not Supported

2107cmdType::ctSetValue

2108 [MEDIACMD::dwPosition](#) - Level (0-65535)

2109 [MEDIACMD::dwAudioChannels](#) - Channels affected

2110cmdType::ctGetValue
2111[MEDIACMD::dwPosition](#) - Level (0-65535)
2112[MEDIACMD::dwStart](#) - Minimum level (usually 0)
2113[MEDIACMD::dwEnd](#) - Maximum level (usually 65535)
[MEDIACMD::dwAudioChannels](#) - Channels affected

2114
gsAudOutPhase Audio output phase

2115cmdType::ctSetValue
2116[MEDIACMD::dwPosition](#) - Phase offset (default = 0) (0-65520 = degrees * 182)
2117[MEDIACMD::dwAudioChannels](#) - Channels affected
2118cmdType::ctGetValue
2119[MEDIACMD::dwPosition](#) - Phase offset (0-65520 = degrees * 182)
2120[MEDIACMD::dwStart](#) - Minimum phase available (usually 0)
2121[MEDIACMD::dwEnd](#) - Maximum phase available (usually 65520 = 360 * 182)
[MEDIACMD::dwAudioChannels](#) - Channels affected

2122
gsAudOutAdvancePhase Audio advance phase (advanced cue head master) - Not Supported

2123cmdType::ctSetValue
2124[MEDIACMD::dwPosition](#) - Phase offset (default = 0) (0-65520 = degrees * 182)
2125[MEDIACMD::dwAudioChannels](#) - Channels affected
2126cmdType::ctGetValue
2127[MEDIACMD::dwPosition](#) - Phase offset (0-65520 = degrees * 182)
2128[MEDIACMD::dwStart](#) - Minimum phase available (usually 0)
2129[MEDIACMD::dwEnd](#) - Maximum phase available (usually 65520 = 360 * 182)
[MEDIACMD::dwAudioChannels](#) - Channels affected

2130
gsAudCrossFadeTime Audio cross-fade time (clip effect overlap) - Not Supported

2131cmdType::ctSetValue
2132[MEDIACMD::dwPosition](#) - Length of cross-fade in milliseconds
2133cmdType::ctGetValue
2134[MEDIACMD::dwPosition](#) - Length of cross-fade in milliseconds
2135[MEDIACMD::dwStart](#) - Minimum cross-fade length (usually 0 = cut)
[MEDIACMD::dwEnd](#) - Maximum cross-fade length (depends on device)

2136
gsAudLtcEnable Enable LTC on an audio channel

2137cmdType::ctSetValue
2138[MEDIACMD::dwPosition](#) - [GS_ENABLE](#) or [GS_DISABLE](#)
2139[MEDIACMD::dwAudioChannels](#) - Bit for channel to use for LTC
2140cmdType::ctGetValue
[MEDIACMD::dwPosition](#) - Is LTC enabled

2141
gsAudInLtcChannel Set audio channel to use for LTC input if enabled. Currently will set LTC output to same channel on all VVW drivers.

2142cmdType::ctSetValue
2143[MEDIACMD::dwAudioChannels](#) - Bit for channel to use for LTC
2144cmdType::ctGetValue
[MEDIACMD::dwAudioChannels](#) - Bit channel is using for LTC

2145
gsAudOutLtcChannel Set audio channel to use for LTC output if enabled. Currently will set LTC input to same channel on all VVW drivers.

2146cmdType::ctSetValue
2147[MEDIACMD::dwAudioChannels](#) - Bit for channel to use for LTC
2148cmdType::ctGetValue
[MEDIACMD::dwAudioChannels](#) - Bit channel is using for LTC

2149
gsAudDtmfEnable Enable DTMF on an audio channel

2150cmdType::ctSetValue
2151[MEDIACMD::dwPosition](#) - [GS_ENABLE](#) or [GS_DISABLE](#)
2152[MEDIACMD::dwAudioChannels](#) - Bit for channel to use for DTMF
2153cmdType::ctGetValue
[MEDIACMD::dwPosition](#) - Is DTMF enabled

2154
gsAudInDtmfChannel Set audio channel to use for DTMF input if enabled. Currently will set DTMF output to same channel on all VVW drivers.

2155cmdType::ctSetValue
2156[MEDIACMD::dwAudioChannels](#) - Bit for channel to use for DTMF
2157cmdType::ctGetValue
[MEDIACMD::dwAudioChannels](#) - Bit channel is using for DTMF

2158
gsAudOutDtmfChannel Set audio channel to use for DTMF output if enabled. Currently will set DTMF input to same channel on all VVW drivers.

2159cmdType::ctSetValue
2160[MEDIACMD::dwAudioChannels](#) - Bit for channel to use for DTMF
2161cmdType::ctGetValue
[MEDIACMD::dwAudioChannels](#) - Bit channel is using for DTMF

2162
gsAudWavePeakRMS Return the last known RMS and peak value of the audio output. Max 2 channels returned per call. 2 channels should always be requested

2163cmdType::ctSetValue
2164- Not Supported
2165cmdType::ctGetValue
2166-in- [MEDIACMD::dwAudioChannels](#) - Requested channels to check
2167[MEDIACMD::dwStart](#) - HUnsigned short=RMS channel +1, LOunsigned short=RMS channel +0

(range 0-65535)
2168 [MEDIACMD::dwEnd](#) - HIunsigned short=Peak channel +1, LOunsigned short=Peak channel +0
(range 0-65535)
[MEDIACMD::dwPosition](#) - duplicates [MEDIACMD::dwStart](#)

2169

gsAudInputBitRate Get / Set the current bit rate for recording audio per call

2170cmdType::ctSetValue

2171- Not Supported

2172cmdType::ctGetValue

2173-in- [MEDIACMD::dwAudioChannels](#) - Requested channels to check

gsAudInputSampleRate Set the audio input sample rate

2174cmdType::ctSetValue

2175 [MEDIACMD::dwPosition](#) - New sample rate (typically 48000 or 96000)

2176cmdType::ctGetValue

2177 [MEDIACMD::dwPosition](#) - Current sample rate (use cmdType::ctValueSupported to get list)

2178 [MEDIACMD::dwStart](#) - Lowest supported sample rate

[MEDIACMD::dwEnd](#) - Highest supported sample rate

2179

gsAudInputMode Audio input mode

2180cmdType::ctSetValue

2181 [MEDIACMD::dwPosition](#) - See Sony defines

2182cmdType::ctGetValue

[MEDIACMD::dwPosition](#) - See Sony defines

2183

gsAudInputHeadRoom Audio input head room

2184cmdType::ctSetValue

2185 [MEDIACMD::dwPosition](#) - Set required headroom

2186cmdType::ctGetValue

2187 [MEDIACMD::dwPosition](#) - Current headroom

2188 [MEDIACMD::dwStart](#) - Min headroom

[MEDIACMD::dwEnd](#) - Max Headroom ;)

2189

gsAudInputOriginal Audio input original - see Sony def

2190cmdType::ctSetValue

2191 [MEDIACMD::dwPosition](#) -

2192cmdType::ctGetValue

[MEDIACMD::dwPosition](#) -

2193

gsAudInputErrorProtect Enable and disable audio input error protection

2194cmdType::ctSetValue

2195 [MEDIACMD::dwPosition](#) - GS_TRUE protection enabled, else GS_FALSE

2196cmdType::ctGetValue

[MEDIACMD::dwPosition](#) - GS_TRUE protection enabled, else GS_FALSE

2197

gsAudInputCopyright Does the audio input bit stream contain a copyright flag

2198cmdType::ctSetValue

2199[MEDIACMD::dwPosition](#) - GS_TRUE copyright flag is set, else GS_FALSE

2200cmdType::ctGetValue

[MEDIACMD::dwPosition](#) - GS_TRUE copyright flag is set, else GS_FALSE

2201

gsAudInputSlave Audio input is in slave mode

2202cmdType::ctSetValue

2203[MEDIACMD::dwPosition](#) - GS_TRUE if in slave mode, else GS_FALSE

2204cmdType::ctGetValue

[MEDIACMD::dwPosition](#) - GS_TRUE if in slave mode, else GS_FALSE

2205

gsAudInputBass Audio bass setting, hardware dependent

2206cmdType::ctSetValue

2207[MEDIACMD::dwPosition](#) - 0..65536, 32768 being nominal and -1 being default

2208[MEDIACMD::dwAudioChannels](#) - Bit(s) in use

2209cmdType::ctGetValue

2210[MEDIACMD::dwPosition](#) - 0..65536, 32768 being nominal and -1 being not supported

[MEDIACMD::dwAudioChannels](#) - Bit(s) in use

2211

gsAudInputTreble Audio treble setting, hardware dependent

2212cmdType::ctSetValue

2213[MEDIACMD::dwPosition](#) - 0..65536, 32768 being nominal and -1 being default

2214[MEDIACMD::dwAudioChannels](#) - Bit(s) in use

2215cmdType::ctGetValue

2216[MEDIACMD::dwPosition](#) - 0..65536, 32768 being nominal and -1 being not supported

[MEDIACMD::dwAudioChannels](#) - Bit(s) in use

2217

gsAudInputStatus What audio channels are available, selected and valid

2218cmdType::ctSetValue

2219Not Available

2220cmdType::ctGetValue

2221[MEDIACMD::dwPosition](#) - Audio channel bits with valid inputs

2222[MEDIACMD::dwStart](#) - Audio channel bits on primary audio selection

2223[MEDIACMD::dwEnd](#) - Audio channel bits being monitored

[MEDIACMD::dwAudioChannels](#) - Audio channel bits available

2224

gsAudioMappingInput Changes the audio mapping on the input

2225cmdType::ctSetValue

2226Not Available

2227cmdType::ctGetValue

2228[MEDIACMD::dwPosition](#) - Bit array of the mapping

[MEDIACMD::dwAudioChannels](#) - Audio channel to map

2229

gsAudioMappingOutput Changes the audio mapping on the output

2230cmdType::ctSetValue

2231Not Available

2232cmdType::ctGetValue

2233[MEDIACMD::dwPosition](#) - Bit array of the mapping

[MEDIACMD::dwAudioChannels](#) - Audio channel to map

2234

gsAudMonitorSelect Selects the pair of channels to monitored (usually RCA analog)

2235cmdType::ctSetValue

2236[MEDIACMD::dwPosition](#) - Numeric pair 1/2=0, 3/4=1, 5/6=2, 7/8=3, etc

2237cmdType::ctGetValue

2238[MEDIACMD::dwPosition](#) - Numeric pair 1/2=0, 3/4=1, 5/6=2, 7/8=3, etc

2239[MEDIACMD::dwStart](#) - Available pairs (bitwise)

[MEDIACMD::dwEnd](#) - Highest possible value (like position)

2240

gsAudChannelsEncoded Selects the channels that are encoded (need raw capture/playback) for Dolby (or doubley)

2241cmdType::ctSetValue

2242[MEDIACMD::dwPosition](#) - Bit pair 1/2=0x03, 3/4=0x0C, 5/6=0x30, 7/8=0xC0, etc

2243cmdType::ctGetValue

2244[MEDIACMD::dwPosition](#) - Bit pair 1/2=0x03, 3/4=0x0C, 5/6=0x30, 7/8=0xC0, etc

[MEDIACMD::dwStart](#) - Available pairs

2245

gsAudAudioScrub Enable or disable audio output at non play speed (scrub)

2246cmdType::ctSetValue

2247[MEDIACMD::dwPosition](#) - 1-enable audio scrub, 0-disable

2248cmdType::ctGetValue

2249[MEDIACMD::dwPosition](#) - 1-enable audio scrub, 0-disable

[MEDIACMD::dwStart](#) - 1 is available

2250

gsVidFreeze Freeze the video output

2251cmdType::ctSetValue

2252[MEDIACMD::dwPosition](#) - Freeze type [GS_VIDFREEZE_NOT_FROZEN](#),
[GS_VIDFREEZE_FIELD0](#), [GS_VIDFREEZE_FIELD1](#), [GS_VIDFREEZE_FRAME](#)

2253[MEDIACMD::dwVideoChannels](#) - Bit(s) for the channel(s) to freeze

2254cmdType::ctGetValue

[MEDIACMD::dwAudioChannels](#) - Bit(s) for the channel(s) currently frozen

2255

gsVidPreReadMode Set DDR into pre read (read before write) mode - requires 2 or more channels

2256cmdType::ctSetValue

2257 [MEDIACMD::dwPosition](#) - Channel to use as output
2258 [MEDIACMD::dwStart](#) - [GS_ENABLE](#) or [GS_DISABLE](#)
2259 [MEDIACMD::dwVideoChannels](#) - Channels to record
2260 [MEDIACMD::dwAudioChannels](#) - Channels to record
2261 [MEDIACMD::dwInfoChannels](#) - Channels to record
2262 cmdType::ctGetValue
2263 [MEDIACMD::dwStart](#) - [GS_ENABLE](#) or [GS_DISABLE](#) - if not enabled, the rest does not matter
2264 [MEDIACMD::dwPosition](#) - Channel in use as output
2265 [MEDIACMD::dwVideoChannels](#) - Channels recording
2266 [MEDIACMD::dwAudioChannels](#) - Channels recording
[MEDIACMD::dwInfoChannels](#) - Channels recording

2267

gsVidEditField First field recorded in edit

2268 cmdType::ctSetValue

2269 [MEDIACMD::dwPosition](#) - Edit start field ([GS_FIELD2](#) for second, else first)

2270 cmdType::ctGetValue

[MEDIACMD::dwPosition](#) - Edit start field ([GS_FIELD2](#) for second, else first)

2271

gsVidRecFrame Record frames or fields

2272 cmdType::ctSetValue

2273 [MEDIACMD::dwPosition](#) - [GS_FIELD](#) record single field, else record frames (default - frames (both fields))

2274 cmdType::ctGetValue

[MEDIACMD::dwPosition](#) - [GS_FIELD](#) recording single field, else recording frames (default - frames (both fields))

2275

gsVidPlayFrame Play frames or fields

2276 cmdType::ctSetValue

2277 [MEDIACMD::dwPosition](#) - [GS_FIELD](#) play single field, else play frames (default - frames (both fields))

2278 cmdType::ctGetValue

[MEDIACMD::dwPosition](#) - [GS_FIELD](#) recording fields, else recording frames (default - frames (both fields))

2279

gsVidNoEE Disable video edit to edit passthrough

2280 cmdType::ctSetValue

2281 [MEDIACMD::dwPosition](#) - [GS_TRUE](#) Always in playback mode, else if [GS_FALSE](#) then will passthrough video

2282 cmdType::ctGetValue

[MEDIACMD::dwPosition](#) - [GS_TRUE](#) Always in playback mode, else if [GS_FALSE](#) then will passthrough video

2283

gsVidSuperimpose Enable superimposed tc/state/menu output in video

2284 cmdType::ctSetValue

2285 [MEDIACMD::dwPosition](#) - [GS_TRUE](#) Superimpose, else normal video

2286 [MEDIACMD::dwStart](#) - SuperImpose Type 0 = VTR Style 1= Film Full 2= Film basic
2287 [MEDIACMD::dwEnd](#) - SuperImpose on VGA only
2288 [MEDIACMD::dwVideoChannels](#) - Height to start Imposing
2289 [MEDIACMD::dwAudioChannels](#) - Width to start Imposing
2290 [MEDIACMD::lSpeed](#) - Color of watermark (NOT SUPERIMPOSE TEXT YET)
2291 cmdType::ctGetValue
2292 [MEDIACMD::dwPosition](#) - [GS_TRUE](#) Superimpose, [GS_FALSE](#) Normal Video, [GS_NOT_SUPPORTED](#) cannot superimpose
2293 [MEDIACMD::dwStart](#) - SuperImpose Type 0 = VTR Style 1= Film Full 2= Film basic
2294 [MEDIACMD::dwEnd](#) - SuperImpose on VGA only
2295 [MEDIACMD::dwVideoChannels](#) - Height to start Imposing
[MEDIACMD::dwAudioChannels](#) - Width to start Imposing

2296
gsVidAnalogMonitorSDType Select the output type of the analog SD (Composite, SMPTE, RGB)

2297 cmdType::ctSetValue
2298 [MEDIACMD::dwPosition](#) - [GS_VIDSELECT_COMPONENT_YUV](#)
[GS_VIDSELECT_COMPONENT_YUV_M2](#) [GS_VIDSELECT_COMPONENT_YUV_SMPTE](#)
[GS_VIDSELECT_COMPONENT_RGB](#) [GS_VIDSELECT_COMPOSITE](#)
2299 cmdType::ctGetValue
[MEDIACMD::dwPosition](#) - [GS_VIDSELECT_COMPONENT_YUV](#)
[GS_VIDSELECT_COMPONENT_YUV_M2](#) [GS_VIDSELECT_COMPONENT_YUV_SMPTE](#)
[GS_VIDSELECT_COMPONENT_RGB](#) [GS_VIDSELECT_COMPOSITE](#)

2300
gsVidAnalogMonitorHDType Select the output type of the analog HD (RGB, SMPTE, xVidRGB)

2301 cmdType::ctSetValue
2302 [MEDIACMD::dwPosition](#) - [GS_VIDSELECT_COMPONENT_YUV](#)
[GS_VIDSELECT_COMPONENT_YUV_M2](#) [GS_VIDSELECT_COMPONENT_YUV_SMPTE](#)
[GS_VIDSELECT_COMPONENT_RGB](#) [GS_VIDSELECT_XVID_RGB](#)
2303 cmdType::ctGetValue
[MEDIACMD::dwPosition](#) - [GS_VIDSELECT_COMPONENT_YUV](#)
[GS_VIDSELECT_COMPONENT_YUV_M2](#) [GS_VIDSELECT_COMPONENT_YUV_SMPTE](#)
[GS_VIDSELECT_COMPONENT_RGB](#) [GS_VIDSELECT_XVID_RGB](#)

2304
gsVidAnalogMonitorMethod Set the type of up down convert to do

2305 cmdType::ctSetValue
2306 [MEDIACMD::dwPosition](#) - [GS_ANALOGMONITORMETHOD_DIRECT](#),
[GS_ANALOGMONITORMETHOD_SD](#), [GS_ANALOGMONITORMETHOD_HD720](#),
[GS_ANALOGMONITORMETHOD_HD1080](#), [GS_ANALOGMONITORMETHOD_FLIP720](#),
[GS_ANALOGMONITORMETHOD_FLIP1080](#)
2307 cmdType::ctGetValue
[MEDIACMD::dwPosition](#) - [GS_ANALOGMONITORMETHOD_DIRECT](#),
[GS_ANALOGMONITORMETHOD_SD](#), [GS_ANALOGMONITORMETHOD_HD720](#),
[GS_ANALOGMONITORMETHOD_HD1080](#), [GS_ANALOGMONITORMETHOD_FLIP720](#),
[GS_ANALOGMONITORMETHOD_FLIP1080](#) or [GS_NOT_SUPPORTED](#)

2308
gsVidAnalogMonitorUpMode Select the method for upconverting to HD

2309cmdType::ctSetValue

2310MEDIACMD::dwPosition - [GS_UPCONVERT_ANAMORPHIC](#),
[GS_UPCONVERT_PILLARBOX](#), [GS_UPCONVERT_ZOOM14x9](#),
[GS_UPCONVERT_LETTERBOX](#), [GS_UPCONVERT_ZOOMWIDE](#)

2311cmdType::ctGetValue

[MEDIACMD::dwPosition](#) - [GS_UPCONVERT_ANAMORPHIC](#), [GS_UPCONVERT_PILLARBOX](#),
[GS_UPCONVERT_ZOOM14x9](#), [GS_UPCONVERT_LETTERBOX](#), [GS_UPCONVERT_ZOOMWIDE](#)

2312

gsVidAnalogMonitorDownMode Select the method for downconverting to SD

2313cmdType::ctSetValue

2314MEDIACMD::dwPosition - [GS_DOWNCONVERT_LETTERBOX](#),
[GS_DOWNCONVERT_CROP](#), [GS_DOWNCONVERT_ANAMORPHIC](#)

2315cmdType::ctGetValue

[MEDIACMD::dwPosition](#) - [GS_DOWNCONVERT_LETTERBOX](#), [GS_DOWNCONVERT_CROP](#),
[GS_DOWNCONVERT_ANAMORPHIC](#)

2316

gsVidPanScanZoom Set/Get the current pan scan pos and zoom

2317cmdType::ctSetValue

2318MEDIACMD::dwPosition - when to do it (0xFFFFFFFF == immediate)

2319MEDIACMD::dwStart - X (upper bit 0x80000000 is FLIP)

2320MEDIACMD::dwEnd - Y (upper bit 0x80000000 is FLIP)

2321MEDIACMD::dwSpeed - Z

2322MEDIACMD::dwAudioChannels - X Aspect

2323MEDIACMD::dwVideoChannels - Y Aspect

2324MEDIACMD::dwInfoChannels - Rotate

2325cmdType::ctGetValue

2326MEDIACMD::dwPosition - when to do it

2327MEDIACMD::dwStart - X (upper bit 0x80000000 is FLIP)

2328MEDIACMD::dwEnd - Y (upper bit 0x80000000 is FLIP)

2329MEDIACMD::dwSpeed - Z

2330MEDIACMD::dwAudioChannels - X Aspect

2331MEDIACMD::dwVideoChannels - Y Aspect

[MEDIACMD::dwInfoChannels](#) - Rotate

2332

gsVidSlowMotionMode Slow motion mode - 'use field duplication'

2333cmdType::ctSetValue

2334MEDIACMD::dwPosition - 1 enabled, 0 disabled

2335cmdType::ctGetValue

2336MEDIACMD::dwPosition - 1 enabled, 0 disabled

[MEDIACMD::dwStart](#) - 1 if available

2337

gsVidVariCamMode Varicam record/playback mode

2338cmdType::ctSetValue

2339MEDIACMD::dwPosition - [GS_FRAMEDROPMODE_VARICAM_MASK_FPS](#)

2340MEDIACMD::dwPosition - [GS_FRAMEDROPMODE_VARICAM_MASK_FPS](#)

2341cmdType::ctGetValue

2342 [MEDIACMD::dwPosition](#) - GS_FRAMEDROPMODE_VARICAM_MASK_FPS
[MEDIACMD::dwStart](#) - 1 if available

2343
gsVidCustomSuperimpose Add custom superimpose elements

2344 cmdType::ctSetValue
2345 [MEDIACMD::dwPosition](#) - 0 = clear all custom elements,
2346 1 = Text element 2 = Line 3 = Box (line) 4 = Rectangle (filled) 5 = Circle (line) 6 = Circle (filled)
7 = Oval 8 = Oval (filled)
2347 [MEDIACMD::dwStart](#) - X position
2348 [MEDIACMD::dwEnd](#) - Y position
2349 [MEDIACMD::lSpeed](#) - Modifier (size for text)
2350 [MEDIACMD::dwVideoChannels](#) - Width (not for text)
2351 [MEDIACMD::dwAudioChannels](#) - Height (not for text)
2352 [MEDIACMD::dwInfoChannels](#) - Color
2353 cmdType::ctGetValue
2354 [MEDIACMD::dwPosition](#) - Number of custom elements, 0 = no custom superimpose
2355 [MEDIACMD::dwStart](#) - Available width of frame
2356 [MEDIACMD::dwEnd](#) - Available height of frame
2357 [MEDIACMD::dwVideoChannels](#) - Width of text element
[MEDIACMD::dwAudioChannels](#) - Height of text element

2358

gsVidInSelect Select video input

2359 cmdType::ctSetValue
2360 [MEDIACMD::dwPosition](#) - Video input to use [GS_VIDSELECT_COMPOSITE](#),
[GS_VIDSELECT_COMPOSITE_2](#), [GS_VIDSELECT_SVIDEO](#),
[GS_VIDSELECT_COMPONENT_YUV](#), [GS_VIDSELECT_COMPONENT_YUV_M2](#),
[GS_VIDSELECT_COMPONENT_YUV_SMPTE](#), [GS_VIDSELECT_COMPONENT_RGB](#),
[GS_VIDSELECT_D1_SERIAL](#), [GS_VIDSELECT_D1_PARALLEL](#), [GS_VIDSELECT_SDTI](#),
[GS_VIDSELECT_NONE](#)
2361 cmdType::ctGetValue
2362 [MEDIACMD::dwPosition](#) - Video input to use [GS_VIDSELECT_COMPOSITE](#),
[GS_VIDSELECT_COMPOSITE_2](#), [GS_VIDSELECT_SVIDEO](#),
[GS_VIDSELECT_COMPONENT_YUV](#), [GS_VIDSELECT_COMPONENT_YUV_M2](#),
[GS_VIDSELECT_COMPONENT_YUV_SMPTE](#), [GS_VIDSELECT_COMPONENT_RGB](#),
[GS_VIDSELECT_D1_SERIAL](#), [GS_VIDSELECT_D1_PARALLEL](#), [GS_VIDSELECT_SDTI](#),
[GS_VIDSELECT_NONE](#)
[MEDIACMD::dwStart](#) - Supported video inputs (bit array using defines from dwPosition)

2363

gsVidInLockType Select video input genlock type

2364 cmdType::ctSetValue
2365 [MEDIACMD::dwPosition](#) - Video input lock type use [GS_VIDLOCKTYPE_VTR](#) or
[GS_VIDLOCKTYPE_BROADCAST](#)
2366 cmdType::ctGetValue
2367 [MEDIACMD::dwPosition](#) - Video input lock type use [GS_VIDLOCKTYPE_VTR](#) or
[GS_VIDLOCKTYPE_BROADCAST](#)
[MEDIACMD::dwStart](#) - Supported video inputs (bit array using defines from dwPosition)

2368

gsVidInSetup Input TBC - Setup (~brightness) Normal range: 0-65535 (0x0000-0xffff)

2369cmdType::ctSetValue
2370[MEDIACMD::dwPosition](#) - Video input TBC Setup
2371cmdType::ctGetValue
2372[MEDIACMD::dwPosition](#) - Video input TBC Setup
2373[MEDIACMD::dwStart](#) - Lowest possible value (usually 0)
[MEDIACMD::dwEnd](#) - Highest possible value (usually 65535)

2374
gsVidInVideo Input TBC - Video (~contrast) Normal range: 0-65535 (0x0000-0xffff)

2375cmdType::ctSetValue
2376[MEDIACMD::dwPosition](#) - Video input TBC Video
2377cmdType::ctGetValue
2378[MEDIACMD::dwPosition](#) - Video input TBC Video
2379[MEDIACMD::dwStart](#) - Lowest possible value (usually 0)
[MEDIACMD::dwEnd](#) - Highest possible value (usually 65535)

2380
gsVidInHue Input TBC - Hue (~color angle) degrees * 182. Normal range: 0-65520 (0x0000-0xffff0)

2381cmdType::ctSetValue
2382[MEDIACMD::dwPosition](#) - Video input TBC Hue
2383cmdType::ctGetValue
2384[MEDIACMD::dwPosition](#) - Video input TBC Hue
2385[MEDIACMD::dwStart](#) - Lowest possible value (usually 0)
[MEDIACMD::dwEnd](#) - Highest possible value (usually 65520)

2386
gsVidInChroma Input TBC - Chroma (~saturation) Normal range: 0-65535 (0x0000-0xffff)

2387cmdType::ctSetValue
2388[MEDIACMD::dwPosition](#) - Video input TBC Chroma
2389cmdType::ctGetValue
2390[MEDIACMD::dwPosition](#) - Video input TBC Chroma
2391[MEDIACMD::dwStart](#) - Lowest possible value (usually 0)
[MEDIACMD::dwEnd](#) - Highest possible value (usually 65535)

2392
gsVidInUChroma Input TBC - U Chroma or Cb or Y'CrCb Normal range: 0-65535 (0x0000-0xffff) Normally only affects the component video or D1 Serial inputs.

2393cmdType::ctSetValue
2394[MEDIACMD::dwPosition](#) - Video input TBC U (Cb) Chroma
2395cmdType::ctGetValue
2396[MEDIACMD::dwPosition](#) - Video input TBC U (Cb) Chroma
2397[MEDIACMD::dwStart](#) - Lowest possible value (usually 0)
[MEDIACMD::dwEnd](#) - Highest possible value (usually 65535)

2398
gsVidInVChroma Input TBC - V Chroma or Cr or Y'CrCb Normal range: 0-65535 (0x0000-0xffff) Normally only affects the component video or D1 Serial inputs.

2399cmdType::ctSetValue

2400 [MEDIACMD::dwPosition](#) - Video input TBC V (Cr) Chroma
2401 cmdType::ctGetValue
2402 [MEDIACMD::dwPosition](#) - Video input TBC V (Cr) Chroma
2403 [MEDIACMD::dwStart](#) - Lowest possible value (usually 0)
[MEDIACMD::dwEnd](#) - Highest possible value (usually 65535)

2404
gsVidInColorKiller Remove color from input signal (black and white luminance data only)

2405 cmdType::ctSetValue
2406 [MEDIACMD::dwPosition](#) - If 1, signal will have no chroma, if 0, normal signal
2407 cmdType::ctGetValue
[MEDIACMD::dwPosition](#) - If 1, signal will have no chroma, if 0, normal signal

2408
gsVidInAGC Automatic gain control

2409 cmdType::ctSetValue
2410 [MEDIACMD::dwPosition](#) - If 1, signal adjust gain automatically, if 0, will use cmdGetSetValue::gsVidInSetup and cmdGetSetValue::gsVidInVideo
2411 cmdType::ctGetValue
[MEDIACMD::dwPosition](#) - If 1, signal adjust gain automatically, if 0, will use cmdGetSetValue::gsVidInSetup and cmdGetSetValue::gsVidInVideo

2412
gsVidInBandwidth Maximum input bandwidth setting

2413 cmdType::ctSetValue
2414 [MEDIACMD::dwPosition](#) - Uses [GS_VIDBAND_STANDARD](#), [GS_VIDBAND_MEDIUM](#), [GS_VIDBAND_HIGH](#), [GS_VIDBAND_NOTCH](#)
2415 cmdType::ctGetValue
2416 [MEDIACMD::dwPosition](#) - Uses [GS_VIDBAND_STANDARD](#), [GS_VIDBAND_MEDIUM](#), [GS_VIDBAND_HIGH](#), [GS_VIDBAND_NOTCH](#)
[MEDIACMD::dwStart](#) - Bit array of allowable values as defined for dwPosition above

2417
gsVidInBlack Black type (NTSC only)

2418 cmdType::ctSetValue
2419 [MEDIACMD::dwPosition](#) - Uses [GS_VIDBLACK_SETUP](#), [GS_VIDBLACK_CRYSTAL](#), [GS_VIDBLACK_SUPER](#)
2420 cmdType::ctGetValue
2421 [MEDIACMD::dwPosition](#) - Uses [GS_VIDBLACK_SETUP](#), [GS_VIDBLACK_CRYSTAL](#), [GS_VIDBLACK_SUPER](#)
[MEDIACMD::dwStart](#) - Bit array of allowable values as defined for dwPosition above

2422
gsVidInWhite White type (NTSC only)

2423 cmdType::ctSetValue
2424 [MEDIACMD::dwPosition](#) - Uses [GS_VIDWHITE_CLAMP](#), [GS_VIDWHITE_SCALE](#), [GS_VIDWHITE_FREE](#)
2425 cmdType::ctGetValue
2426 [MEDIACMD::dwPosition](#) - Uses [GS_VIDWHITE_CLAMP](#), [GS_VIDWHITE_SCALE](#),

[GS_VIDWHITE_FREE](#)
[MEDIACMD::dwStart](#) - Bit array of allowable values as defined for dwPosition above

2427
gsVidInCoring Input digital signal coring. Removal of low order bits to remove DAC aliasing

2428cmdType::ctSetValue
2429[MEDIACMD::dwPosition](#) - Remove bottom 0, 1 or 2 bits of digitized signal
2430cmdType::ctGetValue
2431[MEDIACMD::dwPosition](#) - Remove bottom 0, 1 or 2 bits of digitized signal
[MEDIACMD::dwStart](#) - Bit array of allowable values as defined for dwPosition above (0 always supported)

2432
gsVidInPeaking Remove (smooth) 100% signal spikes

2433cmdType::ctSetValue
2434[MEDIACMD::dwPosition](#) - 0 leave signal intact, 1 smooth
2435cmdType::ctGetValue
2436[MEDIACMD::dwPosition](#) - 0 leave signal intact, 1 smooth
[MEDIACMD::dwStart](#) - Bit array of allowable values as defined for dwPosition above (0 always supported)

2437
gsVidInSharpness Set video transition sharpness

2438cmdType::ctSetValue
2439[MEDIACMD::dwPosition](#) - Depends on cmdType::ctGetValue [MEDIACMD::dwStart](#) [MEDIACMD::dwEnd](#) (typically 0-7, 0-100, 0-65535)
2440cmdType::ctGetValue
2441[MEDIACMD::dwPosition](#) - Video digitizing sharpness
2442[MEDIACMD::dwStart](#) - Lowest possible sharpness
[MEDIACMD::dwEnd](#) - Highest possible sharpness

2443
gsVidInGamma Set video gamma curve

2444cmdType::ctSetValue
2445[MEDIACMD::dwPosition](#) - Depends on cmdType::ctGetValue [MEDIACMD::dwStart](#) [MEDIACMD::dwEnd](#) (typically -32768->+32768)
2446cmdType::ctGetValue
2447[MEDIACMD::dwPosition](#) - Gamma curve weighting or offset
2448[MEDIACMD::dwStart](#) - Lowest possible sharpness
[MEDIACMD::dwEnd](#) - Highest possible sharpness

2449
gsVidInSignalFormat Video input signal format. May be incorrect depending on some hardware setups.

2450cmdType::ctSetValue
2451- Not supported, please use [gsSignalFormat](#) to set channel format to match input
2452cmdType::ctGetValue
[MEDIACMD::dwPosition](#) - [GS_SIGFORM_NTSC](#) [GS_SIGFORM_PAL](#) [GS_SIGFORM_CCIR_NTSC](#)
[GS_SIGFORM_CCIR_PAL](#) [GS_SIGFORM_1035i_30_260M](#) [GS_SIGFORM_1035i_30X_260M](#)

[GS_SIGFORM_1080i_30](#) [GS_SIGFORM_1080i_30X](#) [GS_SIGFORM_1080i_25](#)
[GS_SIGFORM_1080i_24](#) [GS_SIGFORM_1080i_24X](#) [GS_SIGFORM_1080_30](#)
[GS_SIGFORM_1080_30X](#) [GS_SIGFORM_1080_25](#) [GS_SIGFORM_1080_24](#) [GS_SIGFORM_1080_24X](#)
[GS_SIGFORM_720_60](#) [GS_SIGFORM_720_60X](#) [GS_SIGFORM_NOT_PRESENT](#)

2453

gsVidInQuality Set video transition sharpness

2454cmdType::ctSetValue

2455[MEDIACMD::dwPosition](#) - Depends on cmdType::ctGetValue [MEDIACMD::dwStart](#)
[MEDIACMD::dwEnd](#) (typically 0-100)

2456cmdType::ctGetValue

2457[MEDIACMD::dwPosition](#) - quality level

2458[MEDIACMD::dwStart](#) - Lowest possible quality from codec

[MEDIACMD::dwEnd](#) - Highest possible quality from codec

2459

gsVidSetup Main TBC - Setup (~brightness) Normal range: 0-65535 (0x0000-0xffff)

2460cmdType::ctSetValue

2461[MEDIACMD::dwPosition](#) - TBC Setup

2462cmdType::ctGetValue

2463[MEDIACMD::dwPosition](#) - TBC Setup

2464[MEDIACMD::dwStart](#) - Lowest possible value (usually 0)

[MEDIACMD::dwEnd](#) - Highest possible value (usually 65535)

2465

gsVidVideo Main TBC - Video (~contrast) Normal range: 0-65535 (0x0000-0xffff)

2466cmdType::ctSetValue

2467[MEDIACMD::dwPosition](#) - TBC Video

2468cmdType::ctGetValue

2469[MEDIACMD::dwPosition](#) - TBC Video

2470[MEDIACMD::dwStart](#) - Lowest possible value (usually 0)

[MEDIACMD::dwEnd](#) - Highest possible value (usually 65535)

2471

gsVidHue Main TBC - Hue (~color angle) degrees * 182. Normal range: 0-65520
(0x0000-0xffff)

2472cmdType::ctSetValue

2473[MEDIACMD::dwPosition](#) - TBC Hue

2474cmdType::ctGetValue

2475[MEDIACMD::dwPosition](#) - Hue

2476[MEDIACMD::dwStart](#) - Lowest possible value (usually 0)

[MEDIACMD::dwEnd](#) - Highest possible value (usually 65520)

2477

gsVidChroma Main TBC - Chroma (~saturation) Normal range: 0-65535 (0x0000-
0xffff)

2478cmdType::ctSetValue

2479[MEDIACMD::dwPosition](#) - TBC Chroma

2480cmdType::ctGetValue

2481[MEDIACMD::dwPosition](#) - TBC Chroma

2482 [MEDIACMD::dwStart](#) - Lowest possible value (usually 0)
[MEDIACMD::dwEnd](#) - Highest possible value (usually 65535)

2483

gsVidUChroma Main TBC - U Chroma or Cb or Y'CrCb Normal range: 0-65535 (0x0000-0xffff) Normally only affects the component video or D1 Serial paths.

2484 cmdType::ctSetValue

2485 [MEDIACMD::dwPosition](#) - TBC U (Cb) Chroma

2486 cmdType::ctGetValue

2487 [MEDIACMD::dwPosition](#) - TBC U (Cb) Chroma

2488 [MEDIACMD::dwStart](#) - Lowest possible value (usually 0)

[MEDIACMD::dwEnd](#) - Highest possible value (usually 65535)

2489

gsVidVChroma Main TBC - V Chroma or Cr or Y'CrCb Normal range: 0-65535 (0x0000-0xffff) Normally only affects the component video or D1 Serial paths.

2490 cmdType::ctSetValue

2491 [MEDIACMD::dwPosition](#) - TBC V (Cr) Chroma

2492 cmdType::ctGetValue

2493 [MEDIACMD::dwPosition](#) - TBC V (Cr) Chroma

2494 [MEDIACMD::dwStart](#) - Lowest possible value (usually 0)

[MEDIACMD::dwEnd](#) - Highest possible value (usually 65535)

2495

gsVidBandwidth Maximum channel bandwidth setting

2496 cmdType::ctSetValue

2497 [MEDIACMD::dwPosition](#) - Uses [GS_VIDBAND_STANDARD](#), [GS_VIDBAND_MEDIUM](#), [GS_VIDBAND_HIGH](#), [GS_VIDBAND_NOTCH](#)

2498 cmdType::ctGetValue

2499 [MEDIACMD::dwPosition](#) - Uses [GS_VIDBAND_STANDARD](#), [GS_VIDBAND_MEDIUM](#), [GS_VIDBAND_HIGH](#), [GS_VIDBAND_NOTCH](#)

[MEDIACMD::dwStart](#) - Bit array of allowable values as defined for dwPosition above

2500

gsVidBlackSetup Black type (NTSC only)

2501 cmdType::ctSetValue

2502 [MEDIACMD::dwPosition](#) - Uses [GS_VIDBLACK_SETUP](#), [GS_VIDBLACK_CRYSTAL](#), [GS_VIDBLACK_SUPER](#)

2503 cmdType::ctGetValue

2504 [MEDIACMD::dwPosition](#) - Uses [GS_VIDBLACK_SETUP](#), [GS_VIDBLACK_CRYSTAL](#), [GS_VIDBLACK_SUPER](#)

[MEDIACMD::dwStart](#) - Bit array of allowable values as defined for dwPosition above

2505

gsVidColor Remove color from signal path (black and white luminance data only)

2506 cmdType::ctSetValue

2507 [MEDIACMD::dwPosition](#) - If 1, signal will have no chroma, if 0, normal signal

2508 cmdType::ctGetValue

[MEDIACMD::dwPosition](#) - If 1, signal will have no chroma, if 0, normal signal

2509

gsVideoInputRouting Route Output video from to the target channel

2510cmdType::ctSetValue

2511[MEDIACMD::dwPosition](#) - -1 for no routing vvwChannel # to target

2512cmdType::ctGetValue

2513[MEDIACMD::dwPosition](#) - Routing Channel targeted

2514[MEDIACMD::lSpeed](#) - Route Audio

2515[MEDIACMD::dwVideoChannels](#) - Internal Use Only do not set

[MEDIACMD::dwAudioChannels](#) - Internal Use Only do not set

2516

gsVidOutSelect Select video output

2517cmdType::ctSetValue

2518[MEDIACMD::dwPosition](#) - Video output to use [GS_VIDSELECT_COMPOSITE](#),
[GS_VIDSELECT_COMPOSITE_2](#), [GS_VIDSELECT_SVIDEO](#),
[GS_VIDSELECT_COMPONENT_YUV](#), [GS_VIDSELECT_COMPONENT_YUV_M2](#),
[GS_VIDSELECT_COMPONENT_YUV_SMPTE](#), [GS_VIDSELECT_COMPONENT_RGB](#),
[GS_VIDSELECT_D1_SERIAL](#), [GS_VIDSELECT_D1_PARALLEL](#), [GS_VIDSELECT_SDTI](#),
[GS_VIDSELECT_NONE](#)

2519cmdType::ctGetValue

2520[MEDIACMD::dwPosition](#) - Current video output [GS_VIDSELECT_COMPOSITE](#),
[GS_VIDSELECT_COMPOSITE_2](#), [GS_VIDSELECT_SVIDEO](#),
[GS_VIDSELECT_COMPONENT_YUV](#), [GS_VIDSELECT_COMPONENT_YUV_M2](#),
[GS_VIDSELECT_COMPONENT_YUV_SMPTE](#), [GS_VIDSELECT_COMPONENT_RGB](#),
[7GS_VIDSELECT_D1_SERIAL](#), [GS_VIDSELECT_D1_PARALLEL](#), [GS_VIDSELECT_SDTI](#),
[GS_VIDSELECT_NONE](#)

[MEDIACMD::dwStart](#) - Supported video inputs (bit array using defines from dwPosition)

2521

gsVidOutGenlock Enable genlock (video black timing signal)

2522cmdType::ctSetValue

2523[MEDIACMD::dwPosition](#) - 1 using external ref genlock, 0 free running on internal clock (see
gsGetSetCmdValue::gsVidOutGenlockSource)

2524cmdType::ctGetValue

2525[MEDIACMD::dwPosition](#) - 1 using external ref genlock, 0 free running on internal clock (see
gsGetSetCmdValue::gsVidOutGenlockSource)

[MEDIACMD::dwStart](#) - If 1, external genlock supported

2526

gsVidOutGenlockSource Select genlock (video black timing signal) source

2527cmdType::ctSetValue

2528[MEDIACMD::dwPosition](#) - Genlock source to use [GS_LOCKSRC_NONE](#),
[GS_LOCKSRC_EXTIN](#), [GS_LOCKSRC_INPUT](#), [GS_LOCKSRC_CVBS](#) (composite video),
[GS_LOCKSRC_SVIDEO](#) (svhs), [GS_LOCKSRC_IN_Y](#) (y of component in),
[GS_LOCKSRC_SDI](#) (D1 Digital In)

2529cmdType::ctGetValue

2530[MEDIACMD::dwPosition](#) - Genlock source to use [GS_LOCKSRC_NONE](#),
[GS_LOCKSRC_EXTIN](#), [GS_LOCKSRC_INPUT](#), [GS_LOCKSRC_CVBS](#) (composite video),
[GS_LOCKSRC_SVIDEO](#) (svhs), [GS_LOCKSRC_IN_Y](#) (y of component in),
[GS_LOCKSRC_SDI](#) (D1 Digital In)

[MEDIACMD::dwStart](#) - Supported genlock inputs (bit array using defines from dwPosition)

2531

gsVidOutLockType Select genlock type (quality)

2532cmdType::ctSetValue

2533[MEDIACMD::dwPosition](#) - Genlock lock type use [GS_VIDLOCKTYPE_VTR](#) or [GS_VIDLOCKTYPE_BROADCAST](#)

2534cmdType::ctGetValue

2535[MEDIACMD::dwPosition](#) - Genlock lock type use [GS_VIDLOCKTYPE_VTR](#) or [GS_VIDLOCKTYPE_BROADCAST](#)

[MEDIACMD::dwStart](#) - Supported video inputs (bit array using defines from dwPosition)

2536

gsVidOutHPhase Horizontal output phase

2537cmdType::ctSetValue

2538[MEDIACMD::dwPosition](#) - Depends on cmdType::ctGetValue [MEDIACMD::dwStart](#) [MEDIACMD::dwEnd](#) (typically 0->65535 or -32768->32768)

2539cmdType::ctGetValue

2540[MEDIACMD::dwPosition](#) - Horizontal phase setting

2541[MEDIACMD::dwStart](#) - Lowest possible horizontal phase

[MEDIACMD::dwEnd](#) - Highest possible horizontal phase

2542

gsVidOutSubCarrier Video genlock subcarrier phase timing

2543cmdType::ctSetValue

2544[MEDIACMD::dwPosition](#) - Depends on cmdType::ctGetValue [MEDIACMD::dwStart](#) [MEDIACMD::dwEnd](#) (typically 0->65520 == degrees * 182)

2545cmdType::ctGetValue

2546[MEDIACMD::dwPosition](#) - Sub carrier phase setting

2547[MEDIACMD::dwStart](#) - Lowest possible sub carrier phase

[MEDIACMD::dwEnd](#) - Highest possible sub carrier phase

2548

gsVidOutCoring Digital output signal coring. Removal of low order bits to remove DAC aliasing

2549cmdType::ctSetValue

2550[MEDIACMD::dwPosition](#) - Remove bottom 0, 1 or 2 bits of digitized signal

2551cmdType::ctGetValue

2552[MEDIACMD::dwPosition](#) - Remove bottom 0, 1 or 2 bits of digitized signal

[MEDIACMD::dwStart](#) - Bit array of allowable values as defined for dwPosition above (0 always supported)

2553

gsVidOutPeaking Remove (smooth) 100% signal spikes on output

2554cmdType::ctSetValue

2555[MEDIACMD::dwPosition](#) - 0 leave signal intact, 1 smooth

2556cmdType::ctGetValue

2557[MEDIACMD::dwPosition](#) - 0 leave signal intact, 1 smooth

[MEDIACMD::dwStart](#) - Bit array of allowable values as defined for dwPosition above (0 always supported)

2558

gsVidOutAdjust1 Generic advanced adjustment 1 (hardware dependent)

2559cmdType::ctSetValue

2560[MEDIACMD::dwPosition](#) - Depends on cmdType::ctGetValue [MEDIACMD::dwStart](#)
[MEDIACMD::dwEnd](#) (typically 0->65535 or -32768->32768)

2561cmdType::ctGetValue

2562[MEDIACMD::dwPosition](#) - Adjust 1 setting

2563[MEDIACMD::dwStart](#) - Lowest possible adjust 1 setting

[MEDIACMD::dwEnd](#) - Highest possible adjust 1 setting

2564

gsVidOutAdjust2 Generic advanced adjustment 2 (hardware dependent)

2565cmdType::ctSetValue

2566[MEDIACMD::dwPosition](#) - Depends on cmdType::ctGetValue [MEDIACMD::dwStart](#)
[MEDIACMD::dwEnd](#) (typically 0->65535 or -32768->32768)

2567cmdType::ctGetValue

2568[MEDIACMD::dwPosition](#) - Adjust 2 setting

2569[MEDIACMD::dwStart](#) - Lowest possible adjust 2 setting

[MEDIACMD::dwEnd](#) - Highest possible adjust 2 setting

2570

gsVidOutGenlockDelay Genlock output delay (not currently used)

2571cmdType::ctSetValue

2572[MEDIACMD::dwPosition](#) - Depends on cmdType::ctGetValue [MEDIACMD::dwStart](#)
[MEDIACMD::dwEnd](#) (typically 0->65535 or -32768->32768)

2573cmdType::ctGetValue

2574[MEDIACMD::dwPosition](#) - Genlock timing delay

2575[MEDIACMD::dwStart](#) - Lowest possible delay

[MEDIACMD::dwEnd](#) - Highest possible delay

2576

gsVidOutLockSignalFormat Video output genlock input signal format. May be incorrect depending on some hardware setups.

2577cmdType::ctSetValue

2578- Not supported, please use [gsSignalFormat](#) to set channel format to match input

2579cmdType::ctGetValue

[MEDIACMD::dwPosition](#) - [GS_SIGFORM_NTSC](#) [GS_SIGFORM_PAL](#) [GS_SIGFORM_CCIR_NTSC](#)
[GS_SIGFORM_CCIR_PAL](#) [GS_SIGFORM_1035i_30_260M](#) [GS_SIGFORM_1035i_30X_260M](#)
[GS_SIGFORM_1080i_30](#) [GS_SIGFORM_1080i_30X](#) [GS_SIGFORM_1080i_25](#)
[GS_SIGFORM_1080i_24](#) [GS_SIGFORM_1080i_24X](#) [GS_SIGFORM_1080_30](#)
[GS_SIGFORM_1080_30X](#) [GS_SIGFORM_1080_25](#) [GS_SIGFORM_1080_24](#) [GS_SIGFORM_1080_24X](#)
[GS_SIGFORM_720_60](#) [GS_SIGFORM_720_60X](#) [GS_SIGFORM_NOT_PRESENT](#)

2580

gsVidOutDisableDualLink When video input is in DualLink, the out switches to dual link to. If this is set, then the output will stay in single link and convert the dual link 4:4:4 to 4:2:2

2581cmdType::ctSetValue

2582[MEDIACMD::dwPosition](#) - [GS_TRUE](#), [GS_FALSE](#) match input

2583cmdType::ctGetValue

[MEDIACMD::dwPosition](#) - [GS_TRUE](#), [GS_FALSE](#)

2584

gsVidOutReferenceWipeMix Set the output of the Kona to show a wipe or dissolve against the current frame

2585cmdType::ctSetValue

2586[MEDIACMD::dwPosition](#) - 0=dissolve, 1=Wipe

2587[MEDIACMD::dwStart](#) - Wipe Type, 0=horiz,1=vert,2=upperright,3=upperleft

2588[MEDIACMD::dwEnd](#) - Wipe amount, 0..65535 (0..100%) where it is a percentage of stored frame (e.g. 0=showinput,65535=showframe)

2589cmdType::ctGetValue

[MEDIACMD::dwPosition](#) - 0=dissolve, 1=Wipe

2590

gsDisableGenlockForInfiniteLoop Disable genlocking of board to allow out->in connection for capture of output

2591cmdType::ctSetValue

2592[MEDIACMD::dwPosition](#) - 1 to disable, 0 default

2593cmdType::ctGetValue

[MEDIACMD::dwPosition](#) - 1 to disable, 0 default

2594

gsCompChVerticalRes Size of picture Y (Vertical)

2595cmdType::ctSetValue

2596[MEDIACMD::dwPosition](#) - Vertical size of video frame

2597cmdType::ctGetValue

[MEDIACMD::dwPosition](#) - Vertical size of video frame

2598

gsMpegVerticalRes Alias for [gsCompChVerticalRes](#).

gsCompChHorizontalRes Size of picture X (Horizontal)

2599cmdType::ctSetValue

2600[MEDIACMD::dwPosition](#) - Horizontal size of video frame

2601cmdType::ctGetValue

[MEDIACMD::dwPosition](#) - Horizontal size of video frame

2602

gsMpegHorizontalRes Alias for [gsCompChHorizontalRes](#).

gsCompChChromaFormat Chroma type 4:0:0, 4:2:0, 4:2:2, 4:4:4

2603cmdType::ctSetValue

2604[MEDIACMD::dwPosition](#) - One of [GS_MPEG_CHROMA_FORMAT_420](#), [GS_MPEG_CHROMA_FORMAT_422](#), [GS_MPEG_CHROMA_FORMAT_444](#)

2605cmdType::ctGetValue

[MEDIACMD::dwPosition](#) - One of [GS_MPEG_CHROMA_FORMAT_420](#), [GS_MPEG_CHROMA_FORMAT_422](#), [GS_MPEG_CHROMA_FORMAT_444](#)

2606

gsMpegChromaFormat Alias for [gsCompChChromaFormat](#).

gsCompChDCPrecision DC Precision (mostly MPEG) 8..12

2607cmdType::ctSetValue

2608MEDIACMD::dwPosition - [GS_MPEG_DC_PRECISION_8](#), [GS_MPEG_DC_PRECISION_9](#),
[GS_MPEG_DC_PRECISION_10](#), [GS_MPEG_DC_PRECISION_11](#)

2609cmdType::ctGetValue

[MEDIACMD::dwPosition](#) - [GS_MPEG_DC_PRECISION_8](#), [GS_MPEG_DC_PRECISION_9](#),
[GS_MPEG_DC_PRECISION_10](#), [GS_MPEG_DC_PRECISION_11](#)

2610

gsMpegDCPrecision Alias for [gsCompChDCPrecision](#).

gsCompChAspectRatio Video signal aspect ratio

2611cmdType::ctSetValue

2612MEDIACMD::dwPosition - [GS_ASPECT_RATIO_SQUARE](#), [GS_ASPECT_RATIO_4x3](#),
[GS_ASPECT_RATIO_16x9](#), [GS_ASPECT_RATIO_2_21x1](#)

2613cmdType::ctGetValue

[MEDIACMD::dwPosition](#) - [GS_ASPECT_RATIO_SQUARE](#), [GS_ASPECT_RATIO_4x3](#),
[GS_ASPECT_RATIO_16x9](#), [GS_ASPECT_RATIO_2_21x1](#)

2614

gsMpegAspectRatio Alias for [gsCompChAspectRatio](#).

gsCompChStandard MPEG file stream standard

2615cmdType::ctSetValue

2616MEDIACMD::dwPosition - [GS_MPEG_STANDARD_SYSTEM](#),
[GS_MPEG_STANDARD_PROGRAM](#), [GS_MPEG_STANDARD_TRANSPORT](#),
[GS_MPEG_STANDARD_ELEMENTARY](#)

2617cmdType::ctGetValue

[MEDIACMD::dwPosition](#) - [GS_MPEG_STANDARD_SYSTEM](#), [GS_MPEG_STANDARD_PROGRAM](#),
[GS_MPEG_STANDARD_TRANSPORT](#), [GS_MPEG_STANDARD_ELEMENTARY](#)

2618

gsMpegStandard Alias for [gsCompChStandard](#).

gsCompChLanguageCode Video/Audio Language Code

2619cmdType::ctSetValue

2620MEDIACMD::dwPosition - [GS_MPEG_LANGUAGE_ENGLISH](#), etc

2621cmdType::ctGetValue

[MEDIACMD::dwPosition](#) - [GS_MPEG_LANGUAGE_ENGLISH](#), etc

2622

gsMpegLanguageCode Alias for [gsCompChLanguageCode](#).

gsCompChCCFormat Closed Captioning Format

2623cmdType::ctSetValue

2624MEDIACMD::dwPosition - [GS_MPEG_CC_FORMAT_CCUBE](#),
[GS_MPEG_CC_FORMAT_ATSC](#), [GS_MPEG_CC_FORMAT_CCUBE_REORDER](#),
[GS_MPEG_CC_FORMAT_ATSC_REORDER](#)

2625cmdType::ctGetValue

[MEDIACMD::dwPosition](#) - [GS_MPEG_CC_FORMAT_CCUBE](#), [GS_MPEG_CC_FORMAT_ATSC](#),
[GS_MPEG_CC_FORMAT_CCUBE_REORDER](#), [GS_MPEG_CC_FORMAT_ATSC_REORDER](#)

2626

gsMpegCCFormat Alias for [gsCompChCCFormat](#).

gsCompChConcealmentVector MPEG Concealment Vector

2627cmdType::ctSetValue

2628[MEDIACMD::dwPosition](#) - Not sure - Argus Encoder

2629cmdType::ctGetValue

[MEDIACMD::dwPosition](#) - Not sure - Argus Encoder

2630

gsMpegConcealmentVector Alias for [gsCompChConcealmentVector](#).

gsCompChClosedGop Set encoding to closed GOP or open GOP

2631cmdType::ctSetValue

2632[MEDIACMD::dwPosition](#) - 0 = open GOP, 1 = closed GOP

2633cmdType::ctGetValue

[MEDIACMD::dwPosition](#) - 0 = open GOP, 1 = closed GOP

2634

gsMpegClosedGop Alias for [gsCompChClosedGop](#).

gsCompChAdjustGopTC Set the next GOP start time code value

2635cmdType::ctSetValue

2636[MEDIACMD::dwPosition](#) - Time code in frames (used def tctype)

2637cmdType::ctGetValue

[MEDIACMD::dwPosition](#) - Time code in frames (used def tctype)

2638

gsMpegAdjustGopTC Alias for [gsCompChAdjustGopTC](#).

gsCompChAltCoEffTable Set MPEG encoder to use alternate co-efficient tables

2639cmdType::ctSetValue

2640[MEDIACMD::dwPosition](#) - 1/0

2641cmdType::ctGetValue

[MEDIACMD::dwPosition](#) - 1/0

2642

gsMpegAltCoEffTable Alias for [gsCompChAltCoEffTable](#).

gsCompChNonLinearQuant Set encoder to use non linear quantization

2643cmdType::ctSetValue

2644[MEDIACMD::dwPosition](#) - 1/0

2645cmdType::ctGetValue

[MEDIACMD::dwPosition](#) - 1/0

2646

gsMpegNonLinearQuant Alias for [gsCompChNonLinearQuant](#).

gsCompChMuxRate Set the multiplexer (overall) bit rate

2647cmdType::ctSetValue

2648[MEDIACMD::dwPosition](#) - Bits per second

2649cmdType::ctGetValue

[MEDIACMD::dwPosition](#) - Bits per second

2650

gsMpegMuxRate Alias for [gsCompChMuxRate](#).

gsCompChAudPacketSize Audio packet size

2651cmdType::ctSetValue
2652[MEDIACMD::dwPosition](#) - Size of an audio packet in unsigned chars
2653cmdType::ctGetValue
[MEDIACMD::dwPosition](#) - Size of an audio packet in unsigned chars

2654
gsMpegAudPacketSize Alias for [gsCompChAudPacketSize](#).
gsCompChVidPacketSize Video packet size

2655cmdType::ctSetValue
2656[MEDIACMD::dwPosition](#) - Size of a video packet in unsigned chars
2657cmdType::ctGetValue
[MEDIACMD::dwPosition](#) - Size of a video packet in unsigned chars

2658
gsMpegVidPacketSize Alias for [gsCompChVidPacketSize](#).
gsCompChAudioStreamID Stream ID for AUDIO 0xc0 (0x1c0)

2659cmdType::ctSetValue
2660[MEDIACMD::dwPosition](#) - Audio Stream ID
2661cmdType::ctGetValue
[MEDIACMD::dwPosition](#) - Audio Stream ID

2662
gsMpegAudioStreamID Alias for [gsCompChVideoStreamID](#).
gsCompChVideoStreamID Stream ID for VIDEO 0xe0 (0x1e0)

2663cmdType::ctSetValue
2664[MEDIACMD::dwPosition](#) - Video Stream ID
2665cmdType::ctGetValue
[MEDIACMD::dwPosition](#) - Video Stream ID

2666
gsMpegVideoStreamID Alias for [gsCompChVideoStreamID](#).
gsCompChAudioStreamPID Program ID of the video stream within a transport container

2667cmdType::ctSetValue
2668[MEDIACMD::dwPosition](#) - Video program id (PID)
2669cmdType::ctGetValue
[MEDIACMD::dwPosition](#) - Video program id (PID)

2670
gsMpegAudioStreamPID Alias for [gsCompChAudioStreamPID](#).
gsCompChVideoStreamPID Program ID of the audio stream within a transport container

2671cmdType::ctSetValue

2672 [MEDIACMD::dwPosition](#) - Audio program id (PID)
 2673 cmdType::ctGetValue

[MEDIACMD::dwPosition](#) - Audio program id (PID)
 2674
gsMpegVideoStreamPID Alias for [gsCompChVideoStreamPID](#).
gsCompChAllowSettings Allow settings to be changed. Used to determine if settings can be changed (on the fly, without restart),

2675 cmdType::ctSetValue
 2676 [MEDIACMD::dwPosition](#) - 0 disable changes, 1 allow changes
 2677 cmdType::ctGetValue

[MEDIACMD::dwPosition](#) - 0 disable changes, 1 allow changes
 2678
gsMpegAllowSettings Alias for [gsCompChAllowSettings](#).
gsCompChFourCC Fourcc code for compression. Set one video or audio channel set to return fourcc code in dwPosition

2679 cmdType::ctSetValue - See fccDef.h for know fourcc
 2680
 cmdType::ctGetValue - See fccDef.h for know fourcc

 2681
gsCompChBitCount bit count for compression 8 / 10 / 24 / 32 set to return bit count in dwPosition

2682 cmdType::ctSetValue - See fccDef.h for know fourcc
 2683
 cmdType::ctGetValue - See fccDef.h for know fourcc

 2684
gsCompChSizeImage Size of each image in unsigned chars set to return fourcc code in dwPosition

2685 cmdType::ctSetValue - See fccDef.h for know fourcc
 2686
 cmdType::ctGetValue - See fccDef.h for know fourcc

 2687
gsCompChRate rate of frame set to return fourcc code in dwPosition

2688 cmdType::ctSetValue - See fccDef.h for know fourcc
 2689
 cmdType::ctGetValue - See fccDef.h for know fourcc

 2690
gsCompChScale Scale for frame rate set to return fourcc code in dwPosition

2691 cmdType::ctSetValue - See fccDef.h for know fourcc
 2692
 cmdType::ctGetValue - See fccDef.h for know fourcc

 2693
gsCompChPitch unsigned chars per video line set to return fourcc code in dwPosition

2694cmdType::ctSetValue - See fccDef.h for know fourcc
2695
cmdType::ctGetValue - See fccDef.h for know fourcc

2696
gsVideoEncodeFormat Encoding compression format i.e. avi mov dpx set to return fourcc code in dwPosition

2697cmdType::ctSetValue - See defines for file types
2698MEDIACMD::dwPosition - [VIDEOWRITETYPE_AVI](#), [VIDEOWRITETYPE_MOV](#),
[VIDEOWRITETYPE_WMV](#), [VIDEOWRITETYPE_GEN](#),
[VIDEOWRITETYPE_SONY_HD_MXF](#), [VIDEOWRITETYPE_SONY_SR_MXF](#),
[VIDEOWRITETYPE_HDR](#), [VIDEOWRITETYPE_YUV](#), [VIDEOWRITETYPE_RAW](#),
[VIDEOWRITETYPE_TGA](#), [VIDEOWRITETYPE_BMP](#), [VIDEOWRITETYPE_TIFF](#),
[VIDEOWRITETYPE_DPX](#), [VIDEOWRITETYPE_AVCI_MXF](#) [VIDEOWRITETYPE_MPG](#),
[VIDEOWRITETYPE_4224](#), [VIDEOWRITETYPE_SONY_MXF](#),
[VIDEOWRITETYPE_P2_MXF](#), [VIDEOWRITETYPE_AVID_MXF](#)
[VIDEOWRITETYPE_ARRI](#), [VIDEOWRITETYPE_JP2K](#), [VIDEOWRITETYPE_OP1a_MXF](#),
[VIDEOWRITETYPE_DCP_MXF](#), [VIDEOWRITETYPE_TS](#), [VIDEOWRITETYPE_MP4](#),
[VIDEOWRITETYPE_FLASH](#), [VIDEOWRITETYPE_DNG](#)

2699
cmdType::ctGetValue - See defines for file types

2700
2701MEDIACMD::dwPosition - [VIDEOWRITETYPE_AVI](#), [VIDEOWRITETYPE_MOV](#),
[VIDEOWRITETYPE_WMV](#), [VIDEOWRITETYPE_GEN](#),
[VIDEOWRITETYPE_SONY_HD_MXF](#), [VIDEOWRITETYPE_SONY_SR_MXF](#),
[VIDEOWRITETYPE_HDR](#), [VIDEOWRITETYPE_YUV](#), [VIDEOWRITETYPE_RAW](#),
[VIDEOWRITETYPE_TGA](#), [VIDEOWRITETYPE_BMP](#), [VIDEOWRITETYPE_TIFF](#),
[VIDEOWRITETYPE_DPX](#), [VIDEOWRITETYPE_AVCI_MXF](#) [VIDEOWRITETYPE_MPG](#),
[VIDEOWRITETYPE_4224](#), [VIDEOWRITETYPE_SONY_MXF](#),
[VIDEOWRITETYPE_P2_MXF](#), [VIDEOWRITETYPE_AVID_MXF](#)
[VIDEOWRITETYPE_ARRI](#), [VIDEOWRITETYPE_JP2K](#), [VIDEOWRITETYPE_OP1a_MXF](#),
[VIDEOWRITETYPE_DCP_MXF](#), [VIDEOWRITETYPE_TS](#), [VIDEOWRITETYPE_MP4](#),
[VIDEOWRITETYPE_FLASH](#), [VIDEOWRITETYPE_DNG](#)

2702MEDIACMD::dwStart - Bit array of available file formats
gsAudioEncodeFormat Encoding compression format i.e. WAV, AIFF set to return fourcc code in dwPosition

2703cmdType::ctSetValue - See defines for file types
2704MEDIACMD::dwPosition - The file type and the internal format
2705AUDIOWRITETYPE_INTERNAL, AUDIOWRITETYPE_WAVE, AUDIOWRITETYPE_AIFF
AUDIOWRITETYPE_MONO, AUDIOWRITETYPE_STEREO, AUDIOWRITETYPE_MULTI

2706
2707cmdType::ctGetValue - See fccDef.h for know fourcc
2708MEDIACMD::dwPosition - The file type and the internal format
2709AUDIOWRITETYPE_INTERNAL, AUDIOWRITETYPE_WAVE, AUDIOWRITETYPE_AIFF
AUDIOWRITETYPE_MONO, AUDIOWRITETYPE_STEREO, AUDIOWRITETYPE_MULTI

MEDIACMD::dwStart - Bit array of available file formats

2710
gsCompChannelChangeMs last compression change ms for updating the clip bin set to return fourcc code in dwPosition

2711cmdType::ctSetValue - See fccDef.h for know fourcc
2712
cmdType::ctGetValue - See fccDef.h for know fourcc

2713
gsAlphaChromaSource single link, dual link or alpha set to return fourcc code in dwPosition

2714cmdType::ctSetValue - See fccDef.h for know fourcc
2715
cmdType::ctGetValue - See fccDef.h for know fourcc

2716
gsCompressionType RGBA BGRA ycber set to return fourcc code in dwPosition

2717cmdType::ctSetValue - See fccDef.h for know fourcc
2718
cmdType::ctGetValue - See fccDef.h for know fourcc

2719
gsVideoStandard

2720cmdType::ctSetValue -
2721
cmdType::ctGetValue -

2722
gsResetChannel Reset the channel to the new setup

2723cmdType::ctSetValue -
2724
cmdType::ctGetValue -

2725
gsEnableHDSDFormat

2726cmdType::ctSetValue -
2727
cmdType::ctGetValue -

2728
gsVBlankEnable Enable capture of vertical blank?

2729cmdType::ctSetValue -
2730[MEDIACMD::dwPosition](#) - 1 (bit_0) capture vertical blank, 2 (bit_1) save vertical blank
2731
2732cmdType::ctGetValue -
[MEDIACMD::dwPosition](#) - 1 (bit_0) capture vertical blank, 2 (bit_1) save vertical blank

2733
gsLUTEnable Enable/Disable LUTs

2734cmdType::ctSetValue -
2735[MEDIACMD::dwPosition](#) - 1 (bit_0) play enable, 2 (bit_1) linear / log if not set, 4 (bit_2) record enable

2736
 2737cmdType::ctGetValue -
[MEDIACMD::dwPosition](#) - 1 (bit_0) play enable, 2 (bit_1) linear / log if not set, 4 (bit_2) record enable

2738
gsAudioFileType

2739cmdType::ctSetValue -
 2740
 cmdType::ctGetValue -

2741
gsAudioBitSize

2742cmdType::ctSetValue -
 2743
 cmdType::ctGetValue -

2744
gsAudioFrequency

2745cmdType::ctSetValue -
 2746
 cmdType::ctGetValue -

2747
gsEnableOverlappedWrites Enabled/disable/ mediafile overlapped writes

2748cmdType::ctGetValue
 2749[MEDIACMD::dwPosition](#) - Enabled 0,1
 2750cmdType::ctSetValue
[MEDIACMD::dwPosition](#) - Enabled 0,1

2751
gsMatchOutputToClip Match video output to current clip settings

2752cmdType::ctGetValue
 2753[MEDIACMD::dwPosition](#) - Enabled 0,1
 2754cmdType::ctSetValue
[MEDIACMD::dwPosition](#) - Enabled 0,1

2755
gsAllowIndependentChanConfig Allow each channel to be configure separately for file type, bit depth, etc.

2756cmdType::ctGetValue
 2757[MEDIACMD::dwPosition](#) - Enabled 0,1
 2758cmdType::ctSetValue
[MEDIACMD::dwPosition](#) - Enabled 0,1

2759
gsSignalFormat Channel Compression format

2760cmdType::ctSetValue
 2761[MEDIACMD::dwPosition](#) - [GS_SIGFORM_CCIR_NTSC](#) [GS_SIGFORM_CCIR_PAL](#)

[GS_SIGFORM_1035i_30_260M](#) [GS_SIGFORM_1035i_30X_260M](#) [GS_SIGFORM_1080i_30](#)
[GS_SIGFORM_1080i_30X](#) [GS_SIGFORM_1080i_25](#) [GS_SIGFORM_1080i_24](#)
[GS_SIGFORM_1080i_24X](#) [GS_SIGFORM_1080_30](#) [GS_SIGFORM_1080_30X](#)
[GS_SIGFORM_1080_25](#) [GS_SIGFORM_1080_24](#) [GS_SIGFORM_1080_24X](#)
[GS_SIGFORM_720_60](#) [GS_SIGFORM_720_60X](#) [GS_SIGFORM_CUSTOM](#)

2762cmdType::ctGetValue

2763MEDIACMD::dwPosition - [GS_SIGFORM_NTSC](#) [GS_SIGFORM_PAL](#)
[GS_SIGFORM_CCIR_NTSC](#) [GS_SIGFORM_CCIR_PAL](#) [GS_SIGFORM_1035i_30_260M](#)
[GS_SIGFORM_1035i_30X_260M](#) [GS_SIGFORM_1080i_30](#) [GS_SIGFORM_1080i_30X](#)
[GS_SIGFORM_1080i_25](#) [GS_SIGFORM_1080i_24](#) [GS_SIGFORM_1080i_24X](#)
[GS_SIGFORM_1080_30](#) [GS_SIGFORM_1080_30X](#) [GS_SIGFORM_1080_25](#)
[GS_SIGFORM_1080_24](#) [GS_SIGFORM_1080_24X](#) [GS_SIGFORM_720_60](#)
[GS_SIGFORM_720_60X](#) [GS_SIGFORM_CUSTOM](#)

[MEDIACMD::dwStart](#) - Bit array of supported types

2764

gsCompType Channel Compression format/type

2765cmdType::ctSetValue

2766MEDIACMD::dwPosition -

2767[GS_COMPTYPE_SOFTWARE](#) Software passed codec on main processor

2768[GS_COMPTYPE_BAYER](#) Raw Bayer frame Arri LMP Weisscam Phantom

2769[GS_COMPTYPE_H264](#) h264 I or IBP

2770[GS_COMPTYPE_JPEG2000](#) - JPEG 2000

2771[GS_COMPTYPE_CINEFORM_3D](#) CineForm 3D Wavelet codec

2772[GS_COMPTYPE_BGR](#) BGR uncompressed

2773[GS_COMPTYPE_HDCAM](#) HDCam 10 bit 4:4:4/4:2:2 MPEG-4

2774[GS_COMPTYPE_MPEG](#) MPEG 2 long GOP hardware compatible codec

2775[GS_COMPTYPE_DV25](#) Hardware DV25, DVCPRO, DVCPRO25

2776[GS_COMPTYPE_DV50](#) Hardware DV50, DVCPRO50

2777[GS_COMPTYPE_DVSD](#) Hardware Standard DV Bluebook, DVPRO, DVSD

2778[GS_COMPTYPE_DV100](#) High Def DV codec

2779[GS_COMPTYPE_CINEFORM](#) CineForm normal 2D compression

2780[GS_COMPTYPE_YCRCB_V210](#) 10 Bit YCbCr

2781[GS_COMPTYPE_RGB](#) RGB 24 bit

2782[GS_COMPTYPE_DNxHD](#) Avid DNxHD

2783[GS_COMPTYPE_AVCi](#) AVC Intra Panasonic

2784[GS_COMPTYPE_PRORES](#) Apple ProRes

2785[GS_COMPTYPE_BGRA_INVERT](#) BGRA (TGA) vertically inverted

2786[GS_COMPTYPE_DPX_YCBCR10](#) DPX YCbCr 10 bit

2787[GS_COMPTYPE_ARGB](#) ARGB

2788[GS_COMPTYPE_RGBA](#) RGBA

2789[GS_COMPTYPE_ABGR](#) Uncompressed A BGR - TIFF

2790[GS_COMPTYPE_BGRA](#) Uncompressed BGR A - BMP/TGA

2791[GS_COMPTYPE_YCRCB_422](#) Uncompressed Y'CrCb 4:2:2 (DVS, VG)

2792[GS_COMPTYPE_YCRCB_422A](#) Uncompressed Y'CrCb 4:2:2A (DVS, Dual VG)

2793[GS_COMPTYPE_YCRCB_444](#) Uncompressed Y'CrCb 4:4:4 (DVS, Dual VG)

2794[GS_COMPTYPE_YCRCB_444A](#) Uncompressed Y'CrCb 4:4:4A (DVS, Dual VG)

2795[GS_COMPTYPE_STEREO](#) Uncompressed Y'CrCb 4:4:4A (DVS, Dual VG) or 3D 8, 10, 30 or
32 bit

2796[GS_COMPTYPE_YCRCB_420](#) Uncompressed Y'CrCb 4:2:0

2797[GS_COMPTYPE_DPX_RGB10](#) DPX 10 bit rgb

2798[GS_COMPTYPE_ALT](#) Use as generic alternative for use through AVCodec

2799cmdType::ctGetValue

2800MEDIACMD::dwPosition -

2801[GS_COMPTYPE_SOFTWARE](#) Software passed codec on main processor

2802[GS_COMPTYPE_BAYER](#) Raw Bayer frame Arri LMP Weisscam Phantom
 2803[GS_COMPTYPE_H264](#) h264 I or IBP
 2804[GS_COMPTYPE_JPEG2000](#) - JPEG 2000
 2805[GS_COMPTYPE_CINEFORM_3D](#) CineForm 3D Wavelet codec
 2806[GS_COMPTYPE_BGR](#) BGR uncompressed
 2807[GS_COMPTYPE_HDCAM](#) HDCam 10 bit 4:4:4/4:2:2 MPEG-4
 2808[GS_COMPTYPE_MPEG](#) MPEG 2 long GOP hardware compatible codec
 2809[GS_COMPTYPE_DV25](#) Hardware DV25, DVCPRO, DVCPRO25
 2810[GS_COMPTYPE_DV50](#) Hardware DV50, DVCPRO50
 2811[GS_COMPTYPE_DVSD](#) Hardware Standard DV Bluebook, DVPRO, DVSD
 2812[GS_COMPTYPE_DV100](#) High Def DV codec
 2813[GS_COMPTYPE_CINEFORM](#) CineForm normal 2D compression
 2814[GS_COMPTYPE_YCRCB_V210](#) 10 Bit YCbCr
 2815[GS_COMPTYPE_RGB](#) RGB 24 bit
 2816[GS_COMPTYPE_DNxHD](#) Avid DNxHD
 2817[GS_COMPTYPE_AVCi](#) AVC Intra Panasonic
 2818[GS_COMPTYPE_PRORES](#) Apple ProRes
 2819[GS_COMPTYPE_BGRA_INVERT](#) BGRA (TGA) vertically inverted
 2820[GS_COMPTYPE_DPX_YCBCR10](#) DPX YCbCr 10 bit
 2821[GS_COMPTYPE_ARGB](#) ARGB
 2822[GS_COMPTYPE_RGBA](#) RGBA
 2823[GS_COMPTYPE_ABGR](#) Uncompressed A BGR - TIFF
 2824[GS_COMPTYPE_BGRA](#) Uncompressed BGR A - BMP/TGA
 2825[GS_COMPTYPE_YCRCB_422](#) Uncompressed Y'CrCb 4:2:2 (DVS, VG)
 2826[GS_COMPTYPE_YCRCB_422A](#) Uncompressed Y'CrCb 4:2:2A (DVS, Dual VG)
 2827[GS_COMPTYPE_YCRCB_444](#) Uncompressed Y'CrCb 4:4:4 (DVS, Dual VG)
 2828[GS_COMPTYPE_YCRCB_444A](#) Uncompressed Y'CrCb 4:4:4A (DVS, Dual VG)
 2829[GS_COMPTYPE_STEREO](#) Uncompressed Y'CrCb 4:4:4A (DVS, Dual VG) or 3D 8, 10, 30 or 32 bit
 2830[GS_COMPTYPE_YCRCB_420](#) Uncompressed Y'CrCb 4:2:0
 2831[GS_COMPTYPE_DPX_RGB10](#) DPX 10 bit rgb
 2832[GS_COMPTYPE_ALT](#) Use as generic alternative for use through AVCodec
[MEDIACMD::dwStart](#) - Bit array of supported types

2833

gsCompRateSize Compression setting by total throughput

2834cmdType::ctSetValue

2835[MEDIACMD::dwPosition](#) - Size of compressed stream in kilounsigned chars per second

2836cmdType::ctGetValue

2837[MEDIACMD::dwPosition](#) - Size of compressed stream in kilounsigned chars per second

2838[MEDIACMD::dwStart](#) - Smallest size possible

[MEDIACMD::dwEnd](#) - Largest size possible

2839

gsCompRateRatio Compression setting by compression ratio

2840cmdType::ctSetValue

2841[MEDIACMD::dwPosition](#) - Ratio * 100 (e.g. 2:1 = 200)

2842cmdType::ctGetValue

2843[MEDIACMD::dwPosition](#) - Ratio * 100 (e.g. 2:1 = 200)

2844[MEDIACMD::dwStart](#) - Smallest available ration * 100

[MEDIACMD::dwEnd](#) - Largest available ration * 100

2845

gsCompRatePercent Compression setting by compression percentage of original size

2846cmdType::ctSetValue
2847[MEDIACMD::dwPosition](#) - Percentage * 100 (e.g. 50% compression = 5000)
2848cmdType::ctGetValue
2849[MEDIACMD::dwPosition](#) - Percentage * 100 (e.g. 50% compression = 5000)
2850[MEDIACMD::dwStart](#) - Smallest available percentage (usually 0)
[MEDIACMD::dwEnd](#) - Largest available percentage (usually 10000)

2851
gsCompGOPSize Number of frames per 'group of pictures'. For MPEG compression as well as defining keyframe interval for Cinepac, Indeo, MPEG-4, etc.

2852cmdType::ctSetValue
2853[MEDIACMD::dwPosition](#) - Number of frames between keyframes of MPEG 'GOP' frame length
2854cmdType::ctGetValue
2855[MEDIACMD::dwPosition](#) - Number of frames between keyframes of MPEG 'GOP' frame length
2856[MEDIACMD::dwStart](#) - Minimum possible size of group of pictures (usually 0)
[MEDIACMD::dwEnd](#) - Largest possible size of group of pictures (up to 10000 for MPEG 4)

2857
gsCompIFactor Number of I Frame elements per GOP

2858cmdType::ctSetValue
2859Not Supported
2860cmdType::ctGetValue
[MEDIACMD::dwPosition](#) - Kilounsigned chars available on drive

2861
gsCompBFactor Number of B Frame elements per GOP

2862cmdType::ctSetValue
2863Not Supported
2864cmdType::ctGetValue
[MEDIACMD::dwPosition](#) - Kilounsigned chars available on drive

2865
gsCompPFactor Number of P Frame elements per GOP

2866cmdType::ctSetValue
2867Not Supported
2868cmdType::ctGetValue
[MEDIACMD::dwPosition](#) - Kilounsigned chars available on drive

2869
gsCompRefPeriod Reference period to determine amount and order of P and B frames

2870cmdType::ctSetValue
2871Not Supported
2872cmdType::ctGetValue
[MEDIACMD::dwPosition](#) - Kilounsigned chars available on drive

2873
gsTotalStorageAvail Total storage available on current recording drive in kilo unsigned chars

2874cmdType::ctSetValue
2875Not Supported
2876cmdType::ctGetValue
[MEDIACMD::dwPosition](#) - Kilounsigned chars available on drive

2877
gsTotalStorageFree Total storage free on current recording drive in kilo unsigned chars

2878cmdType::ctSetValue
2879Not Supported
2880cmdType::ctGetValue
[MEDIACMD::dwPosition](#) - Kilounsigned chars free on drive

2881
gsTotalTimeAvail Total recording time available on current recording drive at current compression level

2882cmdType::ctSetValue
2883Not Supported
2884cmdType::ctGetValue
[MEDIACMD::dwPosition](#) - Number of frames available to record to

2885
gsTotalTimeFree Total recording time free on current recording drive at current compression level

2886cmdType::ctSetValue
2887Not Supported
2888cmdType::ctGetValue
[MEDIACMD::dwPosition](#) - Number of frames free to record to

2889
gsVtrType VTR emulation ID type

2890cmdType::ctSetValue
2891[MEDIACMD::dwPosition](#) - Any unsigned short VTR ID - See Control key in registry docs and LocalConfig.exe
2892cmdType::ctGetValue
2893[MEDIACMD::dwPosition](#) - Any unsigned short VTR ID - See Control key in registry docs and LocalConfig.exe
[MEDIACMD::arbID](#) - String description of VTR (short (8 char), then long)

2894
gsHSDITransferType HSDSI signal wrapped in HD-SDI (for Arri, Weisscam, LMP, SDTI, etc)

2895cmdType::ctSetValue
2896[MEDIACMD::dwPosition](#) - HSDSI bayer transfer type: [GS_HSDIBAYER_DUALBIT](#), [GS_HSDIBAYER_DUALLINKBIT](#), [GS_HSDIBAYER_ARRI_D21](#), [GS_HSDIBAYER_ARRI_ALEXA](#), [GS_HSDIBAYER_WIESS_ONEFAME](#), [GS_HSDIBAYER_WIESS_2K1536](#), [GS_HSDIBAYER_WIESS_TWOFAME](#), [GS_HSDIBAYER_WIESS_QUADFRAME](#), [GS_HSDIBAYER_WIESS_TWO2K1536](#)

2897cmdType::ctGetValue
[MEDIACMD::dwPosition](#) - HSDSI bayer transfer type: [GS_HSDIBAYER_DUALBIT](#), [GS_HSDIBAYER_DUALLINKBIT](#), [GS_HSDIBAYER_ARRI_D21](#),

[GS_HSDIBAYER_ARRI_ALEXA](#), [GS_HSDIBAYER_WIESS_ONEFAME](#),
[GS_HSDIBAYER_WIESS_2K1536](#), [GS_HSDIBAYER_WIESS_TWOFAME](#),
[GS_HSDIBAYER_WIESS_QUADFRAME](#), [GS_HSDIBAYER_WIESS_TWO2K1536](#)

2898

gsLocal Front panel/GUI Interface Local Mode

2899cmdType::ctSetValue

2900[MEDIACMD::dwPosition](#) - If 1 then local control available, else remote only

2901cmdType::ctGetValue

[MEDIACMD::dwPosition](#) - If 1 then local control available, else remote only

2902

gsSupportedFileTypes Supported read/write file types

2903cmdType::ctSetValue

2904[MEDIACMD::dwPosition](#) - Currently [GS_SUPFILE_AVI](#), [GS_SUPFILE_ODML](#),
[GS_SUPFILE_QT](#), [GS_SUPFILE_OMFI](#), [GS_SUPFILE_FIX](#), [GS_SUPFILE_AUDONLY](#),
[GS_SUPFILE_STILLS](#), [GS_SUPFILE_UNK](#), [GS_SUPFILE_ANY](#)

2905cmdType::ctGetValue

2906[MEDIACMD::dwPosition](#) - Currently [GS_SUPFILE_AVI](#), [GS_SUPFILE_ODML](#),
[GS_SUPFILE_QT](#), [GS_SUPFILE_OMFI](#), [GS_SUPFILE_FIX](#), [GS_SUPFILE_AUDONLY](#),
[GS_SUPFILE_STILLS](#), [GS_SUPFILE_UNK](#), [GS_SUPFILE_ANY](#)

[MEDIACMD::dwStart](#) - Bit array of supported types per dwPosition above.

2907

gsIgnoreFileTypes File types for this channel to ignore

2908cmdType::ctSetValue

2909[MEDIACMD::dwPosition](#) - Currently [GS_SUPFILE_AVI](#), [GS_SUPFILE_ODML](#),
[GS_SUPFILE_QT](#), [GS_SUPFILE_OMFI](#), [GS_SUPFILE_FIX](#), [GS_SUPFILE_AUDONLY](#),
[GS_SUPFILE_STILLS](#), [GS_SUPFILE_UNK](#), [GS_SUPFILE_ANY](#)

2910cmdType::ctGetValue

[MEDIACMD::dwPosition](#) - Currently [GS_SUPFILE_AVI](#), [GS_SUPFILE_ODML](#), [GS_SUPFILE_QT](#),
[GS_SUPFILE_OMFI](#), [GS_SUPFILE_FIX](#), [GS_SUPFILE_AUDONLY](#), [GS_SUPFILE_STILLS](#),
[GS_SUPFILE_UNK](#), [GS_SUPFILE_ANY](#)

2911

gsRecInhibit Disable recording on this channel or this channel does not support recording.

2912cmdType::ctSetValue

2913[MEDIACMD::dwPosition](#) - 1 to disable recording, or 0 to enable

2914cmdType::ctGetValue

[MEDIACMD::dwPosition](#) - 1 to disable recording, or 0 to enable (play only channels always return 1)

2915

gsRecDrive Select recording drive

2916cmdType::ctSetValue

2917[MEDIACMD::dwPosition](#) - Bit representing drive where 0=C:, 1=D: etc

2918cmdType::ctGetValue

2919[MEDIACMD::dwPosition](#) - Bit representing drive where 0=C:, 1=D: etc

[MEDIACMD::dwStart](#) - Bit array of available drives

2920

gsRecFileName Change the default record filename

2921cmdType::ctSetValue
2922[MEDIACMD::arbID](#) - New next record filename
2923[MEDIACMD::cfFlags](#) - must be set to cfUseClipID
2924cmdType::ctGetValue
2925[MEDIACMD::dwPosition](#) - gsTrue/gsFalse
2926[MEDIACMD::arbID](#) - Next record filename
[MEDIACMD::cfFlags](#) - must be set to cfUseClipID

2927

gsRecRate Recording rate by throughput in kilobytes per second

2928cmdType::ctSetValue
2929[MEDIACMD::dwPosition](#) - Target size of recorded stream in kilounsigned chars per second
2930cmdType::ctGetValue
2931[MEDIACMD::dwPosition](#) - Target size of recorded stream in kilounsigned chars per second
2932[MEDIACMD::dwStart](#) - Smallest size possible
[MEDIACMD::dwEnd](#) - Largest size possible

2933

gsRecFileFormat Default video/stream record file type

2934cmdType::ctSetValue
2935[MEDIACMD::dwPosition](#) - Uses mftXXX enum from MediaReactorTypes.h
2936cmdType::ctGetValue
[MEDIACMD::dwPosition](#) - Uses mftXXX enum from MediaReactorTypes.h

2937

gsRecAudFileFormat Default audio record file type

2938cmdType::ctSetValue
2939[MEDIACMD::dwPosition](#) - Uses mftXXX enum from MediaReactorTypes.h
2940cmdType::ctGetValue
[MEDIACMD::dwPosition](#) - Uses mftXXX enum from MediaReactorTypes.h

2941

gsDelInhibit Disable file deletion on this channel

2942cmdType::ctSetValue
2943[MEDIACMD::dwPosition](#) - 1 to disable delete command, or 0 to enable
2944cmdType::ctGetValue
[MEDIACMD::dwPosition](#) - 1 to disable delete command, or 0 to enable

2945

gsInsInhibit Allows/Inhibits clips being Deleted from Bin or TC Space

2946cmdType::ctSetValue
2947[MEDIACMD::dwPosition](#) - TRUE/FALSE
2948cmdType::ctGetValue
[MEDIACMD::dwPosition](#) - TRUE/FALSE

2949

gsConvertFileFormat Allows/Inhibits clips being added to Bin or TC Space

2950cmdType::ctSetValue
2951[MEDIACMD::dwPosition](#) - TRUE/FALSE
2952cmdType::ctGetValue
2953[MEDIACMD::dwPosition](#) - TRUE/FALSE
2954

gsConvertAudFileFormat Default audio conversion file type

2955cmdType::ctSetValue
2956[MEDIACMD::dwPosition](#) - Uses mftXXX enum from MediaReactorTypes.h
2957cmdType::ctGetValue
[MEDIACMD::dwPosition](#) - Uses mftXXX enum from MediaReactorTypes.h

2958

gsDefStillLen Default length, in frames, for a still graphics file being added as a clip

2959cmdType::ctSetValue
2960[MEDIACMD::dwPosition](#) - Duration in frames
2961cmdType::ctGetValue
[MEDIACMD::dwPosition](#) - Duration in frames

2962

gsSysTime Current reference system time for house VITC or house LTC if available, if not then from system clock interpolated with performance counter.

2963cmdType::ctSetValue
2964[MEDIACMD::dwPosition](#) - Current time code position
2965[MEDIACMD::dwStart](#) - Current milliseconds position
2966[MEDIACMD::dwEnd](#) - Current date
2967cmdType::ctGetValue
2968[MEDIACMD::dwPosition](#) - Current time code position
2969[MEDIACMD::dwStart](#) - Current milliseconds position
[MEDIACMD::dwEnd](#) - Current date

2970

gsDSyncMs Current reference system time for house VITC or house LTC if available, if not then from system clock interpolated with performance counter.

2971cmdType::ctSetValue
2972[MEDIACMD::dwPosition](#) - Current dysnc milliseconds
2973cmdType::ctGetValue
[MEDIACMD::dwPosition](#) - Current dysnc milliseconds

2974

gsHwPort Current hardware port used by channel. Mostly for COMx: port selection of CTL and EXT channels.

2975cmdType::ctSetValue
2976[MEDIACMD::dwPosition](#) - New com port
2977cmdType::ctGetValue
2978[MEDIACMD::dwPosition](#) - Current com port
[MEDIACMD::dwStart](#) - Available com ports as bit array

2979

gsPBEE Playback only output or allow edit to edit

2980cmdType::ctSetValue
2981MEDIACMD::dwPosition - [GS_PBEE_AUTO](#) (playback or e to e), [GS_PBEE_PB](#) (playback only)
2982cmdType::ctGetValue
2983MEDIACMD::dwPosition - [GS_PBEE_AUTO](#) (playback or e to e), [GS_PBEE_PB](#) (playback only), [GS_PBEE_DEFAULT](#) (device default read only)
MEDIACMD::dwStart - Bit array of available commands per dwPosition settings above

2984
gsServoRefSelect Video reference for servo select

2985cmdType::ctSetValue
2986MEDIACMD::dwPosition - [GS_SERVOREF_AUTO](#) (ext is avail, else int), [GS_SERVOREF_EXT](#) (always external)
2987cmdType::ctGetValue
2988MEDIACMD::dwPosition - [GS_SERVOREF_AUTO](#) (ext is avail, else int), [GS_SERVOREF_EXT](#) (always external), [GS_SERVOREF_DEFAULT](#) (device default read only)
MEDIACMD::dwStart - Bit array of available commands per dwPosition settings above

2989
gsHeadSelect Head select

2990cmdType::ctSetValue
2991MEDIACMD::dwPosition - [GS_HEADSEL_RECPLAY](#), [GS_HEADSEL_PLAY](#)
2992cmdType::ctGetValue
2993MEDIACMD::dwPosition - [GS_HEADSEL_RECPLAY](#), [GS_HEADSEL_PLAY](#), [GS_HEADSEL_DEFAULT](#) (device default read only)
MEDIACMD::dwStart - Bit array of available commands per dwPosition settings above

2994
gsColorFrame Colour frame select

2995cmdType::ctSetValue
2996MEDIACMD::dwPosition - [GS_CLRFRM_2FLD](#), [GS_CLRFRM_4FLD](#), [GS_CLRFRM_8FLD](#)
2997cmdType::ctGetValue
2998MEDIACMD::dwPosition - [GS_CLRFRM_2FLD](#), [GS_CLRFRM_4FLD](#), [GS_CLRFRM_8FLD](#), [GS_CLRFRM_DEFAULT](#)
MEDIACMD::dwStart - Bit array of available commands per dwPosition settings above

2999
gsVidRefDisable Video reference disable

3000cmdType::ctSetValue
3001MEDIACMD::dwPosition - [GS_VIDREF_DISABLE](#), [GS_VIDREF_ENABLE](#)
3002cmdType::ctGetValue
3003MEDIACMD::dwPosition - [GS_VIDREF_DISABLE](#), [GS_VIDREF_ENABLE](#)
MEDIACMD::dwStart - Bit array of available commands per dwPosition settings above

3004
gsPlayCountDelay Get Play count delay for the VTR interp

3005cmdType::ctSetValue
3006Not Supported
3007cmdType::ctGetValue

[MEDIACMD::dwPosition](#) stores the value 7 for default

3008

gsEmulateEditBumping Use fake edit mode for MPEG bumping. Basically, all non play speed commands will be emulated, and once a play (lock) is reached the card will be synced to that time and play. Dangerous if sync does not happen quickly...

3009cmdType::ctSetValue

3010[MEDIACMD::dwPosition](#) 1 turns on fake edit, 0 turns off

3011cmdType::ctGetValue

[MEDIACMD::dwPosition](#) 1 fake edit on, 0 fake edit of

3012

cmdaltNearestKeyFrame Special command alt value for position requests, goes to key frame nearest to requested frame

3013cmdType::ctPause, cmdType::ctPlay

[MEDIACMD::dwPosition](#) Target Position

3014

cmdaltNextKeyFrame Special command alt value for position requests, goes to key frame after the requested frame

3015cmdType::ctPause, cmdType::ctPlay

[MEDIACMD::dwPosition](#) Target Position

3016

cmdaltPrevKeyFrame Special command alt value for position requests, goes to key frame before the requested frame

3017cmdType::ctPause, cmdType::ctPlay

[MEDIACMD::dwPosition](#) Target Position

3018

cmdaltStartOfMessage Special command alt value for position requests, goes to first frame of actual (non-black) video after the requested frame

3019cmdType::ctPause, cmdType::ctPlay

[MEDIACMD::dwPosition](#) Target Position

3020

gsVidInputValid Special command is video input valid GS_TRUE / GS_FALSE

3021cmdType::ctGetValue, cmdType::ctSetValue

3022[MEDIACMD::dwPosition](#) -True / False is input Valid

[MEDIACMD::dwStart](#) -signal format of the input

3023

gsVidGenlockValid Special command is genlock input valid GS_TRUE / GS_FALSE

3024cmdType::ctGetValue, cmdType::ctSetValue

3025[MEDIACMD::dwPosition](#) -True / False is genlock Valid

[MEDIACMD::dwStart](#) -signal format of the genlock signal

3026

gsSerialEditMode Special command to set/get edit mode for slow MPEG boards

3027cmdType::ctGetValue
3028[MEDIACMD::dwPosition](#) - [GS_SERIALEDITMODE_NONE](#),
[GS_SERIALEDITMODE_IGNORE](#), [GS_SERIALEDITMODE_FAKE](#)
3029cmdType::ctSetValue
[MEDIACMD::dwPosition](#) - [GS_SERIALEDITMODE_NONE](#), [GS_SERIALEDITMODE_IGNORE](#),
[GS_SERIALEDITMODE_FAKE](#)

3030
gsSerialProtocols Enable/Disable serial types for ext or ctl channels

3031cmdType::ctGetValue
3032[MEDIACMD::dwPosition](#) - [GS_SERIALPROTOCOLS_SONY422](#),
[GS_SERIALPROTOCOLS_ODETICS](#), [GS_SERIALPROTOCOLS_VDCP](#)
3033cmdType::ctSetValue
3034[MEDIACMD::dwPosition](#) - [GS_SERIALPROTOCOLS_SONY422](#),
[GS_SERIALPROTOCOLS_ODETICS](#), [GS_SERIALPROTOCOLS_VDCP](#)
[MEDIACMD::dwStart](#) - Bit array of supported protocols

3035
gsPauseBeforeStop Enable/Disable pause being called on first stop (ee) command. A stop in stop will stop (ee)

3036cmdType::ctGetValue
3037[MEDIACMD::dwPosition](#) - 1 enabled
3038cmdType::ctSetValue
[MEDIACMD::dwPosition](#) - 1 enabled

3039
gsPauseDelay Pause delay in frames. Wait this many frames after pause received before pausing

3040cmdType::ctGetValue
3041[MEDIACMD::dwPosition](#) - # frames, default 0 (fast as possible)
3042cmdType::ctSetValue
[MEDIACMD::dwPosition](#) - # frames, default 0 (fast as possible)

3043
gsFrontPanel Enable/disable/setup front panel

3044cmdType::ctGetValue
3045[MEDIACMD::dwPosition](#) - enable (1), disable (0)
3046[MEDIACMD::dwStart](#) - com port connected to panel (0 for true usb/vga)
3047[MEDIACMD::dwEnd](#) - Panel type, set startup
3048cmdType::ctSetValue
3049[MEDIACMD::dwPosition](#) - enable (1), disable (0)
[MEDIACMD::dwStart](#) - com port connected to panel (0 for true usb/vga)

3050
gsFrontPanelComPort Enable/disable/setup front panel

3051cmdType::ctGetValue
3052[MEDIACMD::dwPosition](#) - Comport Number i.e. 2,3,4

3053cmdType::ctSetValue

[MEDIACMD::dwPosition](#) - Comport Number i.e. 2,3,4

3054

gsVDCPPreroll Sent from the ctl module when a VDCP controller sets a new disk preroll value. It is in frames, but is not nec. When commands that use disk preroll are received, the correct offset is calculated by the ctl and sent, so this value is for info only. Do on use it. Version (4) 206 or greater

3055cmdType::ctGetValue

3056Not Supported

3057cmdType::ctSetValue

[MEDIACMD::dwPosition](#) - disk preroll frame send from ctl

3058

gsChannelAdd Add a new channel (int, ext, ctl, net client)

gsChannelDel Delete a channel (int, ext, ctl, net client)

gsChannelEnable Enable/Disable channel

gsChannelAddress If channel supports, a network address

gsChannelPort If channel supports, a network port

gsChannelComPort If channel supports, a com port

gsChannelTarget Target of channels command for ctl and network

gsChannelPath Path to channel support files (e.g. HTML files for the HTTP server)

gsChannelType Type of channel (read only?)

gsChannelUserName UserName

gsChannelPassWord PassWord

gsUserData0 Set/Get user data. Could be string, start, end and/or position kept in a mediacmd the cmdalt returned will be the time it was set or -1

3059cmdType::ctSetValue

3060[MEDIACMD::dwPosition](#) - Any user data

3061[MEDIACMD::dwStart](#) - Any user data

3062[MEDIACMD::dwEnd](#) - Any user data

3063[MEDIACMD::arbID](#) - Any user data

3064cmdType::ctGetValue

3065[MEDIACMD::dwCmdAlt](#) - Ms time the user data was set or -1 for not set

3066[MEDIACMD::dwPosition](#) - Any user data

3067[MEDIACMD::dwStart](#) - Any user data

3068[MEDIACMD::dwEnd](#) - Any user data

[MEDIACMD::arbID](#) - Any user data

3069

gsUserData1 See gsUserData0.

gsUserData2 See gsUserData0.

gsUserData3 See gsUserData0.

gsUserData4 See gsUserData0.

gsUserData5 See gsUserData0.

gsUserData6 See gsUserData0.

gsUserData7 See gsUserData0.
gsUserData8 See gsUserData0.
gsUserData9 See gsUserData0.
gsErrorLog Enable/Disable/Flush error log

3070cmdType::ctSetValue
 3071[MEDIACMD::dwPosition](#) - 1 Enable, 0 Disable, -1 Flush
 3072[MEDIACMD::dwStart](#) - Start messages
 3073[MEDIACMD::dwEnd](#) - Maximum messages
 3074cmdType::ctGetValue
 3075[MEDIACMD::dwPosition](#) - 1 Enable, 0 Disable
 3076[MEDIACMD::dwStart](#) - Total messages
[MEDIACMD::dwEnd](#) - Maximum messages

3077
gsErrorLogName Get/Set Error Log Name
 3078cmdType::ctSetValue
 3079[MEDIACMD::arbID](#) - New Log Name
 3080cmdType::ctGetValue
[MEDIACMD::arbID](#) - Current Log Name

3081
gsErrorLogStartMs Gets the starting ms value in message units
 3082cmdType::ctGetValue
[MEDIACMD::dwPosition](#) - First message ms (ever)

3083
gsErrorLogCurrentMs Gets current ms time per log entries (for relative and date absolute)
 3084cmdType::ctGetValue
[MEDIACMD::dwPosition](#) - Current ms

3085
gsErrorLogLastChange Gets the last change value, modified with each new entry.
 3086cmdType::ctGetValue
 3087[MEDIACMD::dwPosition](#) - Last change value
 3088[MEDIACMD::dwStart](#) - First available message number
gsErrorLogMessage Gets/Set an error message to/from the log
 3089cmdType::ctGetValue
 3090[MEDIACMD::dwPosition](#) - Send -1 for first num, send last returned number to get next, Return num
 3091[MEDIACMD::lSpeed](#) - Send format type
 3092[MEDIACMD::dwStart](#) - Error Code
 3093[MEDIACMD::dwCmdAlt](#) - Time of message ms
 3094[MEDIACMD::cfFlags](#) - cfPreview causes RAW return
 3095[MEDIACMD::dwStart](#) - First available message number
 3096cmdType::ctSetValue
 3097[MEDIACMD::dwStart](#) - Error Code
 3098[MEDIACMD::arbID](#) - Raw message
gsSysBufferLevel Buffer levels for playback/record (real time)

3099cmdType::ctSetValue
 3100Not supported

3101cmdType::ctGetValue
3102[MEDIACMD::dwPosition](#) - Buffers stored in board
3103[MEDIACMD::dwStart](#) - Buffers in queue to board
3104[MEDIACMD::dwEnd](#) - Buffers in queue from disk
3105. [MEDIACMD::dwInfoChannels](#) - total system buffers available in board + memory / system
frame size
3106[MEDIACMD::dwVideoChannels](#) - Selected channel for buffer request
[MEDIACMD::dwAudioChannels](#) - Selected channel for buffer request

3107

gsSysMemoryUsage Memory usage

3108cmdType::ctSetValue
3109Not supported
3110cmdType::ctGetValue
3111[MEDIACMD::dwPosition](#) - Memory we are using
3112[MEDIACMD::dwStart](#) - Memory used in system
[MEDIACMD::dwEnd](#) - Total memory in system

3113

gsSysCPUUsage CPU usage

3114cmdType::ctSetValue
3115Not supported
3116cmdType::ctGetValue
3117[MEDIACMD::dwPosition](#) - CPU we are using
3118[MEDIACMD::dwStart](#) - CPU used in system
[MEDIACMD::dwEnd](#) - Portion of CPU in kernel mode

3119

gsDroppedFrames Dropped frames

3120cmdType::ctSetValue
3121Not supported
3122cmdType::ctGetValue
3123[MEDIACMD::dwPosition](#) - Last drop number (includes off speed play that should drop)
3124[MEDIACMD::lSpeed](#) - millisecond time of last drop
3125[MEDIACMD::dwStart](#) - Total playback dropped (since first run)
[MEDIACMD::dwEnd](#) - Total record dropped (since first run)

3126

gsProxyMode Sets/Returns AutoProxy mode

3127cmdType::ctGetValue
3128[MEDIACMD::dwPosition](#) - Proxy mode 0=disabled,
3129cmdType::ctSetValue
3130[MEDIACMD::dwPosition](#) - Proxy mode 0=disabled,
gsProxyStatus Returns the status of any proxy generation

3131[MEDIACMD::ctCmd](#) -
3132cmdType::ctPlay - creating a proxy from file on disk
3133cmdType::ctRecord - creating a proxy from a recording file
3134[MEDIACMD::dwCmdAlt](#) - Last change in list ms
3135[MEDIACMD::dwPosition](#) - Current encode frame

3136 [MEDIACMD::dwEnd](#) - current length of file
3137 [MEDIACMD::dwStart](#) - current processor percentage
3138 [MEDIACMD::arbID](#) - current encode path + name
gsGetNextProxy Return the next proxy source file name. If the previous name is set to NULL then return the first clip in the list.

3139 cmdType::ctSetValue
3140 - not supported
3141 cmdType::ctGetValue
3142-in- [MEDIACMD::arbID](#) - Last returned file name or NULL
3143-out- [MEDIACMD::arbID](#) - Next file name in proxy list
3144 [MEDIACMD::dwPosition](#) - Not used yet
3145 [MEDIACMD::dwStart](#) - Not used yet
[MEDIACMD::dwEnd](#) - Not used yet

3146
gsAddProxy Add a new proxy file to the list. Also see [gsAddProxyAndOutputName](#)

3147 [MEDIACMD::arbID](#) - File name to proxy
3148 [MEDIACMD::dwStart](#) - Start frame of proxy
3149 [MEDIACMD::dwEnd](#) - End frame of proxy
gsPromoteProxy Set a proxy file to next in list

3150 [MEDIACMD::arbID](#) - File to promote
gsRemoveProxy Remove a proxy file from the list

3151 [MEDIACMD::arbID](#) - File to remove
gsProxyCPUUsage Get Set the max CPU percentage

3152 cmdType::ctSetValue
3153 [MEDIACMD::dwPosition](#) - 0 No CPU limit, 1..100 max CPU usage
3154 cmdType::ctGetValue
3155 [MEDIACMD::dwPosition](#) - Current CPU usage
3156 [MEDIACMD::dwStart](#) - Current MAX CPU usage
3157 [MEDIACMD::dwEnd](#) - Default CPU Max Usage
gsTransferToArchive Get Set a transfer into the archives

3158 cmdType::ctSetValue
3159 [MEDIACMD::arbID](#) - file / clip to transfer
3160 cmdType::ctGetValue
3161 [MEDIACMD::dwPosition](#) - progress
3162 [MEDIACMD::dwStart](#) - # in queue
gsTransferFromArchive Get Set a transfer from the archives

3163 cmdType::ctSetValue
3164 [MEDIACMD::arbID](#) - file / clip to transfer
3165 cmdType::ctGetValue
3166 [MEDIACMD::dwPosition](#) - progress
3167 [MEDIACMD::dwStart](#) - # in queue

gsGetNextArchiveClip Get Archive list (NULL string = first)

3168cmdType::ctSetValue
3169[MEDIACMD::arbID](#) - file / clip transfered
3170cmdType::ctGetValue
3171[MEDIACMD::dwPosition](#) - # of Clips in archive total
3172[MEDIACMD::dwCmdAlt](#) - ms Since last Update of List (getLastChangeMs)
gsGetNextTransferToArchiveClip Get Archive list (NULL string = first)

3173cmdType::ctSetValue
3174[MEDIACMD::arbID](#) - file / clip transferring
3175cmdType::ctGetValue
3176[MEDIACMD::dwPosition](#) - # of Clips transferring to archive
3177[MEDIACMD::dwCmdAlt](#) - ms Since last Update of List (getLastChangeMs)
gsGetNextTransferFromArchiveClip Get Archive list (NULL string = first)

3178cmdType::ctSetValue
3179[MEDIACMD::arbID](#) - file / clip transferring
3180cmdType::ctGetValue
3181[MEDIACMD::dwPosition](#) - # of Clips transferring from archive
3182[MEDIACMD::dwCmdAlt](#) - ms Since last Update of List (getLastChangeMs)
gsAddProxyAndOutputName Add a new proxy file with output name and i/o. Extension of [gsAddProxy](#)

3183[MEDIACMD::arbID](#)[0] - "File name to proxy"\0"Ouptut name to use"\0
3184[MEDIACMD::dwStart](#) - Start frame of proxy
3185[MEDIACMD::dwEnd](#) - End frame of proxy
gsVWVersion Get VVW version number

3186cmdType::ctSetValue
3187Not Supported
3188cmdType::ctGetValue
[MEDIACMD::arbID](#) - Zero terminated ansi string with version number

3189
gsMEVersion Get MediaReactor version number

3190cmdType::ctSetValue
3191Not Supported
3192cmdType::ctGetValue
[MEDIACMD::arbID](#) - Zero terminated ansi string with version number

3193
gsVWType Get VVW type description

3194cmdType::ctSetValue
3195Not Supported
3196cmdType::ctGetValue
[MEDIACMD::arbID](#) - Zero terminated ansi string with VVW machine type

3197

gsVWChannelType Get VVW channel type

3198cmdType::ctSetValue

3199Not Supported

3200cmdType::ctGetValue

[MEDIACMD::arbID](#) - Zero terminated ansi string with channel type

3201

gsVWChannelName Get/Set VVW channel name

3202cmdType::ctSetValue

3203[MEDIACMD::dwPosition](#) - 0 Get Current, 1 Get Default

3204[MEDIACMD::arbID](#) - Zero terminated ansi string with desired channel name

3205cmdType::ctGetValue

[MEDIACMD::arbID](#) - Zero terminated ansi string with current channel name

3206

gsVWLicense Get VVW License status

3207cmdType::ctSetValue

3208Not supported

3209cmdType::ctGetValue

3210[MEDIACMD::dwPosition](#) - License type (-1=invalid, 0=perm, 1=days, 2=runs, 3=users)

3211[MEDIACMD::dwStart](#) - 0 for perm, # for days/runs/users

3212[MEDIACMD::dwEnd](#) - License flags (app dependent)

3213[MEDIACMD::lSpeed](#) - License level (app dependent)

[MEDIACMD::arbID](#) - Zero terminated ansi string about current license

3214

gsMonitor Setup on screen monitor (VGA Monitor)

3215cmdType::ctSetValue

3216[MEDIACMD::dwPosition](#) - 1 to enable, 0 to disable

3217[MEDIACMD::dwEnd](#) - Left corner

3218[MEDIACMD::dwStart](#) - Top corner

3219[MEDIACMD::dwCmdAlt](#) - Size (0 = Default, 1 = Full, 2 = Half, 3 = Quarter)

3220cmdType::ctGetValue

3221[MEDIACMD::dwPosition](#) - 1 to enable, 0 to disable

3222[MEDIACMD::dwEnd](#) - Left corner

3223[MEDIACMD::dwStart](#) - Top corner

[MEDIACMD::dwCmdAlt](#) - Size (0 = Default, 1 = Full, 2 = Half, 3 = Quarter)

3224

gsMonitorHwnds Set handles to windows

3225cmdType::ctSetValue

3226[MEDIACMD::dwPosition](#) - Handle to target window or -1

3227[MEDIACMD::dwEnd](#) - Handle to owner application window

3228[MEDIACMD::dwStart](#) - Window flags

3229cmdType::ctGetValue

3230[MEDIACMD::dwPosition](#) - Handle to target window

3231[MEDIACMD::dwEnd](#) - Handle to owner application window or -1

[MEDIACMD::dwStart](#) - Window flags

3232

gsHwnds Alias for [gsMonitorHwnds](#) for older apps.

gsMonitorDisplay Turns VGA display on / off without killing the window Can use this later to set refresh rates - aspect ratios or what not

3233cmdType::ctSetValue
3234[MEDIACMD::dwPosition](#) - GS_TRUE, GS_FALSE
3235[MEDIACMD::dwEnd](#) -
3236[MEDIACMD::dwStart](#) -
3237cmdType::ctGetValue
3238[MEDIACMD::dwPosition](#) - GS_TRUE, GS_FALSE
3239[MEDIACMD::dwEnd](#) -

[MEDIACMD::dwStart](#)

3240

gsMonitorGrab Get a capture of the current output (input passthrough or current clip output). Use #cmdType::ctGetValue to get a preview.

3241cmdType::ctSetValue
3242[MEDIACMD::dwPosition](#) - Type of capture that is going to be used
3243[MEDIACMD::arbID](#) - Optional, depends on dwPosition
3244cmdType::ctGetValue
3245[MEDIACMD::dwPosition](#) - Preview type [GS_MONITORGRAB_NONE](#),
[GS_MONITORGRAB_TYPE_BMP](#), [GS_MONITORGRAB_TYPE_JPG](#),
[GS_MONITORGRAB_SIZE_FULL](#), [GS_MONITORGRAB_SIZE_HALF](#),
[GS_MONITORGRAB_SIZE_QUARTER](#), [GS_MONITORGRAB_TO_MEMORY](#),
[GS_MONITORGRAB_TO_UNC_PATH](#), [GS_MONITORGRAB_TO_HTTP](#),
[GS_MONITORGRAB_TO_NETWORK](#)
3246[MEDIACMD::arbID](#) - Image data, if returned not saved
3247To create a grab type, combine on type (bmp, jpg) with a size (full, half, quarter) and a target (memory, path, http, network).
3248Depending on target, the arbID member will be filled in as follows:
3249GS_MONITORGRAB_TO_MEMORY - arbID not used
3250GS_MONITORGRAB_TO_UNC_PATH - arbID unified naming conventions (UNC) path
3251GS_MONITORGRAB_TO_HTTP - arbID contains the name
GS_MONITORGRAB_TO_NETWORK - Not implemented

3252

gsUtilityMonitorDraw Get/Set a pointer to the Utility Monitor DTDraw class

3253cmdType::ctSetValue
3254[MEDIACMD::dwPosition](#) - Pointer in DWORD to DTDraw class (always RGB32)
3255
3256cmdType::ctGetValue
3257[MEDIACMD::dwPosition](#) - Pointer in DWORD to DTDraw class (always RGB32)

3258

gsUtilityMonitorDrawSetup Get/Set the layout of the utility monitor

3259cmdType::ctSetValue
3260[MEDIACMD::dwPosition](#) - Main layout of screen // Normal quad screen
3261AVUM_CONFIG_QUAD_SPLIT 0
3262// One full screen on left, 3 1/4 on right
3263AVUM_CONFIG_ONELEFT_THREERIGHT 1

3264// One full screen on top, 3 1/2 on bottom
3265AVUM_CONFIG_ONETOP_THREEBOTTOM 2
3266
3267
3268cmdType::ctGetValue
3269[MEDIACMD::dwPosition](#) - Main layout of screen

3270

gsWaveVectorSetup Get/Set a waveform, vectorscope, etc on DTDraw, VGA output or Main output

3271cmdType::ctSetValue

3272[MEDIACMD::dwPosition](#) - Pointer to the DTDraw structure to draw on

3273[MEDIACMD::lSpeed](#) - Top 32 bits of 64 bit pointer to the DTDraw structure to draw on

3274[MEDIACMD::dwStart](#) - Enable or disable [GS_ENABLE](#) [GS_DISABLE](#)

3275[MEDIACMD::dwEnd](#) - Target #[GS_WAVEVECTOR_TARGET_DTDRAW](#),
#[GS_WAVEVECTOR_TARGET_VGA](#), #[GS_WAVEVECTOR_TARGET_OUTPUT](#)

3276cmdType::ctGetValue

3277[MEDIACMD::dwPosition](#) - Returns the DTDraw structure it is drawing on

3278[MEDIACMD::lSpeed](#) - Top 32 bits of 64 bit pointer to the DTDraw structure to draw on

3279[MEDIACMD::dwStart](#) - Is enabled or disabled [GS_ENABLE](#) [GS_DISABLE](#)
[MEDIACMD::dwEnd](#) - Target #[GS_WAVEVECTOR_TARGET_DTDRAW](#),
#[GS_WAVEVECTOR_TARGET_VGA](#), #[GS_WAVEVECTOR_TARGET_OUTPUT](#)

3280

gsWaveVectorType Get/Set type of waveform vector

3281cmdType::ctSetValue

3282[MEDIACMD::dwPosition](#) - Waveform vector type(s) [GS_WAVEVECTOR_PICTURE](#),
[GS_WAVEVECTOR_VECTORSCOPE](#), [GS_WAVEVECTOR_WAVEFORM](#)

3283[MEDIACMD::dwStart](#) - Channels to enable [GS_WAVEVECTOR_CHANNEL_R](#),
[GS_WAVEVECTOR_CHANNEL_G](#), [GS_WAVEVECTOR_CHANNEL_B](#),
[GS_WAVEVECTOR_CHANNEL_A](#), [GS_WAVEVECTOR_CHANNEL_Y](#),
[GS_WAVEVECTOR_CHANNEL_CR](#), [GS_WAVEVECTOR_CHANNEL_CB](#)

3284[MEDIACMD::dwEnd](#) -

3285cmdType::ctGetValue

3286[MEDIACMD::dwPosition](#) - Waveform vector type(s) [GS_WAVEVECTOR_PICTURE](#),
[GS_WAVEVECTOR_VECTORSCOPE](#), [GS_WAVEVECTOR_WAVEFORM](#)

3287[MEDIACMD::dwStart](#) - Channels to enable [GS_WAVEVECTOR_CHANNEL_R](#),
[GS_WAVEVECTOR_CHANNEL_G](#), [GS_WAVEVECTOR_CHANNEL_B](#),
[GS_WAVEVECTOR_CHANNEL_A](#), [GS_WAVEVECTOR_CHANNEL_Y](#),
[GS_WAVEVECTOR_CHANNEL_CR](#), [GS_WAVEVECTOR_CHANNEL_CB](#)

[MEDIACMD::dwEnd](#)

3288

gsWaveVectorArea Get/Set area to use as source

3289cmdType::ctSetValue

3290[MEDIACMD::dwPosition](#) -

3291[MEDIACMD::dwStart](#) - Start line 0..(height-1)

3292[MEDIACMD::dwEnd](#) - End line 1..height

3293[MEDIACMD::dwVideoChannels](#) - Start pixel 0..(width-1) - not supported yet

3294[MEDIACMD::dwAudioChannels](#) - End pixel 1..width - not supported yet

3295cmdType::ctGetValue

3296 [MEDIACMD::dwPosition](#) -
3297 [MEDIACMD::dwStart](#) - Start line 0..(height-1)
3298 [MEDIACMD::dwEnd](#) - End line 1..height
3299 [MEDIACMD::dwVideoChannels](#) - Start pixel 0..(width-1) - not supported yet
[MEDIACMD::dwAudioChannels](#) - End pixel 1..width - not supported yet

3300
gsWaveVectorLastChangeMs Get last update time in milliseconds

3301 cmdType::ctGetValue
3302 [MEDIACMD::dwPosition](#) - Time of last update in milliseconds
gsMonitorLoadBuffers Load buffers (for opengl right now)

3303 cmdType::ctSetValue
3304 Load buffers now
3305 cmdType::ctGetValue
[MEDIACMD::dwPosition](#) - Get last load time

3306
gsDirGetList Not to be used. See VVWXMLNextDirEntry and VVWXMLFileInfo
gsDirGetInfo Not to be used. See VVWXMLNextDirEntry and VVWXMLFileInfo
gsDirGetFileInfo Not to be used. See VVWXMLNextDirEntry and VVWXMLFileInfo
gsDirGetFileGrab Not to be used. See VVWXMLNextDirEntry and VVWXMLFileInfo
gsVgaDisplayEnable Allow VGA display to function

3307 cmdType::ctGetValue
3308 [MEDIACMD::dwPosition](#) - Enabled 0,1
3309 [MEDIACMD::dwStart](#) - Fullscreen Enabled 0,1
3310 [MEDIACMD::dwEnd](#) - Reduced Frame Rate Setting (1= 1/1 2 = 1/2 3 = 1/3 or 4 = 1/4)
3311 cmdType::ctSetValue
3312 [MEDIACMD::dwPosition](#) - Enabled 0,1
[MEDIACMD::dwEnd](#) - Reduced Frame Rate Setting (1= 1/1 2 = 1/2 3 = 1/3 or 4 = 1/4)

3313
gsVgaDirectXConfig Allow VGA display to function

3314 cmdType::ctGetValue
3315 [MEDIACMD::dwPosition](#) - Allow DirectX
3316 [MEDIACMD::dwStart](#) - Bit Array for YUV, RGB and Overlay allow [GS_DXRGB_OVERLAY](#)
[GS_DXRGB_DIRECT](#) [GS_DXYUV_OVERLAY](#) [GS_DXYUV_DIRECT](#)
3317 cmdType::ctSetValue
3318 [MEDIACMD::dwPosition](#) - Allow DirectX
[MEDIACMD::dwStart](#) - Bit Array for YUV, RGB and Overlay allow [GS_DXRGB_OVERLAY](#)
[GS_DXRGB_DIRECT](#) [GS_DXYUV_OVERLAY](#) [GS_DXYUV_DIRECT](#)

3319
gsVga3DConfig Setup 3D VGA Output

3320 cmdType::ctGetValue
3321 [MEDIACMD::dwPosition](#) - 3D Display Type [GS_3DVGA_LEFTEYE](#)
[GS_3DVGA_RIGHTEYE](#) [GS_3DVGA_ANAGLYPH_REDCYAN](#)
[GS_3DVGA_ANAGLYPH_REDBLUE](#) [GS_3DVGA_ANAGLYPH_AMBERBLUE](#)

[GS_3DVGA_ANAGLYPH_GREENMAGENTA](#) [GS_3DVGA_INTERLACED](#)
[GS_3DVGA_ONIONSkin](#) [GS_3DVGA_DIFFERENCE](#) [GS_3DVGA_OVERUNDER](#)
[GS_3DVGA_SIDEBySIDE](#) [GS_3DVGA_SPLIT](#) [GS_3DVGA_MIRROR](#)
[GS_3DVGA_BUTTERFLY](#) [GS_3DVGA_AMINUSB_THRESHOLD](#) [GS_3DVGA DISSOLVE](#)
[GS_3DVGA_WIPE](#) [GS_3DVGA_FLAG_SPLITVERT](#) [GS_3DVGA_FLAG_LENTICULAR](#)
3322 [GS_3DVGA_FLAG_INVERT](#) [GS_3DVGA_FLAG_FLIPLEFTVERT](#)
[GS_3DVGA_FLAG_FLIPRIGHTVERT](#) [GS_3DVGA_FLAG_FLIPLEFTHORIZ](#)
[GS_3DVGA_FLAG_FLIPRIGHTHORIZ](#) [GS_3DVGA_LUMA_DIFF](#)
3323 [MEDIACMD::dwStart](#) - Bit Array of available 3D display types
3324 [cmdType::ctSetValue](#)
3325 [MEDIACMD::dwPosition](#) - 3D Display Type [GS_3DVGA_LEFTEYE](#)
[GS_3DVGA_RIGHTEYE](#) [GS_3DVGA_ANAGLYPH_REDCYAN](#)
[GS_3DVGA_ANAGLYPH_REDBLUE](#) [GS_3DVGA_ANAGLYPH_AMBERBLUE](#)
[GS_3DVGA_ANAGLYPH_GREENMAGENTA](#) [GS_3DVGA_INTERLACED](#)
[GS_3DVGA_ONIONSkin](#) [GS_3DVGA_DIFFERENCE](#) [GS_3DVGA_OVERUNDER](#)
[GS_3DVGA_SIDEBySIDE](#) [GS_3DVGA_SPLIT](#) [GS_3DVGA_MIRROR](#)
[GS_3DVGA_BUTTERFLY](#) [GS_3DVGA_AMINUSB_THRESHOLD](#) [GS_3DVGA DISSOLVE](#)
[GS_3DVGA_WIPE](#) [GS_3DVGA_FLAG_SPLITVERT](#) [GS_3DVGA_FLAG_LENTICULAR](#)
[GS_3DVGA_FLAG_INVERT](#) [GS_3DVGA_FLAG_FLIPLEFTVERT](#)
[GS_3DVGA_FLAG_FLIPRIGHTVERT](#) [GS_3DVGA_FLAG_FLIPLEFTHORIZ](#)
[GS_3DVGA_FLAG_FLIPRIGHTHORIZ](#) [GS_3DVGA_LUMA_DIFF](#)

3326

gsVga3DWipeType Setup 3D VGA type for [GS_3DVGA_WIPE](#) (use SMPTE)

3327 [cmdType::ctGetValue](#)

3328 [MEDIACMD::dwPosition](#) - Wipe type

3329 [MEDIACMD::dwStart](#) - Wipe range low (0)

3330 [MEDIACMD::dwEnd](#) - Wipe range high (65536)

3331 [cmdType::ctSetValue](#)

[MEDIACMD::dwPosition](#) - Wipe type

3332

gsVga3DMix Setup 3D VGA mix value

3333 [cmdType::ctGetValue](#)

3334 [MEDIACMD::dwPosition](#) - Mix value (16 bit)

3335 [MEDIACMD::dwStart](#) - Mix range low (0)

3336 [MEDIACMD::dwEnd](#) - Mix range high (65536)

3337 [cmdType::ctSetValue](#)

[MEDIACMD::dwPosition](#) - Mix value (16 bit)

3338

gsVga3DThreshold Setup 3D VGA Threshold

3339 [cmdType::ctGetValue](#)

3340 [MEDIACMD::dwPosition](#) - Threshold value (16 bit)

3341 [MEDIACMD::dwStart](#) - Threshold range low (0)

3342 [MEDIACMD::dwEnd](#) - Threshold range high (65536)

3343 [cmdType::ctSetValue](#)

[MEDIACMD::dwPosition](#) - Threshold value (16 bit)

3344

gsVga3DSplitHorizontal Setup 3D VGA horizontal split

3345cmdType::ctGetValue
3346[MEDIACMD::dwPosition](#) - Split position in pixels
3347[MEDIACMD::dwStart](#) - Left (0)
3348[MEDIACMD::dwEnd](#) - Right (width)
3349cmdType::ctSetValue

[MEDIACMD::dwPosition](#) - Split position in pixels

3350
gsVga3DSplitVertical Setup 3D VGA vertical split

3351cmdType::ctGetValue
3352[MEDIACMD::dwPosition](#) - Split position in lines
3353[MEDIACMD::dwStart](#) - Left (0)
3354[MEDIACMD::dwEnd](#) - Right (height)
3355cmdType::ctSetValue

[MEDIACMD::dwPosition](#) - Split position in lines

3356
gsVga3DGridSize Overlay a grid on the display, either percentage or sizes

3357cmdType::ctGetValue
3358[MEDIACMD::dwPosition](#) - Percentage size (if [GS_NOT_SUPPORTED](#) / -1, then use start / end)
3359[MEDIACMD::dwStart](#) - Horizontal pixels to next line in grid
3360[MEDIACMD::dwEnd](#) - Vertical pixels to next line in grid
3361cmdType::ctSetValue
3362[MEDIACMD::dwPosition](#) - Percentage size
3363[MEDIACMD::dwStart](#) - Horizontal lines
[MEDIACMD::dwEnd](#) - Vertical pixels to next line in grid

3364
gsVga3DGridType Overlay a grid on the display

3365cmdType::ctGetValue
3366[MEDIACMD::dwPosition](#) - Set type to (0=off,1=percent,2=pixel)
3367cmdType::ctSetValue

[MEDIACMD::dwPosition](#) - Current Type

3368
gsVgaFullscreenEnable Allow Fullscreen VGA on secondary monitor

3369cmdType::ctGetValue
3370[MEDIACMD::dwPosition](#) - Set type to (0=off,1=on)
3371cmdType::ctSetValue
3372[MEDIACMD::dwPosition](#) - Current Value

[MEDIACMD::dwStart](#) - number of monitors

3373
gsLimitAvailableChannels Allow maximum number of channels to be set

3374cmdType::ctGetValue
3375[MEDIACMD::dwPosition](#) - # channels
3376cmdType::ctSetValue

[MEDIACMD::dwPosition](#) - # channels

3377
gsVGAZoomPan Zoom and pan the VGA (Overlay only right now) for

bbReplay/Officiating

e.g. 1920

1.0 = factor 1 1920

2.0 = factor 655 960

3.0 = factor 1310 640

4.0 = factor 1965 240

3378cmdType::ctGetValue

3379[MEDIACMD::dwPosition](#) - Zoom level 1..65500 = 1..100

3380[MEDIACMD::dwStart](#) - Pan X = 0..Width

3381[MEDIACMD::dwEnd](#) - Pan Y = 0..Height

3382cmdType::ctSetValue

3383[MEDIACMD::dwPosition](#) - Zoom level 1..65500 = 1..100

3384[MEDIACMD::dwStart](#) - Pan X = 0..Width

[MEDIACMD::dwEnd](#) - Pan Y = 0..Height

3385

gsChannelsExist Check if channels exist

3386cmdType::ctSetValue

3387Not Supported

3388cmdType::ctGetValue

3389[MEDIACMD::dwVideoChannels](#) - Possible Video Channels

3390[MEDIACMD::dwAudioChannels](#) - Possible Audio Channels

[MEDIACMD::dwInfoChannels](#) - Possible Info Channels

3391

gsClipMode Get/Set clip mode state (else time code mode)

3392cmdType::ctSetValue

3393[MEDIACMD::dwPosition](#) - [GS_CLIPMODE_CLIPSPACE](#), [GS_CLIPMODE_TCSPACE](#),

[GS_CLIPMODE_SINGLE](#), [GS_CLIPMODE_FILM](#) current types

3394cmdType::ctGetValue

[MEDIACMD::dwPosition](#) - [GS_CLIPMODE_CLIPSPACE](#), [GS_CLIPMODE_TCSPACE](#),

[GS_CLIPMODE_SINGLE](#), [GS_CLIPMODE_FILM](#) current types

3395

gsRecOffset Get/Set record offset for VVW3x00 replay mode

3396cmdType::ctSetValue

3397[MEDIACMD::dwPosition](#) - Time code offset or 0 to reset

3398cmdType::ctGetValue

[MEDIACMD::dwPosition](#) - Time code offset or 0 to reset

3399

gsChanCapabilities Get channel capabilities

3400cmdType::ctSetValue

3401Not Supported

3402cmdType::ctGetValue

3403[MEDIACMD::dwPosition](#) - Bitwise array [GS_CHANCAP_PLAY](#), [GS_CHANCAP_REVPLAY](#),

[GS_CHANCAP_PAUSE](#), [GS_CHANCAP_JOG](#), [GS_CHANCAP_SHUTTLE](#),
[GS_CHANCAP_SEEK](#), [GS_CHANCAP_PREVIEW](#), [GS_CHANCAP_STOP](#),
[GS_CHANCAP_ETOE](#), [GS_CHANCAP_RECORD](#), [GS_CHANCAP_EDIT](#),
[GS_CHANCAP_RECSTOP](#), [GS_CHANCAP_SELECTPRESET](#), [GS_CHANCAP_EJECT](#),
[GS_CHANCAP_LOOP](#), [GS_CHANCAP_VGAPREVIEW](#), [GS_CHANCAP_AUDPREVIEW](#),
[GS_CHANCAP_FILE](#), [GS_CHANCAP_NET](#), [GS_CHANCAP_CLIPSPACE](#),
[GS_CHANCAP_TCSPACE](#), [GS_CHANCAP_ALL](#)

3404

gsLastChangeMs Get last change millisecond time from clip space, tc space or file

3405cmdType::ctSetValue

3406Not Supported

3407cmdType::ctGetValue

[MEDIACMD::dwPosition](#) - last change in ms aligned with Dsync

3408

gsGPIIn Get the state of the GPI ins, reset them with set. Only GPIs that are included in the 'Mask' will be effected. If you want to set the GPI 2, you need to set dwPosition to 0x00000002 and the dwVideoChannels (mask) to 0x00000002. Setting the dwPosition to 0x00000000 and the dwVideoChannels to 0x00000002 will turn off the GPI. If the dwVideoChannels is 0, then nothing will change.

3409cmdType::ctSetValue - reset in events to nothing

3410[MEDIACMD::dwPosition](#) - GPI (0-31 / 1-32)

3411[MEDIACMD::dwStart](#) - GPI (32-63 / 33-64)

3412[MEDIACMD::dwEnd](#) - GPI (64-95 / 65-96)

3413[MEDIACMD::dwVideoChannels](#) - Mask for 0-31

3414[MEDIACMD::dwAudioChannels](#) - Mask for 32-63

3415[MEDIACMD::dwInfoChannels](#) - Mask for 64-95

3416cmdType::ctGetValue

3417[MEDIACMD::dwPosition](#) - GPI (0-31 / 1-32)

3418[MEDIACMD::dwStart](#) - GPI (32-63 / 33-64)

3419[MEDIACMD::dwEnd](#) - GPI (64-95 / 65-96)

3420[MEDIACMD::dwVideoChannels](#) - Mask for 0-31

3421[MEDIACMD::dwAudioChannels](#) - Mask for 32-63

3422[MEDIACMD::dwInfoChannels](#) - Mask for 64-95

3423[MEDIACMD::lSpeed](#) - last change in ms aligned with Dsync

A 1 in the GPI bitwise array mean it triggered, 0 means it has not.

3424

gsGPIOut Get the state of the GPI outs, Only GPIs that are included in the 'Mask' will be effected. If you want to set the GPI 2, you need to set dwPosition to 0x00000002 and the dwVideoChannels (mask) to 0x00000002. Setting the dwPosition to 0x00000000 and the dwVideoChannels to 0x00000002 will turn off the GPI. If the dwVideoChannels is 0, then nothing will change.

3425cmdType::ctSetValue - set the GPIs up or down or pulse

3426[MEDIACMD::dwPosition](#) - GPI (0-31 / 1-32)

3427[MEDIACMD::dwStart](#) - GPI (32-63 / 33-64)

3428[MEDIACMD::dwEnd](#) - GPI (64-95 / 65-96)

3429[MEDIACMD::dwVideoChannels](#) - Mask for 0-31

3430[MEDIACMD::dwAudioChannels](#) - Mask for 32-63

3431[MEDIACMD::dwInfoChannels](#) - Mask for 64-95

3432cmdType::ctGetValue - get the current GPI output state.

3433[MEDIACMD::dwPosition](#) - GPI (0-31 / 1-32)

3434[MEDIACMD::dwStart](#) - GPI (32-63 / 33-64)

3435[MEDIACMD::dwEnd](#) - GPI (64-95 / 65-96)

3436[MEDIACMD::dwVideoChannels](#) - Mask for 0-31

3437[MEDIACMD::dwAudioChannels](#) - Mask for 32-63

3438[MEDIACMD::dwInfoChannels](#) - Mask for 64-95

3439[MEDIACMD::lSpeed](#) - Last Change Ms

A 1 in the GPI bitwise array means on or triggered, 0 is down or off.

3440

gsCurrentMs Get the current millisecond counter on the machine (NOT aligned to anything). NOTE: This is handled directly for network channels on server side

3441cmdType::ctSetValue

3442Not Supported

3443cmdType::ctGetValue

[MEDIACMD::dwPosition](#) - the current millisecond counter

3444

gsClipModePreroll Get/Set whether we are adding on minute of black to start and end of each clip

3445cmdType::ctSetValue

3446[MEDIACMD::dwPosition](#) - [GS_TRUE](#), [GS_FALSE](#)

3447cmdType::ctGetValue

[MEDIACMD::dwPosition](#) - [GS_TRUE](#), [GS_FALSE](#)

3448

gsClipModeBackup Get number of backups in the current media space Set the number to back up from

3449cmdType::ctSetValue

3450[MEDIACMD::dwPosition](#) - Which backup to make active

3451[MEDIACMD::dwStart](#) - Clip mode to use (-1 = Current) [GS_CLIPMODE_CLIPSPACE](#),
[GS_CLIPMODE_TCSPACE](#)

3452cmdType::ctGetValue

3453[MEDIACMD::dwPosition](#) - Number of current backups available

[MEDIACMD::dwStart](#) - Clip mode to use (-1 = Current) [GS_CLIPMODE_CLIPSPACE](#),
[GS_CLIPMODE_TCSPACE](#)

3454

gsSaveCurrent

gsLoadClipSpace Load a new clip space (or new if file does not exist)

3455cmdType::ctSetValue

3456[MEDIACMD::arbID](#) - File name of new or existing clip space

3457cmdType::ctGetValue

[MEDIACMD::arbID](#) - returns current file name of clip space

3458

gsLoadTCSpace Load a new tc space (or new if file does not exist)

3459cmdType::ctSetValue

3460[MEDIACMD::arbID](#) - File name of new or existing tc space

3461cmdType::ctGetValue
[MEDIACMD::arbID](#) - returns current file name of tc space

3462
gsLoadFilmSpace Load a new film (or new if file does not exist)

3463cmdType::ctSetValue
3464[MEDIACMD::arbID](#) - File name of new or existing film
3465cmdType::ctGetValue
[MEDIACMD::arbID](#) - returns current file name of film

3466
gsLoadEditSpace Load a new edit (or new if file does not exist)

3467cmdType::ctSetValue
3468[MEDIACMD::arbID](#) - File name of new or existing edit clip
3469cmdType::ctGetValue
[MEDIACMD::arbID](#) - returns current file name of edit clip

3470
gsSaveClipSpaceToDisk Save the clip to disk using the current names

3471cmdType::ctSetValue
3472[MEDIACMD::arbID](#) - Optional new file name to save to
3473cmdType::ctGetValue
Not supported

3474
gsSaveTCSpaceToDisk Save the tc space to disk using the current names

3475cmdType::ctSetValue
3476[MEDIACMD::arbID](#) - Optional new file name to save to
3477cmdType::ctGetValue
Not supported

3478
gsUserLogIn Log in user (must have rights on the local machine)

3479cmdType::ctSetValue
3480[MEDIACMD::arbID](#) - User name zero terminated followed by passunsigned short zero terminated.
3481cmdType::ctGetValue
[MEDIACMD::arbID](#) - Returns user name zero terminated.

3482
gsUserLastChangeMs Last change in user status

3483cmdType::ctSetValue Not Supported
3484cmdType::ctGetValue
[MEDIACMD::dwPosition](#) - Last change in users (status or number)

3485
gsUserList Return a list of currently logged in users

3486cmdType::ctSetValue Not Supported

3487cmdType::ctGetValue
3488MEDIACMD::dwPosition - 0..max user
3489MEDIACMD::arbID - User name zero terminated, location info zero terminated
MEDIACMD::dwStart - User rights NULL when all users have been returned

3490
gsUserAdd Allow a user access to the unit - ONLY AVAILABLE ON LOCAL MACHINE

3491cmdType::ctSetValue
3492MEDIACMD::arbID - User name zero terminated, passunsigned short zero terminated
3493MEDIACMD::dwStart - User Rights
cmdType::ctGetValue Not Supported

3494
gsUserDel Remove a user's access to the unit - ONLY AVAILABLE ON LOCAL MACHINE

3495cmdType::ctSetValue
3496MEDIACMD::arbID - User name zero terminated.
cmdType::ctGetValue Not Supported

3497
gsUserRights Change a users rights - ONLY AVAILABLE ON LOCAL MACHINE

3498cmdType::ctSetValue
3499MEDIACMD::arbID - User name zero terminated.
3500MEDIACMD::dwPosition - the user rights
3501cmdType::ctGetValue
3502MEDIACMD::arbID - User name zero terminated (current if NULL)
MEDIACMD::dwPosition - the user rights

3503
gsUserPasswd Change current users passunsigned short (must be logged in)

3504cmdType::ctSetValue
3505MEDIACMD::arbID - new passunsigned short zero terminated new passunsigned short zero terminated.
cmdType::ctGetValue Not Supported

3506
gsPiconFrame Create 1 small jpg image (picon) for a file frame

3507cmdType::ctSetValue
3508MEDIACMD::dwPosition - frame number
3509MEDIACMD::arbID - file and directory

3510cmdType::ctGetValue
3511MEDIACMD::dwPosition - (GS_TRUE | GS_FALSE) is set picon
3512MEDIACMD::arbID - file and directory
gsJpegFrame Get a full resolution jpg of a frame

3513cmdType::ctSetValue
3514MEDIACMD::dwPosition - frame number

3515 [MEDIACMD::arbID](#) - file and directory

3516 cmdType::ctGetValue

3517 [MEDIACMD::dwPosition](#) - (GS_TRUE | GS_FALSE) is set jpeg

3518 [MEDIACMD::arbID](#) - file and directory

gsImageDirectory Default image directory

3519 cmdType::ctSetValue

3520 [MEDIACMD::arbID](#) - directory to use (NULL = Use Network Directory)

3521 cmdType::ctGetValue

3522 [MEDIACMD::arbID](#) - directory using for images

gsFrameInfo Get a frame's info

3523 cmdType::ctGetValue

3524 [MEDIACMD::dwPosition](#) - IN = frame number, OUT = Width

3525 [MEDIACMD::dwStart](#) - Height

3526 [MEDIACMD::dwEnd](#) - Bits

3527 [MEDIACMD::arbID](#) - Encoding

3528 return frame number

gsRawFrame Get a raw frame in frame's default format

3529 cmdType::ctSetValue

3530 [MEDIACMD::dwPosition](#) - frame number

3531 [MEDIACMD::dwStart](#) - Format (original, RGBA)

3532 [MEDIACMD::dwEnd](#) - Frame Size

3533 [MEDIACMD::arbID](#) - Image Data

3534 cmdType::ctGetValue

3535 [MEDIACMD::dwPosition](#) - frame number

3536 [MEDIACMD::dwStart](#) - Format (original, RGBA)

3537 [MEDIACMD::dwEnd](#) - Frame Size

3538 [MEDIACMD::arbID](#) - Image Data

gsPreallocateEditFile Create a black empty file for '!:edit'

3539 cmdType::ctSetValue

3540 [MEDIACMD::dwPosition](#) - userbits

3541 [MEDIACMD::dwStart](#) - starting time code

3542 [MEDIACMD::dwEnd](#) - duration

3543 [MEDIACMD::arbID](#) - Filename

3544 cmdType::ctGetValue

3545 [MEDIACMD::dwPosition](#) - current frame writing (0..dwEnd), -1 if no file writing

3546 [MEDIACMD::dwStart](#) - starting time code

3547 [MEDIACMD::dwEnd](#) - duration

3548 [MEDIACMD::arbID](#) - Filename, null if no file writing

gsCreateEditFile Create a file from the clip ::edit

3549 cmdType::ctSetValue

3550 [MEDIACMD::dwPosition](#) - userbits
3551 [MEDIACMD::dwStart](#) - starting time code
3552 [MEDIACMD::dwEnd](#) - duration
3553 [MEDIACMD::arbID](#) - Filename

3554 cmdType::ctGetValue
3555 [MEDIACMD::dwPosition](#) - current frame writing (0..dwEnd), -1 if no file writing
3556 [MEDIACMD::dwStart](#) - starting time code
3557 [MEDIACMD::dwEnd](#) - duration
3558 [MEDIACMD::arbID](#) - Filename, null if no file writing
gsPreviewFrame Get a preview frame from AvHAL (usually via the network)

3559 cmdType::ctSetValue
3560 [MEDIACMD::dwPosition](#) - Frame size
3561 [MEDIACMD::arbID](#) - Image Data

3562 cmdType::ctGetValue
3563 [MEDIACMD::dwPosition](#) - Frame Ms
3564 [MEDIACMD::dwStart](#) - frame Width
3565 [MEDIACMD::dwEnd](#) - Frame height
3566 [MEDIACMD::lSpeed](#) - Frame Pitch
3567 [MEDIACMD::arbID](#) - BYTE * for frame
gsVWService Service

3568 cmdType::ctSetValue
3569
3570 cmdType::ctGetValue

3571
gsInsertQueue For clip copy and translation lists: Set removes an item, Get returns the list like a cliplist

3572 cmdType::ctSetValue
3573
3574 cmdType::ctGetValue

3575
gsXlatQueue For clip copy queue manipulation

3576 cmdType::ctSetValue
3577
3578 cmdType::ctGetValue

3579
gsXMLRateScale Set the rate and scale in an XML file for later opening

3580 cmdType::ctSetValue
3581
3582 cmdType::ctGetValue

3583

gsXMLFileProperties Set the rate and scale in an XML file for later opening

3584cmdType::ctSetValue

3585cmdType::ctGetValue

3586

gsDTProjectToXml Set the clip file EDL or settings for XML export

3587cmdType::ctSetValue

3588[MEDIACMD::dwPosition](#) - XML Value below

3589[MEDIACMD::dwStart](#) - DWORD setting

3590[MEDIACMD::arbID](#) - Filename or string value

3591cmdType::ctGetValue

3592

gsApplicationID Set the calling application to allow for app specific behaviors of the DDR

3593cmdType::ctSetValue

3594[MEDIACMD::dwPosition](#) - [GS_APP_NONE](#), [GS_APP_QUICKCLIP](#),
[GS_APP_QUICKCLIPXO](#), [GS_APP_VTRID](#), [GS_APP_MEDIANXS](#),
[GS_APP_DTREPLAYLIVE](#), [GS_APP_DTOUCH](#)

3595

3596cmdType::ctGetValue

3597[MEDIACMD::dwPosition](#) - [GS_APP_NONE](#), [GS_APP_QUICKCLIP](#),
[GS_APP_QUICKCLIPXO](#), [GS_APP_VTRID](#), [GS_APP_MEDIANXS](#),
[GS_APP_DTREPLAYLIVE](#), [GS_APP_DTOUCH](#)

3598

gsInlay Set the Video Inlay enabled = 1 or disabled = 0

3599cmdType::ctSetValue

3600[MEDIACMD::dwPosition](#) - Enable Inlay

3601cmdType::ctGetValue

[MEDIACMD::dwPosition](#) - Inlay Enabled

3602

gsInlayFile Set the Video Inlay File

3603cmdType::ctSetValue

3604[MEDIACMD::arbID](#) - Inlay Source File

3605cmdType::ctGetValue

[MEDIACMD::arbID](#) - Inlay Source File

3606

gsInlaySourceArea Set the Inlay Source Rect

3607cmdType::ctSetValue

3608[MEDIACMD::dwPosition](#) - Inlay Source X

3609[MEDIACMD::dwStart](#) - Inlay Source Y

3610[MEDIACMD::dwEnd](#) - Inlay Source Height

3611[MEDIACMD::lSpeed](#) - Inlay Source Width

3612cmdType::ctGetValue

3613[MEDIACMD::dwPosition](#) - Inlay Source X

3614[MEDIACMD::dwStart](#) - Inlay Source Y

3615[MEDIACMD::dwEnd](#) - Inlay Source Height

[MEDIACMD::lSpeed](#) - Inlay Source Width

3616

gsInlayDestinationArea Set the Inlay Destination Rect

3617cmdType::ctSetValue

3618[MEDIACMD::dwPosition](#) - Inlay Destination X

3619[MEDIACMD::dwStart](#) - Inlay Destination Y

3620[MEDIACMD::dwEnd](#) - Inlay Destination Height

3621[MEDIACMD::lSpeed](#) - Inlay Destination Width

3622cmdType::ctGetValue

3623[MEDIACMD::dwPosition](#) - Inlay Destination X

3624[MEDIACMD::dwStart](#) - Inlay Destination Y

3625[MEDIACMD::dwEnd](#) - Inlay Destination Height

[MEDIACMD::lSpeed](#) - Inlay Destination Width

3626

gsInlayOffset Set the frame offset of the Inlay Source vs Destination

3627cmdType::ctSetValue

3628[MEDIACMD::dwPosition](#) - Inlay frame offset

3629cmdType::ctGetValue

[MEDIACMD::dwPosition](#) - Inlay frame offset

3630

gsInlayTcType Set the frame of the Inlay Source as DF / NDF

3631cmdType::ctSetValue

3632[MEDIACMD::dwPosition](#) - TC_TYPE_DF / TC_TYPE_NDF

3633cmdType::ctGetValue

[MEDIACMD::dwPosition](#) - Inlay tc source (TC_TYPE_DF / TC_TYPE_NDF)

3634

gsExportClip Exports a section of a file

3635cmdType::ctSetValue

3636 [MEDIACMD::dwStart](#) = In point;
3637 [MEDIACMD::dwEnd](#) = Out point;
3638 [MEDIACMD::dwPosition](#) = Camera Number;
3639 [MEDIACMD::lSpeed](#) = Mark Number;
3640 [MEDIACMD::arbID](#)[0] = Source File Name;

3641 cmdType::ctGetValue
3642 [MEDIACMD::dwStart](#) = This meld finished;
3643 [MEDIACMD::dwEnd](#) = This meld total;
[MEDIACMD::dwPosition](#) - Progress x / 1000

3644
gsExportClipDirectory Sets Export Directory

3645 cmdType::ctSetValue
3646 [MEDIACMD::arbID](#)[0] = New directory Name;

3647 cmdType::ctGetValue
[MEDIACMD::arbID](#)[0] = directory Name;

3648
gsExportFileName Sets Export FileName

3649 cmdType::ctSetValue
3650 [MEDIACMD::arbID](#)[0] = New FileName

3651 cmdType::ctGetValue
[MEDIACMD::arbID](#)[0] = FileName;

3652
gsImportClipDirectory Sets Import Directory

3653 cmdType::ctSetValue
3654 [MEDIACMD::arbID](#)[0] = New directory Name;

3655 cmdType::ctGetValue
[MEDIACMD::arbID](#)[0] = directory Name;

3656
gsFileSegmentSize Set File Length for segmentation support

3657 cmdType::ctSetValue
3658 [MEDIACMD::dwPosition](#) = new segment length;

3659 cmdType::ctGetValue
[MEDIACMD::dwPosition](#) = segment length;

3660
gsRecordFileUpdateFrames Set Clip Info update frame rate

3661 cmdType::ctSetValue
3662 [MEDIACMD::dwPosition](#) = new update rate in frames;

3663cmdType::ctGetValue
[MEDIACMD::dwPosition](#) = update rate in frames;

3664
gsCheckFrameTimeStamp Set AvHal to monitor incoming frame rate to make ms adjustments This always starts up as off, hurricane sets this on on the dialog init

3665cmdType::ctSetValue
3666[MEDIACMD::dwPosition](#) = on / off;

3667cmdType::ctGetValue
[MEDIACMD::dwPosition](#) = not supported;

3668
gsShutdownApplication Close the DDR and currently running application

3669cmdType::ctSetValue
3670[MEDIACMD::dwPosition](#) - Must be 0x01010101
3671[MEDIACMD::dwStart](#) - Must be 0xA5A5A5A5
3672[MEDIACMD::dwEnd](#) - Must be 0x5F5F5F5F
3673
3674cmdType::ctGetValue Not supported
3675
gsShutdownSystem Close the DDR and currently running application

3676cmdType::ctSetValue
3677[MEDIACMD::dwPosition](#) - Must be 0x11111111
3678[MEDIACMD::dwStart](#) - Must be 0x5F5F5F5F
3679[MEDIACMD::dwEnd](#) - Must be 0xA5A5A5A5
3680[MEDIACMD::cfFlags](#) - If 0x1A1A1A1A, call restart
3681
3682cmdType::ctGetValue Not supported
3683
gsCleanRecordWipeDrive Clean off the root, or the whold record derive

3684cmdType::ctSetValue
3685[MEDIACMD::dwPosition](#) - Must be GS_CLEANRECORDWIPE_ROOTDIR || GS_CLEANRECORDWIPE_WHOLEDRIIVE
3686[MEDIACMD::dwStart](#) - Must be 0x5F5F5F5F
3687[MEDIACMD::dwEnd](#) - Must be 0xA5A5A5A5
3688
3689cmdType::ctGetValue Not supported
3690
gsInstallSystem Initiate software install

3691cmdType::ctSetValue
3692[MEDIACMD::dwPosition](#) - Must be 0x2B2B2B2B
3693[MEDIACMD::dwStart](#) - Must be 0x11111111
3694[MEDIACMD::dwEnd](#) - Must be 0x4E4E4E4E
3695[MEDIACMD::arbID](#) - Networkpath to install file (must have shared drive for this)
3696
3697cmdType::ctGetValue Not supported

3698

gsInternalGetImageOffset Get the optimal offset for a video frame to allow a header to be added

Definition at line 519 of file mediacmd.h.

enum [cmdinf](#)

Enumerator:

infLtc LTC time code user bit channel.

infVtc VITC time code user bit channel.

infSrcCtl Incoming source control time code.

infSrcLtc Incoming source LTC time code user bits.

infSrcVtc Incoming source VITC time code user bits.

infRecTime Record time of day.

infRecDate Record Data.

infCC Close caption information.

infAuth Authorization information.

infCopyright Copyright information.

infOwner Ownership information.

infSourceName Source media name.

infProxyName Source proxy name (if any)

infI3 Unused inf13 - inf21.

inf14

inf15

inf16

inf17

inf18

inf19

inf20

inf21

infVB0

infVB1

infVB2

infVB3

infVB4

infVB5

infVB6

infVB7

infVB8

infVB9

Definition at line 470 of file mediacmd.h.

enum [cmdType](#)

The legal commands for a [MEDIACMD](#) structure. Set [MEDIACMD::ctCmd](#) to one of these values and expect it to be set to one of these values on a valid return

Enumerator:

ctStop Stop - Stop all playback, and normally place all channels into passthrough

ctPause Pause - Halt all channels. Display current video frame and silent audio
Seek - With [cmdFlags::cfUsePosition](#) and [MEDIACMD::dwPosition](#), go to that frame and Pause

ctPlay Play - Play all channels. May be modified by ([cmdFlags::cfUseSpeed](#) + [MEDIACMD::lSpeed](#)) and
([cmdFlags::cfUsePosition](#) or [cmdFlags::cfUsePositionOffset](#)) and [MEDIACMD::dwPosition](#)) or
([cmdFlags::cfUseStart](#) or [cmdFlags::cfUseStartOffset](#)) and [MEDIACMD::dwStart](#)) or
([cmdFlags::cfUseEnd](#) or [cmdFlags::cfUseEndOffset](#)) and [MEDIACMD::dwEnd](#)) as well as
[MEDIACMD::dwCmdAlt](#) with certain [cmdFlags](#) to play from-top, at speed or combinations of the above.

ctRecord Record - Record one or a combination of video/audio/info to disk. May be modified, as with [ctPlay](#) by flags and structure members such as
([cmdFlags::cfUsePosition](#) or [cmdFlags::cfUsePositionOffset](#)) and [MEDIACMD::dwPosition](#)) or
([cmdFlags::cfUseStart](#) or [cmdFlags::cfUseStartOffset](#)) and [MEDIACMD::dwStart](#)) or
([cmdFlags::cfUseEnd](#) or [cmdFlags::cfUseEndOffset](#)) and [MEDIACMD::dwEnd](#)) as well as
([cmdFlags::cfUseClipID](#) and [MEDIACMD::arbID](#)) or
([cmdFlags::cfDeferred](#) or [cmdFlags::cfOverrideDeferred](#)) or [MEDIACMD::dwCmdAlt](#) with certain [cmdFlags](#) to record from-top, at speed or combinations of the above.

ctRecStop Record Stop - Set the channel into a record ready state, normally passthrough with the recording file pre-allocated, and possible pass start, end and name information. See
([cmdFlags::cfUseStart](#) [cmdFlags::cfUseStartOffset](#) and [MEDIACMD::dwStart](#)),
([cmdFlags::cfUseEnd](#) [cmdFlags::cfUseEndOffset](#) and [MEDIACMD::dwEnd](#)),
([cmdFlags::cfUseClipID](#) and [MEDIACMD::arbID](#)) for record setups.

ctEject Eject - Stop the channel and unload removable media, if possible, else same as stop

ctTransfer Transfer - Transfer media from one channel to another. Normally used to transfer internal media to or from an external tape device.

ctInsert Insert Clip or Timecode Area - Used in time code space (TCSpace.h) and Clip Space (ClipSpace.h) to add new clips or areas. Inserted media is defined by (cmdFlags::cfUseStart - [MEDIACMD::dwStart](#), cmdFlags::cfUseEnd - [MEDIACMD::dwEnd](#)) for clip being added and (cmdFlags::cfUsePosition - [MEDIACMD::dwPosition](#)) for target. Also (cmdFlags::cfUseClipID and [MEDIACMD::arbID](#)) may be used to specify a file name. #cmdFlags::cfUsePresets and [MEDIACMD::dwVideoChannels](#), [MEDIACMD::dwAudioChannels](#), [MEDIACMD::dwInfoChannels](#) are also respected if set. #cmdFlags::cfRipple may also be used to insert over

NOTE - The ctTransfer command is ALWAYS sent to the target with the SOURCE channel in the [MEDIACMD::dwCmdAlt](#) member and (cmdFlags::cfUseCmdAlt) set UNLESS one of the devices is slow/high latency/sloppy (read VTR), in which case it always receives the command so it can master the transfer and the (cmdFlags::cfInvert) is used to set the direction.

ctBlank Blank a Timecode Area - Used in time code space (TCSpace.h) to set an area to black and silent audio. (cmdFlags::cfUseStart - [MEDIACMD::dwStart](#), cmdFlags::cfUseEnd - [MEDIACMD::dwEnd](#)) set the area to be blanked. cmdFlags::cfUsePresets and [MEDIACMD::dwVideoChannels](#), [MEDIACMD::dwAudioChannels](#), [MEDIACMD::dwInfoChannels](#) are also respected if set. cmdFlags::cfRipple may also be used to remove blank area. With this command, no media is removed from storage.

ctDelete Delete a clip (ClipSpace.h) or an area (TCSpace.h). Deletes the media from storage and from the current space. For ClipSpace, cmdFlags::cfUseClipID and [MEDIACMD::arbID](#) must be specified, and if any sub clip or super clips exist, the ID will be removed but the media will not be deleted. For TCSpace, cmdFlags::cfUseStart and [MEDIACMD::dwStart](#) with cmdFlags::cfUseEnd and [MEDIACMD::dwEnd](#) should be used to specify the time code segment to be deleted. cmdFlags::cfUsePresets and [MEDIACMD::dwVideoChannels](#), [MEDIACMD::dwAudioChannels](#), [MEDIACMD::dwInfoChannels](#) may also be used to delete specific channels. If cmdFlags::cfRipple is set, then the TCSpace will close over the deleted material, changing all timecode location beyond the deletion point by minus the size of the deletion.

ctTrim Trim a clip or area - Currently not implemented. Use cmdType::ctSetValue and #cmdGetSetValue::gsClipInfo to trim a clip, or a combination of cmdType::ctInsert, cmdType::ctDelete, cmdType::ctBlank to trim a tcspace area.

ctChanSelect Channel select - select active channels, preview passthrough channels (to preview and edit) recording channels (to create a split edit a la CMX) Requires cmdFlags::cfUsePresets and [MEDIACMD::dwVideoChannels](#), [MEDIACMD::dwAudioChannels](#), [MEDIACMD::dwInfoChannels](#)

ctGetState Get the current state of the controlled channel(s) - Fills the user supplied [MEDIACMD](#) structure with the current state. Look for cmdType::ctError, cmdType::ctStop, cmdType::ctEject, cmdType::ctPause, cmdType::ctPlay, cmdType::ctRecStop, cmdType::ctRecord for basic state. For valid fields, check

3699cmdFlags::cfDeferred : we have a deferred clip

3700cmdFlags::cfTimeMs : [MEDIACMD::dwCmdAlt](#) has millisecond performance counter info

3701cmdFlags::cfUseSpeed : [MEDIACMD::ISpeed](#) has the valid current speed

3702cmdFlags::cfUsePresets : [MEDIACMD::dwVideoChannels](#), [MEDIACMD::dwAudioChannels](#), [MEDIACMD::dwInfoChannels](#) contain preset information
3703cmdFlags::cfUsePosition : [MEDIACMD::dwPosition](#) contains current position
3704cmdFlags::cfUseStart : [MEDIACMD::dwStart](#) has starting frame position
3705cmdFlags::cfUseEnd : [MEDIACMD::dwEnd](#) has end frame position (+1 the Out is never included)
3706cmdFlags::cfUseClipID : [MEDIACMD::arbID](#) has current clip name (8 char for Louth and Odetics)
3707cmdFlags::cfFields : [MEDIACMD::dwPosition](#), [MEDIACMD::dwStart](#) and [MEDIACMD::dwEnd](#) are in fields if they are valid
3708cmdFlags::cfNoReturn : return is invalid.

ctSetState Set the current state - Used for control type channels such as Serial 422 control (vwwCtl.h) and network control (vwwNet.h). Tells the controller or user what our current state is. The state should be reported honestly, as it is the receiver's responsibility to transition states in an appropriate way for its controller. For actual channels (vwwInt.h, vwwExt.h, vwwNet.h-as controller, vwwDS2.h, etc), the state should be set by using one of the transport commands (cmdType::ctPlay etc) above.

ctGetValue Get a non transport setting - Used for one time setups on channel. Includes audio levels, video proc amps, audio/video input, compression type and level and many others See: [cmdGetSetValue](#) for possible commands

ctSetValue Set a non transport setting - Used for one time setups on channel. Includes audio levels, video proc amps, audio/video input, compression type and level and many others See: [cmdGetSetValue](#) for possible commands

ctValueSupported Check support for a non transport setting - Used for one time setups on channel. Includes audio levels, video proc amps, audio/video input, compression type and level and many others See: [cmdGetSetValue](#) for possible commands

ctError Indicates that an error in the channel has occurred. Return only. See [MEDIACMD::dwCmdAlt](#) for error code and [MEDIACMD::arbID](#) for message if any. These members will be valid if cmdFlags::cfUseCmdAlt and cmdFlags::cfUseClipID are set

ctTerminate Terminate Close A Channel - Only used by remote devices that cannot close the channel directly such as vwwNet.h. Channel may not actually close when this is called, but the communications pipe will be closed and wait for another connection.

ctAbort Abort the current operation - Use to abort operations that would normally ignore extraneous commands such as non-linear playback sequences, records or if the channel just seems to be stuck. Makes a good panic button.

ctEdit Edit - this is an internal state set by an edit record (e.g. a record that is not using all channels, so some channels are playing). The two cases we are going to support with AJA will be 'record video + play all audio' and 'play video + record some audio'. In the second case audio channels that a VTR would normally play (not armed) will not be

played, and will not be recorded.

ctSwitch Switch file source (later live source?) without changing mode or speed (for replay)

Definition at line 97 of file `mediacmd.h`.

enum [cmdVidChan](#)

Video channel bit array for [MEDIACMD::dwVideoChannels](#)

Enumerator:

vidChan0
vidChan1
vidChan2
vidChan3
vidChan4
vidChan5
vidChan6
vidChan7
vidChan8
vidChan9
vidChan10
vidChan11
vidChan12
vidChan13
vidChan14
vidChan15
vidChan16
vidChan17
vidChan18
vidChan19
vidChan20
vidChan21
vidChan22
vidChan23
vidChan24
vidChan25
vidChan26
vidChan27
vidChan28
vidChan29
vidChan30
vidChan31

Definition at line 440 of file `mediacmd.h`.

E:/drastic/api/mediacmd/src/timecode.h File Reference

Defines

3709#define [TC2_MIN_STRING_SIZE](#) 15

```

3710#define TC2\_TCTYPE\_MASK 0x000000FF
Mask to retrieve time code type from dwFlags.
3711#define TC2\_TCTYPE\_FILM 0x00000001
Film 24 FPS time code.
3712#define TC2\_MARK\_FILM '-'
3713#define TC2\_CHAR\_FILM 'F'
3714#define TC2\_TCTYPE\_NDF 0x00000002
Non Drop Frame 30 FPS time code.
3715#define TC2\_MARK\_NDF ':'
3716#define TC2\_CHAR\_NDF 'N'
3717#define TC2\_TCTYPE\_DF 0x00000004
Drop Frame 29.97 FPS time code.
3718#define TC2\_MARK\_DF ';'
3719#define TC2\_CHAR\_DF 'D'
3720#define TC2\_TCTYPE\_PAL 0x00000008
PAL 25 FPS time code.
3721#define TC2\_MARK\_PAL ':'
3722#define TC2\_CHAR\_PAL 'P'
3723#define TC2\_TCTYPE\_50 0x00000010
Double PAL 50 FPS.
3724#define TC2\_MARK\_50 ':'
3725#define TC2\_CHAR\_50 'p'
3726#define TC2\_TCTYPE\_5994 0x00000020
720p DROP 59.94 FPS
3727#define TC2\_MARK\_5994 ':'
3728#define TC2\_CHAR\_5994 'd'
3729#define TC2\_TCTYPE\_5997 0x00000022
720p DROP 59.97 FPS
3730#define TC2\_MARK\_5997 ':'
3731#define TC2\_CHAR\_5997 'd'
3732#define TC2\_TCTYPE\_60 0x00000040
720p 60 FPS
3733#define TC2\_MARK\_60 ':'
3734#define TC2\_CHAR\_60 'n'
3735#define TC2\_TCTYPE\_NTSCFILM 0x00000080
23.98 FILM for NTSC 23.98 FPS (This is actually 24)
3736#define TC2\_MARK\_NTSCFILM '_'
3737#define TC2\_CHAR\_NTSCFILM 'f'
3738#define TC2\_TCTYPE\_2398 0x00000084
23.98 TRUE (actual 23.98 drop per Avid)
3739#define TC2\_MARK\_2398 '+'
3740#define TC2\_CHAR\_2398 't'
3741#define TC2\_TCTYPE\_100 0x00000044
Hundredths of a second HH:MM:SS:/100 100 FPS effective.
3742#define TC2\_MARK\_100 '^'
3743#define TC2\_CHAR\_100 '0'
3744#define TC2\_TCTYPE\_IRIG 0x00000045
IRIG time code, uses both time code and user bits.
3745#define TC2\_MARK\_IRIG ':'

```

```

3746#define TC2_CHAR_IRIG 'I'
3747#define TC2_IRIG_IS_VALID 0xA0000000
IRIG is valid marker.
3748#define TC2_IRIGFLAGS_MASK 0x0F000000
IRIG mask for sub type.
3749#define TC2_IRIGFLAGS_TYPE_A 0x01000000
IRIG type A 10 fps, 0.1 sec interval.
3750#define TC2_IRIGFLAGS_TYPE_B 0x02000000
IRIG type B 1 fps, 1 sec interval.
3751#define TC2_IRIGFLAGS_TYPE_D 0x03000000
IRIG type D 1 fph, 1 hour interval.
3752#define TC2_IRIGFLAGS_TYPE_E 0x04000000
IRIG type E 6 fpm, 10 sec interval.
3753#define TC2_IRIGFLAGS_TYPE_G 0x05000000
IRIG type A 100 fps, 10 millisec interval.
3754#define TC2_IRIGFLAGS_TYPE_H 0x06000000
IRIG type A 1 fpm, 1 minute interval.
3755#define TC2_IRIGSHIFT_CPS 0
3756#define TC2_IRIGMASK_CPS 0x00003FFF
3757#define TC2_IRIGSHIFT_SEC 14
3758#define TC2_IRIGMASK_SEC 0x000FC000
3759#define TC2_IRIGSHIFT_MIN 20
3760#define TC2_IRIGMASK_MIN 0x03F00000
3761#define TC2_IRIGSHIFT_HOUR 26
3762#define TC2_IRIGMASK_HOUR 0xFC000000
3763#define TC2_IRIGSHIFT_DAY 0
3764#define TC2_IRIGMAKS_DAY 0x000001FF
3765#define TC2_IRIGSHIFT_YEAR 9
3766#define TC2_IRIGMASK_YEAR 0x0001FE00
3767#define TC2_IRIGMASK_FLAGS 0xFF000000
3768#define TC2_IRIGMASK_VALID 0xF0000000
3769#define TC2_FTYPE_FIELD 0x10000000
Frame conversion to/from time code is in fields.
3770#define TC2_STRTYPE_MASK 0x00000F00
Basic sting tc representation types area in dwFlags.
3771#define TC2_STRTYPE_ASCII 0x00000100
ANSI ASCII values (human readable)
3772#define TC2_STRTYPE_BCD 0x00000200
Binary Coded Decimal (RS-422 style)
3773#define TC2_STRTYPE_HEX 0x00000400
Hexadecimal pack in DWORD 0xHHMMSSFF.
3774#define TC2_STRTYPE_GOP 0x00000800
MPEG GOP time code encoded.
3775#define TC2_STRTYPE_INVERT 0x00001000
Invert the order from HHMMSSFF to FFSSMMHH.
3776#define TC2_STRTYPE_OVER_24_HOUR 0x00002000
Allow time codes larger then 24 hours - 1 frame.
3777#define TC2_STREXT_MASK 0xFFFF0000

```

Extended string handling area in dwTypes.

3778#define [TC2_STREXT_MARKS](#) 0x00010000

Add ':' marks between hours, minutes, seconds, frames in ASCII strings.

3779#define [TC2_STREXT_LEADING](#) 0x00020000

Fill in the leading zeros for time code less than 10 hours.

3780#define [TC2_STREXT_TYPE](#) 0x00040000

Add a single letter type ID at the end of the string 'N', 'D', 'P' or 'F'.

3781#define [TC2_STREXT_ALLCOLON](#) 0x00080000

Use only colons. Normally type is indicated by the second/frames separator.

3782#define [TC2_STREXT_FLAG](#) 0x00100000

Add Drop Frame flags for BCD time codes.

3783#define [TC2_STREXT_CF](#) 0x00200000

Add color frame flag for BCD time codes.

3784#define [TC2_STREXT_MAX30](#) 0x00400000

Do not allow frame values greater than 29 (24 in PAL)

3785#define [TC2_STREXT_REVERSE](#) 0x00800000

If generating LTC/VITC, are we playing in reverse.

3786#define [TC2_STREXT_TYPE_MASK](#) 0x0F000000

*For IRIG TC2_IRIGFLAGS_TYPE_**

3787#define [TC2_STREXT_ISBSTR](#) 0x10000000

Is the ASCII string a BSTR (ActiveX, VB)

3788#define [TC2_STREXT_SHOW_MONTH_YEAR](#) 0x20000000

Show month/year, if available.

3789#define [TC2_STREXT_SHIFT7](#) 0x40000000

Is it a GOP tc that will insert directly into a string?

3790#define [TC2_STREXT_SAVEBITS](#) 0x80000000

Do we need to save the non time code bits when writing GOP tc.

3791#define [TC2_GOP25TC_TO_HEXINVERT](#)(__dwGOP25Bit_) (((__dwGOP25Bit_ & 0xF80000) << 5) | ((__dwGOP25Bit_ & 0x07E000) << 3) | ((__dwGOP25Bit_ & 0x000FC0) << 2) | (__dwGOP25Bit_ & 0x00003F));

3792#define [TC2_GOP25TC_TO_HEXINVERT_ADD_DF](#)(__dwGOP25Bit_) (__dwGOP25Bit_ | 0x20000000)

3793#define [TC2_HEXINVERT_TO_GOP25TC](#)(__dwHEXInvert_) (((__dwHEXInvert_ & 0x1F000000) >> 5) | ((__dwHEXInvert_ & 0x003F0000) >> 3) | ((__dwHEXInvert_ & 0x00003F00) >> 2) | (__dwHEXInvert_ & 0x000000003F) | 0x01001000)

3794#define [TC2_HEXINVERT_TO_GOP25TC_ADD_DF](#)(__dwGOP25Bit_) (__dwGOP25Bit_ | 0x1000000)

3795#define [TC2_ILLEGAL](#) 0xFFFFFFFF

Illegal time code for invalid requests.

3796#define [TC2_SAFE_MAX_FRAME](#) ((2591999*2)+1)

Safe max frames (use > for illegal) at 60fps.

Functions

3797**DWORD** __stdcall [tc2Maximum](#) (**DWORD** dwFlags)

3798**DWORD** __stdcall [tctype2String](#) (**DWORD** dwFlags, **LPSTR** pszTcType)

3799**DWORD** __stdcall [tc2String](#) (**DWORD** dwTcIn, **DWORD** dwFlags, **LPSTR** pszTcOut)

3800**DWORD** __stdcall [tc2Frame](#) (**LPSTR** pszTcIn, **DWORD** dwFlags, **DWORD** *pdwTcOut)

3801 [DWORD](#) __stdcall [ub2String](#) ([DWORD](#) dwUbIn, [DWORD](#) dwFlags, [LPSTR](#) pszUbOut)
3802 [DWORD](#) __stdcall [ub2Value](#) ([LPSTR](#) pszUbIn, [DWORD](#) dwFlags, [DWORD](#) *pdwUbOut)
3803 [DWORD](#) __stdcall [tc2AddColons](#) ([LPSTR](#) pszTcOut, [LPSTR](#) pszTcIn, [DWORD](#) dwFlags)

Define Documentation

#define TC2_CHAR_100 '0'

Definition at line 77 of file timecode.h.

#define TC2_CHAR_2398 't'

Definition at line 73 of file timecode.h.

#define TC2_CHAR_50 'p'

Definition at line 53 of file timecode.h.

#define TC2_CHAR_5994 'd'

Definition at line 57 of file timecode.h.

#define TC2_CHAR_5997 'd'

Definition at line 61 of file timecode.h.

#define TC2_CHAR_60 'n'

Definition at line 65 of file timecode.h.

#define TC2_CHAR_DF 'D'

Definition at line 45 of file timecode.h.

#define TC2_CHAR_FILM 'F'

Definition at line 37 of file timecode.h.

#define TC2_CHAR_IRIG 'I'

Definition at line 81 of file timecode.h.

#define TC2_CHAR_NDF 'N'

Definition at line 41 of file timecode.h.

```
#define TC2_CHAR_NTSCFILM 'f'
```

Definition at line 69 of file timecode.h.

```
#define TC2_CHAR_PAL 'P'
```

Definition at line 49 of file timecode.h.

```
#define TC2_FTYPE_FIELD 0x10000000
```

Frame conversion to/from time code is in fields.

Definition at line 129 of file timecode.h.

```
#define TC2_GOP25TC_TO_HEXINVERT( __dwGOP25Bit_ ) (((__dwGOP25Bit_ & 0xF80000) << 5) | ((__dwGOP25Bit_ & 0x07E000) << 3) | ((__dwGOP25Bit_ & 0x000FC0) << 2) | (__dwGOP25Bit_ & 0x00003F));
```

Convert a 25 bit GOP time code to an inverted hexadecimal time code

Definition at line 185 of file timecode.h.

```
#define TC2_GOP25TC_TO_HEXINVERT_ADD_DF( __dwGOP25Bit_ ) (__dwGOP25Bit_ | 0x20000000)
```

Convert a 25 bit GOP time code to an inverted hexadecimal time code with the drop frame flag.

Definition at line 189 of file timecode.h.

```
#define TC2_HEXINVERT_TO_GOP25TC( __dwHEXInvert_ ) (((__dwHEXInvert_ & 0x1F000000) >> 5) | ((__dwHEXInvert_ & 0x003F0000) >> 3) | ((__dwHEXInvert_ & 0x00003F00) >> 2) | (__dwHEXInvert_ & 0x000000003F) | 0x01001000)
```

Convert an inverted hexadecimal time code to a 25 bit GOP time code

Definition at line 192 of file timecode.h.

```
#define TC2_HEXINVERT_TO_GOP25TC_ADD_DF( __dwGOP25Bit_ ) (__dwGOP25Bit_ | 0x1000000)
```

Convert an inverted hexadecimal time code to a 25 bit GOP time code with the drop frame flag set

Definition at line 196 of file timecode.h.

```
#define TC2_ILLEGAL 0xFFFFFFFF
```

Illegal time code for invalid requests.

Definition at line 204 of file timecode.h.

#define TC2_IRIG_IS_VALID 0xA0000000

IRIG is valid marker.

IRIG time code representation : User Bits (ranges are inclusive) 31~28 - 4 bits = 0xA Is valid
IRIG 27~24 - 4 bits, IRIG type flag 23~18 - 6 bits, unused 17~9 - 8 bits, year (up to 99) 8~0 -
9 bits, day (up to 366 from 1) Time Code (ranges are inclusive) 31~27 - 5 bits, hours (up to
24) 26~21 - 6 bits, minutes (up to 60) 20~15 - 6 bits, seconds (up to 60) 14~0 - 14 bits, 10000
cps max, see type

Definition at line 97 of file timecode.h.

#define TC2_IRIGFLAGS_MASK 0x0F000000

IRIG mask for sub type.

Definition at line 99 of file timecode.h.

#define TC2_IRIGFLAGS_TYPE_A 0x01000000

IRIG type A 10 fps, 0.1 sec interval.

Definition at line 101 of file timecode.h.

#define TC2_IRIGFLAGS_TYPE_B 0x02000000

IRIG type B 1 fps, 1 sec interval.

Definition at line 103 of file timecode.h.

#define TC2_IRIGFLAGS_TYPE_D 0x03000000

IRIG type D 1 fph, 1 hour interval.

Definition at line 105 of file timecode.h.

#define TC2_IRIGFLAGS_TYPE_E 0x04000000

IRIG type E 6 fpm, 10 sec interval.

Definition at line 107 of file timecode.h.

#define TC2_IRIGFLAGS_TYPE_G 0x05000000

IRIG type A 100 fps, 10 millisecc interval.

Definition at line 109 of file timecode.h.

#define TC2_IRIGFLAGS_TYPE_H 0x06000000

IRIG type A 1 fpm, 1 minute interval.

Definition at line 111 of file timecode.h.

#define TC2_IRIGMAKS_DAY 0x000001FF

Definition at line 122 of file timecode.h.

#define TC2_IRIGMASK_CPS 0x00003FFF

Definition at line 114 of file timecode.h.

#define TC2_IRIGMASK_FLAGS 0xFF00000

Definition at line 125 of file timecode.h.

#define TC2_IRIGMASK_HOUR 0xFC00000

Definition at line 120 of file timecode.h.

#define TC2_IRIGMASK_MIN 0x03F0000

Definition at line 118 of file timecode.h.

#define TC2_IRIGMASK_SEC 0x000FC00

Definition at line 116 of file timecode.h.

#define TC2_IRIGMASK_VALID 0xF000000

Definition at line 126 of file timecode.h.

#define TC2_IRIGMASK_YEAR 0x0001FE0

Definition at line 124 of file timecode.h.

#define TC2_IRIGSHIFT_CPS 0

Definition at line 113 of file timecode.h.

#define TC2_IRIGSHIFT_DAY 0

Definition at line 121 of file timecode.h.

#define TC2_IRIGSHIFT_HOUR 26

Definition at line 119 of file timecode.h.

```
#define TC2_IRIGSHIFT_MIN 20
```

Definition at line 117 of file timecode.h.

```
#define TC2_IRIGSHIFT_SEC 14
```

Definition at line 115 of file timecode.h.

```
#define TC2_IRIGSHIFT_YEAR 9
```

Definition at line 123 of file timecode.h.

```
#define TC2_MARK_100 '^'
```

Definition at line 76 of file timecode.h.

```
#define TC2_MARK_2398 '+'
```

Definition at line 72 of file timecode.h.

```
#define TC2_MARK_50 '.'
```

Definition at line 52 of file timecode.h.

```
#define TC2_MARK_5994 ';' 
```

Definition at line 56 of file timecode.h.

```
#define TC2_MARK_5997 ':'
```

Definition at line 60 of file timecode.h.

```
#define TC2_MARK_60 ':'
```

Definition at line 64 of file timecode.h.

```
#define TC2_MARK_DF ';' 
```

Definition at line 44 of file timecode.h.

```
#define TC2_MARK_FILM '-'
```

Definition at line 36 of file timecode.h.

#define TC2_MARK_IRIG ''

Definition at line 80 of file timecode.h.

#define TC2_MARK_NDF ':'

Definition at line 40 of file timecode.h.

#define TC2_MARK_NTSCFILM '_'

Definition at line 68 of file timecode.h.

#define TC2_MARK_PAL ''

Definition at line 48 of file timecode.h.

#define TC2_MIN_STRING_SIZE 15

Minimum required size of a timecode string

HH:MM:SS:XX D/0

Definition at line 29 of file timecode.h.

#define TC2_SAFE_MAX_FRAME ((2591999*2)+1)

Safe max frames (use > for illegal) at 60fps.

Definition at line 207 of file timecode.h.

#define TC2_STREXT_ALLCOLON 0x00080000

Use only colons. Normally type is indicated by the second/frames separator.

Definition at line 155 of file timecode.h.

#define TC2_STREXT_CF 0x00200000

Add color frame flag for BCD time codes.

Definition at line 159 of file timecode.h.

#define TC2_STREXT_FLAG 0x00100000

Add Drop Frame flags for BCD time codes.

Definition at line 157 of file timecode.h.

#define TC2_STREXT_ISBSTR 0x10000000

Is the ASCII string a BSTR (ActiveX, VB)
Definition at line 167 of file timecode.h.

#define TC2_STREXT_LEADING 0x00020000

Fill in the leading zeros for time code less than 10 hours.
Definition at line 151 of file timecode.h.

#define TC2_STREXT_MARKS 0x00010000

Add ':' marks between hours, minutes, seconds, frames in ASCII strings.
Definition at line 149 of file timecode.h.

#define TC2_STREXT_MASK 0xFFFF0000

Extended string handling area in dwTypes.
Definition at line 147 of file timecode.h.

#define TC2_STREXT_MAX30 0x00400000

Do not allow frame values greater than 29 (24 in PAL)
Definition at line 161 of file timecode.h.

#define TC2_STREXT_REVERSE 0x00800000

If generating LTC/VITC, are we playing in reverse.
Definition at line 163 of file timecode.h.

#define TC2_STREXT_SAVEBITS 0x80000000

Do we need to save the non time code bits when writing GOP tc.
Definition at line 173 of file timecode.h.

#define TC2_STREXT_SHIFT7 0x40000000

Is it a GOP tc that will insert directly into a string?
Definition at line 171 of file timecode.h.

#define TC2_STREXT_SHOW_MONTH_YEAR 0x20000000

Show month/year, if available.
Definition at line 169 of file timecode.h.

#define TC2_STREXT_TYPE 0x00040000

Add a single letter type ID at the end of the string ' N', ' D', ' P' or ' F'.
Definition at line 153 of file timecode.h.

#define TC2_STREXT_TYPE_MASK 0x0F000000

For IRIG TC2_IRIGFLAGS_TYPE_*.
Definition at line 165 of file timecode.h.

#define TC2_STRTYPE_ASCII 0x00000100

ANSI ASCII values (human readable)
Definition at line 134 of file timecode.h.

#define TC2_STRTYPE_BCD 0x00000200

Binary Coded Decimal (RS-422 style)
Definition at line 136 of file timecode.h.

#define TC2_STRTYPE_GOP 0x00000800

MPEG GOP time code encoded.
Definition at line 140 of file timecode.h.

#define TC2_STRTYPE_HEX 0x00000400

Hexadecimal pack in DWORD 0xHHMMSSFF.
Definition at line 138 of file timecode.h.

#define TC2_STRTYPE_INVERT 0x00001000

Invert the order from HHMMSSFF to FFSSMMHH.
Definition at line 142 of file timecode.h.

#define TC2_STRTYPE_MASK 0x00000F00

Basic sting tc representation types area in dwFlags.
Definition at line 132 of file timecode.h.

#define TC2_STRTYPE_OVER_24_HOUR 0x00002000

Allow time codes larger than 24 hours - 1 frame.

Definition at line 144 of file timecode.h.

#define TC2_TCTYPE_100 0x00000044

Hundredths of a second HH:MM:SS:/100 100 FPS effective.

Definition at line 75 of file timecode.h.

#define TC2_TCTYPE_2398 0x00000084

23.98 TRUE (actual 23.98 drop per Avid)

Definition at line 71 of file timecode.h.

#define TC2_TCTYPE_50 0x00000010

Double PAL 50 FPS.

Definition at line 51 of file timecode.h.

#define TC2_TCTYPE_5994 0x00000020

720p DROP 59.94 FPS

Definition at line 55 of file timecode.h.

#define TC2_TCTYPE_5997 0x00000022

720p DROP 59.97 FPS

Definition at line 59 of file timecode.h.

#define TC2_TCTYPE_60 0x00000040

720p 60 FPS

Definition at line 63 of file timecode.h.

#define TC2_TCTYPE_DF 0x00000004

Drop Frame 29.97 FPS time code.

Definition at line 43 of file timecode.h.

#define TC2_TCTYPE_FILM 0x00000001

Film 24 FPS time code.

Definition at line 35 of file timecode.h.

#define TC2_TCTYPE_IRIG 0x00000045

IRIG time code, uses both time code and user bits.

Definition at line 79 of file timecode.h.

#define TC2_TCTYPE_MASK 0x000000FF

Mask to retrieve time code type from dwFlags.

Definition at line 33 of file timecode.h.

#define TC2_TCTYPE_NDF 0x00000002

Non Drop Frame 30 FPS time code.

Definition at line 39 of file timecode.h.

#define TC2_TCTYPE_NTSCFILM 0x00000080

23.98 FILM for NTSC 23.98 FPS (This is actually 24)

Definition at line 67 of file timecode.h.

#define TC2_TCTYPE_PAL 0x00000008

PAL 25 FPS time code.

Definition at line 47 of file timecode.h.

Function Documentation

DWORD __stdcall tc2AddColons (LPSTR pszTcOut, LPSTR pszTcIn, DWORD dwFlags)

copy a string and add colons

DWORD __stdcall tc2Frame (LPSTR pszTcIn, DWORD dwFlags, DWORD * pdwTcOut)

Convert the string pszTcIn to a numeric count pdwTcOut as specified by dwFlags

DWORD __stdcall tc2Maximum (DWORD dwFlags)

Return the maximum possible value for flag combination for a day (24 hours)

DWORD __stdcall tc2String (DWORD dwTcIn, DWORD dwFlags, LPSTR pszTcOut)

Convert the numeric count dwTcIn to the string pszTcOut as specified by dwFlags

DWORD __stdcall tctype2String (**DWORD** dwFlags, **LPSTR** pszTcType)

Set the string pszTcType as specified by dwFlags

DWORD __stdcall ub2String (**DWORD** dwUbIn, **DWORD** dwFlags, **LPSTR** pszUbOut)

Convert the numeric count dwUbIn to the string pszUbOut as specified by dwFlags

DWORD __stdcall ub2Value (**LPSTR** pszUbIn, **DWORD** dwFlags, **DWORD** * pdwUbOut)

Convert the string pszUbIn to a numeric count pdwUbOut as specified by dwFlags

E:/drastic/api/mediacmd/src/vvwif.cpp File Reference

```
#include <unistd.h>
#include <dtutil.h>
#include <malloc.h>
#include <string.h>
#include <sys/time.h>
#include <stdio.h>
#include "dtsystemtypes.h"
#include "mediacmd.h"
#include "dtnetdirect.h"
#include "vwvif.h"
```

Classes

3804struct [tagVVWX_CHANNEL](#)

Defines

```
3805#define dtOSChangeDir
3806#define dtOSLeaveCriticalSection
3807#define MSGAREA "vwvIF"
3808#define VVW_ABS_MAX_CHANNELS 256
3809#define VVW_CONTROL_MAX_EXTERNAL 64U
3810#define VVW_CONTROL_MAX_LOCAL 64U
3811#define VVW_DSINC_CHANNEL 65536
3812#define VVW_CONTROL_START_EXTERNAL _VVW_CONTROL_MAX_LOCAL
3813#define VVW_CONTROL_START_NETWORK (_VVW_CONTROL_MAX_EXTERNAL +
_VVW_CONTROL_MAX_EXTERNAL)
3814#define VVWX_CHANNELFLAG_ISOPEN 0x00000001
3815#define dwLastStatusUpdateMs mCmdState.dwCmdAlt
Last time a status was called.
3816#define MAX_DIRECT_NETWORK_CHANNELS 256
```

Typedefs

```
3817typedef struct tagVVWX\_CHANNEL VVWX_CHANNEL
3818typedef struct tagVVWX\_CHANNEL * pVVWX_CHANNEL
```

Functions

```
3819uint32_t __stdcall strcpyClipIF (char *szClipDst, char *szClipSrc)
3820void __stdcall initMediaCmd (PMEDIACMD pCmd, cmdType ctCmd)
```

3821 long __stdcall [vwwOpeningChannel](#) ()
3822 uint32_t __stdcall [vwwOpenNetworkChannel](#) (char *szAddress, uint32_t dwPort, uint32_t dwChannel)
3823 uint32_t __stdcall [vwwCloseNetworkChannel](#) (VWVIFOPAQUE vvwChannel)
3824 uint32_t __stdcall [vwwCheckNetworkChannel](#) (VWVIFOPAQUE vvwChannel)
3825 VWVHANDLE __stdcall [vwwChannelToHandle](#) (VWVIFOPAQUE vvwChannel)
3826 uint32_t __stdcall [vwwHandleToChannel](#) (VWVHANDLE hVvw)
3827 uint32_t __stdcall [vwwEnableChannels](#) (uint32_t lInternal0_31, uint32_t lInternal32_63, uint32_t lExternal64_95, uint32_t lExternal96_127, uint32_t lNetwork128_159, uint32_t lNetwork160_191)
3828 uint32_t __stdcall [vwwReleaseChannels](#) ()
3829 uint32_t __stdcall [vwwIsOpen](#) (void)
3830 uint32_t __stdcall [vwwGetMaxChannels](#) (void)
3831 uint32_t __stdcall [vwwGetMaxInternalChannels](#) (void)
3832 uint32_t __stdcall [vwwGetChannelName](#) (VWVIFOPAQUE vvwChannel, char *szChannelName)
3833 uint32_t __stdcall [vwwGetChannelType](#) (VWVIFOPAQUE vvwChannel)
3834 [BOOL](#) __stdcall [vwwSetUseAbsoluteTC](#) ([BOOL](#) bUseAbsoluteTC)
3835 uint32_t __stdcall [vwwWaitForState](#) (VWVIFOPAQUE vvwChannel, uint32_t ctCmd)
3836 uint32_t __stdcall [vwwPlay](#) (VWVIFOPAQUE vvwChannel)
3837 uint32_t __stdcall [vwwPlayOffsetAt](#) (VWVIFOPAQUE vvwChannel, uint32_t dwPosition, long lOffset, uint32_t dwMS)
3838 uint32_t __stdcall [vwwPlayAtSpeed](#) (VWVIFOPAQUE vvwChannel, uint32_t lVWVSpeed, uint32_t lEnd)
3839 uint32_t __stdcall [vwwPlayFromTo](#) (VWVIFOPAQUE vvwChannel, uint32_t lFrom, uint32_t lTo, int fDeferred, int fLoop)
3840 uint32_t __stdcall [vwwLoadClip](#) (VWVIFOPAQUE vvwChannel, char *sz8CharClipName, uint32_t lStartFrame)
3841 uint32_t __stdcall [vwwSwitchClip](#) (VWVIFOPAQUE vvwChannel, char *sz8CharClipName, uint32_t lPosition, [BOOL](#) bUseFrameCount)
3842 uint32_t __stdcall [vwwPlayClip](#) (VWVIFOPAQUE vvwChannel, char *sz8CharClipName, int fDeferred)
3843 uint32_t __stdcall [vwwPlayClipFromTo](#) (VWVIFOPAQUE vvwChannel, char *sz8CharClipName, uint32_t lFrom, uint32_t lTo, int fDeferred)
3844 uint32_t __stdcall [vwwPlayAtMs](#) (VWVIFOPAQUE vvwChannel, uint32_t lMs)
3845 uint32_t __stdcall [vwwRecordAtMs](#) (VWVIFOPAQUE vvwChannel, uint32_t lMs, uint32_t lStart, uint32_t lEnd)
3846 uint32_t __stdcall [vwwFastForward](#) (VWVIFOPAQUE vvwChannel)
3847 uint32_t __stdcall [vwwFastRewind](#) (VWVIFOPAQUE vvwChannel)
3848 uint32_t __stdcall [vwwPause](#) (VWVIFOPAQUE vvwChannel)
3849 uint32_t __stdcall [vwwSeek](#) (VWVIFOPAQUE vvwChannel, uint32_t lFrame)
3850 uint32_t __stdcall [vwwSeekRelative](#) (VWVIFOPAQUE vvwChannel, uint32_t lFrameOffset)
3851 uint32_t __stdcall [vwwStop](#) (VWVIFOPAQUE vvwChannel)
3852 uint32_t __stdcall [vwwRecord](#) (VWVIFOPAQUE vvwChannel)
3853 uint32_t __stdcall [vwwRecordFromTo](#) (VWVIFOPAQUE vvwChannel, uint32_t lFrom, uint32_t lTo)
3854 uint32_t __stdcall [vwwRecordStop](#) (VWVIFOPAQUE vvwChannel, char *sz8CharClipName, uint32_t lDuration)
3855 uint32_t __stdcall [vwwRecordStopFileName](#) (VWVIFOPAQUE vvwChannel, char *sz8CharClipName, char *sz256FileName, uint32_t lDuration)
3856 uint32_t __stdcall [vwwSetRecordPresets](#) (VWVIFOPAQUE vvwChannel, uint32_t lIvidEdit, uint32_t lAudEdit, uint32_t lInfEdit)
3857 uint32_t __stdcall [vwwCleanRecordWipeDrive](#) (VWVIFOPAQUE vvwChannel, uint32_t lWipeFolder)
3858 uint32_t __stdcall [vwwEject](#) (VWVIFOPAQUE vvwChannel)
3859 uint32_t __stdcall [vwwMediaCmd](#) (VWVIFOPAQUE vvwChannel, void *pMediaCmd)

3860uint32_t __stdcall [vwwTransfer](#) (VWVIFOPAQUE vvwChannel, uint32_t ITargetChannel, uint32_t IPosition, uint32_t IStart, uint32_t IEnd, uint32_t IVidEdit, uint32_t IAudEdit, uint32_t IInfEdit, char *szClipName, int fToTape)

3861uint32_t __stdcall [vwwUpdateStatus](#) (VWVIFOPAQUE vvwChannel)

3862uint32_t __stdcall [vwwGetState](#) (VWVIFOPAQUE vvwChannel)

3863uint32_t __stdcall [vwwGetFlags](#) (VWVIFOPAQUE vvwChannel)

3864uint32_t __stdcall [vwwGetSpeed](#) (VWVIFOPAQUE vvwChannel)

3865uint32_t __stdcall [vwwGetPosition](#) (VWVIFOPAQUE vvwChannel)

3866uint32_t __stdcall [vwwGetLastMs](#) (VWVIFOPAQUE vvwChannel)

3867uint32_t __stdcall [vwwGetStart](#) (VWVIFOPAQUE vvwChannel)

3868uint32_t __stdcall [vwwGetEnd](#) (VWVIFOPAQUE vvwChannel)

3869uint32_t __stdcall [vwwGetClipName](#) (VWVIFOPAQUE vvwChannel, char *sz8CharClipName)

3870uint32_t __stdcall [vwwGetFileName](#) (VWVIFOPAQUE vvwChannel, char *sz260CharFileName)

3871uint32_t __stdcall [vwwGetCurTC](#) (VWVIFOPAQUE vvwChannel, char *sz14ByteTC)

3872uint32_t __stdcall [vwwGetCurState](#) (VWVIFOPAQUE vvwChannel, char *sz14ByteState)

3873char * __stdcall [vwwGetNextClip](#) (VWVIFOPAQUE vvwChannel, char *sz8CharLastClipCurClip)

3874uint32_t __stdcall [vwwGetClipInfo](#) (VWVIFOPAQUE vvwChannel, char *szClipName, uint32_t *IStart, uint32_t *IEnd, uint32_t *IVidEdit, uint32_t *IAudEdit, uint32_t *IInfEdit, char *szFileName)

3875uint32_t __stdcall [vwwSetMetaData](#) (VWVIFOPAQUE vvwChannel, char *sz8CharClipName, uint32_t vwwInfoRequest, uint32_t nValue, char *szValue)

3876uint32_t __stdcall [vwwGetMetaData](#) (VWVIFOPAQUE vvwChannel, char *sz8CharClipName, char *sz260CharFileName, uint32_t vwwInfoRequest, char *szValue)

3877uint32_t __stdcall [vwwGetNextClipEx](#) (VWVIFOPAQUE vvwChannel, uint32_t *ICreation, uint32_t *ILastModification, uint32_t *IFileSize, uint32_t *IDiskFragments)

3878uint32_t __stdcall [vwwCopyClip](#) (VWVIFOPAQUE vvwChannel, char *szSourceClip, char *szDestClip, uint32_t IStart, uint32_t IEnd)

3879uint32_t __stdcall [vwwEDLResetToStart](#) (VWVIFOPAQUE vvwChannel)

3880uint32_t __stdcall [vwwEDLGetEdit](#) (VWVIFOPAQUE vvwChannel, uint32_t *IRecordIn, uint32_t *IPlayIn, uint32_t *IPlayOut, uint32_t *IVidEdit, uint32_t *IAudEdit, uint32_t *IInfEdit, char *szClipName, char *szFileName, [BOOL](#) bClipInfo)

3881uint32_t __stdcall [vwwEDLSetExtendedInfo](#) (VWVIFOPAQUE vvwChannel, uint32_t IType, uint32_t IRecordIn, uint32_t IValue, char *sz260Comment, uint32_t IDuration)

3882uint32_t __stdcall [vwwEDLGetExtendedInfo](#) (VWVIFOPAQUE vvwChannel, uint32_t IType, uint32_t IRecordIn, char *sz260Comment, uint32_t *plExtra)

3883uint32_t __stdcall [vwwGetLastChangeXferMs](#) (VWVIFOPAQUE vvwChannel)

3884uint32_t __stdcall [vwwGetLastChangeMs](#) (VWVIFOPAQUE vvwChannel, uint32_t IClipSpace, uint32_t *INumClips)

3885uint32_t __stdcall [vwwInsert](#) (VWVIFOPAQUE vvwChannel, char *szClipName, char *szFileName, uint32_t IPosition, uint32_t IStart, uint32_t IEnd, uint32_t IVidEdit, uint32_t IAudEdit, uint32_t IInfEdit, int fRipple)

3886uint32_t __stdcall [vwwBlank](#) (VWVIFOPAQUE vvwChannel, char *szClipName, uint32_t IStart, uint32_t IEnd, uint32_t IVidEdit, uint32_t IAudEdit, uint32_t IInfEdit, int fRipple)

3887uint32_t __stdcall [vwwBlankAllClipIds](#) (VWVIFOPAQUE vvwChannel)

3888uint32_t __stdcall [vwwDelete](#) (VWVIFOPAQUE vvwChannel, char *szClipName, uint32_t IStart, uint32_t IEnd, uint32_t IVidEdit, uint32_t IAudEdit, uint32_t IInfEdit, int fRipple)

3889uint32_t __stdcall [vwwTrim](#) (VWVIFOPAQUE vvwChannel, uint32_t IPosition, uint32_t IStartOffset, uint32_t IEndOffset, uint32_t IVidEdit, uint32_t IAudEdit, uint32_t IInfEdit, int fRipple)

3890uint32_t __stdcall [vwwValueSupported](#) (VWVIFOPAQUE vvwChannel, uint32_t IValueType)

3891uint32_t __stdcall [vwwValueGet](#) (VWVIFOPAQUE vvwChannel, uint32_t IValueType, uint32_t *plMin, uint32_t *plMax)

3892uint32_t __stdcall [vwwValueSet](#) (VWVIFOPAQUE vvwChannel, uint32_t IValueType, uint32_t ISetting)

3893uint32_t __stdcall [vwwValueSet2](#) (VWVIFOPAQUE vvwChannel, uint32_t IValueType, uint32_t

ISetting, uint32_t IStart, uint32_t IEnd, uint32_t IVidChan, uint32_t IAudChan, uint32_t IInfChan)
 3894uint32_t __stdcall [vwwGetClipMode](#) (VWVIFOPAQUE vvwChannel)
 3895uint32_t __stdcall [vwwSetClipMode](#) (VWVIFOPAQUE vvwChannel, uint32_t ISetting)
 3896uint32_t __stdcall [vwwGetTCType](#) (VWVIFOPAQUE vvwChannel)
 3897uint32_t __stdcall [vwwSetTCType](#) (VWVIFOPAQUE vvwChannel, uint32_t ISetting)
 3898uint32_t __stdcall [vwwGetTCSrc](#) (VWVIFOPAQUE vvwChannel)
 3899uint32_t __stdcall [vwwSetTCSrc](#) (VWVIFOPAQUE vvwChannel, uint32_t ISetting)
 3900uint32_t __stdcall [vwwGetAutoMode](#) (VWVIFOPAQUE vvwChannel)
 3901uint32_t __stdcall [vwwSetAutoMode](#) (VWVIFOPAQUE vvwChannel, uint32_t ISetting)
 3902uint32_t __stdcall [vwwGetCurrentPresets](#) (VWVIFOPAQUE vvwChannel, uint32_t *pIVidEdit,
 uint32_t *pIAudEdit, uint32_t *pIInfEdit)
 3903uint32_t __stdcall [vwwGetAvailablePresets](#) (VWVIFOPAQUE vvwChannel, uint32_t *pIVidEdit,
 uint32_t *pIAudEdit, uint32_t *pIInfEdit)
 3904uint32_t __stdcall [vwwGetAudioInput](#) (VWVIFOPAQUE vvwChannel)
 3905uint32_t __stdcall [vwwSetAudioInput](#) (VWVIFOPAQUE vvwChannel, uint32_t ISetting)
 3906uint32_t __stdcall [vwwGetAudioInputLevel](#) (VWVIFOPAQUE vvwChannel)
 3907uint32_t __stdcall [vwwSetAudioInputLevel](#) (VWVIFOPAQUE vvwChannel, uint32_t ISetting)
 3908uint32_t __stdcall [vwwGetAudioOutput](#) (VWVIFOPAQUE vvwChannel)
 3909uint32_t __stdcall [vwwSetAudioOutput](#) (VWVIFOPAQUE vvwChannel, uint32_t ISetting)
 3910uint32_t __stdcall [vwwGetAudioOutputLevel](#) (VWVIFOPAQUE vvwChannel)
 3911uint32_t __stdcall [vwwSetAudioOutputLevel](#) (VWVIFOPAQUE vvwChannel, uint32_t ISetting)
 3912uint32_t __stdcall [vwwGetAudioPeakRMS](#) (VWVIFOPAQUE vvwChannel, uint32_t IAudEdit,
 uint32_t *pIPeaks)
 3913uint32_t __stdcall [vwwGetVideoInput](#) (VWVIFOPAQUE vvwChannel)
 3914uint32_t __stdcall [vwwSetVideoInput](#) (VWVIFOPAQUE vvwChannel, uint32_t ISetting)
 3915uint32_t __stdcall [vwwGetVideoOutput](#) (VWVIFOPAQUE vvwChannel)
 3916uint32_t __stdcall [vwwSetVideoOutput](#) (VWVIFOPAQUE vvwChannel, uint32_t ISetting)
 3917uint32_t __stdcall [vwwGetVideoInputSetup](#) (VWVIFOPAQUE vvwChannel)
 3918uint32_t __stdcall [vwwSetVideoInputSetup](#) (VWVIFOPAQUE vvwChannel, uint32_t ISetting)
 3919uint32_t __stdcall [vwwGetVideoInputVideo](#) (VWVIFOPAQUE vvwChannel)
 3920uint32_t __stdcall [vwwSetVideoInputVideo](#) (VWVIFOPAQUE vvwChannel, uint32_t ISetting)
 3921uint32_t __stdcall [vwwGetVideoInputHue](#) (VWVIFOPAQUE vvwChannel)
 3922uint32_t __stdcall [vwwSetVideoInputHue](#) (VWVIFOPAQUE vvwChannel, uint32_t ISetting)
 3923uint32_t __stdcall [vwwGetVideoInputChroma](#) (VWVIFOPAQUE vvwChannel)
 3924uint32_t __stdcall [vwwSetVideoInputChroma](#) (VWVIFOPAQUE vvwChannel, uint32_t ISetting)
 3925uint32_t __stdcall [vwwGetVideoTBCSetup](#) (VWVIFOPAQUE vvwChannel)
 3926uint32_t __stdcall [vwwSetVideoTBCSetup](#) (VWVIFOPAQUE vvwChannel, uint32_t ISetting)
 3927uint32_t __stdcall [vwwGetVideoTBCVideo](#) (VWVIFOPAQUE vvwChannel)
 3928uint32_t __stdcall [vwwSetVideoTBCVideo](#) (VWVIFOPAQUE vvwChannel, uint32_t ISetting)
 3929uint32_t __stdcall [vwwGetVideoTBCHue](#) (VWVIFOPAQUE vvwChannel)
 3930uint32_t __stdcall [vwwSetVideoTBCHue](#) (VWVIFOPAQUE vvwChannel, uint32_t ISetting)
 3931uint32_t __stdcall [vwwGetVideoTBCChroma](#) (VWVIFOPAQUE vvwChannel)
 3932uint32_t __stdcall [vwwSetVideoTBCChroma](#) (VWVIFOPAQUE vvwChannel, uint32_t ISetting)
 3933uint32_t __stdcall [vwwGetVideoGenlock](#) (VWVIFOPAQUE vvwChannel)
 3934uint32_t __stdcall [vwwSetVideoGenlock](#) (VWVIFOPAQUE vvwChannel, uint32_t ISetting)
 3935uint32_t __stdcall [vwwGetVideoGenlockSource](#) (VWVIFOPAQUE vvwChannel)
 3936uint32_t __stdcall [vwwSetVideoGenlockSource](#) (VWVIFOPAQUE vvwChannel, uint32_t
 ISetting)
 3937uint32_t __stdcall [vwwGetCompressionRate](#) (VWVIFOPAQUE vvwChannel)
 3938uint32_t __stdcall [vwwSetCompressionRate](#) (VWVIFOPAQUE vvwChannel, uint32_t ISetting)
 3939uint32_t __stdcall [vwwGetTotalTime](#) (VWVIFOPAQUE vvwChannel)
 3940uint32_t __stdcall [vwwGetFreeTime](#) (VWVIFOPAQUE vvwChannel)
 3941uint32_t __stdcall [vwwGetTotalStorage](#) (VWVIFOPAQUE vvwChannel)
 3942uint32_t __stdcall [vwwGetFreeStorage](#) (VWVIFOPAQUE vvwChannel)
 3943uint32_t __stdcall [vwwGetCurMs](#) (VWVIFOPAQUE vvwChannel)
 3944uint32_t __stdcall [vwwGetChannelCapabilities](#) (VWVIFOPAQUE vvwChannel)

```

3945char * __stdcall vwwGetVWVVersion ()
3946char * __stdcall vwwGetMRVersion ()
3947char * __stdcall vwwGetVWVType (VWVIFOPAQUE vwwChannel)
3948uint32_t __stdcall vwwGetSuperImpose (VWVIFOPAQUE vwwChannel)
3949uint32_t __stdcall vwwSetSuperImpose (VWVIFOPAQUE vwwChannel, uint32_t lSetting)
3950uint32_t __stdcall vwwSetErrorLog (uint32_t lSetting)
3951uint32_t __stdcall vwwGetErrorLogMs ()
3952uint32_t __stdcall vwwGetNumberOfErrors (uint32_t *plError)
3953uint32_t __stdcall vwwGetErrorLength (uint32_t *plLastError, uint32_t *plErrorLength)
3954uint32_t __stdcall vwwGetError (uint32_t *lLastError, uint32_t *lSeverity, char *szError, short
    *psYear, short *psMonth, short *psDay, short *psHour, short *psMinute, short *psSecond, short
    *psMilliSecond)
3955uint32_t __stdcall vwwGetPiconName (char *szFileName)
3956DWORD __stdcall dtGpiSetModeD (VWVHANDLE hDSync, DWORD dwInOut, DWORD
    dwCmd, DWORD dwSetOn, BOOL bSet, BOOL bTrace)
3957DWORD __stdcall dtGpiSetMode (VWVHANDLE hDSync, DWORD dwInOut, DWORD
    dwCmd, DWORD dwSetOn, BOOL bSet)
3958uint32_t __stdcall vwwXMLProjectSave (char *lpszProjectName)
3959uint32_t __stdcall vwwXMLProjectCheckOpen (char *lpszProjectName)
3960uint32_t __stdcall vwwXMLProjectOpen (char *lpszProjectName, int bDeleteConflictingFiles)
3961uint32_t __stdcall vwwGetNumberOfBackUps (VWVIFOPAQUE vwwChannel, uint32_t
    *pdwBackUps, uint32_t dwClipMode)
3962uint32_t __stdcall vwwSetBackUpNumber (VWVIFOPAQUE vwwChannel, uint32_t dwBackUp,
    uint32_t dwClipMode)
3963uint32_t __stdcall vwwGetLicenseOptions (VWVIFOPAQUE vwwChannel, uint32_t
    *pdwOptions, uint32_t *pdwLicenseValid)
3964void __stdcall vwwFreeString (char *szString)

```

Variables

```

3965pVWVX_CHANNEL garChannels = NULL
3966uint32_t gdwVWVIFInOpen = 0
3967uint32_t gdwOpeningChannel = 0
3968uint32_t gdwMaxChannels = 0
3969uint32_t gdwMaxInternalChannels = 0
3970uint32_t gdwLastModeTC = 0
3971char m_szTCPIPAddr [1025] = ""
3972uint32_t m_dwPORT = 1234

```

Define Documentation

```
#define _VW_ABS_MAX_CHANNELS 256
```

Definition at line 113 of file vwwif.cpp.

```
#define _VW_CONTROL_MAX_EXTERNAL 64U
```

Definition at line 114 of file vwwif.cpp.

```
#define _VW_CONTROL_MAX_LOCAL 64U
```

Definition at line 115 of file vvwif.cpp.

```
#define _VW_CONTROL_START_EXTERNAL _VW_CONTROL_MAX_LOCAL
```

Definition at line 117 of file vvwif.cpp.

```
#define _VW_CONTROL_START_NETWORK (_VW_CONTROL_MAX_EXTERNAL +  
_VW_CONTROL_MAX_EXTERNAL)
```

Definition at line 118 of file vvwif.cpp.

```
#define _VW_DSYNC_CHANNEL 65536
```

Definition at line 116 of file vvwif.cpp.

```
#define dtOSChangeDir
```

: This file is included in VVW.DLL, VVWNET2.DLL (possibly others) and is the main file in the direct MediaCMD sdk. This makes sure that the protocol used in all our apps and our oems, has the same tested basis. Because of this, any changes must be test, at minimum, in vvw.dll, vvwnet.dll and the MediaCmd SDK. To add compile specific code to this interface, use the following defines _WIN32 - Windows XP-7, 32 or 64 bit WIN64 - Windows XP-7 64 bit Darwin - Apple OS-X Linux - CentOS or Ubuntu

BUILDING_VVW_DLL - Any VVW dll, but especially VVW.DLL or VVWNET2.DLL

BUILDING_ACTUAL_VVW_DLL - Only defined for VVW.DLL specifically

BUILDING_VVWNET2_DLL - Only defined for VVWNET2.DLL specifically

Definition at line 57 of file vvwif.cpp.

```
#define dtOSLeaveCriticalSection
```

Definition at line 58 of file vvwif.cpp.

```
#define dwLastStatusUpdateMs mCmdState.dwCmdAlt
```

Last time a status was called.

Definition at line 139 of file vvwif.cpp.

```
#define MAX_DIRECT_NETWORK_CHANNELS 256
```

Definition at line 148 of file vvwif.cpp.

```
#define MSGAREA "vwvIF"
```

Definition at line 74 of file vvwif.cpp.

```
#define VVWX_CHANNELFLAG_ISOPEN 0x00000001
```

Definition at line 137 of file vvwif.cpp.

Typedef Documentation

```
typedef struct tagVVWX\_CHANNEL * pVVWX\_CHANNEL
```

```
typedef struct tagVVWX\_CHANNEL VVWX\_CHANNEL
```

Function Documentation

```
DWORD __stdcall dtGpiSetMode (VVWHANDLE hDSync, DWORD dwInOut, DWORD dwCmd, DWORD dwSetOn, BOOL bSet)
```

Definition at line 3043 of file vvwif.cpp.

```
DWORD __stdcall dtGpiSetModeD (VVWHANDLE hDSync, DWORD dwInOut, DWORD dwCmd, DWORD dwSetOn, BOOL bSet, BOOL bTrace)
```

Definition at line 3026 of file vvwif.cpp.

```
void __stdcall initMediaCmd (PMEDIACMD pCmd, cmdType ctCmd)
```

Setup a mediacmd pointer to basic defaults

Parameters:

<i>pCmd</i>	- pointer to the command structure to initialize
<i>ctCmd</i>	- the command to set the structure to

Definition at line 206 of file vvwif.cpp.

```
uint32_t __stdcall strcpyClipIF (char * szClipDst, char * szClipSrc)
```

Definition at line 168 of file vvwif.cpp.

```
uint32_t __stdcall vvwBlank (VVWIFOPAQUE vwChannel, char * szClipName, uint32_t IStart, uint32_t IEnd, uint32_t IVidEdit, uint32_t IAudEdit, uint32_t IInfEdit, int fRipple)
```

Definition at line 2203 of file vvwif.cpp.

```
uint32_t __stdcall vvwBlankAllClipIds (VVWIFOPAQUE vwChannel)
```

Definition at line 2245 of file vvwif.cpp.

VVWHANDLE __stdcall vvwChannelToHandle (VVWIFOPAQUE vvwChannel)

Convert a 0..3 channel to 0, 64, 65536 Mostly internal Undocumented

Definition at line 606 of file vvwif.cpp.

uint32_t __stdcall vvwCheckNetworkChannel (VVWIFOPAQUE vvwChannel)

Check that we are still conneted through the network

Definition at line 583 of file vvwif.cpp.

uint32_t __stdcall vvwCleanRecordWipeDrive (VVWIFOPAQUE vvwChannel, uint32_t IWipeFolder)

Reset record file(s) and/or delete all files found in the record folder

Definition at line 1255 of file vvwif.cpp.

uint32_t __stdcall vvwCloseNetworkChannel (VVWIFOPAQUE vvwChannel)

Private close for the direct source MediaCMD API access

Definition at line 538 of file vvwif.cpp.

uint32_t __stdcall vvwCopyClip (VVWIFOPAQUE vvwChannel, char * szSourceClip, char * szDestClip, uint32_t IStart, uint32_t IEnd)

Create a virtual copy of a clip, changing the in and out points if necessary. To use the whole clip, set IStart to 0 and the end to -1. Returns 0 if successful, else an error code.

Definition at line 1981 of file vvwif.cpp.

uint32_t __stdcall vvwDelete (VVWIFOPAQUE vvwChannel, char * szClipName, uint32_t IStart, uint32_t IEnd, uint32_t IVidEdit, uint32_t IAudEdit, uint32_t IInfEdit, int fRipple)

Definition at line 2261 of file vvwif.cpp.

uint32_t __stdcall vvwEDLGetEdit (VVWIFOPAQUE vvwChannel, uint32_t * IRecordIn, uint32_t * IPlayIn, uint32_t * IPlayOut, uint32_t * IVidEdit, uint32_t * IAudEdit, uint32_t * IInfEdit, char * szClipName, char * szFileName, BOOL bClipInfo)

Returns an edit line from the VTR space of an internal channel. The function will continue to return the next edit in the timecode space until the last edit is returned, after which an error will be returned. To reset to the start of the Edl use EDLResetToStart. Returns 0 if successfule else an Error code.

Definition at line 2028 of file vvwif.cpp.

uint32_t __stdcall vvwEDLGetExtendedInfo (VVWIFOPAQUE vvwChannel, uint32_t IType, uint32_t IRecordIn, char * sz260Comment, uint32_t * plExtra)

Get Comment

Definition at line 2096 of file vvwif.cpp.

uint32_t __stdcall vvwEDLResetToStart (VVWIFOPAQUE vvwChannel)

Reset the edl returns in VTR mode to the first element of the list.

Definition at line 2008 of file vvwif.cpp.

uint32_t __stdcall vvwEDLSetExtendedInfo (VWVIFOPAQUE vvwChannel, uint32_t IType, uint32_t IRecordIn, uint32_t IValue, char * sz260Comment, uint32_t IDuration)

Set Comment

Definition at line 2073 of file vvwif.cpp.

uint32_t __stdcall vvwEject (VWVIFOPAQUE vvwChannel)

Eject the current media

Definition at line 1279 of file vvwif.cpp.

uint32_t __stdcall vvwEnableChannels (uint32_t IInternal0_31, uint32_t IInternal32_63, uint32_t IExternal64_95, uint32_t IExternal96_127, uint32_t INetwork128_159, uint32_t INetwork160_191)

Set the allowable channels

Definition at line 634 of file vvwif.cpp.

uint32_t __stdcall vvwFastForward (VWVIFOPAQUE vvwChannel)

Play Fast Forward

Definition at line 1031 of file vvwif.cpp.

uint32_t __stdcall vvwFastRewind (VWVIFOPAQUE vvwChannel)

Play Fast Reverse

Definition at line 1046 of file vvwif.cpp.

void __stdcall vvwFreeString (char * szString)

Free a string value returned by the channel.

Definition at line 4788 of file vvwif.cpp.

uint32_t __stdcall vvwGetAudioInput (VWVIFOPAQUE vvwChannel)

ADD FUNCTION lAudIn Get the current audio input.

[GS_AUDSELECT_UNBALANCED_10](#)

[GS_AUDSELECT_UNBALANCED_4](#)

[GS_AUDSELECT_BALANCED_10](#)

[GS_AUDSELECT_BALANCED_4](#)

[GS_AUDSELECT_SPDIF](#) [GS_AUDSELECT_AES_EBU](#) [GS_AUDSELECT_EMBEDDED](#)

Definition at line 2572 of file vvwif.cpp.

uint32_t __stdcall vvwGetAudioInputLevel (VWVIFOPAQUE vvwChannel)

Get the current audio input level

Definition at line 2584 of file vvwif.cpp.

uint32_t __stdcall vvwGetAudioOutput (VWVIFOPAQUE vvwChannel)

Get the current audio Output – See Get/SetAudioInput

[GS_AUDSELECT_UNBALANCED_10](#)

[GS_AUDSELECT_UNBALANCED_4](#)

[GS_AUDSELECT_BALANCED_10](#)

[GS_AUDSELECT_BALANCED_4](#)

[GS_AUDSELECT_SPDIF](#) [GS_AUDSELECT_AES_EBU](#) [GS_AUDSELECT_EMBEDDED](#)

Definition at line 2596 of file vvwif.cpp.

uint32_t __stdcall vvwGetAudioOutputLevel (VWIFOPAQUE vvwChannel)

Get the current audio output level.

Definition at line 2608 of file vvwif.cpp.

uint32_t __stdcall vvwGetAudioPeakRMS (VWIFOPAQUE vvwChannel, uint32_t lAudEdit, uint32_t * plPeaks)

Returns the RMS and Peak audio levels of the input (stop/record) or output (play/pause).

Definition at line 2620 of file vvwif.cpp.

uint32_t __stdcall vvwGetAutoMode (VWIFOPAQUE vvwChannel)

Calls ValueXXX with gsAutoMode. Required for play lists, deferred commands and auto edit commands on VTRs.

Definition at line 2477 of file vvwif.cpp.

uint32_t __stdcall vvwGetAvailablePresets (VWIFOPAQUE vvwChannel, uint32_t * plVidEdit, uint32_t * plAudEdit, uint32_t * plInfEdit)

ADD FUNCTIONS lVidEdit, lAudEdit, lInfEdit Returns the supported audio, video and info presets for a channel.

Definition at line 2533 of file vvwif.cpp.

uint32_t __stdcall vvwGetChannelCapabilities (VWIFOPAQUE vvwChannel)

Get the available commands for a channel.

Definition at line 2817 of file vvwif.cpp.

uint32_t __stdcall vvwGetChannelName (VWIFOPAQUE vvwChannel, char * szChannelName)

Get the current channel name

Definition at line 714 of file vvwif.cpp.

uint32_t __stdcall vvwGetChannelType (VWIFOPAQUE vvwChannel)

Get the current channel type

Definition at line 754 of file vvwif.cpp.

uint32_t __stdcall vvwGetClipInfo (VWIFOPAQUE vvwChannel, char * sz8CharClipName, uint32_t * lStart, uint32_t * lEnd, uint32_t * lVidEdit, uint32_t * lAudEdit, uint32_t * lInfEdit, char * szFileName)

Returns the basic information from szClip. The information is located in lStart, lEnd, lVidEdit, lAudEdit and szFileName as the in point, out point, number of video channels, number of audio channels, and the file name respectively. Returns 0 if successful, else an error code.

Definition at line 1867 of file vvwif.cpp.

uint32_t __stdcall vvwGetClipMode (VWIFOPAQUE vvwChannel)

Calls ValueXXX with gsClipMode. If equal to 1 then the channel is in clip mode, if 0 the channel is in VTR mode.

Definition at line 2432 of file vvwif.cpp.

uint32_t __stdcall vvwGetClipName (VWIFOPAQUE vvwChannel, char * sz8CharClipName)

Only supported in clip Mode. Returns the current clip name, if any. For dll access, the memory must be at least 9 bytes uint32_t (8 character bytes + NULL) and is always ANSI.

Definition at line 1517 of file vvwif.cpp.

uint32_t __stdcall vvwGetCompressionRate (VWIFOPAQUE vvwChannel)

Get the current compression rate

Definition at line 2775 of file vvwif.cpp.

uint32_t __stdcall vvwGetCurMs (VWIFOPAQUE vvwChannel)

Get the current millisecond time.

Parameters:

<i>vwChannel</i>	the mediacmd target being controlled (internal, external, network)
------------------	--

Returns:

the current millisecond counter for that channel
Definition at line 2810 of file vvwif.cpp.

uint32_t __stdcall vvwGetCurrentPresets (VWIFOPAQUE vvwChannel, uint32_t * pIVidEdit, uint32_t * pIAudEdit, uint32_t * pIInfEdit)

ADD FUNCTIONS IVidEdit, IAudEdit, IInfEdit Returns the curent audio, video and info presets for a channel.

Definition at line 2489 of file vvwif.cpp.

uint32_t __stdcall vvwGetCurState (VWIFOPAQUE vvwChannel, char * sz14ByteState)

Returns the current state as a string (e.g. "Play"). For dll access, the memory must always be at least 15 bytes uint32_t (14 byte state + NULL) and is always ANSI.

Stop - Stop all playback, and normally place all channels into passthrough

Pause - Halt all channels. Display current video frame and silence audio

Seek - With cmdFlags::cfUsePosition and MEDICMD::dwPosition, goto that frame and Pause

Play - Play all channels. May be modified by (cmdFlags::cfUseSpeed + [MEDIACMD::ISpeed](#)) and

((cmdFlags::cfUsePosition or cmdFlags::cfUsePositionOffset) and [MEDIACMD::dwPosition](#)) or

((cmdFlags::cfUseStart or cmdFlags::cfUseStartOffset) and [MEDIACMD::dwStart](#)) or

((cmdFlags::cfUseEnd or cmdFlags::cfUseEndOffset) and [MEDIACMD::dwEnd](#)) as well as

[MEDIACMD::dwCmdAlt](#) with certain [cmdFlags](#) to play from-top, at speed or combinations of the above.

Record - Record one or a combination of video/audio/info to disk. May be modified, as with [ctPlay](#) by flags and structure members such as

((cmdFlags::cfUsePosition or cmdFlags::cfUsePositionOffset) and [MEDIACMD::dwPosition](#)) or

((cmdFlags::cfUseStart or cmdFlags::cfUseStartOffset) and [MEDIACMD::dwStart](#)) or ((cmdFlags::cfUseEnd or cmdFlags::cfUseEndOffset) and [MEDIACMD::dwEnd](#)) as well as (cmdFlags::cfUseClipID and [MEDIACMD::arbID](#)) or (cmdFlags::cfDeferred or cmdFlags::cfOverrideDeferred) or [MEDIACMD::dwCmdAlt](#) with certain [cmdFlags](#) to record from-top, at speed or combinations of the above.

Record Stop - Set the channel into a record ready state, normally passthrough with the recording file preallocated, and possible pass start end and name information. See

(cmdFlags::cfUseStart cmdFlags::cfUseStartOffset and [MEDIACMD::dwStart](#))

(cmdFlags::cfUseEnd cmdFlags::cfUseEndOffset and [MEDIACMD::dwEnd](#))

(cmdFlags::cfUseClipID and [MEDIACMD::arbID](#)) for record setups.

Eject - Stop the channel and unload removable media, if possible, else same as stop

Transfer - Transfer media from one channel to another. Normally used to transfer internal media to or from an external tape device.

Insert Clip or Timecode Area - Used in time code space (TCspace.h) and Clip Space (ClipSpace.h) to add new clips or areas. Inserted media is defined by (cmdFlags::cfUseStart - [MEDIACMD::dwStart](#), cmdFlags::cfUseEnd - [MEDIACMD::dwEnd](#)) for clip being added and cmdFlags::cfUsePosition - [MEDIACMD::dwPosition](#) for target. Also (cmdFlags::cfUseClipID and [MEDIACMD::arbID](#)) may be used to specify a file name. cmdFlags::cfUsePresets and [MEDIACMD::dwVideoChannels](#), [MEDIACMD::dwAudioChannels](#), [MEDIACMD::dwInfoChannels](#) are also respected if set. cmdFlags::cfRipple may also be used to insert over

NOTE - The ctTransfer command is ALWAYS sent to the target with the SOURCE channel in the [MEDIACMD::dwCmdAlt](#) member and cmdFlags::cfUseCmdAlt set UNLESS one of the devices is slow/high latency/sloppy (read VTR), in which case it always receives the command so it can master the transfer and the cmdFlags::cfInvert is used to set the direction.

Blank a Timecode Area - Used in time code space (TCspace.h) to set an area to black and silent audio. (cmdFlags::cfUseStart - [MEDIACMD::dwStart](#), cmdFlags::cfUseEnd - [MEDIACMD::dwEnd](#)) set the area to be blanked. cmdFlags::cfUsePresets and [MEDIACMD::dwVideoChannels](#), [MEDIACMD::dwAudioChannels](#), [MEDIACMD::dwInfoChannels](#) are also respected if set. cmdFlags::cfRipple may also be used to remove blank area. With this command, no media is removed from storage.

Delete a clip (ClipSpace.h) or an area (TCspace.h). Deletes the media from storage and from the current space. For ClipSpace, cmdFlags::cfUseClipID and [MEDIACMD::arbID](#) must be specified, and if any sub clip or super clips exist, the id will be removed but the media will not be deleted.

For TCspace, cmdFlags::cfUseStart and [MEDIACMD::dwStart](#) with cmdFlags::cfUseEnd and [MEDIACMD::dwEnd](#) should be used to specify the time code segment to be deleted.

cmdFlags::cfUsePresets and [MEDIACMD::dwVideoChannels](#), [MEDIACMD::dwAudioChannels](#), [MEDIACMD::dwInfoChannels](#) may also be used to delete specific channels. If cmdFlags::cfRipple is set, then the TCspace will close over the deleted material, changing all timecode location beyond the deletion point by minus the size of the deletion.

Trim a clip or area - Currently not implemented. Use cmdType::ctSetValue and #GS_CLIP_INFO to trim a clip, or a combination of cmdType::ctInsert, cmdType::ctDelete, cmdType::ctBlank to trim a tspace area.

Channel select - select active channels, preview passthrough channels (to preview and edit) recording channels (to create a split edit ala CMX) Requires cmdFlags::cfUsePresets and [MEDIACMD::dwVideoChannels](#), [MEDIACMD::dwAudioChannels](#),

[MEDIACMD::dwInfoChannels](#)

Get the current state of the controlled channel(s) - Fills the user supplied [MEDIACMD](#) structure with the current state. Look for `cmdType::ctError`, `cmdType::ctStop`, `cmdType::ctEject`, `cmdType::ctPause`, `cmdType::ctPlay`, `cmdType::ctRecStop`, `cmdType::ctRecord` for basic state. For valid fields, check `cmdFlags::cfDeferred` : we have a deferred clip `cmdFlags::cfTimeMs` : [MEDIACMD::dwCmdAlt](#) has millisecond performance counter info `cmdFlags::cfUseSpeed` : [MEDIACMD::lSpeed](#) has the valid current speed `cmdFlags::cfUsePresets` : [MEDIACMD::dwVideoChannels](#), [MEDIACMD::dwAudioChannels](#), [MEDIACMD::dwInfoChannels](#) contain preset information `cmdFlags::cfUsePosition` : [MEDIACMD::dwPosition](#) contains current position `cmdFlags::cfUseStart` : [MEDIACMD::dwStart](#) has starting frame position `cmdFlags::cfUseEnd` : [MEDIACMD::dwEnd](#) has end frame position (+1 the out is never included) `cmdFlags::cfUseClipID` : [MEDIACMD::arbID](#) has current clip name (8 char for louth and odetics) `cmdFlags::cfFields` : [MEDIACMD::dwPosition](#), [MEDIACMD::dwStart](#) and [MEDIACMD::dwEnd](#) are in fields if they are valid `cmdFlags::cfNoReturn` : return is invalid.

Set the current state - Used for control type channels such as Serial 422 control (`vvwCtl.h`) and network control (`vvwNet.h`). Tells the controller or user what our current state is. The state should be reported honestly, as it is the receivers responsibility to transition states in an appropriate way for its controller. For actual channels (`vvwInt.h`, `vvwExt.h`, `vvwNet.h`-as controller, `vvwDS2.h`, etc), the state should be set by using one of the transport commands (`cmdType::ctPlay` etc) above. Get a non transport setting - Used for one time setups on channel. Includes audio levels, video proc amps, audio/video input, compression type and level and many others See: [cmdGetSetValue](#) for possible commands

Set a non transport setting - Used for one time setups on channel. Includes audio levels, video proc amps, audio/video input, compression type and level and many others See: [cmdGetSetValue](#) for possible commands

Check support for a non transport setting - Used for one time setups on channel. Includes audio levels, video proc amps, audio/video input, compression type and level and many others See: [cmdGetSetValue](#) for possible commands

Indicates the an error in the channel has occurred. Return only. See [MEDIACMD::dwCmdAlt](#) for error code and [MEDIACMD::arbID](#) for message if any. These members will be valid if `cmdFlags::cfUseCmdAlt` and `cmdFlags::cfUseClipID` are set

Terminate Close A Channel - Only used by remote devices that cannot close the channel directly such as `vvwNet.h`. Channel may not actually close when this is called, but the communications pipe will be closed and wait for another connection.

Abort the current operation - Use to abort operations that would normally ignore extraneous commands such as non-linear playback sequences, records or if the channel just seems to be stuck. Makes a good panic button.

Definition at line 1570 of file `vvwif.cpp`.

uint32_t __stdcall vwvGetCurTC ([VWVIFOPAQUE vwvChannel](#), char * sz14ByteTC)

Returns the current time code as a string (e.g. "00:01:00:00"). For dll access, the memory must always be at least 15 bytes `uint32_t` (14 byte time code plus id + NULL) and is always ANSI.

Definition at line 1546 of file `vvwif.cpp`.

uint32_t __stdcall vwvGetEnd ([VWVIFOPAQUE vwvChannel](#))

Return the current end point or out point if `cfUseEnd` is set.

Definition at line 1503 of file `vvwif.cpp`.

uint32_t __stdcall vvwGetError (uint32_t * pLastError, uint32_t * pSeverity, char * szError, short * psYear, short * psMonth, short * psDay, short * psHour, short * psMinute, short * psSecond, short * psMillisecond)

Get the current error. Sets pointer to the next one automatically

Definition at line 2979 of file vvwif.cpp.

uint32_t __stdcall vvwGetErrorLength (uint32_t * pLastError, uint32_t * pErrorLength)

Get the length of the current error string

Definition at line 2961 of file vvwif.cpp.

uint32_t __stdcall vvwGetErrorLogMs ()

Get the ms time the last error was added to the error log

Definition at line 2931 of file vvwif.cpp.

uint32_t __stdcall vvwGetFileName (VWVIFOPAQUE vvwChannel, char * sz260CharFileName)

Returns the current file name, if any. For dll access, the memory must be at least 261 bytes uint32_t (260 bytes max path + NULL) and is always ANSI.

Definition at line 1533 of file vvwif.cpp.

uint32_t __stdcall vvwGetFlags (VWVIFOPAQUE vvwChannel)

Returns the current flags

cfDeferred = 1, // 0x00000001 This is a delayed cfOverrideDeferred = 1 << 30, // 0x40000000 Override all previous deferred commands cfTimeMs = 1 << 1, // 0x00000002 Use Millisecond time for delayed time, not fields cfTimeTarget = 1 << 2, // 0x00000004 Delayed time is offset from current time code cfTimeHouseClock = 1 << 3, // 0x00000008 Delayed time is based on absolute (real) time cfUseSpeed = 1 << 4, // 0x00000010 Set the new speed cfUsePresets = 1 << 5, // 0x00000020 Use video and audio edit presets cfUsePosition = 1 << 6, // 0x00000040 Use the position setting cfUsePositionOffset = 1 << 7, // 0x00000080 Position is an offset cfUseStart = 1 << 8, // 0x00000100 Start a new timecode cfUseStartOffset = 1 << 9, // 0x00000200 Start is an offset from current tc cfUseEnd = 1 << 10, // 0x00000400 End command as specified cfUseEndOffset = 1 << 11, // 0x00000800 End is and offset from current tc cfUseAllIDs = 1 << 12, // 0x00001000 Use all clip IDs cfUseClipID = 1 << 13, // 0x00002000 Use new clip ID, otherwise use last or none cfNoClipFiles = 1 << 14, // 0x00004000 Use new clip ID, otherwise use last or none cfNoTCSpaces = 1 << 15, // 0x00008000 Use new clip ID, otherwise use last or none cfUseCmdAlt = 1 << 16, // 0x00010000 Use the dwCmdAlt cfIsShuttle = 1 << 17, // 0x00020000 Use speed in play for shuttle cfFields = 1 << 20, // 0x00100000 Position, start and end are fields, not frames cfRipple = 1 << 21, // 0x00200000 Ripple for insert or delete cfLoop = 1 << 22, // 0x00400000 Loop the clip or in out cfTrigger = 1 << 23, // 0x00800000 Trigger using dsync class cfPreview = 1 << 24, // 0x01000000 Preview set (EE, non rt play) cfInvert = 1 << 28, // 0x10000000 Invert a transfer cfTest = 1 << 29, // 0x20000000 See if the command exists cfNoReturn = 1 << 31, // 0x80000000 No return mediacmd is required

Definition at line 1434 of file vvwif.cpp.

uint32_t __stdcall vvwGetFreeStorage (VWVIFOPAQUE vvwChannel)

Returns the amount of available storage for recording in megabytes.

Definition at line 2803 of file vvwif.cpp.

uint32_t __stdcall vvwGetFreeTime (VWVIFOPAQUE vvwChannel)

Returns the remaining number of frames of storage available at current compression rate.

Definition at line 2793 of file vvwif.cpp.

uint32_t __stdcall vvwGetLastChangeMs (VWVIFOPAQUE vvwChannel, uint32_t IClipSpace, uint32_t * INumClips)

Returns the millisecond time of the last change in the current mode (clip or vtr).

Definition at line 2135 of file vvwif.cpp.

uint32_t __stdcall vvwGetLastChangeXferMs (VWVIFOPAQUE vvwChannel)

Returns the millisecond time of the last change in the transfer queue

Definition at line 2120 of file vvwif.cpp.

uint32_t __stdcall vvwGetLastMs (VWVIFOPAQUE vvwChannel)

Returns the millisecond time the last status occurred (time of the last vertical blank).

Definition at line 1476 of file vvwif.cpp.

uint32_t __stdcall vvwGetLicenseOptions (VWVIFOPAQUE vvwChannel, uint32_t * pdwOptions, uint32_t * pdwLicenseValid)

Definition at line 3148 of file vvwif.cpp.

uint32_t __stdcall vvwGetMaxChannels (void)

Get the maximum channels available

Definition at line 689 of file vvwif.cpp.

uint32_t __stdcall vvwGetMaxInternalChannels (void)

Get the maximum channels available

Definition at line 702 of file vvwif.cpp.

uint32_t __stdcall vvwGetMetaData (VWVIFOPAQUE vvwChannel, char * sz8CharClipName, char * sz260CharFileName, uint32_t vvwInfoRequest, char * szValue)

Returns the meta data from szClip. Returns 0 if successful, else an error code.

Definition at line 1941 of file vvwif.cpp.

char* __stdcall vvwGetMRVersion ()

Returns the version string of the MediaReactor subsystem.

Definition at line 2846 of file vvwif.cpp.

char* __stdcall vvwGetNextClip (VWVIFOPAQUE vvwChannel, char * sz8CharLastClipCurClip)

Clip Mode Only. Returns the next clip identifier. To get the first clip, szLastClip should be an

empty string. Once the last clip available has been returned, GetNextClip will return an error or NULL for unix/dll access. Please note: For unix/dll access, the sz8CharLastClipCurClip memory area is used for the new clip. The previous clip name is therefore lost and the memory is not allocated by the vvw. Returns 0 if successful, else an error code.

Definition at line 1834 of file vvwif.cpp.

uint32_t __stdcall vvwGetNextClipEx ([VWIFOPAQUE](#) vvwChannel, uint32_t * ICreation, uint32_t * ILastModification, uint32_t * IFileSize, uint32_t * IDiskFragments)

Definition at line 1970 of file vvwif.cpp.

uint32_t __stdcall vvwGetNumberOfBackUps ([VWIFOPAQUE](#) vvwChannel, uint32_t * pdwBackUps, uint32_t dwClipMode)

Definition at line 3113 of file vvwif.cpp.

uint32_t __stdcall vvwGetNumberOfErrors (uint32_t * pIErrors)

Get the number of current errors

Definition at line 2944 of file vvwif.cpp.

uint32_t __stdcall vvwGetPiconName (char * szFileName)

Get the name of a file from the media file's name

Definition at line 3007 of file vvwif.cpp.

uint32_t __stdcall vvwGetPosition ([VWIFOPAQUE](#) vvwChannel)

Returns the current position if the cfUsePosition flag is set, otherwise invalid.

Definition at line 1462 of file vvwif.cpp.

uint32_t __stdcall vvwGetSpeed ([VWIFOPAQUE](#) vvwChannel)

Returns the current VVW speed if the cfUseSpeed flag is set, otherwise pause or full play speed. VVW speeds are based on 65520 as the play speed. To translate to decimal number where 1.0 represents play, use the following formula: $D1Speed = ((double)VVWSpeed / 65520.0)$ For percentages, where 100.0 represents play speed, use the following formula: $Dpercent = (((double)VVWSpeed * 100.0) / 65520.0) = ((double)VVWSpeed / 655.2)$ XML: See <MediaCmd> root element, <Speed> sub-element

Typical VVW speeds (note speeds are linear): Pause 0% 0 Play 100% 65520 Half Play 50% 32760 Rev Play -100% -65520 Rev 2 x Play -200% 131040 10 x Play 1000% 655200 Max Play 90000% 5896800 Max Rev -90000%-5896800

Definition at line 1447 of file vvwif.cpp.

uint32_t __stdcall vvwGetStart ([VWIFOPAQUE](#) vvwChannel)

Returns the current start or in point if the cfUseStart flag is set.

Definition at line 1490 of file vvwif.cpp.

uint32_t __stdcall vvwGetState ([VWIFOPAQUE](#) vvwChannel)

Returns the current state

ctStop 0 // Stop all action ctPause 1 // Pause, Seek ctPlay 2 // Play at specified speed (includes pause) ctRecord 3 // Record at specified speed ctRecStop 4 // Stop ready for recording ctEject 5 // Eject the current media ctError 17 // An error has occurred ctAbort 19 // Abort any queued commands

Definition at line 1415 of file vvwif.cpp.

uint32_t __stdcall vvwGetSuperImpose ([VWIFOPAQUE](#) vvwChannel)

Get the status of the super imposed time code overlay

Definition at line 2889 of file vvwif.cpp.

uint32_t __stdcall vvwGetTCSOURCE ([VWIFOPAQUE](#) vvwChannel)

Calls ValueXXX with gsTcSource (VITC, LTC, Control, Clip). [GS_TCSOURCE_LTC](#), [GS_TCSOURCE_VITC](#), [GS_TCSOURCE_CTL](#) or [GS_TCSOURCE_CLIP](#)

Definition at line 2465 of file vvwif.cpp.

uint32_t __stdcall vvwGetTCType ([VWIFOPAQUE](#) vvwChannel)

Calls ValueXXX with gsTcType (drop frame, non drop frame, pal). [TC2_TCTYPE_FILM](#), [TC2_TCTYPE_NDF](#), [TC2_TCTYPE_DF](#), [TC2_TCTYPE_PAL](#), [TC2_TCTYPE_50](#), [TC2_TCTYPE_5994](#), [TC2_TCTYPE_60](#), [TC2_TCTYPE_NTSCFILM](#), [TC2_TCTYPE_2398](#), [TC2_TCTYPE_100](#)

Definition at line 2453 of file vvwif.cpp.

uint32_t __stdcall vvwGetTotalStorage ([VWIFOPAQUE](#) vvwChannel)

Returns the total storage connected in megabytes.

Definition at line 2798 of file vvwif.cpp.

uint32_t __stdcall vvwGetTotalTime ([VWIFOPAQUE](#) vvwChannel)

Returns the total number of frames of storage available at current compression rate if the storage space was empty.

Definition at line 2788 of file vvwif.cpp.

uint32_t __stdcall vvwGetVideoGenlock ([VWIFOPAQUE](#) vvwChannel)

Turn the house/reference lock on or off

Definition at line 2751 of file vvwif.cpp.

uint32_t __stdcall vvwGetVideoGenlockSource ([VWIFOPAQUE](#) vvwChannel)

Set the genlock source to input or external reference

Definition at line 2763 of file vvwif.cpp.

uint32_t __stdcall vvwGetVideoInput ([VWIFOPAQUE](#) vvwChannel)

Get the current video input. [GS_VIDSELECT_COMPOSITE](#), [GS_VIDSELECT_COMPOSITE_2](#), [GS_VIDSELECT_SVIDEO](#),

[GS_VIDSELECT_COMPONENT_YUV](#), [GS_VIDSELECT_COMPONENT_YUV_M2](#),
[GS_VIDSELECT_COMPONENT_YUV_SMPTE](#), [GS_VIDSELECT_COMPONENT_RGB](#),
[GS_VIDSELECT_D1_SERIAL](#), [GS_VIDSELECT_D1_PARALLEL](#),
[GS_VIDSELECT_SDTI](#), [GS_VIDSELECT_NONE](#)

Definition at line 2642 of file vvwif.cpp.

uint32_t __stdcall vvwGetVideoInputChroma ([VWIFOPAQUE](#) vvwChannel)

Get the current video input's 'Chroma' TBC setting.

Definition at line 2697 of file vvwif.cpp.

uint32_t __stdcall vvwGetVideoInputHue ([VWIFOPAQUE](#) vvwChannel)

Get the current video input's 'Hue' TBC setting.

Definition at line 2687 of file vvwif.cpp.

uint32_t __stdcall vvwGetVideoInputSetup ([VWIFOPAQUE](#) vvwChannel)

Get the current video input's 'Setup' TBC setting.

Definition at line 2667 of file vvwif.cpp.

uint32_t __stdcall vvwGetVideoInputVideo ([VWIFOPAQUE](#) vvwChannel)

Get the current video input's 'Video' TBC setting.

Definition at line 2677 of file vvwif.cpp.

uint32_t __stdcall vvwGetVideoOutput ([VWIFOPAQUE](#) vvwChannel)

Get the current video output. See [Get/SetVideoInput](#) for settings.

[GS_VIDSELECT_COMPOSITE](#), [GS_VIDSELECT_COMPOSITE_2](#),
[GS_VIDSELECT_SVIDEO](#), [GS_VIDSELECT_COMPONENT_YUV](#),
[GS_VIDSELECT_COMPONENT_YUV_M2](#),
[GS_VIDSELECT_COMPONENT_YUV_SMPTE](#), [GS_VIDSELECT_COMPONENT_RGB](#),
[GS_VIDSELECT_D1_SERIAL](#), [GS_VIDSELECT_D1_PARALLEL](#),
[GS_VIDSELECT_SDTI](#), [GS_VIDSELECT_NONE](#)

Definition at line 2654 of file vvwif.cpp.

uint32_t __stdcall vvwGetVideoTBCChroma ([VWIFOPAQUE](#) vvwChannel)

Get the current global TBC's 'Chroma' setting.

Definition at line 2739 of file vvwif.cpp.

uint32_t __stdcall vvwGetVideoTBCHue ([VWIFOPAQUE](#) vvwChannel)

Get the current global TBC's 'Hue' setting.

Definition at line 2729 of file vvwif.cpp.

uint32_t __stdcall vvwGetVideoTBCSetup ([VWIFOPAQUE](#) vvwChannel)

Get the current global TBC's 'Setup' setting.

Definition at line 2709 of file vvwif.cpp.

uint32_t __stdcall vvwGetVideoTBCVideo (VWIFOPAQUE vvwChannel)

Get the current global TBC's 'Video' setting.

Definition at line 2719 of file vvwif.cpp.

char* __stdcall vvwGetVWVType (VWIFOPAQUE vvwChannel)

Returns the type string of the VVW channel.

Definition at line 2868 of file vvwif.cpp.

char* __stdcall vvwGetVWVVersion ()

Returns the version string of the VVW subsystem.

Definition at line 2824 of file vvwif.cpp.

uint32_t __stdcall vvwHandleToChannel (VWHANDLE hVvw)

Convert a 0, 64, 65536 to 0..3 Mostly internal Undocumented

Definition at line 620 of file vvwif.cpp.

uint32_t __stdcall vvwInsert (VWIFOPAQUE vvwChannel, char * szClipName, char * szFileName, uint32_t IPosition, uint32_t IStart, uint32_t IEnd, uint32_t IVidEdit, uint32_t IAudEdit, uint32_t IInfEdit, int fRipple)

Definition at line 2154 of file vvwif.cpp.

uint32_t __stdcall vvwIsOpen (void)

Has the vvw sub system been open

Definition at line 677 of file vvwif.cpp.

uint32_t __stdcall vvwLoadClip (VWIFOPAQUE vvwChannel, char * sz8CharClipName, uint32_t IStartFrame)

Load a clip

Definition at line 898 of file vvwif.cpp.

uint32_t __stdcall vvwMediaCmd (VWIFOPAQUE vvwChannel, void * pMediaCmd)

MediaCmd direct access

Definition at line 1292 of file vvwif.cpp.

long __stdcall vvwOpeningChannel ()

Definition at line 444 of file vvwif.cpp.

uint32_t __stdcall vvwOpenNetworkChannel (char * szAddress, uint32_t dwPort, uint32_t dwChannel)

Private open for the direct source MediaCMD API access

Definition at line 452 of file vvwif.cpp.

uint32_t __stdcall vvwPause ([VWIFOPAQUE](#) vvwChannel)

Pause displaying current frame

Definition at line 1061 of file vvwif.cpp.

uint32_t __stdcall vvwPlay ([VWIFOPAQUE](#) vvwChannel)

Set the channel into play

Definition at line 807 of file vvwif.cpp.

uint32_t __stdcall vvwPlayAtMs ([VWIFOPAQUE](#) vvwChannel, uint32_t IMs)

Start playback at a specified MS

Definition at line 995 of file vvwif.cpp.

uint32_t __stdcall vvwPlayAtSpeed ([VWIFOPAQUE](#) vvwChannel, uint32_t IVVWSpeed, uint32_t IEnd)

Play at a VVW speed specified

Definition at line 852 of file vvwif.cpp.

uint32_t __stdcall vvwPlayClip ([VWIFOPAQUE](#) vvwChannel, char * sz8CharClipName, int fDeferred)

Play a clip

Definition at line 945 of file vvwif.cpp.

uint32_t __stdcall vvwPlayClipFromTo ([VWIFOPAQUE](#) vvwChannel, char * sz8CharClipName, uint32_t IFrom, uint32_t ITo, int fDeferred)

Play a clip from a frame to a frame

Definition at line 963 of file vvwif.cpp.

uint32_t __stdcall vvwPlayFromTo ([VWIFOPAQUE](#) vvwChannel, uint32_t IFrom, uint32_t ITo, int fDeferred, int fLoop)

Play from a frame to a frame

Definition at line 869 of file vvwif.cpp.

uint32_t __stdcall vvwPlayOffsetAt ([VWIFOPAQUE](#) vvwChannel, uint32_t dwPosition, long IOffset, uint32_t dwMS)

Set the channel into play jumping a amount of frames

vwWaitForState(vvwChannel, ctPlay);

Definition at line 822 of file vvwif.cpp.

uint32_t __stdcall vvwRecord ([VWIFOPAQUE](#) vvwChannel)

Crash record

Definition at line 1125 of file vvwif.cpp.

uint32_t __stdcall vvwRecordAtMs ([VWIFOPAQUE](#) vvwChannel, uint32_t I Ms, uint32_t IStart, uint32_t IEnd)

Start Recording at a specified MS

Definition at line 1012 of file vvwif.cpp.

uint32_t __stdcall vvwRecordFromTo ([VWIFOPAQUE](#) vvwChannel, uint32_t IFrom, uint32_t ITo)

Record from one frame to another

Definition at line 1138 of file vvwif.cpp.

uint32_t __stdcall vvwRecordStop ([VWIFOPAQUE](#) vvwChannel, char * sz8CharClipName, uint32_t IDuration)

Record stop - prepare a record (clip mode only)

Definition at line 1164 of file vvwif.cpp.

uint32_t __stdcall vvwRecordStopFileName ([VWIFOPAQUE](#) vvwChannel, char * sz8CharClipName, char * sz256FileName, uint32_t IDuration)

Record stop - prepare a record (clip mode only)

Definition at line 1189 of file vvwif.cpp.

uint32_t __stdcall vvwReleaseChannels (void)

Release memory allocated to channels

Definition at line 655 of file vvwif.cpp.

uint32_t __stdcall vvwSeek ([VWIFOPAQUE](#) vvwChannel, uint32_t IFrame)

Seek to a frame

Definition at line 1074 of file vvwif.cpp.

uint32_t __stdcall vvwSeekRelative ([VWIFOPAQUE](#) vvwChannel, uint32_t IFrameOffset)

Seek to an offset from the current position

Definition at line 1097 of file vvwif.cpp.

uint32_t __stdcall vvwSetAudioInput ([VWIFOPAQUE](#) vvwChannel, uint32_t ISetting)

ADD FUNCTION lAudIn Set the current audio input.

[GS_AUDSELECT_UNBALANCED_10](#)

[GS_AUDSELECT_UNBALANCED_4](#)

[GS_AUDSELECT_BALANCED_10](#)

[GS_AUDSELECT_BALANCED_4](#)

[GS_AUDSELECT_SPDIF](#) [GS_AUDSELECT_AES_EBU](#) [GS_AUDSELECT_EMBEDDED](#)

Definition at line 2577 of file vvwif.cpp.

uint32_t __stdcall vvwSetAudioInputLevel ([VWIFOPAQUE](#) vvwChannel, uint32_t ISetting)

Get the current audio input level

Definition at line 2589 of file vvwif.cpp.

uint32_t __stdcall vvwSetAudioOutput ([VWIFOPAQUE](#) vvwChannel, uint32_t ISetting)

Set the current audio Output – See Get/SetAudioInput
[GS_AUDSELECT_UNBALANCED_10](#) [GS_AUDSELECT_UNBALANCED_4](#)
[GS_AUDSELECT_BALANCED_10](#) [GS_AUDSELECT_BALANCED_4](#)
[GS_AUDSELECT_SPDIF](#) [GS_AUDSELECT_AES_EBU](#) [GS_AUDSELECT_EMBEDDED](#)

Definition at line 2601 of file vvwif.cpp.

uint32_t __stdcall vvwSetAudioOutputLevel ([VWIFOPAQUE](#) vvwChannel, uint32_t ISetting)

Get the current audio output level.

Definition at line 2613 of file vvwif.cpp.

uint32_t __stdcall vvwSetAutoMode ([VWIFOPAQUE](#) vvwChannel, uint32_t ISetting)

Calls ValueXXX with gsAutoMode. Required for play lists, deferred commands and auto edit commands on VTRs.

Definition at line 2482 of file vvwif.cpp.

uint32_t __stdcall vvwSetBackUpNumber ([VWIFOPAQUE](#) vvwChannel, uint32_t dwBackUp, uint32_t dwClipMode)

Definition at line 3132 of file vvwif.cpp.

uint32_t __stdcall vvwSetClipMode ([VWIFOPAQUE](#) vvwChannel, uint32_t ISetting)

Calls ValueXXX with gsClipMode. If equal to 1 then the channel is in clip mode, if 0 the channel is in VTR mode.

Definition at line 2437 of file vvwif.cpp.

uint32_t __stdcall vvwSetCompressionRate ([VWIFOPAQUE](#) vvwChannel, uint32_t ISetting)

Set the current compression rate

Definition at line 2780 of file vvwif.cpp.

uint32_t __stdcall vvwSetErrorLog (uint32_t ISetting)

Set the error log pointer to the message you want

Definition at line 2916 of file vvwif.cpp.

uint32_t __stdcall vvwSetMetaData ([VWIFOPAQUE](#) vvwChannel, char * sz8CharClipName, uint32_t vvwilInfoRequest, uint32_t nValue, char * szValue)

Sets the meta data for szClip. Returns 0 if successful, else an error code.

Definition at line 1904 of file vvwif.cpp.

uint32_t __stdcall vvwSetRecordPresets ([VWIFOPAQUE](#) vvwChannel, uint32_t IVidEdit, uint32_t IAudEdit, uint32_t IInfEdit)

Set video/audio/info 'channels' to record (preset)

Definition at line 1238 of file vvwif.cpp.

uint32_t __stdcall vvwSetSuperImpose ([VWIFOPAQUE](#) vvwChannel, uint32_t ISetting)

Set the status of the super imposed time code overlay

Definition at line 2902 of file vvwif.cpp.

uint32_t __stdcall vvwSetTCSource ([VWIFOPAQUE](#) vvwChannel, uint32_t ISetting)

Calls ValueXXX with gsTcSource (VITC, LTC, Control, Clip). [GS_TCSOURCE_LTC](#), [GS_TCSOURCE_VITC](#), [GS_TCSOURCE_CTL](#) or [GS_TCSOURCE_CLIP](#)

Definition at line 2470 of file vvwif.cpp.

uint32_t __stdcall vvwSetTCType ([VWIFOPAQUE](#) vvwChannel, uint32_t ISetting)

Calls ValueXXX with gsTcType (drop frame, non drop frame, pal). [TC2_TCTYPE_FILM](#), [TC2_TCTYPE_NDF](#), [TC2_TCTYPE_DF](#), [TC2_TCTYPE_PAL](#), [TC2_TCTYPE_50](#), [TC2_TCTYPE_5994](#), [TC2_TCTYPE_60](#), [TC2_TCTYPE_NTSCFILM](#), [TC2_TCTYPE_2398](#), [TC2_TCTYPE_100](#)

Definition at line 2458 of file vvwif.cpp.

[BOOL](#) __stdcall vvwSetUseAbsoluteTC ([BOOL](#) bUseAbsoluteTC)

Setting this will cause the cfUseFrameCount Flag to be set on transport commands This will cause the interface to ignore LTC/VITC offsets and use absolute Timcode values. Mostly for ease if use in TC Space when still running as a LTC / VITC timecode source.

Definition at line 770 of file vvwif.cpp.

uint32_t __stdcall vvwSetVideoGenlock ([VWIFOPAQUE](#) vvwChannel, uint32_t ISetting)

Turn the house/reference lock on or off

Definition at line 2756 of file vvwif.cpp.

uint32_t __stdcall vvwSetVideoGenlockSource ([VWIFOPAQUE](#) vvwChannel, uint32_t ISetting)

Set the genlock source to input or external reference

Definition at line 2768 of file vvwif.cpp.

uint32_t __stdcall vvwSetVideoInput ([VWIFOPAQUE](#) vvwChannel, uint32_t ISetting)

Set the current video input. [GS_VIDSELECT_COMPOSITE](#), [GS_VIDSELECT_COMPOSITE_2](#), [GS_VIDSELECT_SVIDEO](#), [GS_VIDSELECT_COMPONENT_YUV](#), [GS_VIDSELECT_COMPONENT_YUV_M2](#), [GS_VIDSELECT_COMPONENT_YUV_SMPTE](#), [GS_VIDSELECT_COMPONENT_RGB](#), [GS_VIDSELECT_D1_SERIAL](#), [GS_VIDSELECT_D1_PARALLEL](#), [GS_VIDSELECT_SDTI](#), [GS_VIDSELECT_NONE](#)

Definition at line 2647 of file vvwif.cpp.

uint32_t __stdcall vvwSetVideoInputChroma ([VWIFOPAQUE](#) vvwChannel, uint32_t ISetting)

Set the current video input's 'Chroma' TBC setting.

Definition at line 2702 of file vvwif.cpp.

uint32_t __stdcall vvwSetVideoInputHue ([VWIFOPAQUE](#) vvwChannel, uint32_t ISetting)

Set the current video input's 'Hue' TBC setting.

Definition at line 2692 of file vvwif.cpp.

uint32_t __stdcall vvwSetVideoInputSetup ([VWIFOPAQUE](#) vvwChannel, uint32_t ISetting)

Set the current video input's 'Setup' TBC setting.

Definition at line 2672 of file vvwif.cpp.

uint32_t __stdcall vvwSetVideoInputVideo ([VWIFOPAQUE](#) vvwChannel, uint32_t ISetting)

Set the current video input's 'Video' TBC setting.

Definition at line 2682 of file vvwif.cpp.

uint32_t __stdcall vvwSetVideoOutput ([VWIFOPAQUE](#) vvwChannel, uint32_t ISetting)

Set the current video output. See Get/SetVideoInput for settings.
[GS_VIDSELECT_COMPOSITE](#), [GS_VIDSELECT_COMPOSITE_2](#),
[GS_VIDSELECT_SVIDEO](#), [GS_VIDSELECT_COMPONENT_YUV](#),
[GS_VIDSELECT_COMPONENT_YUV_M2](#),
[GS_VIDSELECT_COMPONENT_YUV_SMPTE](#), [GS_VIDSELECT_COMPONENT_RGB](#),
[GS_VIDSELECT_D1_SERIAL](#), [GS_VIDSELECT_D1_PARALLEL](#),
[GS_VIDSELECT_SDTI](#), [GS_VIDSELECT_NONE](#)

Definition at line 2659 of file vvwif.cpp.

uint32_t __stdcall vvwSetVideoTBCChroma ([VWIFOPAQUE](#) vvwChannel, uint32_t ISetting)

Set the current global TBC's 'Chroma' setting.

Definition at line 2744 of file vvwif.cpp.

uint32_t __stdcall vvwSetVideoTBCHue ([VWIFOPAQUE](#) vvwChannel, uint32_t ISetting)

Set the current global TBC's 'Hue' setting.

Definition at line 2734 of file vvwif.cpp.

uint32_t __stdcall vvwSetVideoTBCSetup ([VWIFOPAQUE](#) vvwChannel, uint32_t ISetting)

Set the current global TBC's 'Setup' setting.

Definition at line 2714 of file vvwif.cpp.

uint32_t __stdcall vvwSetVideoTBCVideo ([VWIFOPAQUE](#) vvwChannel, uint32_t ISetting)

Set the current global TBC's 'Video' setting.

Definition at line 2724 of file vvwif.cpp.

uint32_t __stdcall vvwStop ([VWIFOPAQUE](#) vvwChannel)

Stop - stop play back a show input if supported, else its a pause

Definition at line 1112 of file vvwif.cpp.

uint32_t __stdcall vvwSwitchClip (VWIFOPAQUE vvwChannel, char * sz8CharClipName, uint32_t IPosition, BOOL bUseFrameCount)

Switch to a clip

Definition at line 922 of file vvwif.cpp.

uint32_t __stdcall vvwTransfer (VWIFOPAQUE vvwChannel, uint32_t ITargetChannel, uint32_t IPosition, uint32_t IStart, uint32_t IEnd, uint32_t IVidEdit, uint32_t IAudEdit, uint32_t IInfEdit, char * szClipName, int fToTape)

Transfer media to or from an external VTR.

Definition at line 1328 of file vvwif.cpp.

uint32_t __stdcall vvwTrim (VWIFOPAQUE vvwChannel, uint32_t IPosition, uint32_t IStartOffset, uint32_t IEndOffset, uint32_t IVidEdit, uint32_t IAudEdit, uint32_t IInfEdit, int fRipple)

Definition at line 2302 of file vvwif.cpp.

uint32_t __stdcall vvwUpdateStatus (VWIFOPAQUE vvwChannel)

Retrieve the current status from the controlled device. The status is automatically updated by the interface, but this call ensures that the status is current when you are checking it. Returns 0 if successful, else an error code.

Definition at line 1376 of file vvwif.cpp.

uint32_t __stdcall vvwValueGet (VWIFOPAQUE vvwChannel, uint32_t IValueType, uint32_t * pIMin, uint32_t * pIMax)

Returns the current setting for a get/set value.

Definition at line 2362 of file vvwif.cpp.

uint32_t __stdcall vvwValueSet (VWIFOPAQUE vvwChannel, uint32_t IValueType, uint32_t ISetting)

Sets the get/set value to setting.

Definition at line 2384 of file vvwif.cpp.

uint32_t __stdcall vvwValueSet2 (VWIFOPAQUE vvwChannel, uint32_t IValueType, uint32_t ISetting, uint32_t IStart, uint32_t IEnd, uint32_t IVidChan, uint32_t IAudChan, uint32_t IInfChan)

Sets the get/set value to setting with extended parameters. Please set unused parameters to NULL.

Definition at line 2397 of file vvwif.cpp.

uint32_t __stdcall vvwValueSupported (VWIFOPAQUE vvwChannel, uint32_t IValueType)

Returns the supported attributes of a get/set value (gsClipMode, gsTcSource, etc) or -1 for not supported.

Definition at line 2344 of file vvwif.cpp.

uint32_t __stdcall vvwWaitForState ([VWFIFOPAQUE](#) vvwChannel, uint32_t ctCmd)

Wait for the channel to reach a state (ctPlay, ctRecord, etc). Will wait up to 500 milliseconds

Parameters:

<i>vwChannel</i>	the mediacmd target being controlled (internal, external, network)
<i>ctCmd</i>	the command state we will wait for (ctPlay, ctRecord, ctPause, etc)

Returns:

if state is reached, return 0. If it timed out waiting, return -1
Definition at line 791 of file vvwif.cpp.

uint32_t __stdcall vvwXMLProjectCheckOpen (char * lpszProjectName)

Definition at line 3075 of file vvwif.cpp.

uint32_t __stdcall vvwXMLProjectOpen (char * lpszProjectName, int bDeleteConflictingFiles)

Definition at line 3092 of file vvwif.cpp.

uint32_t __stdcall vvwXMLProjectSave (char * lpszProjectName)

Definition at line 3048 of file vvwif.cpp.

Variable Documentation

[pVWVX_CHANNEL](#) garChannels = NULL

Definition at line 147 of file vvwif.cpp.

uint32_t [gdwLastModeTC](#) = 0

Definition at line 153 of file vvwif.cpp.

uint32_t [gdwMaxChannels](#) = 0

Definition at line 151 of file vvwif.cpp.

uint32_t [gdwMaxInternalChannels](#) = 0

Definition at line 152 of file vvwif.cpp.

uint32_t [gdwOpeningChannel](#) = 0

Definition at line 150 of file vvwif.cpp.

uint32_t [gdwVWIFInOpen](#) = 0

Definition at line 149 of file vvwif.cpp.

uint32_t [m_dwPORT](#) = 1234

Definition at line 162 of file vvwif.cpp.

char [m_szTCPIPAddr](#)[1025] = ""

Definition at line 161 of file vvwif.cpp.

E:/drastic/api/mediacmd/src/vvwif.h File Reference

```
#include <string.h>
#include <sys/timeb.h>
#include <unistd.h>
#include <dtsystemtypes.h>
```

Defines

3973#define [COMPILING_DIRECT_MEDIACMD_SOURCE](#)
3974#define [SPD_FWD_PLAY](#) 65520L
Forward play speed (normal) in VVW (65520) see [MEDIACMD::lSpeed](#).
3975#define [SPD_PAUSE](#) 0L
Pause speed (0%) in VVW (0) see [MEDIACMD::lSpeed](#).
3976#define [SPD_REV_PLAY](#) (-SPD_FWD_PLAY)
Reverse play speed (-100%) in VVW (-65520) see [MEDIACMD::lSpeed](#).
3977#define [SPD_FWD_MAX](#) 5896800
Maximum possible play speed in VVW see [MEDIACMD::lSpeed](#).
3978#define [SPD_REV_MAX](#) (-SPD_FWD_MAX)
Minimum possible play speed in VVW see [MEDIACMD::lSpeed](#).
3979#define [SPD_ILLEGAL](#) 2147483647L
Illegal speed, set [MEDIACMD::lSpeed](#) to this value if not used.

Typedefs

3980typedef uint32_t [VWIFOPAQUE](#)

Functions

3981long __stdcall [vwOpeningChannel](#) ()
3982uint32_t __stdcall [vwOpenNetworkChannel](#) (char *szAddress, uint32_t dwPort, uint32_t dwChannel)
3983uint32_t __stdcall [vwCloseNetworkChannel](#) ([VWIFOPAQUE](#) vvwChannel)
3984uint32_t __stdcall [vwCheckNetworkChannel](#) ([VWIFOPAQUE](#) vvwChannel)
3985uint32_t __stdcall [vwGetCurMs](#) ([VWIFOPAQUE](#) vvwChannel)
3986VWVHANDLE __stdcall [vwChannelToHandle](#) ([VWIFOPAQUE](#) vvwChannel)
3987uint32_t __stdcall [vwHandleToChannel](#) (VWVHANDLE hVvw)

3988uint32_t __stdcall [vwwEnableChannels](#) (uint32_t IInternal0_31, uint32_t IInternal32_63, uint32_t IExternal64_95, uint32_t IExternal96_127, uint32_t INetwork128_159, uint32_t INetwork160_191)

3989uint32_t __stdcall [vwwReleaseChannels](#) (void)

3990uint32_t __stdcall [vwwIsOpen](#) (void)

3991uint32_t __stdcall [vwwGetMaxChannels](#) (void)

3992uint32_t __stdcall [vwwGetMaxInternalChannels](#) (void)

3993uint32_t __stdcall [vwwGetChannelName](#) (VWIFOPAQUE vvwChannel, char *szChannelName)

3994uint32_t __stdcall [vwwGetChannelType](#) (VWIFOPAQUE vvwChannel)

3995[BOOL](#) __stdcall [vwwSetUseAbsoluteTC](#) (BOOL bUseAbsoluteTC)

3996uint32_t __stdcall [vwwShowConfigDialog](#) (VWIFOPAQUE vvwChannel, uint32_t hWnd)

3997uint32_t __stdcall [vwwWaitForState](#) (VWIFOPAQUE vvwChannel, uint32_t ctCmd)

3998uint32_t __stdcall [vwwPlay](#) (VWIFOPAQUE vvwChannel)

3999uint32_t __stdcall [vwwPlayOffsetAt](#) (VWIFOPAQUE vvwChannel, uint32_t dwPosition, long lOffset, uint32_t dwMS)

4000uint32_t __stdcall [vwwPlayAtSpeed](#) (VWIFOPAQUE vvwChannel, uint32_t IVVWSpeed, uint32_t IEnd)

4001uint32_t __stdcall [vwwPlayFromTo](#) (VWIFOPAQUE vvwChannel, uint32_t IFrom, uint32_t ITo, int fDeferred, int fLoop)

4002uint32_t __stdcall [vwwLoadClip](#) (VWIFOPAQUE vvwChannel, char *sz8CharClipName, uint32_t IStartFrame)

4003uint32_t __stdcall [vwwSwitchClip](#) (VWIFOPAQUE vvwChannel, char *sz8CharClipName, uint32_t IPosition, [BOOL](#) bUseFrameCount)

4004uint32_t __stdcall [vwwPlayClip](#) (VWIFOPAQUE vvwChannel, char *sz8CharClipName, int fDeferred)

4005uint32_t __stdcall [vwwPlayClipFromTo](#) (VWIFOPAQUE vvwChannel, char *sz8CharClipName, uint32_t IFrom, uint32_t ITo, int fDeferred)

4006uint32_t __stdcall [vwwFastForward](#) (VWIFOPAQUE vvwChannel)

4007uint32_t __stdcall [vwwFastRewind](#) (VWIFOPAQUE vvwChannel)

4008uint32_t __stdcall [vwwPause](#) (VWIFOPAQUE vvwChannel)

4009uint32_t __stdcall [vwwSeek](#) (VWIFOPAQUE vvwChannel, uint32_t IFrame)

4010uint32_t __stdcall [vwwSeekRelative](#) (VWIFOPAQUE vvwChannel, uint32_t IFrameOffset)

4011uint32_t __stdcall [vwwStop](#) (VWIFOPAQUE vvwChannel)

4012uint32_t __stdcall [vwwRecord](#) (VWIFOPAQUE vvwChannel)

4013uint32_t __stdcall [vwwRecordFromTo](#) (VWIFOPAQUE vvwChannel, uint32_t IFrom, uint32_t ITo)

4014uint32_t __stdcall [vwwRecordStop](#) (VWIFOPAQUE vvwChannel, char *sz8CharClipName, uint32_t IDuration)

4015uint32_t __stdcall [vwwRecordStopFileName](#) (VWIFOPAQUE vvwChannel, char *sz8CharClipName, char *sz256CharFileName, uint32_t IDuration)

4016uint32_t __stdcall [vwwSetRecordPresets](#) (VWIFOPAQUE vvwChannel, uint32_t IVidEdit, uint32_t lAudEdit, uint32_t lInfEdit)

4017uint32_t __stdcall [vwwCleanRecordWipeDrive](#) (VWIFOPAQUE vvwChannel, uint32_t IWipeFolder)

4018uint32_t __stdcall [vwwEject](#) (VWIFOPAQUE vvwChannel)

4019uint32_t __stdcall [vwwMediaCmd](#) (VWIFOPAQUE vvwChannel, void *pMediaCmd)

4020uint32_t __stdcall [vwwTransfer](#) (VWIFOPAQUE vvwChannel, uint32_t ITargetChannel, uint32_t IPosition, uint32_t IStart, uint32_t IEnd, uint32_t IVidEdit, uint32_t lAudEdit, uint32_t lInfEdit, char *szClipName, int fToTape)

4021uint32_t __stdcall [vwwUpdateStatus](#) (VWIFOPAQUE vvwChannel)

4022uint32_t __stdcall [vwwGetState](#) (VWIFOPAQUE vvwChannel)

4023uint32_t __stdcall [vwwGetFlags](#) (VWIFOPAQUE vvwChannel)

4024uint32_t __stdcall [vwwGetSpeed](#) (VWIFOPAQUE vvwChannel)

4025uint32_t __stdcall [vwwGetPosition](#) (VWIFOPAQUE vvwChannel)

4026uint32_t __stdcall [vwwGetLastMs](#) (VWIFOPAQUE vvwChannel)

4027uint32_t __stdcall [vwwGetStart](#) (VWVIFOPAQUE vwwChannel)
 4028uint32_t __stdcall [vwwGetEnd](#) (VWVIFOPAQUE vwwChannel)
 4029uint32_t __stdcall [vwwGetClipName](#) (VWVIFOPAQUE vwwChannel, char *sz8CharClipName)
 4030uint32_t __stdcall [vwwGetFileName](#) (VWVIFOPAQUE vwwChannel, char *sz260CharFileName)
 4031uint32_t __stdcall [vwwGetCurTC](#) (VWVIFOPAQUE vwwChannel, char *sz14ByteTC)
 4032uint32_t __stdcall [vwwGetCurState](#) (VWVIFOPAQUE vwwChannel, char *sz14ByteState)
 4033char * __stdcall [vwwGetNextClip](#) (VWVIFOPAQUE vwwChannel, char *sz8CharLastClipCurClip)
 4034uint32_t __stdcall [vwwGetClipInfo](#) (VWVIFOPAQUE vwwChannel, char *sz8CharClipName, uint32_t *lStart, uint32_t *lEnd, uint32_t *lVidEdit, uint32_t *lAudEdit, uint32_t *lInfEdit, char *szFileName)
 4035uint32_t __stdcall [vwwSetMetaData](#) (VWVIFOPAQUE vwwChannel, char *sz8CharClipName, uint32_t vwwInfoRequest, uint32_t nValue, char *szValue)
 4036uint32_t __stdcall [vwwGetMetaData](#) (VWVIFOPAQUE vwwChannel, char *sz8CharClipName, char *sz260CharFileName, uint32_t vwwInfoRequest, char *szValue)
 4037uint32_t __stdcall [vwwGetNextClipEx](#) (VWVIFOPAQUE vwwChannel, char *sz8CharClipName, uint32_t *lCreation, uint32_t *lLastModification, uint32_t *lFileSize, uint32_t *lDiskFragments)
 4038uint32_t __stdcall [vwwCopyClip](#) (VWVIFOPAQUE vwwChannel, char *szSourceClip, char *szDestClip, uint32_t lStart, uint32_t lEnd)
 4039uint32_t __stdcall [vwwEDLResetToStart](#) (VWVIFOPAQUE vwwChannel)
 4040uint32_t __stdcall [vwwEDLGetEdit](#) (VWVIFOPAQUE vwwChannel, uint32_t *lRecordIn, uint32_t *lPlayIn, uint32_t *lPlayOut, uint32_t *lVidEdit, uint32_t *lAudEdit, uint32_t *lInfEdit, char *szClipName, char *szFileName, [BOOL](#) bClipInfo)
 4041uint32_t __stdcall [vwwEDLSetExtendedInfo](#) (VWVIFOPAQUE vwwChannel, uint32_t lType, uint32_t lRecordIn, uint32_t lValue, char *sz260Comment, uint32_t lDuration)
 4042uint32_t __stdcall [vwwEDLGetExtendedInfo](#) (VWVIFOPAQUE vwwChannel, uint32_t lType, uint32_t lRecordIn, char *sz260Comment, uint32_t *plExtra)
 4043uint32_t __stdcall [vwwGetLastChangeXferMs](#) (VWVIFOPAQUE vwwChannel)
 4044uint32_t __stdcall [vwwGetLastChangeMs](#) (VWVIFOPAQUE vwwChannel, uint32_t lClipSpace, uint32_t *lNumClips)
 4045uint32_t __stdcall [vwwInsert](#) (VWVIFOPAQUE vwwChannel, char *szClipName, char *szFileName, uint32_t lPosition, uint32_t lStart, uint32_t lEnd, uint32_t lVidEdit, uint32_t lAudEdit, uint32_t lInfEdit, int fRipple)
 4046uint32_t __stdcall [vwwBlank](#) (VWVIFOPAQUE vwwChannel, char *szClipName, uint32_t lStart, uint32_t lEnd, uint32_t lVidEdit, uint32_t lAudEdit, uint32_t lInfEdit, int fRipple)
 4047uint32_t __stdcall [vwwBlankAllClipIds](#) (VWVIFOPAQUE vwwChannel)
 4048uint32_t __stdcall [vwwDelete](#) (VWVIFOPAQUE vwwChannel, char *szClipName, uint32_t lStart, uint32_t lEnd, uint32_t lVidEdit, uint32_t lAudEdit, uint32_t lInfEdit, int fRipple)
 4049uint32_t __stdcall [vwwTrim](#) (VWVIFOPAQUE vwwChannel, uint32_t lPosition, uint32_t lStartOffset, uint32_t lEndOffset, uint32_t lVidEdit, uint32_t lAudEdit, uint32_t lInfEdit, int fRipple)
 4050uint32_t __stdcall [vwwValueSupported](#) (VWVIFOPAQUE vwwChannel, uint32_t lValueType)
 4051uint32_t __stdcall [vwwValueGet](#) (VWVIFOPAQUE vwwChannel, uint32_t lValueType, uint32_t *plMin, uint32_t *plMax)
 4052uint32_t __stdcall [vwwValueSet](#) (VWVIFOPAQUE vwwChannel, uint32_t lValueType, uint32_t lSetting)
 4053uint32_t __stdcall [vwwValueSet2](#) (VWVIFOPAQUE vwwChannel, uint32_t lValueType, uint32_t lSetting, uint32_t lStart, uint32_t lEnd, uint32_t lVidChan, uint32_t lAudChan, uint32_t lInfChan)
 4054uint32_t __stdcall [vwwGetClipMode](#) (VWVIFOPAQUE vwwChannel)
 4055uint32_t __stdcall [vwwSetClipMode](#) (VWVIFOPAQUE vwwChannel, uint32_t lSetting)
 4056uint32_t __stdcall [vwwGetTCType](#) (VWVIFOPAQUE vwwChannel)
 4057uint32_t __stdcall [vwwSetTCType](#) (VWVIFOPAQUE vwwChannel, uint32_t lSetting)
 4058uint32_t __stdcall [vwwGetTCSOURCE](#) (VWVIFOPAQUE vwwChannel)
 4059uint32_t __stdcall [vwwSetTCSOURCE](#) (VWVIFOPAQUE vwwChannel, uint32_t lSetting)
 4060uint32_t __stdcall [vwwGetAutoMode](#) (VWVIFOPAQUE vwwChannel)
 4061uint32_t __stdcall [vwwSetAutoMode](#) (VWVIFOPAQUE vwwChannel, uint32_t lSetting)

4062uint32_t __stdcall [vwwGetCurrentPresets](#) (VWIFOPAQUE vvwChannel, uint32_t *plVidEdit, uint32_t *plAudEdit, uint32_t *plInfEdit)

4063uint32_t __stdcall [vwwGetAvailablePresets](#) (VWIFOPAQUE vvwChannel, uint32_t *plVidEdit, uint32_t *plAudEdit, uint32_t *plInfEdit)

4064uint32_t __stdcall [vwwGetAudioInput](#) (VWIFOPAQUE vvwChannel)

4065uint32_t __stdcall [vwwSetAudioInput](#) (VWIFOPAQUE vvwChannel, uint32_t lSetting)

4066uint32_t __stdcall [vwwGetAudioInputLevel](#) (VWIFOPAQUE vvwChannel)

4067uint32_t __stdcall [vwwSetAudioInputLevel](#) (VWIFOPAQUE vvwChannel, uint32_t lSetting)

4068uint32_t __stdcall [vwwGetAudioOutput](#) (VWIFOPAQUE vvwChannel)

4069uint32_t __stdcall [vwwSetAudioOutput](#) (VWIFOPAQUE vvwChannel, uint32_t lSetting)

4070uint32_t __stdcall [vwwGetAudioOutputLevel](#) (VWIFOPAQUE vvwChannel)

4071uint32_t __stdcall [vwwSetAudioOutputLevel](#) (VWIFOPAQUE vvwChannel, uint32_t lSetting)

4072uint32_t __stdcall [vwwGetAudioPeakRMS](#) (VWIFOPAQUE vvwChannel, uint32_t lAudEdit, uint32_t *plPeaks)

4073uint32_t __stdcall [vwwGetVideoInput](#) (VWIFOPAQUE vvwChannel)

4074uint32_t __stdcall [vwwSetVideoInput](#) (VWIFOPAQUE vvwChannel, uint32_t lSetting)

4075uint32_t __stdcall [vwwGetVideoOutput](#) (VWIFOPAQUE vvwChannel)

4076uint32_t __stdcall [vwwSetVideoOutput](#) (VWIFOPAQUE vvwChannel, uint32_t lSetting)

4077uint32_t __stdcall [vwwGetVideoInputSetup](#) (VWIFOPAQUE vvwChannel)

4078uint32_t __stdcall [vwwSetVideoInputSetup](#) (VWIFOPAQUE vvwChannel, uint32_t lSetting)

4079uint32_t __stdcall [vwwGetVideoInputVideo](#) (VWIFOPAQUE vvwChannel)

4080uint32_t __stdcall [vwwSetVideoInputVideo](#) (VWIFOPAQUE vvwChannel, uint32_t lSetting)

4081uint32_t __stdcall [vwwGetVideoInputHue](#) (VWIFOPAQUE vvwChannel)

4082uint32_t __stdcall [vwwSetVideoInputHue](#) (VWIFOPAQUE vvwChannel, uint32_t lSetting)

4083uint32_t __stdcall [vwwGetVideoInputChroma](#) (VWIFOPAQUE vvwChannel)

4084uint32_t __stdcall [vwwSetVideoInputChroma](#) (VWIFOPAQUE vvwChannel, uint32_t lSetting)

4085uint32_t __stdcall [vwwGetVideoTBCSetup](#) (VWIFOPAQUE vvwChannel)

4086uint32_t __stdcall [vwwSetVideoTBCSetup](#) (VWIFOPAQUE vvwChannel, uint32_t lSetting)

4087uint32_t __stdcall [vwwGetVideoTBCVideo](#) (VWIFOPAQUE vvwChannel)

4088uint32_t __stdcall [vwwSetVideoTBCVideo](#) (VWIFOPAQUE vvwChannel, uint32_t lSetting)

4089uint32_t __stdcall [vwwGetVideoTBCHue](#) (VWIFOPAQUE vvwChannel)

4090uint32_t __stdcall [vwwSetVideoTBCHue](#) (VWIFOPAQUE vvwChannel, uint32_t lSetting)

4091uint32_t __stdcall [vwwGetVideoTBCChroma](#) (VWIFOPAQUE vvwChannel)

4092uint32_t __stdcall [vwwSetVideoTBCChroma](#) (VWIFOPAQUE vvwChannel, uint32_t lSetting)

4093uint32_t __stdcall [vwwGetVideoGenlock](#) (VWIFOPAQUE vvwChannel)

4094uint32_t __stdcall [vwwSetVideoGenlock](#) (VWIFOPAQUE vvwChannel, uint32_t lSetting)

4095uint32_t __stdcall [vwwGetVideoGenlockSource](#) (VWIFOPAQUE vvwChannel)

4096uint32_t __stdcall [vwwSetVideoGenlockSource](#) (VWIFOPAQUE vvwChannel, uint32_t lSetting)

4097uint32_t __stdcall [vwwGetCompressionRate](#) (VWIFOPAQUE vvwChannel)

4098uint32_t __stdcall [vwwSetCompressionRate](#) (VWIFOPAQUE vvwChannel, uint32_t lSetting)

4099uint32_t __stdcall [vwwGetSuperImpose](#) (VWIFOPAQUE vvwChannel)

4100uint32_t __stdcall [vwwSetSuperImpose](#) (VWIFOPAQUE vvwChannel, uint32_t lSetting)

4101uint32_t __stdcall [vwwGetErrorLogMs](#) ()

4102uint32_t __stdcall [vwwSetErrorLog](#) (uint32_t lSetting)

4103uint32_t __stdcall [vwwGetNumberOfErrors](#) (uint32_t *plErrors)

4104uint32_t __stdcall [vwwGetErrorLength](#) (uint32_t *plLastError, uint32_t *plErrorLength)

4105uint32_t __stdcall [vwwGetError](#) (uint32_t *plLastError, uint32_t *plSeverity, char *szError, short *psYear, short *psMonth, short *psDay, short *psHour, short *psMinute, short *psSecond, short *psMillisecond)

4106uint32_t __stdcall [vwwGetTotalTime](#) (VWIFOPAQUE vvwChannel)

4107uint32_t __stdcall [vwwGetFreeTime](#) (VWIFOPAQUE vvwChannel)

4108uint32_t __stdcall [vwwGetTotalStorage](#) (VWIFOPAQUE vvwChannel)

4109uint32_t __stdcall [vwwGetFreeStorage](#) (VWIFOPAQUE vvwChannel)

4110uint32_t __stdcall [vwwGetChannelCapabilities](#) (VWIFOPAQUE vvwChannel)

4111char * __stdcall [vwwGetVWVVersion](#) ()

```

4112char *__stdcall vwwGetMRVersion ()
4113char *__stdcall vwwGetVWVWType (VWVWIFOPAQUE vwwChannel)
4114uint32_t __stdcall vwwGetPiconName (char *szFileName)
4115uint32_t __stdcall vwwPlayAtMs (VWVWIFOPAQUE vwwChannel, uint32_t lMs)
4116uint32_t __stdcall vwwRecordAtMs (VWVWIFOPAQUE vwwChannel, uint32_t lMs, uint32_t
    lStart, uint32_t lEnd)
4117uint32_t __stdcall vwwXMLProjectSave (char *lpszProjectName)
4118uint32_t __stdcall vwwXMLProjectCheckOpen (char *lpszProjectName)
4119uint32_t __stdcall vwwXMLProjectOpen (char *lpszProjectName, int bDeleteConflictingFiles)
4120uint32_t __stdcall vwwGetNumberOfBackUps (VWVWIFOPAQUE vwwChannel, uint32_t
    *pdwBackUps, uint32_t dwClipMode)
4121uint32_t __stdcall vwwSetBackUpNumber (VWVWIFOPAQUE vwwChannel, uint32_t dwBackUp,
    uint32_t dwClipMode)
4122uint32_t __stdcall vwwGetLicenseOptions (VWVWIFOPAQUE vwwChannel, uint32_t
    *pdwOptions, uint32_t *pdwLicenseValid)
4123void __stdcall vwwFreeString (char *szString)

```

Define Documentation

#define COMPILING_DIRECT_MEDIACMD_SOURCE

Definition at line 41 of file vwwif.h.

#define SPD_FWD_MAX 5896800

Maximum possible play speed in VVW see [MEDIACMD::lSpeed](#).

Definition at line 987 of file vwwif.h.

#define SPD_FWD_PLAY 65520L

Forward play speed (normal) in VVW (65520) see [MEDIACMD::lSpeed](#).

Definition at line 981 of file vwwif.h.

#define SPD_ILLEGAL 2147483647L

Illegal speed, set [MEDIACMD::lSpeed](#) to this value if not used.

Definition at line 991 of file vwwif.h.

#define SPD_PAUSE 0L

Pause speed (0%) in VVW (0) see [MEDIACMD::lSpeed](#).

Definition at line 983 of file vwwif.h.

#define SPD_REV_MAX (-SPD_FWD_MAX)

Minimum possible play speed in VVW see [MEDIACMD::lSpeed](#).

Definition at line 989 of file vvwif.h.

#define SPD_REV_PLAY (-SPD_FWD_PLAY)

Reverse play speed (-100%) in VVW (-65520) see [MEDIACMD::lSpeed](#).

Definition at line 985 of file vvwif.h.

Typedef Documentation

typedef uint32_t [VWIFOPAQUE](#)

NOTE This is an SDK file, so it needs to stay clean and being able to compile independently of the main source tree. Please test any changes under linux, macOS and windows.

Definition at line 32 of file vvwif.h.

Function Documentation

uint32_t __stdcall vvwBlank ([VWIFOPAQUE](#) vvwChannel, char * szClipName, uint32_t IStart, uint32_t IEnd, uint32_t IVidEdit, uint32_t IAudEdit, uint32_t IInfEdit, int fRipple)

Definition at line 2203 of file vvwif.cpp.

uint32_t __stdcall vvwBlankAllClipIds ([VWIFOPAQUE](#) vvwChannel)

Definition at line 2245 of file vvwif.cpp.

VVWHANDLE __stdcall vvwChannelToHandle ([VWIFOPAQUE](#) vvwChannel)

Convert a 0..3 channel to 0, 64, 65536 Mostly internal Undocumented

Definition at line 606 of file vvwif.cpp.

uint32_t __stdcall vvwCheckNetworkChannel ([VWIFOPAQUE](#) vvwChannel)

Check that we are still conneted through the network

Definition at line 583 of file vvwif.cpp.

uint32_t __stdcall vvwCleanRecordWipeDrive ([VWIFOPAQUE](#) vvwChannel, uint32_t IWipeFolder)

Reset Record file and/or delete all files found in the record folder Returns 0 if successful, else an error code.

Reset record file(s) and/or delete all files found in the record folder

Definition at line 1255 of file vvwif.cpp.

uint32_t __stdcall vvwCloseNetworkChannel ([VWIFOPAQUE](#) vvwChannel)

Private close for the direct source MediaCMD API access

Definition at line 538 of file vvwif.cpp.

uint32_t __stdcall vvwCopyClip ([VWIFOPAQUE](#) vvwChannel, char * szSourceClip, char * szDestClip, uint32_t IStart, uint32_t IEnd)

Create a virtual copy of a clip, changing the in and out points if necessary. To use the whole clip, set IStart to 0 and the end to -1. Returns 0 if successful, else an error code.

Definition at line 1981 of file vvwif.cpp.

uint32_t __stdcall vvwDelete ([VWIFOPAQUE](#) vvwChannel, char * szClipName, uint32_t IStart, uint32_t IEnd, uint32_t IVidEdit, uint32_t IAudEdit, uint32_t IInfEdit, int fRipple)

Definition at line 2261 of file vvwif.cpp.

uint32_t __stdcall vvwEDLGetEdit ([VWIFOPAQUE](#) vvwChannel, uint32_t * IRecordIn, uint32_t * IPlayIn, uint32_t * IPlayOut, uint32_t * IVidEdit, uint32_t * IAudEdit, uint32_t * IInfEdit, char * szClipName, char * szFileName, [BOOL](#) bClipInfo)

Returns an edit line from the VTR space of an internal channel. The function will continue to return the next edit in the timecode space until the last edit is returned, after which an error will be returned. To reset to the start of the Edl use EDLResetToStart. Returns 0 if successful else an Error code.

Definition at line 2028 of file vvwif.cpp.

uint32_t __stdcall vvwEDLGetExtendedInfo ([VWIFOPAQUE](#) vvwChannel, uint32_t IType, uint32_t IRecordIn, char * sz260Comment, uint32_t * plExtra)

Get Comment

Definition at line 2096 of file vvwif.cpp.

uint32_t __stdcall vvwEDLResetToStart ([VWIFOPAQUE](#) vvwChannel)

Reset the edl returns in VTR mode to the first element of the list.

Definition at line 2008 of file vvwif.cpp.

uint32_t __stdcall vvwEDLSetExtendedInfo ([VWIFOPAQUE](#) vvwChannel, uint32_t IType, uint32_t IRecordIn, uint32_t IValue, char * sz260Comment, uint32_t IDuration)

Set Comment

Definition at line 2073 of file vvwif.cpp.

uint32_t __stdcall vvwEject ([VWIFOPAQUE](#) vvwChannel)

Eject the current media if it is removable. Normally only used with VTRs. Returns 0 if successful, else an error code.

Eject the current media

Definition at line 1279 of file vvwif.cpp.

uint32_t __stdcall vvwEnableChannels (uint32_t *lInternal0_31*, uint32_t *lInternal32_63*, uint32_t *lExternal64_95*, uint32_t *lExternal96_127*, uint32_t *lNetwork128_159*, uint32_t *lNetwork160_191*)

Enable or disable channels based on the bit array supplied. VVW can contain up to 256 channels per access point. Channels 193-255 are disabled by default. The remaining channels may be enabled (if the corresponding bit is set to 1) or disabled (if the corresponding bit is set to 0) with this call. The first 64 channels (0 through 63) are reserved for internal ddr channels. Then next 64 channels (64 through 127) are reserved for VTR or DDR devices controlled via serial, Odetics or Louth protocol. The remaining channels are for controlling other devices through the network. Please note that a network channel controls all the channels on the network server box, so disabling one network connection may disable more than one channel. Always call GetMaxChannels() after setting the bits to make sure all the channels you expect exist actually exist. This should be the first call made to the activex component.

Set the allowable channels

Definition at line 634 of file vvwif.cpp.

uint32_t __stdcall vvwFastForward ([VVWIFOPAQUE](#) vvwChannel)

Set the channel into its fastest possible forward motion state. Returns 0 if successful, else an error code.

Play Fast Forward

Definition at line 1031 of file vvwif.cpp.

uint32_t __stdcall vvwFastRewind ([VVWIFOPAQUE](#) vvwChannel)

Set the channel into its fastest possible reverse motion state. Returns 0 if successful, else an error code.

Play Fast Reverse

Definition at line 1046 of file vvwif.cpp.

void __stdcall vvwFreeString (char * szString)

Free a string value returned by the channel.

Definition at line 4788 of file vvwif.cpp.

uint32_t __stdcall vvwGetAudioInput ([VVWIFOPAQUE](#) vvwChannel)

ADD FUNCTION lAudIn Get the current audio input.
[GS_AUDSELECT_UNBALANCED_10](#) [GS_AUDSELECT_UNBALANCED_4](#)
[GS_AUDSELECT_BALANCED_10](#) [GS_AUDSELECT_BALANCED_4](#)
[GS_AUDSELECT_SPDIF](#) [GS_AUDSELECT_AES_EBU](#) [GS_AUDSELECT_EMBEDDED](#)

Definition at line 2572 of file vvwif.cpp.

uint32_t __stdcall vvwGetAudioInputLevel ([VVWIFOPAQUE](#) vvwChannel)

Get the current audio input level

Definition at line 2584 of file vvwif.cpp.

uint32_t __stdcall vvwGetAudioOutput ([VWIFOPAQUE](#) vvwChannel)

Get the current audio Output – See Get/SetAudioInput
[GS_AUDSELECT_UNBALANCED_10](#) [GS_AUDSELECT_UNBALANCED_4](#)
[GS_AUDSELECT_BALANCED_10](#) [GS_AUDSELECT_BALANCED_4](#)
[GS_AUDSELECT_SPDIF](#) [GS_AUDSELECT_AES_EBU](#) [GS_AUDSELECT_EMBEDDED](#)

Definition at line 2596 of file vvwif.cpp.

uint32_t __stdcall vvwGetAudioOutputLevel ([VWIFOPAQUE](#) vvwChannel)

Get the current audio output level.

Definition at line 2608 of file vvwif.cpp.

uint32_t __stdcall vvwGetAudioPeakRMS ([VWIFOPAQUE](#) vvwChannel, uint32_t lAudEdit, uint32_t * pIPeaks)

Returns the RMS and Peak audio levels of the input (stop/record) or output (play/pause).

Definition at line 2620 of file vvwif.cpp.

uint32_t __stdcall vvwGetAutoMode ([VWIFOPAQUE](#) vvwChannel)

Calls ValueXXX with gsAutoMode. Required for play lists, deferred commands and auto edit commands on VTRs.

Definition at line 2477 of file vvwif.cpp.

uint32_t __stdcall vvwGetAvailablePresets ([VWIFOPAQUE](#) vvwChannel, uint32_t * pIVidEdit, uint32_t * pIAudEdit, uint32_t * pIInfEdit)

ADD FUNCTIONS IVidEdit, IAudEdit, IInfEdit Returns the supported audio, video and info presets for a channel.

Definition at line 2533 of file vvwif.cpp.

uint32_t __stdcall vvwGetChannelCapabilities ([VWIFOPAQUE](#) vvwChannel)

Get the available commands for a channel.

Definition at line 2817 of file vvwif.cpp.

uint32_t __stdcall vvwGetChannelName ([VWIFOPAQUE](#) vvwChannel, char * szChannelName)

Get the name of the current channel. For unix and dll access, pass a null to get the channel name size, then pass in a pointer that points to a memory size of at least that many bytes (ANSI characters only).

Get the current channel name

Definition at line 714 of file vvwif.cpp.

uint32_t __stdcall vvwGetChannelType ([VWIFOPAQUE](#) vvwChannel)

Returns the basic type of the channel (VTR, Internal, User, House)
VWV_CHANATYPE_HOUSE 0x1 1 VWV_CHANATYPE_INTERNAL 0x2 2
VWV_CHANATYPE_VTR_DDR 0x4 4 VWV_CHANATYPE_UNKNOWN 0xFFFFFFFF -1

Get the current channel type

Definition at line 754 of file vvwif.cpp.

uint32_t __stdcall vvwGetClipInfo (VWIFOPAQUE vvwChannel, char * sz8CharClipName, uint32_t * IStart, uint32_t * IEnd, uint32_t * IVidEdit, uint32_t * IAudEdit, uint32_t * IInfEdit, char * szFileName)

Returns the basic information from szClip. The information is located in IStart, IEnd, IVidEdit, IAudEdit and szFileName as the in point, out point, number of video channels, number of audio channels, and the file name respectively. Returns 0 if successful, else an error code.

Definition at line 1867 of file vvwif.cpp.

uint32_t __stdcall vvwGetClipMode (VWIFOPAQUE vvwChannel)

Calls ValueXXX with gsClipMode. If equal to 1 then the channel is in clip mode, if 0 the channel is in VTR mode.

Definition at line 2432 of file vvwif.cpp.

uint32_t __stdcall vvwGetClipName (VWIFOPAQUE vvwChannel, char * sz8CharClipName)

Only supported in clip Mode. Returns the current clip name, if any. For dll access, the memory must be at least 9 bytes uint32_t (8 character bytes + NULL) and is always ANSI.

Definition at line 1517 of file vvwif.cpp.

uint32_t __stdcall vvwGetCompressionRate (VWIFOPAQUE vvwChannel)

Get the current compression rate

Definition at line 2775 of file vvwif.cpp.

uint32_t __stdcall vvwGetCurMs (VWIFOPAQUE vvwChannel)

Get the current millisecond time.

Parameters:

<i>vwChannel</i>	the mediacmd target being controlled (internal, external, network)
------------------	--

Returns:

the current millisecond counter for that channel

Definition at line 2810 of file vvwif.cpp.

uint32_t __stdcall vvwGetCurrentPresets (VWIFOPAQUE vvwChannel, uint32_t * pIVidEdit, uint32_t * pIAudEdit, uint32_t * pIInfEdit)

ADD FUNCTIONS IVidEdit, IAudEdit, IInfEdit Returns the current audio, video and info presets for a channel.

Definition at line 2489 of file vvwif.cpp.

uint32_t __stdcall vvwGetCurState (VWIFOPAQUE vvwChannel, char * sz14ByteState)

Returns the current state as a string (e.g. "Play"). For dll access, the memory must always be at least 15 bytes uint32_t (14 byte state + NULL) and is always ANSI.

Stop - Stop all playback, and normally place all channels into passthrough

Pause - Halt all channels. Display current video frame and silence audio

Seek - With cmdFlags::cfUsePosition and MEDICMD::dwPosition, goto that frame and Pause

Play - Play all channels. May be modified by (cmdFlags::cfUseSpeed +

[MEDIACMD::lSpeed](#)) and
([cmdFlags::cfUsePosition](#) or [cmdFlags::cfUsePositionOffset](#)) and
[MEDIACMD::dwPosition](#)) or
([cmdFlags::cfUseStart](#) or [cmdFlags::cfUseStartOffset](#)) and [MEDIACMD::dwStart](#)) or
([cmdFlags::cfUseEnd](#) or [cmdFlags::cfUseEndOffset](#)) and [MEDIACMD::dwEnd](#)) as well as
[MEDIACMD::dwCmdAlt](#) with certain [cmdFlags](#) to play from-top, at speed or combinations of
the above.

Record - Record one or a combination of video/audio/info to disk. May be modified, as with
[ctPlay](#) by flags and structure members such as

([cmdFlags::cfUsePosition](#) or [cmdFlags::cfUsePositionOffset](#)) and
[MEDIACMD::dwPosition](#)) or
([cmdFlags::cfUseStart](#) or [cmdFlags::cfUseStartOffset](#)) and [MEDIACMD::dwStart](#)) or
([cmdFlags::cfUseEnd](#) or [cmdFlags::cfUseEndOffset](#)) and [MEDIACMD::dwEnd](#)) as well as
([cmdFlags::cfUseClipID](#) and [MEDIACMD::arbID](#)) or
([cmdFlags::cfDeferred](#) or [cmdFlags::cfOverrideDeferred](#)) or [MEDIACMD::dwCmdAlt](#) with
certain [cmdFlags](#) to record from-top, at speed or combinations of the above.

Record Stop - Set the channel into a record ready state, normally passthrough with the
recording file preallocated, and possible pass start end and name information. See

([cmdFlags::cfUseStart](#) [cmdFlags::cfUseStartOffset](#) and [MEDIACMD::dwStart](#))
([cmdFlags::cfUseEnd](#) [cmdFlags::cfUseEndOffset](#) and [MEDIACMD::dwEnd](#))
([cmdFlags::cfUseClipID](#) and [MEDIACMD::arbID](#)) for record setups.

Eject - Stop the channel and unload removable media, if possible, else same as stop

Transfer - Transfer media from one channel to another. Normally used to transfer internal media
to or from an external tape device.

Insert Clip or Timecode Area - Used in time code space (TCSpace.h) and Clip Space
(ClipSpace.h) to add new clips or areas. Inserted media is defined by ([cmdFlags::cfUseStart](#) -
[MEDIACMD::dwStart](#), [cmdFlags::cfUseEnd](#) - [MEDIACMD::dwEnd](#)) for clip being added
and [cmdFlags::cfUsePosition](#) - [MEDIACMD::dwPosition](#) for target. Also
([cmdFlags::cfUseClipID](#) and [MEDIACMD::arbID](#)) may be used to specify a file name.
[cmdFlags::cfUsePresets](#) and [MEDIACMD::dwVideoChannels](#),
[MEDIACMD::dwAudioChannels](#), [MEDIACMD::dwInfoChannels](#) are also respected if set.
[cmdFlags::cfRipple](#) may also be used to insert over

NOTE - The [ctTransfer](#) command is ALWAYS sent to the target with the SOURCE channel in
the [MEDIACMD::dwCmdAlt](#) member and [cmdFlags::cfUseCmdAlt](#) set UNLESS one of the
devices is slow/high latency/sloppy (read VTR), in which case it always receives the command so
it can master the transfer and the [cmdFlags::cfInvert](#) is used to set the direction.

Blank a Timecode Area - Used in time code space (TCSpace.h) to set an area to black and silent
audio. ([cmdFlags::cfUseStart](#) - [MEDIACMD::dwStart](#), [cmdFlags::cfUseEnd](#) -
[MEDIACMD::dwEnd](#)) set the area to be blanked. [cmdFlags::cfUsePresets](#) and
[MEDIACMD::dwVideoChannels](#), [MEDIACMD::dwAudioChannels](#),
[MEDIACMD::dwInfoChannels](#) are also respected if set. [cmdFlags::cfRipple](#) may also be used to
remove blank area. With this command, no media is removed from storage.

Delete a clip (ClipSpace.h) or an area (TCSpace.h). Deletes the media from storage and from
the current space. For ClipSpace, [cmdFlags::cfUseClipID](#) and [MEDIACMD::arbID](#) must be
specified, and if any sub clip or super clips exist, the id will be removed but the media will
not be deleted.

For TCSpace, `cmdFlags::cfUseStart` and [MEDIACMD::dwStart](#) with `cmdFlags::cfUseEnd` and [MEDIACMD::dwEnd](#) should be used to specify the time code segment to be deleted.

`cmdFlags::cfUsePresets` and [MEDIACMD::dwVideoChannels](#), [MEDIACMD::dwAudioChannels](#), [MEDIACMD::dwInfoChannels](#) may also be used to delete specific channels. If `cmdFlags::cfRipple` is set, then the TCSpace will close over the deleted material, changing all timecode location beyond the deletion point by minus the size of the deletion.

Trim a clip or area - Currently not implemented. Use `cmdType::ctSetValue` and `#GS_CLIP_INFO` to trim a clip, or a combination of `cmdType::ctInsert`, `cmdType::ctDelete`, `cmdType::ctBlank` to trim a tcspace area.

Channel select - select active channels, preview passthrough channels (to preview and edit) recording channels (to create a split edit ala CMX) Requires `cmdFlags::cfUsePresets` and [MEDIACMD::dwVideoChannels](#), [MEDIACMD::dwAudioChannels](#), [MEDIACMD::dwInfoChannels](#)

Get the current state of the controlled channel(s) - Fills the user supplied [MEDIACMD](#) structure with the current state. Look for `cmdType::ctError`, `cmdType::ctStop`, `cmdType::ctEject`, `cmdType::ctPause`, `cmdType::ctPlay`, `cmdType::ctRecStop`, `cmdType::ctRecord` for basic state. For valid fields, check `cmdFlags::cfDeferred` : we have a deferred clip `cmdFlags::cfTimeMs` : [MEDIACMD::dwCmdAlt](#) has millisecond performance counter info `cmdFlags::cfUseSpeed` : [MEDIACMD::!Speed](#) has the valid current speed `cmdFlags::cfUsePresets` : [MEDIACMD::dwVideoChannels](#), [MEDIACMD::dwAudioChannels](#), [MEDIACMD::dwInfoChannels](#) contain preset information `cmdFlags::cfUsePosition` : [MEDIACMD::dwPosition](#) contains current position `cmdFlags::cfUseStart` : [MEDIACMD::dwStart](#) has starting frame position `cmdFlags::cfUseEnd` : [MEDIACMD::dwEnd](#) has end frame position (+1 the out is never included) `cmdFlags::cfUseClipID` : [MEDIACMD::arbID](#) has current clip name (8 char for louth and odetics) `cmdFlags::cfFields` : [MEDIACMD::dwPosition](#), [MEDIACMD::dwStart](#) and [MEDIACMD::dwEnd](#) are in fields if they are valid `cmdFlags::cfNoReturn` : return is invalid.

Set the current state - Used for control type channels such as Serial 422 control (`vvwCtl.h`) and network control (`vvwNet.h`). Tells the controller or user what our current state is. The state should be reported honestly, as it is the receivers responsibility to transition states in an appropriate way for its controller. For actual channels (`vvwInt.h`, `vvwExt.h`, `vvwNet.h`-as controller, `vvwDS2.h`, etc), the state should be set by using one of the transport commands (`cmdType::ctPlay` etc) above. Get a non transport setting - Used for one time setups on channel. Includes audio levels, video proc amps, audio/video input, compression type and level and many others See: [cmdGetSetValue](#) for possible commands

Set a non transport setting - Used for one time setups on channel. Includes audio levels, video proc amps, audio/video input, compression type and level and many others See: [cmdGetSetValue](#) for possible commands

Check support for a non transport setting - Used for one time setups on channel. Includes audio levels, video proc amps, audio/video input, compression type and level and many others See: [cmdGetSetValue](#) for possible commands

Indicates the an error in the channel has occurred. Return only. See [MEDIACMD::dwCmdAlt](#) for error code and [MEDIACMD::arbID](#) for message if any. These members will be valid if `cmdFlags::cfUseCmdAlt` and `cmdFlags::cfUseClipID` are set

Terminate Close A Channel - Only used by remote devices that cannot close the channel directly such as `vvwNet.h`. Channel may not actually close when this is called, but the communications pipe will be closed and wait for another connection.

Abort the current operation - Use to abort operations that would normally ignore extraneous commands such as non-linear playback sequences, records or if the channel just seems to be stuck. Makes a good panic button.

Definition at line 1570 of file vvwif.cpp.

uint32_t __stdcall vvwGetCurTC ([VWVIFOPAQUE](#) vvwChannel, char * sz14ByteTC)

Returns the current time code as a string (e.g. "00:01:00:00"). For dll access, the memory must always be at least 15 bytes uint32_t (14 byte time code plus id + NULL) and is always ANSI.

Definition at line 1546 of file vvwif.cpp.

uint32_t __stdcall vvwGetEnd ([VWVIFOPAQUE](#) vvwChannel)

Return the current end point or out point if cfUseEnd is set.

Definition at line 1503 of file vvwif.cpp.

uint32_t __stdcall vvwGetError (uint32_t * pLastError, uint32_t * pSeverity, char * szError, short * psYear, short * psMonth, short * psDay, short * psHour, short * psMinute, short * psSecond, short * psMillisecond)

Get the current error. Sets pointer to the next one automatically

Definition at line 2979 of file vvwif.cpp.

uint32_t __stdcall vvwGetErrorLength (uint32_t * pLastError, uint32_t * pErrorLength)

Get the length of the current error string

Definition at line 2961 of file vvwif.cpp.

uint32_t __stdcall vvwGetErrorLogMs ()

Get the ms time the last error was added to the error log

Definition at line 2931 of file vvwif.cpp.

uint32_t __stdcall vvwGetFileName ([VWVIFOPAQUE](#) vvwChannel, char * sz260CharFileName)

Returns the current file name, if any. For dll access, the memory must be at least 261 bytes uint32_t (260 bytes max path + NULL) and is always ANSI.

Definition at line 1533 of file vvwif.cpp.

uint32_t __stdcall vvwGetFlags ([VWVIFOPAQUE](#) vvwChannel)

Returns the current flags

cfDeferred = 1, // 0x00000001 This is a delayed cfOverrideDeferred = 1 << 30, // 0x40000000
Override all previous deferred commands cfTimeMs = 1 << 1, // 0x00000002 Use Millisecond
time for delayed time, not fields cfTimeTarget = 1 << 2, // 0x00000004 Delayed time is offset
from current time code cfTimeHouseClock = 1 << 3, // 0x00000008 Delayed time is based on
absolute (real) time cfUseSpeed = 1 << 4, // 0x00000010 Set the new speed cfUsePresets = 1
<< 5, // 0x00000020 Use video and audio edit presets cfUsePosition = 1 << 6, // 0x00000040
Use the position setting cfUsePositionOffset = 1 << 7, // 0x00000080 Position is an offset
cfUseStart = 1 << 8, // 0x00000100 Start a new timecode cfUseStartOffset = 1 << 9, //
0x00000200 Start is an offset from current tc cfUseEnd = 1 << 10, // 0x00000400 End
command as specified cfUseEndOffset = 1 << 11, // 0x00000800 End is and offset from
current tc cfUseAllIDs = 1 << 12, // 0x00001000 Use all clip IDs cfUseClipID = 1 << 13, //
0x00002000 Use new clip ID, otherwise use last or none cfNoClipFiles = 1 << 14, //

0x00004000 Use new clip ID, otherwise use last or none cfNoTCSpaces = 1 << 15, //
0x00008000 Use new clip ID, otherwise use last or none cfUseCmdAlt = 1 << 16, //
0x00010000 Use the dwCmdAlt cfIsShuttle = 1 << 17, // 0x00020000 Use speed in play for
shuttle cfFields = 1 << 20, // 0x00100000 Position, start and end are fields, not frames
cfRipple = 1 << 21, // 0x00200000 Ripple for insert or delete cfLoop = 1 << 22, //
0x00400000 Loop the clip or in out cfTrigger = 1 << 23, // 0x00800000 Trigger using dsync
class cfPreview = 1 << 24, // 0x01000000 Preview set (EE, non rt play) cfInvert = 1 << 28, //
0x10000000 Invert a transfer cfTest = 1 << 29, // 0x20000000 See if the command exists
cfNoReturn = 1 << 31, // 0x80000000 No return mediacmd is required

Definition at line 1434 of file vvwif.cpp.

uint32_t __stdcall vvwGetFreeStorage (VWIFOPAQUE vvwChannel)

Returns the amount of available storage for recording in megabytes.

Definition at line 2803 of file vvwif.cpp.

uint32_t __stdcall vvwGetFreeTime (VWIFOPAQUE vvwChannel)

Returns the remaining number of frames of storage available at current compression rate.

Definition at line 2793 of file vvwif.cpp.

**uint32_t __stdcall vvwGetLastChangeMs (VWIFOPAQUE vvwChannel, uint32_t
IClipSpace, uint32_t * INumClips)**

Returns the millisecond time of the last change in the current mode (clip or vtr).

Definition at line 2135 of file vvwif.cpp.

uint32_t __stdcall vvwGetLastChangeXferMs (VWIFOPAQUE vvwChannel)

Returns the millisecond time of the last change in the transfer queue

Definition at line 2120 of file vvwif.cpp.

uint32_t __stdcall vvwGetLastMs (VWIFOPAQUE vvwChannel)

Returns the millisecond time the last status occurred (time of the last vertical blank).

Definition at line 1476 of file vvwif.cpp.

**uint32_t __stdcall vvwGetLicenseOptions (VWIFOPAQUE vvwChannel, uint32_t *
pdwOptions, uint32_t * pdwLicenseValid)**

Definition at line 3148 of file vvwif.cpp.

uint32_t __stdcall vvwGetMaxChannels (void)

Returns the maximum number of channels available for control. Channels start at 0 and end at max channels – 1. This return is one greater than the largest value available for SetCurChannel(), GetCurChannel() and the IChannel parameter for the DLL interface.

Get the maximum channels available

Definition at line 689 of file vvwif.cpp.

uint32_t __stdcall vvwGetMaxInternalChannels (void)

Returns the maximum number of internal channels available for control. Channels start at 0 and end at 62 – 1. This return is one greater than the largest value available for SetCurChannel(), GetCurChannel() and the lChannel parameter for the DLL interface.

Get the maximum channels available

Definition at line 702 of file vvwif.cpp.

uint32_t __stdcall vvwGetMetaData ([VWIFOPAQUE](#) vvwChannel, char * sz8CharClipName, char * sz260CharFileName, uint32_t vvwInfoRequest, char * szValue)

Returns the meta data from szClip. Returns 0 if successful, else an error code.

Definition at line 1941 of file vvwif.cpp.

char* __stdcall vvwGetMRVersion ()

Returns the version string of the MediaReactor subsystem.

Definition at line 2846 of file vvwif.cpp.

char* __stdcall vvwGetNextClip ([VWIFOPAQUE](#) vvwChannel, char * sz8CharLastClipCurClip)

Clip Mode Only. Returns the next clip identifier. To get the first clip, szLastClip should be an empty string. Once the last clip available has been returned, GetNextClip will return an error or NULL for unix/dll access. Please note: For unix/dll access, the sz8CharLastClipCurClip memory area is used for the new clip. The previous clip name is therefore lost and the memory is not allocated by the vvw. Returns 0 if successful, else an error code.

Definition at line 1834 of file vvwif.cpp.

uint32_t __stdcall vvwGetNextClipEx ([VWIFOPAQUE](#) vvwChannel, char * sz8CharClipName, uint32_t * lCreation, uint32_t * lLastModification, uint32_t * lFileSize, uint32_t * lDiskFragments)

Returns the extended information from szClip. The information is located in lStart, lEnd, lVidEdit, lAudEdit and szFileName as time of creation, last modified date, the file size, and the number of fragments in the file respectively.

uint32_t __stdcall vvwGetNumberOfBackUps ([VWIFOPAQUE](#) vvwChannel, uint32_t * pdwBackUps, uint32_t dwClipMode)

Definition at line 3113 of file vvwif.cpp.

uint32_t __stdcall vvwGetNumberOfErrors (uint32_t * plErrors)

Get the number of current errors

Definition at line 2944 of file vvwif.cpp.

uint32_t __stdcall vvwGetPiconName (char * szFileName)

Get the name of a file from the media file's name

Definition at line 3007 of file vvwif.cpp.

uint32_t __stdcall vvwGetPosition ([VWIFOPAQUE](#) vvwChannel)

Returns the current position if the cfUsePosition flag is set, otherwise invalid.

Definition at line 1462 of file vvwif.cpp.

uint32_t __stdcall vvwGetSpeed ([VWIFOPAQUE](#) vvwChannel)

Returns the current VVW speed if the cfUseSpeed flag is set, otherwise pause or full play speed. VVW speeds are based on 65520 as the play speed. To translate to decimal number where 1.0 represents play, use the following formula: $D1Speed = ((double)VWWSpeed / 65520.0)$ For percentages, where 100.0 represents play speed, use the following formula: $Dpercent = (((double)VWWSpeed * 100.0) / 65520.0) = ((double)VWWSpeed / 655.2)$ XML: See <MediaCmd> root element, <Speed> sub-element

Typical VVW speeds (note speeds are linear): Pause 0% 0 Play 100% 65520 Half Play 50% 32760 Rev Play -100% -65520 Rev 2 x Play-200% 131040 10 x Play 1000% 655200 Max Play 90000% 5896800 Max Rev -90000%-5896800

Definition at line 1447 of file vvwif.cpp.

uint32_t __stdcall vvwGetStart ([VWIFOPAQUE](#) vvwChannel)

Returns the current start or in point if the cfUseStart flag is set.

Definition at line 1490 of file vvwif.cpp.

uint32_t __stdcall vvwGetState ([VWIFOPAQUE](#) vvwChannel)

Returns the current state

ctStop 0 // Stop all action ctPause 1 // Pause, Seek ctPlay 2 // Play at specified speed (includes pause) ctRecord 3 // Record at specified speed ctRecStop 4 // Stop ready for recording ctEject 5 // Eject the current media ctError 17 // An error has occurred ctAbort 19 // Abort any queued commands

Definition at line 1415 of file vvwif.cpp.

uint32_t __stdcall vvwGetSuperImpose ([VWIFOPAQUE](#) vvwChannel)

Get the status of the super imposed time code overlay

Definition at line 2889 of file vvwif.cpp.

uint32_t __stdcall vvwGetTCSource ([VWIFOPAQUE](#) vvwChannel)

Calls ValueXXX with gsTcSource (VITC, LTC, Control, Clip). [GS_TCSOURCE_LTC](#), [GS_TCSOURCE_VITC](#), [GS_TCSOURCE_CTL](#) or [GS_TCSOURCE_CLIP](#)

Definition at line 2465 of file vvwif.cpp.

uint32_t __stdcall vvwGetTCType ([VWIFOPAQUE](#) vvwChannel)

Calls ValueXXX with gsTcType (drop frame, non drop frame, pal). [TC2_TCTYPE_FILM](#), [TC2_TCTYPE_NDF](#), [TC2_TCTYPE_DF](#), [TC2_TCTYPE_PAL](#), [TC2_TCTYPE_50](#), [TC2_TCTYPE_5994](#), [TC2_TCTYPE_60](#), [TC2_TCTYPE_NTSCFILM](#), [TC2_TCTYPE_2398](#), [TC2_TCTYPE_100](#)

Definition at line 2453 of file vvwif.cpp.

uint32_t __stdcall vvwGetTotalStorage ([VWIFOPAQUE](#) vvwChannel)

Returns the total storage connected in megabytes.

Definition at line 2798 of file vvwif.cpp.

uint32_t __stdcall vvwGetTotalTime ([VWIFOPAQUE](#) vvwChannel)

Returns the total number of frames of storage available at current compression rate if the storage space was empty.

Definition at line 2788 of file vvwif.cpp.

uint32_t __stdcall vvwGetVideoGenlock ([VWIFOPAQUE](#) vvwChannel)

Turn the house/reference lock on or off

Definition at line 2751 of file vvwif.cpp.

uint32_t __stdcall vvwGetVideoGenlockSource ([VWIFOPAQUE](#) vvwChannel)

Set the genlock source to input or external reference

Definition at line 2763 of file vvwif.cpp.

uint32_t __stdcall vvwGetVideoInput ([VWIFOPAQUE](#) vvwChannel)

Get the current video input. [GS_VIDSELECT_COMPOSITE](#),
[GS_VIDSELECT_COMPOSITE_2](#), [GS_VIDSELECT_SVIDEO](#),
[GS_VIDSELECT_COMPONENT_YUV](#), [GS_VIDSELECT_COMPONENT_YUV_M2](#),
[GS_VIDSELECT_COMPONENT_YUV_SMPTE](#), [GS_VIDSELECT_COMPONENT_RGB](#),
[GS_VIDSELECT_D1_SERIAL](#), [GS_VIDSELECT_D1_PARALLEL](#),
[GS_VIDSELECT_SDTI](#), [GS_VIDSELECT_NONE](#)

Definition at line 2642 of file vvwif.cpp.

uint32_t __stdcall vvwGetVideoInputChroma ([VWIFOPAQUE](#) vvwChannel)

Get the current video input's 'Chroma' TBC setting.

Definition at line 2697 of file vvwif.cpp.

uint32_t __stdcall vvwGetVideoInputHue ([VWIFOPAQUE](#) vvwChannel)

Get the current video input's 'Hue' TBC setting.

Definition at line 2687 of file vvwif.cpp.

uint32_t __stdcall vvwGetVideoInputSetup ([VWIFOPAQUE](#) vvwChannel)

Get the current video input's 'Setup' TBC setting.

Definition at line 2667 of file vvwif.cpp.

uint32_t __stdcall vvwGetVideoInputVideo ([VWIFOPAQUE](#) vvwChannel)

Get the current video input's 'Video' TBC setting.

Definition at line 2677 of file vvwif.cpp.

uint32_t __stdcall vvwGetVideoOutput ([VWIFOPAQUE](#) vvwChannel)

Get the current video output. See [Get/SetVideoInput](#) for settings.

[GS_VIDSELECT_COMPOSITE](#), [GS_VIDSELECT_COMPOSITE_2](#),
[GS_VIDSELECT_SVIDEO](#), [GS_VIDSELECT_COMPONENT_YUV](#),
[GS_VIDSELECT_COMPONENT_YUV_M2](#),
[GS_VIDSELECT_COMPONENT_YUV_SMPTE](#), [GS_VIDSELECT_COMPONENT_RGB](#),
[GS_VIDSELECT_D1_SERIAL](#), [GS_VIDSELECT_D1_PARALLEL](#),
[GS_VIDSELECT_SDTI](#), [GS_VIDSELECT_NONE](#)

Definition at line 2654 of file vvwif.cpp.

uint32_t __stdcall vvwGetVideoTBCChroma ([VWIFOPAQUE](#) vvwChannel)

Get the current global TBC's 'Chroma' setting.

Definition at line 2739 of file vvwif.cpp.

uint32_t __stdcall vvwGetVideoTBCHue ([VWIFOPAQUE](#) vvwChannel)

Get the current global TBC's 'Hue' setting.

Definition at line 2729 of file vvwif.cpp.

uint32_t __stdcall vvwGetVideoTBCSetup ([VWIFOPAQUE](#) vvwChannel)

Get the current global TBC's 'Setup' setting.

Definition at line 2709 of file vvwif.cpp.

uint32_t __stdcall vvwGetVideoTBCVideo ([VWIFOPAQUE](#) vvwChannel)

Get the current global TBC's 'Video' setting.

Definition at line 2719 of file vvwif.cpp.

char* __stdcall vvwGetVWVType ([VWIFOPAQUE](#) vvwChannel)

Returns the type string of the VVW channel.

Definition at line 2868 of file vvwif.cpp.

char* __stdcall vvwGetVWVVersion ()

Returns the version string of the VVW subsystem.

Definition at line 2824 of file vvwif.cpp.

uint32_t __stdcall vvwHandleToChannel (VWVHANDLE hVvw)

Convert a 0, 64, 65536 to 0..3 Mostly internal Undocumented

Definition at line 620 of file vvwif.cpp.

uint32_t __stdcall vvwInsert ([VWIFOPAQUE](#) vvwChannel, char * szClipName, char * szFileName, uint32_t IPosition, uint32_t IStart, uint32_t IEnd, uint32_t IVidEdit, uint32_t IAudEdit, uint32_t IInfEdit, int fRipple)

Definition at line 2154 of file vvwif.cpp.

uint32_t __stdcall vvwIsOpen (void)

Has the vvw sub system been open

Definition at line 677 of file vvwif.cpp.

uint32_t __stdcall vvwLoadClip ([VWIFOPAQUE](#) vvwChannel, char * sz8CharClipName, uint32_t IStartFrame)

Clip Mode Only. Load a clip into the channel and display the IStartFrame. Returns 0 if successful, else an error code.

Load a clip

Definition at line 898 of file vvwif.cpp.

uint32_t __stdcall vvwMediaCmd ([VWIFOPAQUE](#) vvwChannel, void * pMediaCmd)

MediaCmd direct access

Definition at line 1292 of file vvwif.cpp.

long __stdcall vvwOpeningChannel ()

Definition at line 444 of file vvwif.cpp.

uint32_t __stdcall vvwOpenNetworkChannel (char * szAddress, uint32_t dwPort, uint32_t dwChannel)

Private open for the direct source MediaCMD API access

Definition at line 452 of file vvwif.cpp.

uint32_t __stdcall vvwPause ([VWIFOPAQUE](#) vvwChannel)

Stop playback and display the current frame. Returns 0 if successful, else an error code.

Pause displaying current frame

Definition at line 1061 of file vvwif.cpp.

uint32_t __stdcall vvwPlay ([VWIFOPAQUE](#) vvwChannel)

Play at normal speed. Returns 0 if successful, else an error code.

Set the channel into play

Definition at line 807 of file vvwif.cpp.

uint32_t __stdcall vvwPlayAtMs ([VWIFOPAQUE](#) vvwChannel, uint32_t IMs)

Start playback at a specified MS

Definition at line 995 of file vvwif.cpp.

uint32_t __stdcall vvwPlayAtSpeed ([VWIFOPAQUE](#) vvwChannel, uint32_t IVVWSpeed, uint32_t IEnd)

Play at a particular VVW speed. VVW speeds use a base play speed of 65520. This means that play = 65520, reverse play = -65520, four times play = 262080, half play speed = 32760. Percentage play speeds may be converted to VVW speeds using the PercentageToVVWSpeed() function. For Speed calculations please see GetSpeed() below. Returns 0 if successful, else an error code.

Play at a VVW speed specified

Definition at line 852 of file vvwif.cpp.

uint32_t __stdcall vvwPlayClip (VWIFOPAQUE vvwChannel, char * sz8CharClipName, int fDeferred)

Clip Mode Only. Play the entire clip specified by clip name. If the deferred flag is true, clip playback will only occur once the currently playing clip has finished. If there is no currently playing clip, playback will occur immediately. Returns 0 if successful, else an error code.

Play a clip

Definition at line 945 of file vvwif.cpp.

uint32_t __stdcall vvwPlayClipFromTo (VWIFOPAQUE vvwChannel, char * sz8CharClipName, uint32_t IFrom, uint32_t ITo, int fDeferred)

Clip Mode Only. Play the specified portion of the clip specified by clip name. If the deferred flag is true, clip playback will only occur once the currently playing clip has finished. If there is no clip currently playing, playback will occur immediately. Returns 0 if successful, else an error code.

Play a clip from a frame to a frame

Definition at line 963 of file vvwif.cpp.

uint32_t __stdcall vvwPlayFromTo (VWIFOPAQUE vvwChannel, uint32_t IFrom, uint32_t ITo, int fDeferred, int fLoop)

Play from a frame to another frame. As with editing systems, the 'from' point is included and will be displayed but the to point is NOT included and will not be displayed. This means that the last frame displayed will be IFrom - 1. The deferred flag allows

PlayFromTos to be stacked so that they will play back to back. The deferred flag in the status return should be false before another deferred command is added. Returns 0 if successful, else an error code.

Play from a frame to a frame

Definition at line 869 of file vvwif.cpp.

uint32_t __stdcall vvwPlayOffsetAt (VWIFOPAQUE vvwChannel, uint32_t dwPosition, long IOffset, uint32_t dwMS)

Set the channel into play jumping a amount of frames at a time for no offset, set IOffset to 0. For no position set it to -1. For now start time, set dwMS to -1.

Set the channel into play jumping a amount of frames

vwWaitForState(vvwChannel, ctPlay);

Definition at line 822 of file vvwif.cpp.

uint32_t __stdcall vvwRecord (VWIFOPAQUE vvwChannel)

Start the channel recording. In clip mode a default clip name will be used with a duration set to infinity. The record will stop on any transport command or at the point that the disk is full. Returns 0 if successful, else an error code.

Crash record

Definition at line 1125 of file vvwif.cpp.

uint32_t __stdcall vvwRecordAtMs (VWIFOPAQUE vvwChannel, uint32_t I Ms, uint32_t IStart, uint32_t IEnd)

Start Recording at a specified MS

Definition at line 1012 of file vvwif.cpp.

uint32_t __stdcall vvwRecordFromTo (VWIFOPAQUE vvwChannel, uint32_t IFrom, uint32_t ITo)

Record from a frame value to a frame value. As with editing systems, the 'from' point is included and will be recorded but the to point is NOT included and will not be recorded. This means that the last frame recorded will be IFrom - 1. Returns 0 if successful, else an error code.

Record from one frame to another

Definition at line 1138 of file vvwif.cpp.

uint32_t __stdcall vvwRecordStop (VWIFOPAQUE vvwChannel, char * sz8CharClipName, uint32_t IDuration)

Clip Mode Only. Set the clip name and length of time to record in frames. The record will not actually start until Record() is called. If the IDuration is set to -1 the record will continue until Stop() is called or the channel runs out of space. Returns 0 if successful, else an error code.

Record stop - prepare a record (clip mode only)

Definition at line 1164 of file vvwif.cpp.

uint32_t __stdcall vvwRecordStopFileName (VWIFOPAQUE vvwChannel, char * sz8CharClipName, char * sz256FileName, uint32_t IDuration)

Clip Mode Only. Set the clip name filename and length of time to record in frames. The record will not actually start until Record() is called. If the IDuration is set to -1 the record will continue until Stop() is called or the channel runs out of space. Returns 0 if successful, else an error code.

Record stop - prepare a record (clip mode only)

Definition at line 1189 of file vvwif.cpp.

uint32_t __stdcall vvwReleaseChannels (void)

Release memory allocated to channels

Definition at line 655 of file vvwif.cpp.

uint32_t __stdcall vvwSeek (VWIFOPAQUE vvwChannel, uint32_t IFrame)

Seek to a particular frame and display it to the user. This call will return before the seek is complete. Once the Position return in the status reaches the IFrame, the seek is complete. Returns 0 if successful, else an error code.

Seek to a frame

Definition at line 1074 of file vvwif.cpp.

uint32_t __stdcall vvwSeekRelative (VWIFOPAQUE vvwChannel, uint32_t IFrameOffset)

Seek a certain number of frames from the current position. Positive offsets imply forward

direction, negative offset imply reverse.
Seek to an offset from the current position
Definition at line 1097 of file vvwif.cpp.

uint32_t __stdcall vvwSetAudioInput ([VWIFOPAQUE](#) vvwChannel, uint32_t ISetting)

ADD FUNCTION lAudIn Set the current audio input.
[GS_AUDSELECT_UNBALANCED_10](#) [GS_AUDSELECT_UNBALANCED_4](#)
[GS_AUDSELECT_BALANCED_10](#) [GS_AUDSELECT_BALANCED_4](#)
[GS_AUDSELECT_SPDIF](#) [GS_AUDSELECT_AES_EBU](#) [GS_AUDSELECT_EMBEDDED](#)

Definition at line 2577 of file vvwif.cpp.

uint32_t __stdcall vvwSetAudioInputLevel ([VWIFOPAQUE](#) vvwChannel, uint32_t ISetting)

Get the current audio input level
Definition at line 2589 of file vvwif.cpp.

uint32_t __stdcall vvwSetAudioOutput ([VWIFOPAQUE](#) vvwChannel, uint32_t ISetting)

Set the current audio Output – See Get/SetAudioInput
[GS_AUDSELECT_UNBALANCED_10](#) [GS_AUDSELECT_UNBALANCED_4](#)
[GS_AUDSELECT_BALANCED_10](#) [GS_AUDSELECT_BALANCED_4](#)
[GS_AUDSELECT_SPDIF](#) [GS_AUDSELECT_AES_EBU](#) [GS_AUDSELECT_EMBEDDED](#)

Definition at line 2601 of file vvwif.cpp.

uint32_t __stdcall vvwSetAudioOutputLevel ([VWIFOPAQUE](#) vvwChannel, uint32_t ISetting)

Get the current audio output level.
Definition at line 2613 of file vvwif.cpp.

uint32_t __stdcall vvwSetAutoMode ([VWIFOPAQUE](#) vvwChannel, uint32_t ISetting)

Calls ValueXXX with gsAutoMode. Required for play lists, deferred commands and auto edit commands on VTRs.
Definition at line 2482 of file vvwif.cpp.

uint32_t __stdcall vvwSetBackUpNumber ([VWIFOPAQUE](#) vvwChannel, uint32_t dwBackUp, uint32_t dwClipMode)

Definition at line 3132 of file vvwif.cpp.

uint32_t __stdcall vvwSetClipMode ([VWIFOPAQUE](#) vvwChannel, uint32_t ISetting)

Calls ValueXXX with gsClipMode. If equal to 1 then the channel is in clip mode, if 0 the channel is in VTR mode.
Definition at line 2437 of file vvwif.cpp.

uint32_t __stdcall vvwSetCompressionRate ([VWIFOPAQUE](#) vvwChannel, uint32_t ISetting)

Set the current compression rate

Definition at line 2780 of file vvwif.cpp.

uint32_t __stdcall vvwSetErrorLog (uint32_t *ISetting*)

Set the error log pointer to the message you want

Definition at line 2916 of file vvwif.cpp.

uint32_t __stdcall vvwSetMetaData ([VWIFOPAQUE](#) vvwChannel, char * sz8CharClipName, uint32_t vvwInfoRequest, uint32_t nValue, char * szValue)

Sets the meta data for szClip. Returns 0 if successful, else an error code.

Definition at line 1904 of file vvwif.cpp.

uint32_t __stdcall vvwSetRecordPresets ([VWIFOPAQUE](#) vvwChannel, uint32_t *IvidEdit*, uint32_t *IAudEdit*, uint32_t *IInfEdit*)

Set the channels to record. Using -1 values implies that the Preset should be set to all available channels. Record Presets will remain set until the user changes them. Returns 0 if successful, else an error code.

Set video/audio/info 'channels' to record (preset)

Definition at line 1238 of file vvwif.cpp.

uint32_t __stdcall vvwSetSuperImpose ([VWIFOPAQUE](#) vvwChannel, uint32_t *ISetting*)

Set the status of the super imposed time code overlay

Definition at line 2902 of file vvwif.cpp.

uint32_t __stdcall vvwSetTCSource ([VWIFOPAQUE](#) vvwChannel, uint32_t *ISetting*)

Calls ValueXXX with gsTcSource (VITC, LTC, Control, Clip). [GS_TCSOURCE_LTC](#), [GS_TCSOURCE_VITC](#), [GS_TCSOURCE_CTL](#) or [GS_TCSOURCE_CLIP](#)

Definition at line 2470 of file vvwif.cpp.

uint32_t __stdcall vvwSetTCType ([VWIFOPAQUE](#) vvwChannel, uint32_t *ISetting*)

Calls ValueXXX with gsTcType (drop frame, non drop frame, pal). [TC2_TCTYPE_FILM](#), [TC2_TCTYPE_NDF](#), [TC2_TCTYPE_DF](#), [TC2_TCTYPE_PAL](#), [TC2_TCTYPE_50](#), [TC2_TCTYPE_5994](#), [TC2_TCTYPE_60](#), [TC2_TCTYPE_NTSCFILM](#), [TC2_TCTYPE_2398](#), [TC2_TCTYPE_100](#)

Definition at line 2458 of file vvwif.cpp.

[BOOL](#) __stdcall vvwSetUseAbsoluteTC ([BOOL](#) *bUseAbsoluteTC*)

Setting this will cause the cfUseFrameCount Flag to be set on transport commands This will cause the interface to ignore LTC/VITC offsets and use absolute Timcode values. Mostly for ease if use in TC Space when still running as a LTC / VITC timecode source.

Definition at line 770 of file vvwif.cpp.

uint32_t __stdcall vvwSetVideoGenlock ([VWIFOPAQUE](#) vvwChannel, uint32_t *ISetting*)

Turn the house/reference lock on or off

Definition at line 2756 of file vvwif.cpp.

uint32_t __stdcall vvwSetVideoGenlockSource (VWIFOPAQUE vvwChannel, uint32_t ISetting)

Set the genlock source to input or external reference

Definition at line 2768 of file vvwif.cpp.

uint32_t __stdcall vvwSetVideoInput (VWIFOPAQUE vvwChannel, uint32_t ISetting)

Set the current video input. [GS_VIDSELECT_COMPOSITE](#),
[GS_VIDSELECT_COMPOSITE_2](#), [GS_VIDSELECT_SVIDEO](#),
[GS_VIDSELECT_COMPONENT_YUV](#), [GS_VIDSELECT_COMPONENT_YUV_M2](#),
[GS_VIDSELECT_COMPONENT_YUV_SMPTE](#), [GS_VIDSELECT_COMPONENT_RGB](#),
[GS_VIDSELECT_D1_SERIAL](#), [GS_VIDSELECT_D1_PARALLEL](#),
[GS_VIDSELECT_SDTI](#), [GS_VIDSELECT_NONE](#)

Definition at line 2647 of file vvwif.cpp.

uint32_t __stdcall vvwSetVideoInputChroma (VWIFOPAQUE vvwChannel, uint32_t ISetting)

Set the current video input's 'Chroma' TBC setting.

Definition at line 2702 of file vvwif.cpp.

uint32_t __stdcall vvwSetVideoInputHue (VWIFOPAQUE vvwChannel, uint32_t ISetting)

Set the current video input's 'Hue' TBC setting.

Definition at line 2692 of file vvwif.cpp.

uint32_t __stdcall vvwSetVideoInputSetup (VWIFOPAQUE vvwChannel, uint32_t ISetting)

Set the current video input's 'Setup' TBC setting.

Definition at line 2672 of file vvwif.cpp.

uint32_t __stdcall vvwSetVideoInputVideo (VWIFOPAQUE vvwChannel, uint32_t ISetting)

Set the current video input's 'Video' TBC setting.

Definition at line 2682 of file vvwif.cpp.

uint32_t __stdcall vvwSetVideoOutput (VWIFOPAQUE vvwChannel, uint32_t ISetting)

Set the current video output. See [Get/SetVideoInput](#) for settings.
[GS_VIDSELECT_COMPOSITE](#), [GS_VIDSELECT_COMPOSITE_2](#),
[GS_VIDSELECT_SVIDEO](#), [GS_VIDSELECT_COMPONENT_YUV](#),
[GS_VIDSELECT_COMPONENT_YUV_M2](#),
[GS_VIDSELECT_COMPONENT_YUV_SMPTE](#), [GS_VIDSELECT_COMPONENT_RGB](#),
[GS_VIDSELECT_D1_SERIAL](#), [GS_VIDSELECT_D1_PARALLEL](#),
[GS_VIDSELECT_SDTI](#), [GS_VIDSELECT_NONE](#)

Definition at line 2659 of file vvwif.cpp.

uint32_t __stdcall vvwSetVideoTBCChroma (VWIFOPAQUE vvwChannel, uint32_t ISetting)

Set the current global TBC's 'Chroma' setting.

Definition at line 2744 of file vvwif.cpp.

uint32_t __stdcall vvwSetVideoTBCHue (VWIFOPAQUE vvwChannel, uint32_t ISetting)

Set the current global TBC's 'Hue' setting.

Definition at line 2734 of file vvwif.cpp.

uint32_t __stdcall vvwSetVideoTBCSetup (VWIFOPAQUE vvwChannel, uint32_t ISetting)

Set the current global TBC's 'Setup' setting.

Definition at line 2714 of file vvwif.cpp.

uint32_t __stdcall vvwSetVideoTBCVideo (VWIFOPAQUE vvwChannel, uint32_t ISetting)

Set the current global TBC's 'Video' setting.

Definition at line 2724 of file vvwif.cpp.

uint32_t __stdcall vvwShowConfigDialog (VWIFOPAQUE vvwChannel, uint32_t hWnd)

Show the configuration dialog box for the current channel. If the channel does not have a configuration dialog, this function will return an error. It is not available in Java or unix as the dialog only shows up on the local machine, and cannot be seen through the network.

uint32_t __stdcall vvwStop (VWIFOPAQUE vvwChannel)

Stop the output of the controlled channel and display the input video (not supported on all devices). On unsupported devices stop will be the same as a pause. Returns 0 if successful, else an error code.

Stop - stop play back a show input if supported, else its a pause

Definition at line 1112 of file vvwif.cpp.

uint32_t __stdcall vvwSwitchClip (VWIFOPAQUE vvwChannel, char * sz8CharClipName, uint32_t IPosition, BOOL bUseFrameCount)

Clip Mode Only. Switch to a different clip without changing play/pause mode. Returns 0 if successful, else an error code.

Switch to a clip

Definition at line 922 of file vvwif.cpp.

uint32_t __stdcall vvwTransfer (VWIFOPAQUE vvwChannel, uint32_t ITargetChannel, uint32_t IPosition, uint32_t IStart, uint32_t IEnd, uint32_t IVidEdit, uint32_t IAudEdit, uint32_t IInfEdit, char * szClipName, int fToTape)

Transfer media from one channel to another. Only supported by VTR channels. Currently only implemented for VTR to internal channels or internal channels to VTR channels. To record media from a VTR, the fToTape should be false, to record media onto a VTR the fToTape should be true. The start and end point are from the playback device. The edit will occur at the current timecode location on the recorder. Returns 0 if successful, else an error code.

Transfer media to or from an external VTR.

Definition at line 1328 of file vvwif.cpp.

uint32_t __stdcall vvwTrim (VWIFOPAQUE vvwChannel, uint32_t IPosition, uint32_t IStartOffset, uint32_t IEndOffset, uint32_t IVidEdit, uint32_t IAudEdit, uint32_t IInfEdit, int fRipple)

Definition at line 2302 of file vvwif.cpp.

uint32_t __stdcall vvwUpdateStatus (VWIFOPAQUE vvwChannel)

Retrieve the current status from the controlled device. The status is automatically updated by the interface, but this call ensures that the status is current when you are checking it. Returns 0 if successful, else an error code.

Definition at line 1376 of file vvwif.cpp.

uint32_t __stdcall vvwValueGet (VWIFOPAQUE vvwChannel, uint32_t IValueType, uint32_t * pIMin, uint32_t * pIMax)

Returns the current setting for a get/set value.

Definition at line 2362 of file vvwif.cpp.

uint32_t __stdcall vvwValueSet (VWIFOPAQUE vvwChannel, uint32_t IValueType, uint32_t ISetting)

Sets the get/set value to setting.

Definition at line 2384 of file vvwif.cpp.

uint32_t __stdcall vvwValueSet2 (VWIFOPAQUE vvwChannel, uint32_t IValueType, uint32_t ISetting, uint32_t IStart, uint32_t IEnd, uint32_t IVidChan, uint32_t IAudChan, uint32_t IInfChan)

Sets the get/set value to setting with extended parameters. Please set unused parameters to NULL.

Definition at line 2397 of file vvwif.cpp.

uint32_t __stdcall vvwValueSupported (VWIFOPAQUE vvwChannel, uint32_t IValueType)

Returns the supported attributes of a get/set value (gsClipMode, gsTcSource, etc) or -1 for not supported.

Definition at line 2344 of file vvwif.cpp.

uint32_t __stdcall vvwWaitForState (VWIFOPAQUE vvwChannel, uint32_t ctCmd)

Wait for the channel to reach a state (ctPlay, ctRecord, etc). Will wait up to 500 milliseconds

Parameters:

<i>vwChannel</i>	the mediacmd target being controlled (internal, external, network)
<i>ctCmd</i>	the command state we will wait for (ctPlay, ctRecord, ctPause, etc)

Returns:

if state is reached, return 0. If it timed out waiting, return -1
Definition at line 791 of file vvwif.cpp.

uint32_t __stdcall vvwXMLProjectCheckOpen (char * IpszProjectName)

Definition at line 3075 of file vvwif.cpp.

uint32_t __stdcall vvwXMLProjectOpen (char * *IpszProjectName*, int *bDeleteConflictingFiles*)

Definition at line 3092 of file vvwif.cpp.

uint32_t __stdcall vvwXMLProjectSave (char * *IpszProjectName*)

Definition at line 3048 of file vvwif.cpp.

E:/drastic/api/mediacmd/src/vvwtypes.h File Reference

```
#include "mediacmd.h"
```

Classes

4124struct [RECT16](#)
4125struct [FRAME_INFO](#)
4126struct [DRASTIC_CHANNEL](#)
4127struct [DCLIP](#)
4128struct [DFRAME](#)
4129struct [DPOSSIZENAME](#)
4130struct [VVWSYSTEM](#)
4131struct [VVWVIDEO](#)
4132struct [VVWAUDIO](#)
4133struct [DTDIRECT_WAVEHDR](#)
4134struct [vwvInflmageInfo](#)
4135struct [VVWINFO](#)

Defines

4136#define [DTVVW_PREVIEW](#) 0x80000000
4137#define [VWV_INVALID_FRAME](#) 0x80000001
No frame exists - is the same as TC_ILLEGAL - careful, only valid when tc cannot go negative, else it means -1.
4138#define [VWV_ABS_MAX_FRAME](#) 10692000
Absolute maximum timecode in a time code space (99:59:59:29 NDF) - Actual max is usually 23:59:59:x9 depending on type (see [tc2Maximum\(\)](#))
4139#define [_MAX_CLIP_NAME_SIZE](#) 9
The size of a clip name string (8 char plus NULL term) (1 for safety) - This is for odetics/louth compatibility.
4140#define [FRAMEINFO_DATA_EIA608](#) 1
Data is EIA-608B SD closed caption data.
4141#define [FRAMEINFO_DATA_EIA708](#) 2
Data is EIA-708 HD closed caption data.
4142#define [FRAMEINFO_DATA_RP215_KLV](#) 4
Data is RP-215 KLV data.
4143#define [FRAMEINFO_DATA_VBI_FRAME_ELEM](#) 8

Data is raw line with info per SMPTE 436 VBI Frame Element.

4144#define [FRAMEINFO_MAX_DATA_SIZE](#) 8192
Size of the arbData area.

4145#define [pFI_DataAreaPtr](#) dwVITCAux
Alternate name for dwVITCAux.

4146#define [dwFI_DataAreaSize](#) dwCCData
Alternate name for dwCCData.

4147#define [DFRAME_MAX_EXTRA_DATA_SIZE](#) 1024

4148#define [DFRAME_DATAAREA_RPDPXHEADER](#) "DraStiCteCh-215"
This indicates that fi.dwVITCAux points to a data area and dwCCData contains its size / See below define DFRAME_TYPE_FI_PTR_DATA 0x00008000.

4149#define [DFRAME_DATAAREA_RPDPXHEADER_SIZE](#) 16

4150#define [COPYFRAMEINFO](#)(_ptrFiDst, _ptrFiSrc)
Copy the frame info area, remove pointers.

4151#define [DECLARE_DRASTIC_CHANNEL](#)(_x_)
Declare and initialize a [DRASTIC_CHANNEL](#).

4152#define [INIT_DRASTIC_CHANNEL](#)(_x_)
Initialize a [DRASTIC_CHANNEL](#).

4153#define [INIT_PDRASTIC_CHANNEL](#)(_x_)
Initialize a memory area allocated as a [DRASTIC_CHANNEL](#).

4154#define [DFRAME_TYPE_RECORD](#) 0x00000001
This frame was recorded into (normally by AvHAL) see [DFRAME::dwType](#).

4155#define [DFRAME_TYPE_PLAY](#) 0x00000002
This frame should be played out (normally by AvHAL) see [DFRAME::dwType](#).

4156#define [DFRAME_TYPE_PAUSE](#) (0x00000004 | DFRAME_TYPE_PLAY)
This frame was acquired while in pause and is most likely a seek (normally by AvHAL) see [DFRAME::dwType](#).

4157#define [DFRAME_TYPE_UNCOMPRESSED](#) 0x00000008
This frame has already been decompressed. If avhal is in a compressed mode, by pass the decompression. see [DFRAME::dwType](#).

4158#define [DFRAME_TYPE_UNCTYPE_MASK](#) 0x000000F0
Uncompressed type mask, if DFRAME_TYPE_UNCOMPRESSED is true see [DFRAME::dwType](#).

4159#define [DFRAME_TYPE_UNCTYPE_YCBCR8](#) 0x00000000
Uncompressed YCbCr 8 bit UYVY/yuv2 see [DFRAME::dwType](#).

4160#define [DFRAME_TYPE_UNCTYPE_YUY2](#) 0x00000010
Uncompressed YUY2 8 bit see [DFRAME::dwType](#).

4161#define [DFRAME_TYPE_UNCTYPE_V210](#) 0x00000020
Uncompressed V210 10 bit YCbCr see [DFRAME::dwType](#).

4162#define [DFRAME_TYPE_UNCTYPE_BGR](#) 0x00000030
Windows BGR 8 bit see [DFRAME::dwType](#).

4163#define [DFRAME_TYPE_UNCTYPE_BGRA](#) 0x00000040
Windows BGRA 8 bit see [DFRAME::dwType](#).

4164#define [DFRAME_TYPE_UNCTYPE_DPX10](#) 0x00000050
DPX 10 bit RGB see [DFRAME::dwType](#).

4165#define [DFRAME_TYPE_UNCTYPE_ARRID21](#) 0x00000060

HD-SDI Arri D21 bayer double buffer.

4166#define [DFRAME_TYPE_UNCTYPE_ARRIALEXA](#) 0x00000070
HD-SDI Arri Alexa bayer double buffer.

4167#define [DFRAME_TYPE_RIGHT1ST_INVERT_VERT](#) 0x00000100
3D Stereo, first frame/right eye inverted vertically

4168#define [DFRAME_TYPE_RIGHT1ST_INVERT_HORIZ](#) 0x00000200
3D Stereo, first frame/right eye inverted horizontally

4169#define [DFRAME_TYPE_LEFT2ND_INVERT_VERT](#) 0x00000400
3D Stereo, second frame/left eye inverted vertically

4170#define [DFRAME_TYPE_LEFT2ND_INVERT_HORIZ](#) 0x00000800
3D Stereo, second frame/left eye inverted horizontally

4171#define [DFRAME_TYPE_NOTPHYSHEAP](#) 0x00001000
This indicates PhysHeap did NOT allocate the dframe (not pheap or local heap) see [DFRAME::dwType](#).

4172#define [DFRAME_TYPE_INCLUDES_HEADER](#) 0x00002000
This indicates the header is includes with the video data (e.g. Ari from Alexa) see [DFRAME::dwType](#).

4173#define [DFRAME_TYPE_FI_PTR_DATA](#) 0x00008000
This indicates that FRAME_INFO::dwVITCAux points to a data area and [FRAME_INFO::dwCCData](#) contains its size see them for more info see [DFRAME::dwType](#).

4174#define [DFRAME_TYPE_AUDIO](#) 0x00010000
This frame contains audio data see [DFRAME::dwType](#).

4175#define [DFRAME_TYPE_VIDEO](#) 0x00020000
This frame contains video data see [DFRAME::dwType](#).

4176#define [DFRAME_TYPE_LIVE_VIDEO](#) 0x00040000
This frame contains video data from a differnet channels input source see [DFRAME::dwType](#).

4177#define [DFRAME_PROGRESSIVE](#) 0x00100000
The contents of the frame are progressive (as opposed to interlaced) see [DFRAME::dwType](#).

4178#define [DFRAME_FIELD_INVERT](#) 0x00200000
The fields in the frame are inverted (jaggies) see [DFRAME::dwType](#).

4179#define [DFRAME_TIME_INVERT](#) 0x00400000
The fields in the frame are temporally inverts (jumps back and forth) see [DFRAME::dwType](#).

4180#define [DFRAME_ORIENTATION_INVERT](#) 0x00800000
The contents of the frame are inverted.

4181#define [DFRAME_TYPE_EYES_ARE_FLIPPED](#) 0x01000000
AUDIO: The frame contains a large chunk of audio (allows for optimization in AvHAL) see [DFRAME::dwType](#).

4182#define [DFRAME_NEW_FORMAT](#) 0x08000000
This frame starts a new format that is different from the ones previous. Please get the new format and adjust before displaying see [DFRAME::dwType](#).

4183#define [DFRAME_TYPE_KEYFRAME](#) 0x10000000
This frame is independant of other frames for decode see [DFRAME::dwType](#).

4184#define [DFRAME_TYPE_KEYFRAME_I](#) 0x10000000
This frame is independant of other frames for decode (an MPEG I Frame) see [DFRAME::dwType](#).

4185#define [DFRAME_TYPE_KEYFRAME_B](#) 0x20000000
This frame requires more than one frame to decode (for MPEG a B Frame) see [DFRAME::dwType](#).

4186#define [DFRAME_SKIP_FRAME](#) 0x40000000
This frame should be skipped (decoded, but not displayed) - Used to reach seek frame on a non key frame from key frame see [DFRAME::dwType](#).

4187#define [DFRAME_TYPE_KEYFRAME_P](#) 0x80000000
This frame requires previous keyframe(s) (for MPEG a P Frame) see [DFRAME::dwType](#).

4188#define [DTVHDR_DONE](#) 0x00000001
internal to AvHAL Set by the device driver to indicate it is finished with the data buffer and is returning the buffer to the client. see [DFRAME::dwFlags](#)

4189#define [DTVHDR_PREPARED](#) 0x00000002
internal to AvHAL Indicates whether or not the buffer has been prepared for use. See [DVM_STREAM_PREPAREHEADER](#). see [DFRAME::dwFlags](#)

4190#define [DTVHDR_INQUEUE](#) 0x00000004
internal to AvHAL Set by the driver to indicate the buffer is in the driver's buffer queue. see [DFRAME::dwFlags](#)

4191#define [DTVHDR_KEYFRAME](#) 0x00000008
internal to AvHAL Set by the device driver to indicate a key frame. see [DFRAME::dwFlags](#)

4192#define [DTWHDR_DONE](#) 0x00000001
Set by the device driver to indicate that it is finished with the buffer and is returning it to the application. see [DFRAME::dwFlags](#).

4193#define [DTWHDR_PREPARED](#) 0x00000002
internal to AvHAL Set by Windows to indicate that the buffer has been prepared with the [waveInPrepareHeader](#) or [waveOutPrepareHeader](#) function. see [DFRAME::dwFlags](#)

4194#define [DTWHDR_BEGINLOOP](#) 0x00000004
internal to AvHAL This buffer is the first buffer in a loop. This flag is used only with output buffers. see [DFRAME::dwFlags](#)

4195#define [DTWHDR_ENDLOOP](#) 0x00000008
internal to AvHAL This buffer is the last buffer in a loop. This flag is used only with output buffers. see [DFRAME::dwFlags](#)

4196#define [DTWHDR_INQUEUE](#) 0x00000010
internal to AvHAL Set by Windows to indicate that the buffer is queued for playback. see [DFRAME::dwFlags](#)

4197#define [PDFRAMEFLAGS_CLIPSTILL](#) 0x00000001
This is a single frame clip (implies progressive) that needs to be play for the duration of the clip see [DFRAME::dwFlags](#).

4198#define [PDFRAMEFLAGS_CLIPSTART](#) 0x00000002
This is the first frame of a new clip see [DFRAME::dwFlags](#).

4199#define [PDFRAMEFLAGS_CLIPEND](#) 0x00000004
This is the last frame of the current clip see [DFRAME::dwFlags](#).

4200#define [PDFRAMEFLAGS_FIRSTFRAME](#) 0x00000002
Alias for [PDFRAMEFLAGS_CLIPSTART](#) see [DFRAME::dwFlags](#).

4201#define [PDFRAMEFLAGS_LASTFRAME](#) 0x00000004
Alias for [PDFRAMEFLAGS_CLIPEND](#) see [DFRAME::dwFlags](#).

4202#define [PDFRAMEFLAGS_IMPOSED_MARKED](#) 0x00000008

This Frame has been imposed if need be and watermark.

4203#define [PDFRAMEFLAGS_FIELD_MARK_MASK](#) 0x00030000
Interleaved frame had already been split and it the first field part mask.

4204#define [PDFRAMEFLAGS_FIELD_HASMARK](#) 0x00020000
If set, the first second bit is correct, if not then ignore even off bit.

4205#define [PDFRAMEFLAGS_FIELD_FIRST](#) 0x00000000
This is the first field, if the HASMARK is set.

4206#define [PDFRAMEFLAGS_FIELD_SECOND](#) 0x00010000
This is the second field, if the HASMARK is set.

4207#define [PDFRAMEFLAGS_FIELD_INLINE](#) 0x00040000
This is the second field, if the HASMARK is set.

4208#define [PDFRAMEFLAGS_FIELD_MARKFIRST](#) (0x00000000 |
[PDFRAMEFLAGS_FIELD_HASMARK](#))
Set as first field.

4209#define [PDFRAMEFLAGS_FIELD_MARKSECOND](#) (0x00010000 |
[PDFRAMEFLAGS_FIELD_HASMARK](#))
Set as second field.

4210#define [PDFRAMEFLAGS_FIELD_MARK_TWO_FIELDS](#)
([PDFRAMEFLAGS_FIELD_INLINE](#) | [PDFRAMEFLAGS_FIELD_HASMARK](#))
Set as two fields, one full field followed by the other ie not interlaced.

4211#define [DT_TOP_FIELD](#) 0x10000000
Used in dwFlags - dup above, should condense at some point.

4212#define [DT_BOTTOM_FIELD](#) 0x20000000
Used in dwFlags - dup above, should condense at some point.

4213#define [DT_FRAME](#) 0x30000000
Used in dwFlags - dup above, should condense at some point.

4214#define [SIZEOFDFRAME](#) (((sizeof([DFRAME](#)) >> 2) + 1) << 2)
The DWORD aligned size of a [DFRAME](#), used for more efficient memory allocations.

4215#define [DPOSSIZENAME_VIDEO_FRAME](#) 0x00000001
This is a video frame.

4216#define [DPOSSIZENAME_RECORDING](#) 0x00000004
Is this file type currently recording.

4217#define [DPOSSIZENAME_MONO_AUDIO_FRAME](#) 0x00000100
This is a mono audio chunk.

4218#define [DPOSSIZENAME_STEREO_AUDIO_FRAME](#) 0x00000200
This is a stereo audio chunk.

4219#define [DPOSSIZENAME_QUAD_AUDIO_FRAME](#) 0x00000400

4220#define [DPOSSIZENAME_4_1_AUDIO_FRAME](#) 0x00000800

4221#define [DPOSSIZENAME_5_1_AUDIO_FRAME](#) 0x00001000

4222#define [DPOSSIZENAME_7_1_AUDIO_FRAME](#) 0x00002000

4223#define [DPOSSIZENAME_9_1_AUDIO_FRAME](#) 0x00004000

4224#define [DPOSSIZENAME_FRAME_MASK](#) 0x0000FFFF

4225#define [DPOSSIZENAME_AUD_16_16_BIT](#) 0x00100000
This frame contains audio data see [DFRAME::dwType](#).

4226#define [DPOSSIZENAME_AUD_20_24_BIT](#) 0x00200000
20 bit audio in 24

4227#define [DPOSSIZENAME_AUD_24_24_BIT](#) 0x00400000

24 bit audio in 24

4228#define [DPOSSIZENAME_AUD_24_32_BIT](#) 0x00800000

24/32 bit audio in 32

4229#define [DPOSSIZENAME_AUD_32_32_BIT](#) 0x01000000

32/32 bit audio in 32

4230#define [DPOSSIZENAME_AUD_BIGENDIAN_BIT](#) 0x00080000

Audio is big endian, else little endian.

4231#define [VWXXX_RESERVED_SIZE](#) 256

The size of the reserved area (in DWORDs) within [VWSYSTEM](#), [VWVIDEO](#), [VWAUDIO](#) and [VWINFO](#).

4232#define [VWXXX_NAME_SIZE](#) 256

The size of the name area (in chars) within [VWSYSTEM](#), [VWVIDEO](#), [VWAUDIO](#) and [VWINFO](#).

4233#define [dstreamtypeVIDEO](#) ((DWORD)(unsigned char)('v') | ((DWORD)(unsigned char)('i') << 8) | ((DWORD)(unsigned char)('d') << 16) | ((DWORD)(unsigned char)('s') << 24))

4234#define [dstreamtypeAUDIO](#) ((DWORD)(unsigned char)('a') | ((DWORD)(unsigned char)('u') << 8) | ((DWORD)(unsigned char)('d') << 16) | ((DWORD)(unsigned char)('s') << 24))

4235#define [VW_IS_VWSYSTEM](#) 0x0000

The flag indicating the structure is a [VWSYSTEM](#) structure within a union.

4236#define [VWSYS_SET](#)(__pvvwsys_, __pvwvid_, __pvwaud_, _granularity, _frames)

Set a [VWSYSTEM](#) structure pointer from [VWVIDEO](#) pointer, [VWAUDIO](#) pointer, a granularity size and number of frames.

4237#define [VWSYS_SETVIDONLY](#)(__pvvwsys_, __pvwvid_, _granularity, _frames)

Set a [VWSYSTEM](#) structure pointer from [VWVIDEO](#) pointer, a granularity size and number of frames.

4238#define [VWSYS_SETAUDONLY](#)(__pvvwsys_, __pvwaud_, _granularity, _frames)

Set a [VWSYSTEM](#) structure pointer from [VWAUDIO](#) pointer, a granularity size and number of frames.

4239#define [VWSYS_CLR](#)(__pvvwsys_)

Clear a important 0 of a [VWVIDEO](#) structure pointer.

4240#define [VWXXX_SETSAMPLETOLENGTH](#)(__pvvw_, _length) { (__pvvw_)->dwLength = _length * (__pvvw_)->dwScale; }

This macro is incorrect.

4241#define [VWXXX_GETSAMPLEFROMLENGTH](#)(__pvvw_) ((__pvvw_)->dwLength / (__pvvw_)->dwScale)

This macro is incorrect.

4242#define [VW_IS_VWVIDEO](#) 0x0010

The flag indicating the structure is a [VWVIDEO](#) structure within a union.

4243#define [VWVIDEO_720P_YCBCR10](#) 3456

4244#define [VWVIDEO_2048_YCBCR10](#) 5504

4245#define [MASK_DRFLAGS_FIELD](#) 0x00000001

Field dominance MASK for [VWVIDEO::dwDrFlags](#).

4246#define [DRFLAGS_ZERO_FIELD_DOMINANT](#) 0x00000000

Zero (Second) Field Dominant in [VWVIDEO::dwDrFlags](#).

4247#define [DRFLAGS_FIRST_FIELD_DOMINANT](#) 0x00000001

First Field Dominant in [VWVIDEO::dwDrFlags](#).

4248#define [DRFLAGS_NOT_QUICKCLIP](#) 0x00000002

Preview to QuickclipXO.

```
4249#define MASK_DRFLAGS_KEYFRAME 0x00000010
KeyFrame MASK for VVWVIDEO::dwDrFlags.
4250#define DRFLAGS_HAS_KEYFRAMES 0x00000010
Stream has key frames, else all key frames VVWVIDEO::dwDrFlags.
4251#define MASK_DRFLAGS_VTYPE 0x00000700
Frame type mask (1=interlaced,2=progressive,4=segmentedframe)
4252#define DRFLAGS_VTYPE_INTERLACED 0x00000100
Interlaced video frames.
4253#define DRFLAGS_VTYPE_PROGRESSIVE 0x00000200
Progressive video frames.
4254#define DRFLAGS_VTYPE_SEGMENTEDFRAME 0x00000400
Segmented Frame video frames.
4255#define DRFLAGS_IS_COMPRESS 0x000001000
Is opening for a compression (0== decompression)
4256#define MASK_FCCMODIFIERS 0x00FF0000
Fourcc modifiers MASK for VVWVIDEO::dwDrFlags.
4257#define DRFLAGS_FCC_MJPG_DIGISUITE 0x00000000
Stream is DigiSuite MJPG see VVWVIDEO::dwDrFlags.
4258#define DRFLAGS_FCC_MJPG_DCx0 0x00010000
Stream is Miro DC50 MJPG see VVWVIDEO::dwDrFlags.
4259#define DRFLAGS_FCC_MJPG_DSEEDIT 0x00020000
Stream is DigiSuite Edit MJPG see VVWVIDEO::dwDrFlags.
4260#define DRFLAGS_FCC_MJPG_JPGDIB 0x00040000
Stream is MJPG MS-Dib variant see VVWVIDEO::dwDrFlags.
4261#define DRFLAGS_FCC_MJPG_JFIF 0x00080000
Stream is JFIF jpeg see VVWVIDEO::dwDrFlags.
4262#define DRFLAGS_FCC_USE_INTERN 0x00100000
Stream should use internal codecs see VVWVIDEO::dwDrFlags.
4263#define DRFLAGS_FCC_USE_QT 0x00400000
Stream should use quicktime codecs see VVWVIDEO::dwDrFlags.
4264#define DRFLAGS_FCC_USE_ICM 0x00800000
Stream should use windows icm/vfw codecs see VVWVIDEO::dwDrFlags.
4265#define MASK_CODECPRIATEDATA 0x0F000000
Private Data MASK see VVWVIDEO::dwDrFlags.
4266#define DRFLAGS_CODECPRIATEDATA_AVI 0x01000000
Private data format is AVI see VVWVIDEO::dwDrFlags.
4267#define DRFLAGS_CODECPRIATEDATA_MOV 0x02000000
Private data format is MOV see VVWVIDEO::dwDrFlags.
4268#define DRFLAGS_CODECPRIATEDATA_OMF 0x04000000
Private data format is OMF see VVWVIDEO::dwDrFlags.
4269#define MASK_PREVIEW 0xF0000000
MASK Room for current DTVVW_PREVIEW and more if nec see VVWVIDEO::dwDrFlags.
4270#define VVWVID_PITCHALIGN1(_pvwvid_) { (_pvwvid_)->biPitch = (((_pvwvid_)-
->biWidth * (_pvwvid_)->biBitCount+ 7) / 8); }
```

Set [VWVIDEO::biPitch](#) member to 1 byte alignment based on [VWVIDEO::biWidth](#) and [VWVIDEO::biBitCount](#).

```
4271#define VWVID_PITCHALIGN4(__pvwvid_) { (__pvwvid_)->biPitch = (((__pvwvid_)->biWidth * (__pvwvid_)->biBitCount+ 31) / 32) * 4; }
```

Set [VWVIDEO::biPitch](#) member to 4 byte (DWORD) alignment based on [VWVIDEO::biWidth](#) and [VWVIDEO::biBitCount](#).

```
4272#define VWVID_PITCHALIGN8(__pvwvid_) { (__pvwvid_)->biPitch = (((__pvwvid_)->biWidth * (__pvwvid_)->biBitCount+ 63) / 64) * 8; }
```

Set [VWVIDEO::biPitch](#) member to 8 byte alignment based on [VWVIDEO::biWidth](#) and [VWVIDEO::biBitCount](#).

```
4273#define VWVID_PITCHALIGN16(__pvwvid_) { (__pvwvid_)->biPitch = (((__pvwvid_)->biWidth * (__pvwvid_)->biBitCount+127) / 128) * 16; }
```

Set [VWVIDEO::biPitch](#) member to 16 byte alignment based on [VWVIDEO::biWidth](#) and [VWVIDEO::biBitCount](#).

```
4274#define VWVID_PITCHALIGNANY(__pvwvid_, align_1_4_8_16) { (__pvwvid_)->biPitch = (((__pvwvid_)->biWidth * (__pvwvid_)->biBitCount+((align_1_4_8_16 * 8) - 1)) / (align_1_4_8_16 * 8)) * align_1_4_8_16; }
```

Set [VWVIDEO::biPitch](#) member to alignment specified based on [VWVIDEO::biWidth](#) and [VWVIDEO::biBitCount](#).

```
4275#define VWVID_PITCHMODULO(__pvwvid_) ((__pvwvid_)->biPitch - ((__pvwvid_)->biWidth * (__pvwvid_)->biBitCount + 7) / 8)
```

Set pitch to 1-0 alignment ?

```
4276#define VWVID_SIZEIMAGE(__pvwvid_) { if(!(__pvwvid_)->biPitch) VWVID_PITCHALIGN1(__pvwvid_); (__pvwvid_)->biSizeImage = (__pvwvid_)->biPitch * abs((__pvwvid_)->biHeight); }
```

Set the [VWVIDEO::biSizeImage](#) based on [VWVIDEO::biWidth](#), [VWVIDEO::biHeight](#) and [VWVIDEO::biBitCount](#).

```
4277#define VWVID_SUGGESTEDBUFFERSIZE(__pvwvid_) { if(!(__pvwvid_)->biPitch) VWVID_PITCHALIGN1(__pvwvid_); (__pvwvid_)->dwSuggestedBufferSize = (__pvwvid_)->biPitch * (abs((__pvwvid_)->biHeight) + (abs((__pvwvid_)->biHeight) >> 2)); }
```

Set the [VWVIDEO::dwSuggestedBufferSize](#) based on [VWVIDEO::biWidth](#), [VWVIDEO::biHeight](#) and [VWVIDEO::biBitCount](#).

```
4278#define VWVID_RATESCALE_NTSC_AVI(__pvwvid_) { (__pvwvid_)->dwScale = 1001; (__pvwvid_)->dwRate = 30000; }
```

Set the [VWVIDEO::dwRate](#) and [VWVIDEO::dwScale](#) for 30000/1001 (NTSC)

```
4279#define VWVID_RATESCALE_NTSC(__pvwvid_) { (__pvwvid_)->dwScale = 100; (__pvwvid_)->dwRate = 2997; }
```

Set the [VWVIDEO::dwRate](#) and [VWVIDEO::dwScale](#) for 2997/100 (NTSC)

```
4280#define VWVID_RATESCALE_PAL(__pvwvid_) { (__pvwvid_)->dwScale = 1; (__pvwvid_)->dwRate = 25; }
```

Set the [VWVIDEO::dwRate](#) and [VWVIDEO::dwScale](#) for 25/1 (PAL)

```
4281#define VWVID_RATESCALE_FILM(__pvwvid_) { (__pvwvid_)->dwScale = 1; (__pvwvid_)->dwRate = 24; }
```

Set the [VWVIDEO::dwRate](#) and [VWVIDEO::dwScale](#) for 24/1 (FILM)

```
4282#define VWVID_SET(__pvwvid_, _bitcount, _width, _height, _pitch_align_1_4_8_16, _fccodec, _scale, _rate, _frames)
```

Basic setup of a [VWVIDEO](#) pointer from width, height, align, codec, scale, rate and length in frames.

4283#define [VWVID_SETCLR](#)(__pvwvid_, _bitcount, _width, _height, _pitch_align_1_4_8_16, _fccodec, _scale, _rate, _frames)
Clean and do basic setup of a [VWVIDEO](#) pointer from width, height, align, codec, scale, rate and length in frames.

4284#define [VW_IS_VWAUDIO](#) 0x0100
The flag indicating the structure is a [VWAUDIO](#) structure within a union.

4285#define [DTWAVE_FORMAT_PCM](#) 1
PCM Wave Type, see [fccDef.h](#) for other possible types.

4286#define [DTWAVE_FORMAT_EXTENSIBLE](#) 0xFFFE
4287#define [VWAUD_RECALC](#)(__pvwaud_)
4288#define [VWAUD_SET](#)(__pvwaud_, _formattag, _channels, _samplespersec, _bitspersample, _samples)
Set the [VWAUDIO](#) structure pointer from formattag, channels, samples per sec, bits per sample and number of samples.

4289#define [VWAUD_SETCLR](#)(__pvwaud_, _formattag, _channels, _samplespersec, _bitspersample, _samples)
Clean and set the [VWAUDIO](#) structure pointer from formattag, channels, samples per sec, bits per sample and number of samples.

4290#define [DTWAVEHDR_RIFF](#) 0x46464952
The canonical WAVE format starts with the RIFF header:

4291#define [DTWAVEHDR_WAV](#) 0x45564157
4292#define [DTWAVEHDR_FMT](#)
4293#define [DTWAVEHDR_DATA](#)
36 2 Size of the dwReserved area used. For PCM this will be zero. For other compressors, it may be anything < 256 DWORDs

4294#define [VW_IS_VWINFO](#) 0x1000
The flag indicating the structure is a [VWINFO](#) structure within a union.

4295#define [VWINF_INVALID](#) (-1)
Invalid setting, ignore it.

4296#define [VWINF_CURVETYPE_UNKNOWN](#) 0
dwCurveType

4297#define [VWINF_CURVETYPE_LINEAR](#) 1
4298#define [VWINF_CURVETYPE_LOG](#) 2
4299#define [VWINFOTAG_woVideoWidth](#) "Width"
XML tag name for width.

4300#define [VWINFODESC_woVideoWidth](#) "Width"
4301#define [VWINFOTAG_woVideoHeight](#) "Height"
XML tag name for height.

4302#define [VWINFODESC_woVideoHeight](#) "Height"
4303#define [VWINFOTAG_woVideoPlanes](#) "Planes"
XML tag name for planes.

4304#define [VWINFODESC_woVideoPlanes](#) "Planes"
4305#define [VWINFOTAG_woVideoBitCount](#) "BitCount"
XML tag name for bit count.

4306#define [VWINFODESC_woVideoBitCount](#) "BitCount"
4307#define [VWINFOTAG_woVideoCompression](#) "Compression"
XML tag name for compression (fourcc)

4308#define [VWINFODESC_woVideoCompression](#) "Compression"

4309#define [VVWINFOTAG_woVideoSizeImage](#) "SizeImage"
XML tag name for size of the image in unsigned chars.

4310#define [VVWINFODESC_woVideoSizeImage](#) "SizeImage"
4311#define [VVWINFOTAG_woVideoXPelsPerMeter](#) "XPelsPerMeter"
XML tag name for X pels per meter.

4312#define [VVWINFODESC_woVideoXPelsPerMeter](#) "XPelsPerMeter"
4313#define [VVWINFOTAG_woVideoYPelsPerMeter](#) "YPelsPerMeter"
XML tag name for Y pels per meter.

4314#define [VVWINFODESC_woVideoYPelsPerMeter](#) "YPelsPerMeter"
4315#define [VVWINFOTAG_woVideoClrUsed](#) "ClrUsed"
XML tag name for color elements used.

4316#define [VVWINFODESC_woVideoClrUsed](#) "ClrUsed"
4317#define [VVWINFOTAG_woVideoClrImportant](#) "ClrImportant"
XML tag name for.

4318#define [VVWINFODESC_woVideoClrImportant](#) "ClrImportant"
4319#define [VVWINFOTAG_woVideoReserved](#) "Reserved"
XML tag name for reserved array.

4320#define [VVWINFODESC_woVideoReserved](#) "Reserved"
4321#define [VVWINFOTAG_woVideoFccType](#) "FccType"
XML tag name for four cc type (video/audio)

4322#define [VVWINFODESC_woVideoFccType](#) "FccType"
4323#define [VVWINFOTAG_woVideoFccHandler](#) "FccHandler"
XML tag name for four cc handler.

4324#define [VVWINFODESC_woVideoFccHandler](#) "FccHandler"
4325#define [VVWINFOTAG_woVideoFlags](#) "Flags"
XML tag name for flags.

4326#define [VVWINFODESC_woVideoFlags](#) "Flags"
4327#define [VVWINFOTAG_woVideoCaps](#) "Caps"
XML tag name for capabilities.

4328#define [VVWINFODESC_woVideoCaps](#) "Caps"
4329#define [VVWINFOTAG_woVideoPriority](#) "Priority"
XML tag name for priority.

4330#define [VVWINFODESC_woVideoPriority](#) "Priority"
4331#define [VVWINFOTAG_woVideoLanguage](#) "Language"
XML tag name for language.

4332#define [VVWINFODESC_woVideoLanguage](#) "Language"
4333#define [VVWINFOTAG_woVideoScale](#) "Scale"
XML tag name for scale (fps = rate / scale)

4334#define [VVWINFODESC_woVideoScale](#) "Scale"
4335#define [VVWINFOTAG_woVideoRate](#) "Rate"
XML tag name for rate (fps = rate / scale)

4336#define [VVWINFODESC_woVideoRate](#) "Rate"
4337#define [VVWINFOTAG_woVideoStart](#) "Start"
XML tag name for start frame.

4338#define [VVWINFODESC_woVideoStart](#) "Start"
4339#define [VVWINFOTAG_woVideoLength](#) "Length"
XML tag name for the length in frames.

4340#define [VVWINFODESC_woVideoLength](#) "Length"

4341#define [VVWINFOTAG_woVideoInitialFrames](#) "InitialFrames"
XML tag name for number of initial frames to load.

4342#define [VVWINFODESC_woVideoInitialFrames](#) "InitialFrames"
4343#define [VVWINFOTAG_woVideoSuggestedBufferSize](#) "SuggestedBufferSize"
XML tag name for suggested maximum buffer size.

4344#define [VVWINFODESC_woVideoSuggestedBufferSize](#) "SuggestedBufferSize"
4345#define [VVWINFOTAG_woVideoQuality](#) "Quality"
XML tag name for quality.

4346#define [VVWINFODESC_woVideoQuality](#) "Quality"
4347#define [VVWINFOTAG_woVideoSampleSize](#) "SampleSize"
XML tag name for recommended sample size.

4348#define [VVWINFODESC_woVideoSampleSize](#) "SampleSize"
4349#define [VVWINFOTAG_woVideoEditCount](#) "EditCount"
XML tag name for number of edits done on this file.

4350#define [VVWINFODESC_woVideoEditCount](#) "EditCount"
4351#define [VVWINFOTAG_woVideoFormatChangeCount](#) "FormatChangeCount"
XML tag name for number of format changes.

4352#define [VVWINFODESC_woVideoFormatChangeCount](#) "FormatChangeCount"
4353#define [VVWINFOTAG_woVideoPitch](#) "Pitch"
XML tag name for video line pitch.

4354#define [VVWINFODESC_woVideoPitch](#) "Pitch"
4355#define [VVWINFOTAG_woVideoDrFlags](#) "DrFlags"
XML tag name for internal drastic flags.

4356#define [VVWINFODESC_woVideoDrFlags](#) "DrFlags"
4357#define [VVWINFOTAG_woVideoFileType](#) "FileType"
XML tag name for drastic 'mft' file type.

4358#define [VVWINFODESC_woVideoFileType](#) "FileType"
4359#define [VVWINFOTAG_woVideoResDrastic](#) "ResDrastic"
XML tag name for reserved drastic array of DWORDs.

4360#define [VVWINFODESC_woVideoResDrastic](#) "ResDrastic"
4361#define [VVWINFOTAG_woAudioType](#) "AudioType"
XML tag.

4362#define [VVWINFODESC_woAudioType](#) "AudioType"
4363#define [VVWINFOTAG_woAudioChannels](#) "AudioChannels"
XML tag.

4364#define [VVWINFODESC_woAudioChannels](#) "AudioChannels"
4365#define [VVWINFOTAG_woAudioFrequency](#) "AudioFrequency"
XML tag.

4366#define [VVWINFODESC_woAudioFrequency](#) "AudioFrequency"
4367#define [VVWINFOTAG_woAudioBits](#) "AudioBits"
XML tag.

4368#define [VVWINFODESC_woAudioBits](#) "AudioBits"
4369#define [VVWINFO_TOTAL_ITEMS](#) (vwwiEND_OF_STRINGS +
(vwwiEND_OF_DWORD_V2 - vwwiNumericStart))

4370#define [DT_CustomMetadataBlockType_CINE](#) 0x00000001
4371#define [DT_CustomMetadataBlockType_DPX](#) 0x00000002
4372#define [DT_CustomMetadataBlockType_ILLEGAL](#) 0xFFFFFFFF
4373#define [DT_CustomMetadataBlockType_NONE](#) 0x00000000
4374#define [VVWINFOTAG_szFileName](#) "FileName"

XML tag name for [VVWINFO::szFileName](#).

4375#define [VVWINFODESC_szFileName](#) "FileName"

4376#define [VVWINFOFOTAG_szNativeLocator](#) "NativeLocator"

XML tag name for [VVWINFO::szNativeLocator](#).

4377#define [VVWINFODESC_szNativeLocator](#) "NativeLocator"

4378#define [VVWINFOFOTAG_szUniversalName](#) "UniversalLocator"

XML tag name for [VVWINFO::szUniversalLocator](#).

4379#define [VVWINFODESC_szUniversalName](#) "UniversalLocator"

4380#define [VVWINFOFOTAG_szIP](#) "TCP-IPAddress"

XML tag name for [VVWINFO::szIP](#).

4381#define [VVWINFODESC_szIP](#) "TCP-IPAddress"

4382#define [VVWINFOFOTAG_szSourceLocator](#) "SourceLocator"

XML tag name for [VVWINFO::szSourceLocator](#).

4383#define [VVWINFODESC_szSourceLocator](#) "SourceLocator"

4384#define [VVWINFOFOTAG_szChannel](#) "ChannelIdentifier"

XML tag name for [VVWINFO::szChannelIdentifier](#).

4385#define [VVWINFODESC_szChannel](#) "ChannelIdentifier"

4386#define [VVWINFOFOTAG_szChannelName](#) "ChannelName"

XML tag name for [VVWINFO::szChannelName](#).

4387#define [VVWINFODESC_szChannelName](#) "ChannelName"

4388#define [VVWINFOFOTAG_szChannelDescription](#) "ChannelDescription"

XML tag name for [VVWINFO::szChannelDescription](#).

4389#define [VVWINFODESC_szChannelDescription](#) "ChannelDescription"

4390#define [VVWINFOFOTAG_szTitle](#) "Title"

XML tag name for [VVWINFO::szTitle](#).

4391#define [VVWINFODESC_szTitle](#) "Title"

4392#define [VVWINFOFOTAG_szSubject](#) "Subject"

XML tag name for [VVWINFO::szSubject](#).

4393#define [VVWINFODESC_szSubject](#) "Subject"

4394#define [VVWINFOFOTAG_szCategory](#) "Category"

XML tag name for [VVWINFO::szCategory](#).

4395#define [VVWINFODESC_szCategory](#) "Category"

4396#define [VVWINFOFOTAG_szKeywords](#) "Keywords"

XML tag name for [VVWINFO::szKeywords](#).

4397#define [VVWINFODESC_szKeywords](#) "Keywords"

4398#define [VVWINFOFOTAG_szRatings](#) "Ratings"

XML tag name for [VVWINFO::szRatings](#).

4399#define [VVWINFODESC_szRatings](#) "Ratings"

4400#define [VVWINFOFOTAG_szComments](#) "Comments"

XML tag name for [VVWINFO::szComments](#).

4401#define [VVWINFODESC_szComments](#) "Comments"

4402#define [VVWINFOFOTAG_szDoNotUse](#) "DoNotUse"

XML tag name for [VVWINFO::szDoNotUse](#).

4403#define [VVWINFODESC_szDoNotUse](#) "DoNotUse"

4404#define [VVWINFOFOTAG_szOwner](#) "Owner"

XML tag name for [VVWINFO::szOwner](#).

4405#define [VVWINFODESC_szOwner](#) "Owner"

4406#define [VVWINFOFOTAG_szEditor](#) "Editor"

XML tag name for [VVWINFO::szEditor](#).

4407#define [VVWINFODESC_szEditor](#) "Editor"
4408#define [VVWINFOFOTAG_szSupplier](#) "Supplier"

XML tag name for [VVWINFO::szSupplier](#).

4409#define [VVWINFODESC_szSupplier](#) "Supplier"
4410#define [VVWINFOFOTAG_szSource](#) "Source"

XML tag name for [VVWINFO::szSource](#).

4411#define [VVWINFODESC_szSource](#) "Source"
4412#define [VVWINFOFOTAG_szProject](#) "Project"

XML tag name for [VVWINFO::szProject](#).

4413#define [VVWINFODESC_szProject](#) "Project"
4414#define [VVWINFOFOTAG_szStatus](#) "Status"

XML tag name for [VVWINFO::szStatus](#).

4415#define [VVWINFODESC_szStatus](#) "Status"
4416#define [VVWINFOFOTAG_szAuthor](#) "Author"

XML tag name for [VVWINFO::szAuthor](#).

4417#define [VVWINFODESC_szAuthor](#) "Author"
4418#define [VVWINFOFOTAG_szRevisionNumber](#) "RevisionNumber"

XML tag name for [VVWINFO::szRevisionNumber](#).

4419#define [VVWINFODESC_szRevisionNumber](#) "RevisionNumber"
4420#define [VVWINFOFOTAG_szProduced](#) "Produced"

XML tag name for [VVWINFO::szProduced](#).

4421#define [VVWINFODESC_szProduced](#) "Produced"
4422#define [VVWINFOFOTAG_szAlbum](#) "Album"

XML tag name for [VVWINFO::szAlbum](#).

4423#define [VVWINFODESC_szAlbum](#) "Album"
4424#define [VVWINFOFOTAG_szArtist](#) "Artist"

XML tag name for [VVWINFO::szArtist](#).

4425#define [VVWINFODESC_szArtist](#) "Artist"
4426#define [VVWINFOFOTAG_szComposer](#) "Composer"

XML tag name for [VVWINFO::szComposer](#).

4427#define [VVWINFODESC_szComposer](#) "Composer"
4428#define [VVWINFOFOTAG_szCopyright](#) "Copyright"

XML tag name for [VVWINFO::szCopyright](#).

4429#define [VVWINFODESC_szCopyright](#) "Copyright"
4430#define [VVWINFOFOTAG_szCreationData](#) "CreationData"

XML tag name for [VVWINFO::szCreationData](#).

4431#define [VVWINFODESC_szCreationData](#) "CreationData"
4432#define [VVWINFOFOTAG_szDescription](#) "Description"

XML tag name for [VVWINFO::szDescription](#).

4433#define [VVWINFODESC_szDescription](#) "Description"
4434#define [VVWINFOFOTAG_szDirector](#) "Director"

XML tag name for [VVWINFO::szDirector](#).

4435#define [VVWINFODESC_szDirector](#) "Director"
4436#define [VVWINFOFOTAG_szDisclaimer](#) "Disclaimer"

XML tag name for [VVWINFO::szDisclaimer](#).

4437#define [VVWINFODESC_szDisclaimer](#) "Disclaimer"
4438#define [VVWINFOFOTAG_szEncodedBy](#) "EncodedBy"

XML tag name for [VVWINFO::szEncodedBy](#).

4439#define [VVWINFODESC_szEncodedBy](#) "EncodedBy"

4440#define [VVWINFOFOTAG_szFullName](#) "FullName"

XML tag name for [VVWINFO::szFullName](#).

4441#define [VVWINFODESC_szFullName](#) "FullName"

4442#define [VVWINFOFOTAG_szGenre](#) "Genre"

XML tag name for [VVWINFO::szGenre](#).

4443#define [VVWINFODESC_szGenre](#) "Genre"

4444#define [VVWINFOFOTAG_szHostComputer](#) "HostComputer"

XML tag name for [VVWINFO::szHostComputer](#).

4445#define [VVWINFODESC_szHostComputer](#) "HostComputer"

4446#define [VVWINFOFOTAG_szInformation](#) "Information"

XML tag name for [VVWINFO::szInformation](#).

4447#define [VVWINFODESC_szInformation](#) "Information"

4448#define [VVWINFOFOTAG_szMake](#) "Make"

XML tag name for [VVWINFO::szMake](#).

4449#define [VVWINFODESC_szMake](#) "Make"

4450#define [VVWINFOFOTAG_szModel](#) "Model"

XML tag name for [VVWINFO::szModel](#).

4451#define [VVWINFODESC_szModel](#) "Model"

4452#define [VVWINFOFOTAG_szOriginalArtist](#) "OriginalArtist"

XML tag name for [VVWINFO::szOriginalArtist](#).

4453#define [VVWINFODESC_szOriginalArtist](#) "OriginalArtist"

4454#define [VVWINFOFOTAG_szOriginalFormat](#) "OriginalFormat"

XML tag name for [VVWINFO::szOriginalFormat](#).

4455#define [VVWINFODESC_szOriginalFormat](#) "OriginalFormat"

4456#define [VVWINFOFOTAG_szPerformers](#) "Performers"

XML tag name for [VVWINFO::szPerformers](#).

4457#define [VVWINFODESC_szPerformers](#) "Performers"

4458#define [VVWINFOFOTAG_szProducer](#) "Producer"

XML tag name for [VVWINFO::szProducer](#).

4459#define [VVWINFODESC_szProducer](#) "Producer"

4460#define [VVWINFOFOTAG_szProduct](#) "Product"

XML tag name for [VVWINFO::szProduct](#).

4461#define [VVWINFODESC_szProduct](#) "Product"

4462#define [VVWINFOFOTAG_szSoftware](#) "Software"

XML tag name for [VVWINFO::szSoftware](#).

4463#define [VVWINFODESC_szSoftware](#) "Software"

4464#define [VVWINFOFOTAG_szSpecialPlaybackRequirements](#) "SpecialPlaybackRequirements"

XML tag name for [VVWINFO::szSpecialPlaybackRequirements](#).

4465#define [VVWINFODESC_szSpecialPlaybackRequirements](#) "SpecialPlaybackRequirements"

4466#define [VVWINFOFOTAG_szTrack](#) "Track"

XML tag name for [VVWINFO::szTrack](#).

4467#define [VVWINFODESC_szTrack](#) "Track"

4468#define [VVWINFOFOTAG_szWarning](#) "Warning"

XML tag name for [VVWINFO::szWarning](#).

4469#define [VVWINFODESC_szWarning](#) "Warning"

4470#define [VVWINFOFOTAG_szURLLink](#) "URL"

XML tag name for [VVWINFO::szURLLink](#).

4471#define [VVWINFODESC_szURLLink](#) "URL"

4472#define [VVWINFOFOTAG_szEditData1](#) "EditData1"

XML tag name for [VVWINFO::szEditData1](#).

4473#define [VVWINFODESC_szEditData1](#) "EditData1"

4474#define [VVWINFOFOTAG_szEditData2](#) "EditData2"

XML tag name for [VVWINFO::szEditData2](#).

4475#define [VVWINFODESC_szEditData2](#) "EditData2"

4476#define [VVWINFOFOTAG_szEditData3](#) "EditData3"

XML tag name for [VVWINFO::szEditData3](#).

4477#define [VVWINFODESC_szEditData3](#) "EditData3"

4478#define [VVWINFOFOTAG_szEditData4](#) "EditData4"

XML tag name for [VVWINFO::szEditData4](#).

4479#define [VVWINFODESC_szEditData4](#) "EditData4"

4480#define [VVWINFOFOTAG_szEditData5](#) "EditData5"

XML tag name for [VVWINFO::szEditData5](#).

4481#define [VVWINFODESC_szEditData5](#) "EditData5"

4482#define [VVWINFOFOTAG_szEditData6](#) "EditData6"

XML tag name for [VVWINFO::szEditData6](#).

4483#define [VVWINFODESC_szEditData6](#) "EditData6"

4484#define [VVWINFOFOTAG_szEditData7](#) "EditData7"

XML tag name for [VVWINFO::szEditData7](#).

4485#define [VVWINFODESC_szEditData7](#) "EditData7"

4486#define [VVWINFOFOTAG_szEditData8](#) "EditData8"

XML tag name for [VVWINFO::szEditData8](#).

4487#define [VVWINFODESC_szEditData8](#) "EditData8"

4488#define [VVWINFOFOTAG_szEditData9](#) "EditData9"

XML tag name for [VVWINFO::szEditData9](#).

4489#define [VVWINFODESC_szEditData9](#) "EditData9"

4490#define [VVWINFOFOTAG_szVersionString](#) "VersionString"

XML tag name for [VVWINFO::szVersionString](#).

4491#define [VVWINFODESC_szVersionString](#) "VersionString"

4492#define [VVWINFOFOTAG_szManufacturer](#) "Manufacturer"

XML tag name for [VVWINFO::szManufacturer](#).

4493#define [VVWINFODESC_szManufacturer](#) "Manufacturer"

4494#define [VVWINFOFOTAG_szLanguage](#) "Language"

XML tag name for [VVWINFO::szLanguage](#).

4495#define [VVWINFODESC_szLanguage](#) "Language"

4496#define [VVWINFOFOTAG_szFormat](#) "Format"

XML tag name for [VVWINFO::szFormat](#).

4497#define [VVWINFODESC_szFormat](#) "Format"

4498#define [VVWINFOFOTAG_szInputDevice](#) "InputDevice"

XML tag name for [VVWINFO::szInputDevice](#).

4499#define [VVWINFODESC_szInputDevice](#) "InputDevice"

4500#define [VVWINFOFOTAG_szDeviceModelNum](#) "DeviceModelNum"

XML tag name for [VVWINFO::szDeviceModelNum](#).

4501#define [VVWINFODESC_szDeviceModelNum](#) "DeviceModelNum"

4502#define [VVWINFOFOTAG_szDeviceSerialNum](#) "DeviceSerialNum"

XML tag name for [VVWINFO::szDeviceSerialNum](#).

4503#define [VVWINFODESC_szDeviceSerialNum](#) "DeviceSerialNum"

4504#define [VVWINFOFOTAG_szReel](#) "Reel"

XML tag name for [VVWINFO::szReel](#).

4505#define [VVWINFODESC_szReel](#) "Reel"

4506#define [VVWINFOFOTAG_szShot](#) "Shot"

XML tag name for [VVWINFO::szShot](#).

4507#define [VVWINFODESC_szShot](#) "Shot"

4508#define [VVWINFOFOTAG_szTake](#) "Take"

XML tag name for [VVWINFO::szTake](#).

4509#define [VVWINFODESC_szTake](#) "Take"

4510#define [VVWINFOFOTAG_szSlateInfo](#) "SlateInfo"

XML tag name for [VVWINFO::szSlateInfo](#).

4511#define [VVWINFODESC_szSlateInfo](#) "SlateInfo"

4512#define [VVWINFOFOTAG_szFrameAttribute](#) "FrameAttribute"

XML tag name for [VVWINFO::szFrameAttribute](#).

4513#define [VVWINFODESC_szFrameAttribute](#) "FrameAttribute"

4514#define [VVWINFOFOTAG_szEpisode](#) "Episode"

XML tag name for [VVWINFO::szEpisode](#).

4515#define [VVWINFODESC_szEpisode](#) "Episode"

4516#define [VVWINFOFOTAG_szScene](#) "Scene"

XML tag name for [VVWINFO::szScene](#).

4517#define [VVWINFODESC_szScene](#) "Scene"

4518#define [VVWINFOFOTAG_szDailyRoll](#) "DailyRoll"

XML tag name for [VVWINFO::szDailyRoll](#).

4519#define [VVWINFODESC_szDailyRoll](#) "DailyRoll"

4520#define [VVWINFOFOTAG_szCamRoll](#) "CamRoll"

XML tag name for [VVWINFO::szCamRoll](#).

4521#define [VVWINFODESC_szCamRoll](#) "CamRoll"

4522#define [VVWINFOFOTAG_szSoundRoll](#) "SoundRoll"

XML tag name for [VVWINFO::szSoundRoll](#).

4523#define [VVWINFODESC_szSoundRoll](#) "SoundRoll"

4524#define [VVWINFOFOTAG_szLabRoll](#) "LabRoll"

XML tag name for [VVWINFO::szLabRoll](#).

4525#define [VVWINFODESC_szLabRoll](#) "LabRoll"

4526#define [VVWINFOFOTAG_szKeyNumberPrefix](#) "KeyNumberPrefix"

XML tag name for [VVWINFO::szKeyNumberPrefix](#).

4527#define [VVWINFODESC_szKeyNumberPrefix](#) "KeyNumberPrefix"

4528#define [VVWINFOFOTAG_szInkNumberPrefix](#) "InkNumberPrefix"

XML tag name for [VVWINFO::szInkNumberPrefix](#).

4529#define [VVWINFODESC_szInkNumberPrefix](#) "InkNumberPrefix"

4530#define [VVWINFOFOTAG_szPictureIcon](#) "PictureIcon"

XML tag name for [VVWINFO::szPictureIcon](#).

4531#define [VVWINFODESC_szPictureIcon](#) "PictureIcon"

4532#define [VVWINFOFOTAG_szProxyFile](#) "ProxyFile"

XML tag name for [VVWINFO::szProxyFile](#).

4533#define [VVWINFODESC_szProxyFile](#) "ProxyFile"

4534#define [VVWINFOFOTAG_szCustomMetadataBlockPointer](#) "CustomMetadataBlockPointer"

XML tag name for [VVWINFO::szCustomMetadataBlockPointer](#).

4535#define [VVWINFODESC_szCustomMetadataBlockPointer](#) "CustomMetadataBlockPointer"

4536#define [VVWINFOFOTAG_szImageInfo](#) "ImageInfo"

XML tag name for [VVWINFO::szImageInfo](#).

4537#define [VVWINFODESC_ImageInfo](#) "ImageInfo"

4538#define [VVWINFOFOTAG_szUMID](#) "UMID"

XML tag name for [VVWINFO::szUMID](#).

4539#define [VVWINFODESC_UMID](#) "UMID"

4540#define [VVWINFOFOTAG_szNU_84](#) "NotUsed_84"

XML tag name for [VVWINFO::szNU_84](#).

4541#define [VVWINFOFOTAG_szNU_85](#) "NotUsed_85"

XML tag name for [VVWINFO::szNU_85](#).

4542#define [VVWINFOFOTAG_szNU_86](#) "NotUsed_86"

XML tag name for [VVWINFO::szNU_86](#).

4543#define [VVWINFOFOTAG_szNU_87](#) "NotUsed_87"

XML tag name for [VVWINFO::szNU_87](#).

4544#define [VVWINFOFOTAG_szNU_88](#) "NotUsed_88"

XML tag name for [VVWINFO::szNU_88](#).

4545#define [VVWINFOFOTAG_szNU_89](#) "NotUsed_89"

XML tag name for [VVWINFO::szNU_89](#).

4546#define [VVWINFOFOTAG_szNU_90](#) "NotUsed_90"

XML tag name for [VVWINFO::szNU_90](#).

4547#define [VVWINFOFOTAG_szNU_91](#) "NotUsed_91"

XML tag name for [VVWINFO::szNU_91](#).

4548#define [VVWINFOFOTAG_szNU_92](#) "NotUsed_92"

XML tag name for [VVWINFO::szNU_92](#).

4549#define [VVWINFOFOTAG_szNU_93](#) "NotUsed_93"

XML tag name for [VVWINFO::szNU_93](#).

4550#define [VVWINFOFOTAG_szNU_94](#) "NotUsed_94"

XML tag name for [VVWINFO::szNU_94](#).

4551#define [VVWINFOFOTAG_szNU_95](#) "NotUsed_95"

XML tag name for [VVWINFO::szNU_95](#).

4552#define [VVWINFOFOTAG_szNU_96](#) "NotUsed_96"

XML tag name for [VVWINFO::szNU_96](#).

4553#define [VVWINFOFOTAG_szNU_97](#) "NotUsed_97"

XML tag name for [VVWINFO::szNU_97](#).

4554#define [VVWINFOFOTAG_szNU_98](#) "NotUsed_98"

XML tag name for [VVWINFO::szNU_98](#).

4555#define [VVWINFOFOTAG_szNU_99](#) "NotUsed_99"

XML tag name for [VVWINFO::szNU_99](#).

4556#define [VVWINFOFOTAG_dwTimeCode](#) "TimeCode"

XML tag name for [VVWINFO::dwTimeCode](#).

4557#define [VVWINFODESC_dwTimeCode](#) "TimeCode"

4558#define [VVWINFOFOTAG_dwUserBits](#) "UserBits"

XML tag name for [VVWINFO::dwUserBits](#).

4559#define [VVWINFODESC_dwUserBits](#) "UserBits"

4560#define [VWINFO_TAG_dwVITCTimeCode](#) "VITCTimeCode"
XML tag name for [VWINFO::dwVITCTimeCode](#).

4561#define [VWINFODESC_dwVITCTimeCode](#) "VITCTimeCode"
4562#define [VWINFO_TAG_dwVITCUserBits](#) "VITCUserBits"
XML tag name for [VWINFO::dwVITCUserBits](#).

4563#define [VWINFODESC_dwVITCUserBits](#) "VITCUserBits"
4564#define [VWINFO_TAG_dwVITCLine3](#) "VITCExtraData"
XML tag name for [VWINFO::dwVITCLine3](#).

4565#define [VWINFODESC_dwVITCLine3](#) "VITCExtraData"
4566#define [VWINFO_TAG_dwPosterFrame](#) "PosterFrame"
XML tag name for [VWINFO::dwPosterFrame](#).

4567#define [VWINFODESC_dwPosterFrame](#) "PosterFrame"
4568#define [VWINFO_TAG_dwAFrame](#) "A-Frame"
XML tag name for [VWINFO::dwAFrame](#).

4569#define [VWINFODESC_dwAFrame](#) "A-Frame"
4570#define [VWINFO_TAG_dwAspectRatio](#) "AspectRatio"
XML tag name for [VWINFO::dwAspectRatio](#).

4571#define [VWINFODESC_dwAspectRatio](#) "AspectRatio"
4572#define [VWINFO_TAG_dwOriginalRate](#) "OriginalRate"
XML tag name for [VWINFO::dwOriginalRate](#).

4573#define [VWINFODESC_dwOriginalRate](#) "OriginalRate"
4574#define [VWINFO_TAG_dwOriginalScale](#) "OriginalScale"
XML tag name for [VWINFO::dwOriginalScale](#).

4575#define [VWINFODESC_dwOriginalScale](#) "OriginalScale"
4576#define [VWINFO_TAG_dwConversions](#) "TotalConversions"
XML tag name for [VWINFO::dwTotalConversions](#).

4577#define [VWINFODESC_dwConversions](#) "TotalConversions"
4578#define [VWINFO_TAG_dwVersionNumber](#) "VersionNumber"
XML tag name for [VWINFO::dwVersionNumber](#).

4579#define [VWINFODESC_dwVersionNumber](#) "VersionNumber"
4580#define [VWINFO_TAG_dwFileSize](#) "FileSize"
XML tag name for [VWINFO::dwFileSize](#).

4581#define [VWINFODESC_dwFileSize](#) "FileSize"
4582#define [VWINFO_TAG_dwFileDate](#) "FileDate"
XML tag name for [VWINFO::dwFileDate](#).

4583#define [VWINFODESC_dwFileDate](#) "FileDate"
4584#define [VWINFO_TAG_dwFileTime](#) "FileTime"
XML tag name for [VWINFO::dwFileTime](#).

4585#define [VWINFODESC_dwFileTime](#) "FileTime"
4586#define [VWINFO_TAG_dwSequenceNumber](#) "SequenceNumber"
XML tag name for [VWINFO::dwSequenceNumber](#).

4587#define [VWINFODESC_dwSequenceNumber](#) "SequenceNumber"
4588#define [VWINFO_TAG_dwTotalStreams](#) "TotalStreams"
XML tag name for [VWINFO::dwTotalStreams](#).

4589#define [VWINFODESC_dwTotalStreams](#) "TotalStreams"
4590#define [VWINFO_TAG_dwTotalLength](#) "TotalLength"
XML tag name for [VWINFO::dwTotalLength](#).

4591#define [VWINFODESC_dwTotalLength](#) "TotalLength"

4592#define [VWINFODESC_dwFilmManufacturerCode](#) "FilmManufacturerCode"
XML tag name for [VWINFO::dwFilmManufacturerCode](#).

4593#define [VWINFODESC_dwFilmManufacturerCode](#) "FilmManufacturerCode"
4594#define [VWINFODESC_dwFilmTypeCode](#) "FilmTypeCode"
XML tag name for [VWINFO::dwFilmTypeCode](#).

4595#define [VWINFODESC_dwFilmTypeCode](#) "FilmTypeCode"
4596#define [VWINFODESC_dwWhitePoint](#) "WhitePoint"
XML tag name for [VWINFO::dwWhitePoint](#).

4597#define [VWINFODESC_dwWhitePoint](#) "WhitePoint"
4598#define [VWINFODESC_dwBlackPoint](#) "BlackPoint"
XML tag name for [VWINFO::dwBlackPoint](#).

4599#define [VWINFODESC_dwBlackPoint](#) "BlackPoint"
4600#define [VWINFODESC_dwBlackGain](#) "BlackGain"
XML tag name for [VWINFO::dwBlackGain](#).

4601#define [VWINFODESC_dwBlackGain](#) "BlackGain"
4602#define [VWINFODESC_dwBreakPoint](#) "BreakPoint"
XML tag name for [VWINFO::dwBreakPoint](#).

4603#define [VWINFODESC_dwBreakPoint](#) "BreakPoint"
4604#define [VWINFODESC_dwGamma1000](#) "Gamma1000"
XML tag name for [VWINFO::dwGamma1000](#).

4605#define [VWINFODESC_dwGamma1000](#) "Gamma1000"
4606#define [VWINFODESC_dwTagNumber](#) "TagNumber"
XML tag name for [VWINFO::dwTagNumber](#).

4607#define [VWINFODESC_dwTagNumber](#) "TagNumber"
4608#define [VWINFODESC_dwFlags](#) "Flags"
XML tag name for [VWINFO::dwFlags](#).

4609#define [VWINFODESC_dwFlags](#) "Flags"
4610#define [VWINFODESC_dwTimeCodeType](#) "TimeCodeType"
XML tag name for [VWINFO::dwTimeCodeType](#).

4611#define [VWINFODESC_dwTimeCodeType](#) "TimeCodeType"
4612#define [VWINFODESC_dwLTCTimeCodeType](#) "LTCTimeCodeType"
XML tag name for [VWINFO::dwLTCTimeCodeType](#).

4613#define [VWINFODESC_dwLTCTimeCodeType](#) "LTCTimeCodeType"
4614#define [VWINFODESC_dwVITCTimeCodeType](#) "VITCTimeCodeType"
XML tag name for [VWINFO::dwVITCTimeCodeType](#).

4615#define [VWINFODESC_dwVITCTimeCodeType](#) "VITCTimeCodeType"
4616#define [VWINFODESC_dwProdDate](#) "ProdDate"
XML tag name for [VWINFO::dwProdDate](#).

4617#define [VWINFODESC_dwProdDate](#) "ProdDate"
4618#define [VWINFODESC_dwUniqueID](#) "UniqueID"
XML tag name for [VWINFO::dwUniqueID](#).

4619#define [VWINFODESC_dwUniqueID](#) "UniqueID"
4620#define [VWINFODESC_dwCustomMetadataBlockType](#) "CustomMetadataBlockType"
XML tag name for [VWINFO::dwCustomMetadataBlockType](#).

4621#define [VWINFODESC_dwCustomMetadataBlockType](#) "CustomMetadataBlockType"
4622#define [VWINFODESC_dwCustomMetadataBlockSize](#) "CustomMetadataBlockSize"
XML tag name for [VWINFO::dwCustomMetadataBlockSize](#).

4623#define [VWINFODESC_dwCustomMetadataBlockSize](#) "CustomMetadataBlockSize"

4624#define [VVWINFO::dwNorthSouthEastWest](#) "NorthSouthEastWest"
XML tag name for [VVWINFO::dwNorthSouthEastWest](#).

4625#define [VVWINFODESC_dwNorthSouthEastWest](#) "NorthSouthEastWest"

4626#define [VVWINFO::dwLatitude](#) "Latitude"
XML tag name for [VVWINFO::dwLatitude](#).

4627#define [VVWINFODESC_dwLatitude](#) "Latitude"

4628#define [VVWINFO::dwLongitude](#) "Longitude"
XML tag name for [VVWINFO::dwLongitude](#).

4629#define [VVWINFODESC_dwLongitude](#) "Longitude"

4630#define [VVWINFO::dwNU_38](#) "NumericNotUsed_38"
XML tag name for [VVWINFO::dwNU_38](#).

4631#define [VVWINFO::dwNU_39](#) "NumericNotUsed_39"
XML tag name for [VVWINFO::dwNU_39](#).

4632#define [VVWINFO::dwNU_40](#) "NumericNotUsed_40"
XML tag name for [VVWINFO::dwNU_40](#).

4633#define [VVWINFO::dwNU_41](#) "NumericNotUsed_41"
XML tag name for [VVWINFO::dwNU_41](#).

4634#define [VVWINFO::dwNU_42](#) "NumericNotUsed_42"
XML tag name for [VVWINFO::dwNU_42](#).

4635#define [VVWINFO::dwNU_43](#) "NumericNotUsed_43"
XML tag name for [VVWINFO::dwNU_43](#).

4636#define [VVWINFO::dwNU_44](#) "NumericNotUsed_44"
XML tag name for [VVWINFO::dwNU_44](#).

4637#define [VVWINFO::dwNU_45](#) "NumericNotUsed_45"
XML tag name for [VVWINFO::dwNU_45](#).

4638#define [VVWINFO::dwNU_46](#) "NumericNotUsed_46"
XML tag name for [VVWINFO::dwNU_46](#).

4639#define [VVWINFO::dwNU_47](#) "NumericNotUsed_47"
XML tag name for [VVWINFO::dwNU_47](#).

4640#define [VVWINFO::dwNU_48](#) "NumericNotUsed_48"
XML tag name for [VVWINFO::dwNU_48](#).

4641#define [VVWINFO::dwNU_49](#) "NumericNotUsed_49"
XML tag name for [VVWINFO::dwNU_49](#).

4642#define [DT_META_DATA_CHANGED](#) 0x01

4643#define [GetAudio](#) 0x00000000

4644#define [GetVideo](#) 0x00000001
Flag for mediafile/avhal to get video dframe.

4645#define [GetInf](#) 0x00000002
Flag for mediafile/avhal to get info ([VVWINFO](#))

4646#define [PutAudio](#) GetAudio
Flag for mediafile/avhal to put audio dframe.

4647#define [PutVideo](#) GetVideo
Flag for mediafile/avhal to put audio dframe.

4648#define [PutInf](#) GetInf
Flag for mediafile/avhal to get info ([VVWINFO](#))

4649#define [GetNoFail](#) 0x00000100

Flag for getting second copy or other non queue critical frames.

4650#define [GetCurrent](#) 0x00000000

Flag for avhal to get the current [FRAME_INFO](#).

4651#define [GetQueued](#) 0x00000010

Flag for avhal to get the last queued [FRAME_INFO](#).

4652#define [GetCurrentField](#) 0x00000020

Flag for avhal to get the last queued [FRAME_INFO](#).

4653#define [IsSingleField](#) 0x00000040

Flag for avhal to get the last queued [FRAME_INFO](#).

4654#define [GetVBIVideo](#) 0x00000080

Flag for avhal to get the vbi video queued [FRAME_INFO](#).

4655#define [GetCTLTrack](#) 0x00000200

Flag to get CTL even if VITC/LTC selected.

4656#define [MB_FILE_EXT_LEN](#) 200

String return sizes (mediabase/mediafile plugin)

4657#define [MB_FILE_SHORT_LEN](#) 64

Maximum length for the base or short description (used in drop down)

4658#define [MB_FILE_DESC_LEN](#) 255

Maximum length for the long description.

4659#define [MB_CODEC_SHORT_LEN](#) 64

Maximum length for the short codec name.

4660#define [MB_CODEC_DESC_LEN](#) 255

Maximum length for the full (long) codec description.

4661#define [VVW_DSSYNC_CHANNEL](#) 65536

4662#define [VVW_DSSYNC_LTC_CHANNEL](#) _VVW_DSSYNC_CHANNEL + 1

4663#define [VVW_CONTROL_START_LOCAL](#) 0

4664#define [VVW_CONTROL_MAX_LOCAL](#) 64U

4665#define [VVW_CONTROL_START_EXTERNAL](#) _VVW_CONTROL_MAX_LOCAL

4666#define [VVW_CONTROL_MAX_EXTERNAL](#) 64U

Maximum number of external channels (63..127)

4667#define [VVW_CONTROL_START_NETWORK](#) (_VVW_CONTROL_MAX_EXTERNAL +
_VVW_CONTROL_MAX_EXTERNAL)

4668#define [VVW_CONTROL_MAX_NETWORK](#) 128U

Maximum number of network channels (128..255)

4669#define [VVW_CONTROL_MAX_CONTROL](#) _VVW_CONTROL_MAX_LOCAL

This is the maximum number of control inputs available.

4670#define [VVW_ABS_MAX_CHANNELS](#) 256

Abs Max Channels on any system.

Typedefs

4671typedef struct [FRAME_INFO](#) * [pFRAME_INFO](#)

4672typedef struct [DRASTIC_CHANNEL](#) * [pDRASTIC_CHANNEL](#)

4673typedef struct [DCLIP](#) * [PDCLIP](#)

4674typedef struct [DFRAME](#) * [PDFRAME](#)

4675typedef struct [DPOSSIZENAME](#) * [PDPOSSIZENAME](#)
4676typedef struct [VVWSYSTEM](#) * [pVVWSYSTEM](#)
4677typedef struct [VVWVIDEO](#) * [pVVWVIDEO](#)
4678typedef struct [VVWAUDIO](#) * [pVVWAUDIO](#)
4679typedef struct [DTDIRECT_WAVEHDR](#) * [pDTDIRECT_WAVEHDR](#)
4680typedef struct [vwwInflImageInfo](#) * [pvwwInflImageInfo](#)
4681typedef struct [VVWINFO](#) * [pVVWINFO](#)

Enumerations

4682enum [vwwInfoMetaTypes](#) { [vwwiFileName](#), [vwwiNativeLocator](#), [vwwiUniversalName](#), [vwwiIP](#),
[vwwiSourceLocator](#), [vwwiChannel](#), [vwwiChannelName](#), [vwwiChannelDescription](#), [vwwiTitle](#),
[vwwiSubject](#), [vwwiCategory](#), [vwwiKeywords](#), [vwwiRatings](#), [vwwiComments](#), [vwwiOwner](#),
[vwwiEditor](#), [vwwiSupplier](#), [vwwiSource](#), [vwwiProject](#), [vwwiStatus](#), [vwwiAuthor](#),
[vwwiRevisionNumber](#), [vwwiProduced](#), [vwwiAlbum](#), [vwwiArtist](#), [vwwiComposer](#), [vwwiCopyright](#),
[vwwiCreationData](#), [vwwiDescription](#), [vwwiDirector](#), [vwwiDisclaimer](#), [vwwiEncodedBy](#),
[vwwiFullName](#), [vwwiGenre](#), [vwwiHostComputer](#), [vwwiInformation](#), [vwwiMake](#), [vwwiModel](#),
[vwwiOriginalArtist](#), [vwwiOriginalFormat](#), [vwwiPerformers](#), [vwwiProducer](#), [vwwiProduct](#),
[vwwiSoftware](#), [vwwiSpecialPlaybackRequirements](#), [vwwiTrack](#), [vwwiWarning](#), [vwwiURLLink](#),
[vwwiEditData1](#), [vwwiEditData2](#), [vwwiEditData3](#), [vwwiEditData4](#), [vwwiEditData5](#), [vwwiEditData6](#),
[vwwiEditData7](#), [vwwiEditData8](#), [vwwiEditData9](#), [vwwiVersionString](#), [vwwiManufacturer](#),
[vwwiLanguage](#), [vwwiFormat](#), [vwwiInputDevice](#), [vwwiDeviceModelNum](#), [vwwiDeviceSerialNum](#),
[vwwiReel](#), [vwwiShot](#), [vwwiTake](#), [vwwiSlateInfo](#), [vwwiFrameAttribute](#), [vwwiEpisode](#), [vwwiScene](#),
[vwwiDailyRoll](#), [vwwiCamRoll](#), [vwwiSoundRoll](#), [vwwiLabRoll](#), [vwwiKeyNumberPrefix](#),
[vwwiInkNumberPrefix](#), [vwwiPictureIcon](#), [vwwiProxyFile](#), [vwwiCustomMetadataBlockPointer](#),
[vwwiImageInfo](#), [vwwiUMID](#), [vwwiEND_OF_STRINGS](#), [vwwiNumericStart](#) = 0x1000,
[vwwiTimeCode](#), [vwwiUserBits](#), [vwwiVITCTimeCode](#), [vwwiVITCUserBits](#), [vwwiVITCLine3](#),
[vwwiPosterFrame](#), [vwwiAFrame](#), [vwwiAspectRatio](#), [vwwiOriginalRate](#), [vwwiOriginalScale](#),
[vwwiConversions](#), [vwwiVersionNumber](#), [vwwiFileSize](#), [vwwiFileDate](#), [vwwiFileTime](#),
[vwwiSequenceNumber](#), [vwwiTotalStreams](#), [vwwiTotalLength](#), [vwwiFilmManufacturerCode](#),
[vwwiFilmTypeCode](#), [vwwiWhitePoint](#), [vwwiBlackPoint](#), [vwwiBlackGain](#), [vwwiBreakPoint](#),
[vwwiGamma1000](#), [vwwiTagNumber](#), [vwwiFlags](#), [vwwiTimeCodeType](#), [vwwiLTCTimeCodeType](#),
[vwwiVITCTimeCodeType](#), [vwwiProdDate](#), [vwwiUniqueID](#), [vwwiCustomMetadataBlockType](#),
[vwwiCustomMetadataBlockSize](#), [vwwiNorthSouthEastWest](#), [vwwiLatitude](#), [vwwiLongitude](#),
[vwwiEND_OF_DWORD_V2](#), [vwwiVideoWidth](#) = 0x10000 }

Define Documentation

#define [_MAX_CLIP_NAME_SIZE](#) 9

The size of a clip name string (8 char plus NULL term) (1 for safety) - This is for odetics/louth compatibility.

Definition at line 81 of file [vwwtypes.h](#).

#define [_PDFRAMEFLAGS_CLIPEND](#) 0x00000004

This is the last frame of the current clip see [DFRAME::dwFlags](#).

Definition at line 517 of file [vwwtypes.h](#).

#define _PDFRAMEFLAGS_CLIPSTART 0x00000002

This is the first frame of a new clip see [DFRAME::dwFlags](#).

Definition at line 515 of file vvwtypes.h.

#define _PDFRAMEFLAGS_CLIPSTILL 0x00000001

This is a single frame clip (implies progressive) that needs to be play for the duration of the clip see [DFRAME::dwFlags](#).

Definition at line 513 of file vvwtypes.h.

#define _PDFRAMEFLAGS_FIELD_FIRST 0x00000000

This is the first field, if the HASMARK is set.

Definition at line 529 of file vvwtypes.h.

#define _PDFRAMEFLAGS_FIELD_HASMARK 0x00020000

If set, the first second bit is correct, if not then ignore even off bit.

Definition at line 527 of file vvwtypes.h.

#define _PDFRAMEFLAGS_FIELD_INLINE 0x00040000

This is the second field, if the HASMARK is set.

Definition at line 533 of file vvwtypes.h.

#define _PDFRAMEFLAGS_FIELD_MARK_MASK 0x00030000

Interleaved frame had already been split and it the first field part mask.

Definition at line 525 of file vvwtypes.h.

**#define _PDFRAMEFLAGS_FIELD_MARK_TWO_FIELDS
(_PDFRAMEFLAGS_FIELD_INLINE | _PDFRAMEFLAGS_FIELD_HASMARK)**

Set as two fields, one full field followed by the other ie not interlaced.

Definition at line 539 of file vvwtypes.h.

**#define _PDFRAMEFLAGS_FIELD_MARKFIRST (0x00000000 |
_PDFRAMEFLAGS_FIELD_HASMARK)**

Set as first field.

Definition at line 535 of file vvwtypes.h.

**#define _PDFRAMEFLAGS_FIELD_MARKSECOND (0x00010000 |
_PDFRAMEFLAGS_FIELD_HASMARK)**

Set as second field.

Definition at line 537 of file vvwtypes.h.

#define _PDFRAMEFLAGS_FIELD_SECOND 0x00010000

This is the second field, if the HASMARK is set.

Definition at line 531 of file vvwtypes.h.

#define _PDFRAMEFLAGS_FIRSTFRAME 0x00000002

Alias for [_PDFRAMEFLAGS_CLIPSTART](#) see [DFRAME::dwFlags](#).

Definition at line 519 of file vvwtypes.h.

#define _PDFRAMEFLAGS_IMPOSED_MARKED 0x00000008

This Frame has been imposed if need be and watermarked.

Definition at line 523 of file vvwtypes.h.

#define _PDFRAMEFLAGS_LASTFRAME 0x00000004

Alias for [_PDFRAMEFLAGS_CLIPEND](#) see [DFRAME::dwFlags](#).

Definition at line 521 of file vvwtypes.h.

#define _VW_ABS_MAX_CHANNELS 256

Abs Max Channels on any system.

Definition at line 125 of file vvwtypes.h.

#define _VW_CONTROL_MAX_CONTROL _VW_CONTROL_MAX_LOCAL

This is the maximum number of control inputs available.

Definition at line 123 of file vvwtypes.h.

#define _VW_CONTROL_MAX_EXTERNAL 64U

Maximum number of external channels (63..127)

Definition at line 112 of file vvwtypes.h.

#define _VW_CONTROL_MAX_LOCAL 64U

This is the maximum number of channels that may exist on a machine locally. They are

numbered from 0..63 inclusive to the controlling software

Definition at line 104 of file vvwtypes.h.

#define _VW_CONTROL_MAX_NETWORK 128U

Maximum number of network channels (128..255)

Definition at line 121 of file vvwtypes.h.

#define _VW_CONTROL_START_EXTERNAL _VW_CONTROL_MAX_LOCAL

This is the maximum number of local serial control channels, LTC, VITC or any other info/control channels that may exist on a single machine. They are numbered 64..127

Definition at line 110 of file vvwtypes.h.

#define _VW_CONTROL_START_LOCAL 0

First real channel

Definition at line 98 of file vvwtypes.h.

**#define _VW_CONTROL_START_NETWORK (_VW_CONTROL_MAX_EXTERNAL +
_VW_CONTROL_MAX_EXTERNAL)**

This is the maximum number of network control channels. Each of the other channels exports a network server and may be connected to any of these channels as the user wishes. They are numbered 128..255

Definition at line 119 of file vvwtypes.h.

#define _VW_DSINC_CHANNEL 65536

Channel definition for VVW: Special sync channel

Definition at line 90 of file vvwtypes.h.

#define _VW_DSINC_LTC_CHANNEL _VW_DSINC_CHANNEL + 1

First ltc channel

Definition at line 94 of file vvwtypes.h.

#define _VW_IS_VVWAUDIO 0x0100

The flag indicating the structure is a [VVWAUDIO](#) structure within a union.

Definition at line 1284 of file vvwtypes.h.

#define _VW_IS_VVWINFO 0x1000

The flag indicating the structure is a [VVWINFO](#) structure within a union.

Definition at line 1537 of file vvwtypes.h.

#define _VW_IS_VVWSYSTEM 0x0000

The flag indicating the structure is a [VVWSYSTEM](#) structure within a union.
Definition at line 671 of file vvwtypes.h.

#define _VW_IS_VWVIDEO 0x0010

The flag indicating the structure is a [VWVIDEO](#) structure within a union.
Definition at line 938 of file vvwtypes.h.

#define _VWXXX_NAME_SIZE 256

The size of the name area (in chars) within [VVWSYSTEM](#), [VWVIDEO](#), [VWAUDIO](#) and [VWINFO](#).
Definition at line 654 of file vvwtypes.h.

#define _VWXXX_RESERVED_SIZE 256

The size of the reserved area (in DWORDs) within [VVWSYSTEM](#), [VWVIDEO](#), [VWAUDIO](#) and [VWINFO](#).
Definition at line 652 of file vvwtypes.h.

#define COPYFRAMEINFO(_ptrFiDst, _ptrFiSrc)

```
Value: CopyMemoryFast( _ptrFiDst, _ptrFiSrc, sizeof( FRAME\_INFO )); \
      (_ptrFiDst)->dwVitcAux = 0; \
      (_ptrFiDst)->dwCCData = 0; \
```

Copy the frame info area, remove pointers.
Definition at line 252 of file vvwtypes.h.

#define DECLARE_DRASTIC_CHANNEL(__x_)

```
Value: { 0xFFFFFFFF, 0, 0, 0, 0, 0, 0, 0, \
        &__x_.dwFrame, 1, 3, ctStop, 0, \
        0, 0xFFFFFFFF, &__x_, 0}; \
```

Declare and initialize a [DRASTIC_CHANNEL](#).
Definition at line 287 of file vvwtypes.h.

#define DFRAME_DATAAREA_RPDPXHEADER "DraStiCteCh-215"

This indicates that fi.dwVITCAux points to a data area and dwCCData contains its size / See below define DFRAME_TYPE_FI_PTR_DATA 0x00008000.
Definition at line 248 of file vvwtypes.h.

#define DFRAME_DATAAREA_RPDPXHEADER_SIZE 16

Definition at line 249 of file vvwtypes.h.

#define DFRAME_FIELD_INVERT 0x00200000

The fields in the frame are inverted (jaggies) see [DFRAME::dwType](#).
Definition at line 432 of file vvwtypes.h.

#define DFRAME_MAX_EXTRA_DATA_SIZE 1024

Normal size of the pFI_DataAreaPtr
Definition at line 243 of file vvwtypes.h.

#define DFRAME_NEW_FORMAT 0x08000000

This frame starts a new format that is different from the ones previous. Please get the new format and adjust before displaying see [DFRAME::dwType](#).
Definition at line 444 of file vvwtypes.h.

#define DFRAME_ORIENTATION_INVERT 0x00800000

The contents of the frame are inverted.
Definition at line 436 of file vvwtypes.h.

#define DFRAME_PROGRESSIVE 0x00100000

The contents of the frame are progressive (as opposed to interlaced) see [DFRAME::dwType](#).
Definition at line 430 of file vvwtypes.h.

#define DFRAME_SKIP_FRAME 0x40000000

This frame should be skipped (decoded, but not displayed) - Used to reach seek frame on a non key frame from key frame see [DFRAME::dwType](#).
Definition at line 452 of file vvwtypes.h.

#define DFRAME_TIME_INVERT 0x00400000

The fields in the frame are temporally inverts (jumps back and forth) see [DFRAME::dwType](#).
Definition at line 434 of file vvwtypes.h.

#define DFRAME_TYPE_AUDIO 0x00010000

This frame contains audio data see [DFRAME::dwType](#).
Definition at line 422 of file vvwtypes.h.

#define DFRAME_TYPE_EYES_ARE_FLIPPED 0x01000000

AUDIO: The frame contains a large chunk of audio (allows for optimization in AvHAL) see [DFRAME::dwType](#).

VIDEO 3D: If the codec (e.g. cineform) has flipped the eyes, don't reflip
Definition at line 440 of file vvwtypes.h.

#define DFRAME_TYPE_FI_PTR_DATA 0x00008000

This indicates that `FRAME_INFO::dwVITCAux` points to a data area and [FRAME_INFO::dwCCData](#) contains its size see them for more info see [DFRAME::dwType](#).
Definition at line 420 of file vvwtypes.h.

#define DFRAME_TYPE_INCLUDES_HEADER 0x00002000

This indicates the header is includes with the video data (e.g. Ari from Alexa) see [DFRAME::dwType](#).
Definition at line 418 of file vvwtypes.h.

#define DFRAME_TYPE_KEYFRAME 0x10000000

This frame is independant of other frames for decode see [DFRAME::dwType](#).
Definition at line 446 of file vvwtypes.h.

#define DFRAME_TYPE_KEYFRAME_B 0x20000000

This frame requires more than one frame to decode (for MPEG a B Frame) see [DFRAME::dwType](#).
Definition at line 450 of file vvwtypes.h.

#define DFRAME_TYPE_KEYFRAME_I 0x10000000

This frame is independant of other frames for decode (an MPEG I Frame) see [DFRAME::dwType](#).
Definition at line 448 of file vvwtypes.h.

#define DFRAME_TYPE_KEYFRAME_P 0x80000000

This frame requires previous keyframe(s) (for MPEG a P Frame) see [DFRAME::dwType](#).
Definition at line 454 of file vvwtypes.h.

#define DFRAME_TYPE_LEFT2ND_INVERT_HORIZ 0x00000800

3D Stereo, second frame/left eye inverted horizontally
Definition at line 414 of file vvwtypes.h.

#define DFRAME_TYPE_LEFT2ND_INVERT_VERT 0x00000400

3D Stereo, second frame/left eye inverted vertically
Definition at line 412 of file vvwtypes.h.

#define DFRAME_TYPE_LIVE_VIDEO 0x00040000

This frame contains video data from a different channels input source see [DFRAME::dwType](#).
Definition at line 426 of file vvwtypes.h.

#define DFRAME_TYPE_NOTPHYSHEAP 0x00001000

This indicates PhysHeap did NOT allocate the dframe (not pheap or local heap) see [DFRAME::dwType](#).
Definition at line 416 of file vvwtypes.h.

#define DFRAME_TYPE_PAUSE (0x00000004 | DFRAME_TYPE_PLAY)

This frame was acquired while in pause and is most likely a seek (normally by AvHAL) see [DFRAME::dwType](#).
Definition at line 386 of file vvwtypes.h.

#define DFRAME_TYPE_PLAY 0x00000002

This frame should be played out (normally by AvHAL) see [DFRAME::dwType](#).
Definition at line 384 of file vvwtypes.h.

#define DFRAME_TYPE_RECORD 0x00000001

This frame was recorded into (normally by AvHAL) see [DFRAME::dwType](#).
Definition at line 382 of file vvwtypes.h.

#define DFRAME_TYPE_RIGHT1ST_INVERT_HORIZ 0x00000200

3D Stereo, first frame/right eye inverted horizontally
Definition at line 410 of file vvwtypes.h.

#define DFRAME_TYPE_RIGHT1ST_INVERT_VERT 0x00000100

3D Stereo, first frame/right eye inverted vertically
Definition at line 408 of file vvwtypes.h.

#define DFRAME_TYPE_UNCOMPRESSED 0x00000008

This frame has already been decompressed. If avhal is in a compressed mode, by pass the decompression. see [DFRAME::dwType](#).

Definition at line 388 of file vvwtypes.h.

#define DFRAME_TYPE_UNCTYPE_ARRIALEXA 0x00000070

HD-SDI Arri Alexa bayer double buffer.

Definition at line 406 of file vvwtypes.h.

#define DFRAME_TYPE_UNCTYPE_ARRID21 0x00000060

HD-SDI Arri D21 bayer double buffer.

Definition at line 404 of file vvwtypes.h.

#define DFRAME_TYPE_UNCTYPE_BGR 0x00000030

Windows BGR 8 bit see [DFRAME::dwType](#).

Definition at line 398 of file vvwtypes.h.

#define DFRAME_TYPE_UNCTYPE_BGRA 0x00000040

Windows BGRA 8 bit see [DFRAME::dwType](#).

Definition at line 400 of file vvwtypes.h.

#define DFRAME_TYPE_UNCTYPE_DPX10 0x00000050

DPX 10 bit RGB see [DFRAME::dwType](#).

Definition at line 402 of file vvwtypes.h.

#define DFRAME_TYPE_UNCTYPE_MASK 0x000000F0

Uncompressed type mask, if DFRAME_TYPE_UNCOMPRESSED is true see [DFRAME::dwType](#).

Definition at line 390 of file vvwtypes.h.

#define DFRAME_TYPE_UNCTYPE_V210 0x00000020

Uncompressed V210 10 bit YCbCr see [DFRAME::dwType](#).

Definition at line 396 of file vvwtypes.h.

#define DFRAME_TYPE_UNCTYPE_YCBCR8 0x00000000

Uncompressed YCbCr 8 bit UYVY/yuv2 see [DFRAME::dwType](#).

Definition at line 392 of file vvwtypes.h.

#define DFRAME_TYPE_UNCTYPE_YUY2 0x00000010

Uncompressed YUY2 8 bit see [DFRAME::dwType](#).

Definition at line 394 of file vvwtypes.h.

#define DFRAME_TYPE_VIDEO 0x00020000

This frame contains video data see [DFRAME::dwType](#).

Definition at line 424 of file vvwtypes.h.

#define DPOSSIZENAME_4_1_AUDIO_FRAME 0x00000800

Definition at line 599 of file vvwtypes.h.

#define DPOSSIZENAME_5_1_AUDIO_FRAME 0x00001000

Definition at line 600 of file vvwtypes.h.

#define DPOSSIZENAME_7_1_AUDIO_FRAME 0x00002000

Definition at line 601 of file vvwtypes.h.

#define DPOSSIZENAME_9_1_AUDIO_FRAME 0x00004000

Definition at line 602 of file vvwtypes.h.

#define DPOSSIZENAME_AUD_16_16_BIT 0x00100000

This frame contains audio data see [DFRAME::dwType](#).

16 bit audio

Definition at line 607 of file vvwtypes.h.

#define DPOSSIZENAME_AUD_20_24_BIT 0x00200000

20 bit audio in 24

Definition at line 609 of file vvwtypes.h.

#define DPOSSIZENAME_AUD_24_24_BIT 0x00400000

24 bit audio in 24

Definition at line 611 of file vvwtypes.h.

#define DPOSSIZENAME_AUD_24_32_BIT 0x00800000

24/32 bit audio in 32

Definition at line 613 of file vvwtypes.h.

#define DPOSSIZENAME_AUD_32_32_BIT 0x01000000

32/32 bit audio in 32

Definition at line 615 of file vvwtypes.h.

#define DPOSSIZENAME_AUD_BIGENDIAN_BIT 0x00080000

Audio is big endian, else little endian.

Definition at line 617 of file vvwtypes.h.

#define DPOSSIZENAME_FRAME_MASK 0x0000FFFF

Definition at line 603 of file vvwtypes.h.

#define DPOSSIZENAME_MONO_AUDIO_FRAME 0x00000100

This is a mono audio chunk.

Definition at line 595 of file vvwtypes.h.

#define DPOSSIZENAME_QUAD_AUDIO_FRAME 0x00000400

Definition at line 598 of file vvwtypes.h.

#define DPOSSIZENAME_RECORDING 0x00000004

Is this file type currently recording.

Definition at line 593 of file vvwtypes.h.

#define DPOSSIZENAME_STEREO_AUDIO_FRAME 0x00000200

This is a stereo audio chunk.

Definition at line 597 of file vvwtypes.h.

#define DPOSSIZENAME_VIDEO_FRAME 0x00000001

This is a video frame.

Definition at line 591 of file vvwtypes.h.

#define DRFLAGS_CODECPRIVATE_DATA_AVI 0x01000000

Private data format is AVI see [VWVIDEO::dwDrFlags](#).

Definition at line 1181 of file vvwtypes.h.

#define DRFLAGS_CODECPRIVATE_DATA_MOV 0x02000000

Private data format is MOV see [VWVIDEO::dwDrFlags](#).

Definition at line 1183 of file vvwtypes.h.

#define DRFLAGS_CODECPRIVATE_DATA_OMF 0x04000000

Private data format is OMF see [VWVIDEO::dwDrFlags](#).

Definition at line 1185 of file vvwtypes.h.

#define DRFLAGS_FCC_MJPEG_DCx0 0x00010000

Stream is Miro DC50 MJPG see [VWVIDEO::dwDrFlags](#).

Definition at line 1165 of file vvwtypes.h.

#define DRFLAGS_FCC_MJPEG_DIGISUITE 0x00000000

Stream is DigiSuite MJPG see [VWVIDEO::dwDrFlags](#).

Definition at line 1163 of file vvwtypes.h.

#define DRFLAGS_FCC_MJPEG_DSEEDIT 0x00020000

Stream is DigiSuite Edit MJPG see [VWVIDEO::dwDrFlags](#).

Definition at line 1167 of file vvwtypes.h.

#define DRFLAGS_FCC_MJPEG_JFIF 0x00080000

Stream is JFIF jpeg see [VWVIDEO::dwDrFlags](#).

Definition at line 1171 of file vvwtypes.h.

#define DRFLAGS_FCC_MJPEG_JPGDIB 0x00040000

Stream is MJPG MS-Dib variant see [VWVIDEO::dwDrFlags](#).

Definition at line 1169 of file vvwtypes.h.

#define DRFLAGS_FCC_USE_ICM 0x00800000

Stream should use windows icm/vfw codecs see [VWVIDEO::dwDrFlags](#).

Definition at line 1177 of file vvwtypes.h.

#define DRFLAGS_FCC_USE_INTERN 0x00100000

Stream should use internal codecs see [VWVIDEO::dwDrFlags](#).

Definition at line 1173 of file vvwtypes.h.

#define DRFLAGS_FCC_USE_QT 0x00400000

Stream should use quicktime codecs see [VWVIDEO::dwDrFlags](#).

Definition at line 1175 of file vvwtypes.h.

#define DRFLAGS_FIRST_FIELD_DOMINANT 0x00000001

First Field Dominant in [VWVIDEO::dwDrFlags](#).

Definition at line 1143 of file vvwtypes.h.

#define DRFLAGS_HAS_KEYFRAMES 0x00000010

Stream has key frames, else all key frames [VWVIDEO::dwDrFlags](#).

Definition at line 1149 of file vvwtypes.h.

#define DRFLAGS_IS_COMPRESS 0x000001000

Is opening for a compression (0== decompression)

Definition at line 1159 of file vvwtypes.h.

#define DRFLAGS_NOT_QUICKCLIP 0x00000002

Preview to QuickclipXO.

Definition at line 1145 of file vvwtypes.h.

#define DRFLAGS_VTYPE_INTERLACED 0x00000100

Interlaced video frames.

Definition at line 1153 of file vvwtypes.h.

#define DRFLAGS_VTYPE_PROGRESSIVE 0x00000200

Progressive video frames.

Definition at line 1155 of file vvwtypes.h.

#define DRFLAGS_VTYPE_SEGMENTEDFRAME 0x00000400

Segmented Frame video frames.

Definition at line 1157 of file vvwtypes.h.

#define DRFLAGS_ZERO_FIELD_DOMINANT 0x00000000

Zero (Second) Field Dominant in [VWVIDEO::dwDrFlags](#).

Definition at line 1141 of file vvwtypes.h.

#define DT_BOTTOM_FIELD 0x20000000

Used in dwFlags - dup above, should condense at some point.

Definition at line 543 of file vvwtypes.h.

#define DT_CustomMetadataBlockType_CINE 0x00000001

Definition at line 2039 of file vvwtypes.h.

#define DT_CustomMetadataBlockType_DPX 0x00000002

Definition at line 2040 of file vvwtypes.h.

#define DT_CustomMetadataBlockType_ILLEGAL 0xFFFFFFFF

Definition at line 2041 of file vvwtypes.h.

#define DT_CustomMetadataBlockType_NONE 0x00000000

Definition at line 2042 of file vvwtypes.h.

#define DT_FRAME 0x30000000

Used in dwFlags - dup above, should condense at some point.

Definition at line 545 of file vvwtypes.h.

#define DT_META_DATA_CHANGED 0x01

Definition at line 3319 of file vvwtypes.h.

#define DT_TOP_FIELD 0x10000000

Used in dwFlags - dup above, should condense at some point.

Definition at line 541 of file vvwtypes.h.

#define dtstreamtypeAUDIO ((DWORD)(unsigned char)('a') | ((DWORD)(unsigned char)('u') << 8) | ((DWORD)(unsigned char)('d') << 16) | ((DWORD)(unsigned char)('s') << 24))

Definition at line 658 of file vvwtypes.h.

#define dtstreamtypeVIDEO ((DWORD)(unsigned char)('v') | ((DWORD)(unsigned char)('i') << 8) | ((DWORD)(unsigned char)('d') << 16) | ((DWORD)(unsigned char)('s') << 24))

Definition at line 657 of file vvwtypes.h.

#define DTVHDR_DONE 0x00000001

internal to AvHAL Set by the device driver to indicate it is finished with the data buffer and is returning the buffer to the client. see [DFRAME::dwFlags](#)

Definition at line 494 of file vvwtypes.h.

#define DTVHDR_INQUEUE 0x00000004

internal to AvHAL Set by the driver to indicate the buffer is in the driver's buffer queue. see [DFRAME::dwFlags](#)

Definition at line 498 of file vvwtypes.h.

#define DTVHDR_KEYFRAME 0x00000008

internal to AvHAL Set by the device driver to indicate a key frame. see [DFRAME::dwFlags](#)

Definition at line 500 of file vvwtypes.h.

#define DTVHDR_PREPARED 0x00000002

internal to AvHAL Indicates whether or not the buffer has been prepared for use. See DVM_STREAM_PREPAREHEADER. see [DFRAME::dwFlags](#)

Definition at line 496 of file vvwtypes.h.

#define DTVVW_PREVIEW 0x80000000

The universal preview flag - Top bit of a flag DWORD. This bit should not be used for other purposes as the whole system should know if we are in preview. Main places it is used:

intPreviewOpen(dwFlags) avOpen(dwFlags) mfOpen(dwOpenFlags) dtcodecOpen(fccFlags)
- I think

Definition at line 69 of file vvwtypes.h.

#define DTWAVE_FORMAT_EXTENSIBLE 0xFFFE

Definition at line 1296 of file vvwtypes.h.

#define DTWAVE_FORMAT_PCM 1

PCM Wave Type, see fccDef.h for other possible types.

Definition at line 1295 of file vvwtypes.h.

#define DTWAVEHDR_DATA

36 2 Size of the dwReserved area used. For PCM this will be zero. For other compressors, it may be anything < 256 DWORDs

The "data" subchunk contains the size of the data and the actual sound:

Definition at line 1509 of file vvwtypes.h.

#define DTWAVEHDR_FMT

The "WAVE" format consists of two subchunks: "fmt " and "data": The "fmt " subchunk describes the sound data's format:

Definition at line 1484 of file vvwtypes.h.

#define DTWAVEHDR_RIFF 0x46464952

The canonical WAVE format starts with the RIFF header:

Definition at line 1461 of file vvwtypes.h.

#define DTWAVEHDR_WAV 0x45564157

Definition at line 1475 of file vvwtypes.h.

#define DTWHDR_BEGINLOOP 0x00000004

internal to AvHAL This buffer is the first buffer in a loop. This flag is used only with output buffers. see [DFRAME::dwFlags](#)

Definition at line 507 of file vvwtypes.h.

#define DTWHDR_DONE 0x00000001

Set by the device driver to indicate that it is finished with the buffer and is returning it to the

application. see [DFRAME::dwFlags](#).

Definition at line 503 of file vvwtypes.h.

#define DTWHDR_ENDLOOP 0x00000008

internal to AvHAL This buffer is the last buffer in a loop. This flag is used only with output buffers. see [DFRAME::dwFlags](#)

Definition at line 509 of file vvwtypes.h.

#define DTWHDR_INQUEUE 0x00000010

internal to AvHAL Set by Windows to indicate that the buffer is queued for playback. see [DFRAME::dwFlags](#)

Definition at line 511 of file vvwtypes.h.

#define DTWHDR_PREPARED 0x00000002

internal to AvHAL Set by Windows to indicate that the buffer has been prepared with the waveInPrepareHeader or waveOutPrepareHeader function. see [DFRAME::dwFlags](#)

Definition at line 505 of file vvwtypes.h.

#define dwFI_DataAreaSize dwCCData

Alternate name for dwCCData.

Definition at line 238 of file vvwtypes.h.

#define FRAMEINFO_DATA_EIA608 1

Data is EIA-608B SD closed caption data.

Definition at line 184 of file vvwtypes.h.

#define FRAMEINFO_DATA_EIA708 2

Data is EIA-708 HD closed caption data.

Definition at line 186 of file vvwtypes.h.

#define FRAMEINFO_DATA_RP215_KLV 4

Data is RP-215 KLV data.

Definition at line 188 of file vvwtypes.h.

#define FRAMEINFO_DATA_VBI_FRAME_ELEM 8

Data is raw line with info per SMPTE 436 VBI Frame Element.

Definition at line 190 of file vvwtypes.h.

#define FRAMEINFO_MAX_DATA_SIZE 8192

Size of the arbData area.

Definition at line 198 of file vvwtypes.h.

#define GetAudio 0x00000000

Flag for mediafile/avhal to get audio dframe

Definition at line 3322 of file vvwtypes.h.

#define GetCTLTrack 0x00000200

Flag to get CTL even if VITC/LTC selected.

Definition at line 3346 of file vvwtypes.h.

#define GetCurrent 0x00000000

Flag for avhal to get the current [FRAME_INFO](#).

Definition at line 3336 of file vvwtypes.h.

#define GetCurrentField 0x00000020

Flag for avhal to get the last queued [FRAME_INFO](#).

Definition at line 3340 of file vvwtypes.h.

#define GetInf 0x00000002

Flag for mediafile/avhal to get info ([VWVWINFO](#))

Definition at line 3326 of file vvwtypes.h.

#define GetNoFail 0x00000100

Flag for getting second copy or other non queue critical frames.

Definition at line 3334 of file vvwtypes.h.

#define GetQueued 0x00000010

Flag for avhal to get the last queued [FRAME_INFO](#).

Definition at line 3338 of file vvwtypes.h.

#define GetVBIVideo 0x00000080

Flag for avhal to get the vbi video queued [FRAME_INFO](#).

Definition at line 3344 of file vvwtypes.h.

#define GetVideo 0x00000001

Flag for mediafile/avhal to get video dframe.

Definition at line 3324 of file vvwtypes.h.

#define INIT_DRASTIC_CHANNEL(__x_)

```
Value: { ZeroMemory(&__x_, sizeof(DRASTIC_CHANNEL)); \
        __x_.pdwFrame = &__x_.dwFrame; \
        __x_.ctCmd = ctStop; \
        __x_.pOwnerHandle = &__x_; };
```

Initialize a [DRASTIC_CHANNEL](#).

Definition at line 292 of file vvwtypes.h.

#define INIT_PDRASTIC_CHANNEL(__x_)

```
Value: { ZeroMemory(__x_, sizeof(DRASTIC_CHANNEL)); \
        __x_->pdwFrame = &__x_->dwFrame; \
        __x_->ctCmd = ctStop; \
        __x_->pOwnerHandle = &__x_; };
```

Initialize a memory area allocated as a [DRASTIC_CHANNEL](#).

Definition at line 298 of file vvwtypes.h.

#define IsSingleField 0x00000040

Flag for avhal to get the last queued [FRAME_INFO](#).

Definition at line 3342 of file vvwtypes.h.

#define MASK___DRFLAGS_FIELD 0x00000001

Field dominance MASK for [VWVIDEO::dwDrFlags](#).

Definition at line 1139 of file vvwtypes.h.

#define MASK___DRFLAGS_KEYFRAME 0x00000010

KeyFrame MASK for [VWVIDEO::dwDrFlags](#).

Definition at line 1147 of file vvwtypes.h.

#define MASK___DRFLAGS_VTYPE 0x00000700

Frame type mask (1=interlaced,2=progressive,4=segmentedframe)

Definition at line 1151 of file vvwtypes.h.

#define MASK__FCCMODIFIERS 0x00FF0000

Fourcc modifiers MASK for [VWVIDEO::dwDrFlags](#).

Definition at line 1161 of file vvwtypes.h.

#define MASK__PREVIEW 0xF0000000

MASK Room for current DTVWVW_PREVIEW and more if nec see [VWVIDEO::dwDrFlags](#).

Definition at line 1187 of file vvwtypes.h.

#define MASK__CODECPRIVATEDATA 0x0F000000

Private Data MASK see [VWVIDEO::dwDrFlags](#).

Definition at line 1179 of file vvwtypes.h.

#define MB_CODEC_DESC_LEN 255

Maximum length for the full (long) codec description.

Definition at line 56 of file vvwtypes.h.

#define MB_CODEC_SHORT_LEN 64

Maximum length for the short codec name.

Definition at line 54 of file vvwtypes.h.

#define MB_FILE_DESC_LEN 255

Maximum length for the long description.

Definition at line 52 of file vvwtypes.h.

#define MB_FILE_EXT_LEN 200

String return sizes (mediabase/mediafile plugin)

Maximum length for the supported file extensions string (fmt: *.xxx;*.yyy;*.ttt)

Definition at line 48 of file vvwtypes.h.

#define MB_FILE_SHORT_LEN 64

Maximum length for the base or short description (used in drop down)

Definition at line 50 of file vvwtypes.h.

#define pFI_DataAreaPtr dwVITCAux

Alternate name for dwVITCAux.
Definition at line 225 of file vvwtypes.h.

#define PutAudio GetAudio

Flag for mediafile/avhal to put audio dframe.
Definition at line 3328 of file vvwtypes.h.

#define PutInf GetInf

Flag for mediafile/avhal to get info ([VWVINFO](#))
Definition at line 3332 of file vvwtypes.h.

#define PutVideo GetVideo

Flag for mediafile/avhal to put audio dframe.
Definition at line 3330 of file vvwtypes.h.

#define SIZEOFDFRAME (((sizeof([DFRAME](#)) >> 2) + 1) << 2)

The DWORD aligned size of a [DFRAME](#), used for more effecient memory allocations.
Definition at line 571 of file vvwtypes.h.

#define VVW_ABS_MAX_FRAME 10692000

Absolute maximum timecode in a time code space (99:59:59:29 NDF) - Actual max is usually 23:59:59:x9 depending on type (see [tc2Maximum\(\)](#))
Definition at line 78 of file vvwtypes.h.

#define VVW_INVALID_FRAME 0x80000001

No frame exists - is the same as TC_ILLEGAL - careful, only valid when tc cannot go negative, else it means -1.
Definition at line 75 of file vvwtypes.h.

#define VVWAUD_RECALC(__pvvwaud_)

```
Value: {
    \
    if((__pvvwaud_)->wFormatTag == DTWAVE\_FORMAT\_PCM) {
        \
        (__pvvwaud_)->nBlockAlign = ((__pvvwaud_)->nChannels * ((__pvvwaud_)-
>wBitsPerSample + 7) >> 3);
        \
        (__pvvwaud_)->nAvgBytesPerSec = (__pvvwaud_)->nBlockAlign *
(__pvvwaud_)->nSamplesPerSec;
        \
        (__pvvwaud_)->dwSuggestedBufferSize = (__pvvwaud_)->nAvgBytesPerSec >>
1;
    \
}
```

```

        (__pvvwaud_)->dwSampleSize = (__pvvwaud_)->nBlockAlign; \
        (__pvvwaud_)->dwScale = (__pvvwaud_)->nBlockAlign; \
        (__pvvwaud_)->dwRate = (__pvvwaud_)->nAvgBytesPerSec; \
    } \
}

```

Recalculate audio structure: Uses [VVWAUDIO::nChannels](#), [VVWAUDIO::wBitsPerSample](#) and [VVWAUDIO::nSamplesPerSec](#) to calculate the [VVWAUDIO::nBlockAlign](#), [VVWAUDIO::nAvgBytesPerSec](#), [VVWAUDIO::dwSuggestedBufferSize](#), [VVWAUDIO::dwSampleSize](#), [VVWAUDIO::dwScale](#) and [VVWAUDIO::dwRate](#) members - Only valid of PCM uncompressed audio streams. NOTE: nBlockAlign = nChannels * ((wBitsPerSample + 7) >> 3) gives the minimum container size for the bit size. In our write and internal cases, we always use 32 bits for 20 and 24. This calc returns 24 for 20 and 24. We need this for the read side, so make sure any write size stuff is caught.

Definition at line 1415 of file vvtypes.h.

```

#define VVWAUD_SET( __pvvwaud_, _formattag, _channels, _samplespersec, _bitspersample, _samples)

```

```

    Value: { \
        (__pvvwaud_)->wFormatTag = _formattag; \
        (__pvvwaud_)->fccHandler = _formattag; \
        (__pvvwaud_)->nChannels = _channels; \
        (__pvvwaud_)->nSamplesPerSec = _samplespersec; \
        (__pvvwaud_)->wBitsPerSample = _bitspersample; \
        (__pvvwaud_)->dwLength = _samples; \
        VVWAUD_RECALC((__pvvwaud_)); \
    }

```

Set the [VVWAUDIO](#) structure pointer from formattag, channels, samples per sec, bits per sample and number of samples.

Definition at line 1426 of file vvtypes.h.

```

#define VVWAUD_SETCLR( __pvvwaud_, _formattag, _channels, _samplespersec, _bitspersample, _samples)

```

```

    Value: { \
        VVWAUD_SET((__pvvwaud_), _formattag, _channels, _samplespersec, \
        _bitspersample, _samples); \
        ZeroMemory((__pvvwaud_)->dwReserved, \_VVWXXX\_RESERVED\_SIZE); \
        (__pvvwaud_)->fccType = dstreamtypeAUDIO; \
        (__pvvwaud_)->dwFlags = 0; \
        (__pvvwaud_)->dwCaps = 0; \
        (__pvvwaud_)->wPriority = 0; \
        (__pvvwaud_)->wLanguage = 0; \
        (__pvvwaud_)->dwStart = 0; \
        (__pvvwaud_)->dwInitialFrames = 0; \
        (__pvvwaud_)->dwQuality = 0xFFFFFFFF; \
        (__pvvwaud_)->dwEditCount = 0; \
        (__pvvwaud_)->dwFormatChangeCount = 0; \
        ZeroMemory((__pvvwaud_)->szName, \_VVWXXX\_NAME\_SIZE); \
        (__pvvwaud_)->dwDrFlags = 0; \
        (__pvvwaud_)->dwResDrastic = 0; \
    }

```

Clean and set the [VVWAUDIO](#) structure pointer from formattag, channels, samples per sec, bits per sample and number of samples.

Definition at line 1436 of file vvtypes.h.

```

#define VVWINF_CURVETYPE_LINEAR 1

```

Definition at line 1603 of file vvtypes.h.

#define VVWINF_CURVETYPE_LOG 2

Definition at line 1604 of file vvwtypes.h.

#define VVWINF_CURVETYPE_UNKNOWN 0

dwCurveType

Definition at line 1602 of file vvwtypes.h.

#define VVWINF_INVALID (-1)

Invalid setting, ignore it.

Definition at line 1600 of file vvwtypes.h.

#define VVWINFO_TOTAL_ITEMS (vwiEND_OF_STRINGS + (vwiEND_OF_DWORD_V2 - vwiNumericStart))

Definition at line 2036 of file vvwtypes.h.

#define VVWINFODESC_dwAFrame "A-Frame"

Definition at line 2991 of file vvwtypes.h.

#define VVWINFODESC_dwAspectRatio "AspectRatio"

Definition at line 2999 of file vvwtypes.h.

#define VVWINFODESC_dwBlackGain "BlackGain"

Definition at line 3121 of file vvwtypes.h.

#define VVWINFODESC_dwBlackPoint "BlackPoint"

Definition at line 3113 of file vvwtypes.h.

#define VVWINFODESC_dwBreakPoint "BreakPoint"

Definition at line 3129 of file vvwtypes.h.

#define VVWINFODESC_dwConversions "TotalConversions"

Definition at line 3023 of file vvwtypes.h.

#define VVWINFODESC_dwCustomMetadataBlockSize "CustomMetadataBlockSize"

Definition at line 3204 of file vvwtypes.h.

#define VVWINFODESC_dwCustomMetadataBlockType "CustomMetadataBlockType"

Definition at line 3197 of file vvwtypes.h.

#define VVWINFODESC_dwFileDate "FileDate"

Definition at line 3048 of file vvwtypes.h.

#define VVWINFODESC_dwFileSize "FileSize"

Definition at line 3039 of file vvwtypes.h.

#define VVWINFODESC_dwFileTime "FileTime"

Definition at line 3057 of file vvwtypes.h.

#define VVWINFODESC_dwFilmManufacturerCode "FilmManufacturerCode"

Definition at line 3089 of file vvwtypes.h.

#define VVWINFODESC_dwFilmTypeCode "FilmTypeCode"

Definition at line 3097 of file vvwtypes.h.

#define VVWINFODESC_dwFlags "Flags"

Definition at line 3151 of file vvwtypes.h.

#define VVWINFODESC_dwGamma1000 "Gamma1000"

Definition at line 3137 of file vvwtypes.h.

#define VVWINFODESC_dwLatitude "Latitude"

Definition at line 3218 of file vvwtypes.h.

#define VVWINFODESC_dwLongitude "Longitude"

Definition at line 3225 of file vvwtypes.h.

#define VVWINFODESC_dwLTCTimeCodeType "LTCTimeCodeType"

Definition at line 3167 of file vvwtypes.h.

#define VVWINFODESC_dwNorthSouthEastWest "NorthSouthEastWest"

Definition at line 3211 of file vvwtypes.h.

#define VVWINFODESC_dwOriginalRate "OriginalRate"

Definition at line 3007 of file vvwtypes.h.

#define VVWINFODESC_dwOriginalScale "OriginalScale"

Definition at line 3015 of file vvwtypes.h.

#define VVWINFODESC_dwPosterFrame "PosterFrame"

Definition at line 2983 of file vvwtypes.h.

#define VVWINFODESC_dwProdDate "ProdDate"

Definition at line 3182 of file vvwtypes.h.

#define VVWINFODESC_dwSequenceNumber "SequenceNumber"

Definition at line 3065 of file vvwtypes.h.

#define VVWINFODESC_dwTagNumber "TagNumber"

Definition at line 3144 of file vvwtypes.h.

#define VVWINFODESC_dwTimeCode "TimeCode"

Definition at line 2943 of file vvwtypes.h.

#define VVWINFODESC_dwTimeCodeType "TimeCodeType"

Definition at line 3159 of file vvwtypes.h.

#define VVWINFODESC_dwTotalLength "TotalLength"

Definition at line 3081 of file vvwtypes.h.

#define VVWINFODESC_dwTotalStreams "TotalStreams"

Definition at line 3073 of file vvwtypes.h.

#define VVWINFODESC_dwUniqueID "UniqueID"

Definition at line 3190 of file vvwtypes.h.

#define VVWINFODESC_dwUserBits "UserBits"

Definition at line 2951 of file vvwtypes.h.

#define VVWINFODESC_dwVersionNumber "VersionNumber"

Definition at line 3031 of file vvwtypes.h.

#define VVWINFODESC_dwVITCLine3 "VITCExtraData"

Definition at line 2975 of file vvwtypes.h.

#define VVWINFODESC_dwVITCTimeCode "VITCTimeCode"

Definition at line 2959 of file vvwtypes.h.

#define VVWINFODESC_dwVITCTimeCodeType "VITCTimeCodeType"

Definition at line 3175 of file vvwtypes.h.

#define VVWINFODESC_dwVITCUserBits "VITCUserBits"

Definition at line 2967 of file vvwtypes.h.

#define VVWINFODESC_dwWhitePoint "WhitePoint"

Definition at line 3105 of file vvwtypes.h.

#define VVWINFODESC_ImageInfo "ImageInfo"

Definition at line 2803 of file vvwtypes.h.

#define VVWINFODESC_szAlbum "Album"

Definition at line 2356 of file vvwtypes.h.

#define VVWINFODESC_szArtist "Artist"

Definition at line 2364 of file vvwtypes.h.

#define VVWINFODESC_szAuthor "Author"

Definition at line 2332 of file vvwtypes.h.

#define VVWINFODESC_szCamRoll "CamRoll"

Definition at line 2747 of file vvwtypes.h.

#define VVWINFODESC_szCategory "Category"

Definition at line 2238 of file vvwtypes.h.

#define VVWINFODESC_szChannel "ChannelIdentifier"

Definition at line 2198 of file vvwtypes.h.

#define VVWINFODESC_szChannelDescription "ChannelDescription"

Definition at line 2214 of file vvwtypes.h.

#define VVWINFODESC_szChannelName "ChannelName"

Definition at line 2206 of file vvwtypes.h.

#define VVWINFODESC_szComments "Comments"

Definition at line 2262 of file vvwtypes.h.

#define VVWINFODESC_szComposer "Composer"

Definition at line 2372 of file vvwtypes.h.

#define VVWINFODESC_szCopyright "Copyright"

Definition at line 2381 of file vvwtypes.h.

#define VVWINFODESC_szCreationData "CreationData"

Definition at line 2389 of file vvwtypes.h.

#define VVWINFODESC_szCustomMetadataBlockPointer "CustomMetadataBlockPointer"

Definition at line 2796 of file vvwtypes.h.

#define VVWINFODESC_szDailyRoll "DailyRoll"

Definition at line 2740 of file vvwtypes.h.

#define VVWINFODESC_szDescription "Description"

Definition at line 2397 of file vvwtypes.h.

#define VVWINFODESC_szDeviceModelNum "DeviceModelNum"

Definition at line 2671 of file vvwtypes.h.

#define VVWINFODESC_szDeviceSerialNum "DeviceSerialNum"

Definition at line 2679 of file vvwtypes.h.

#define VVWINFODESC_szDirector "Director"

Definition at line 2405 of file vvwtypes.h.

#define VVWINFODESC_szDisclaimer "Disclaimer"

Definition at line 2413 of file vvwtypes.h.

#define VVWINFODESC_szDoNotUse "DoNotUse"

Definition at line 2271 of file vvwtypes.h.

#define VVWINFODESC_szEditData1 "EditData1"

Definition at line 2559 of file vvwtypes.h.

#define VVWINFODESC_szEditData2 "EditData2"

Definition at line 2567 of file vvwtypes.h.

#define VVWINFODESC_szEditData3 "EditData3"

Definition at line 2575 of file vvwtypes.h.

#define VVWINFODESC_szEditData4 "EditData4"

Definition at line 2583 of file vvwtypes.h.

#define VVWINFODESC_szEditData5 "EditData5"

Definition at line 2591 of file vvwtypes.h.

#define VVWINFODESC_szEditData6 "EditData6"

Definition at line 2599 of file vvwtypes.h.

#define VVWINFODESC_szEditData7 "EditData7"

Definition at line 2607 of file vvwtypes.h.

#define VVWINFODESC_szEditData8 "EditData8"

Definition at line 2615 of file vvwtypes.h.

#define VVWINFODESC_szEditData9 "EditData9"

Definition at line 2623 of file vvwtypes.h.

#define VVWINFODESC_szEditor "Editor"

Definition at line 2289 of file vvwtypes.h.

#define VVWINFODESC_szEncodedBy "EncodedBy"

Definition at line 2422 of file vvwtypes.h.

#define VVWINFODESC_szEpisode "Episode"

Definition at line 2726 of file vvwtypes.h.

#define VVWINFODESC_szFileName "FileName"

Definition at line 2152 of file vvwtypes.h.

#define VVWINFODESC_szFormat "Format"

Definition at line 2655 of file vvwtypes.h.

#define VVWINFODESC_szFrameAttribute "FrameAttribute"

Definition at line 2719 of file vvwtypes.h.

#define VVWINFODESC_szFullName "FullName"

Definition at line 2430 of file vvwtypes.h.

#define VVWINFODESC_szGenre "Genre"

Definition at line 2438 of file vvwtypes.h.

#define VVWINFODESC_szHostComputer "HostComputer"

Definition at line 2446 of file vvwtypes.h.

#define VVWINFODESC_szInformation "Information"

Definition at line 2454 of file vvwtypes.h.

#define VVWINFODESC_szInkNumberPrefix "InkNumberPrefix"

Definition at line 2775 of file vvwtypes.h.

#define VVWINFODESC_szInputDevice "InputDevice"

Definition at line 2663 of file vvwtypes.h.

#define VVWINFODESC_szIP "TCP-IPAddress"

Definition at line 2180 of file vvwtypes.h.

#define VVWINFODESC_szKeyNumberPrefix "KeyNumberPrefix"

Definition at line 2768 of file vvwtypes.h.

#define VVWINFODESC_szKeywords "Keywords"

Definition at line 2246 of file vvwtypes.h.

#define VVWINFODESC_szLabRoll "LabRoll"

Definition at line 2761 of file vvwtypes.h.

#define VVWINFODESC_szLanguage "Language"

Definition at line 2647 of file vvwtypes.h.

#define VVWINFODESC_szMake "Make"

Definition at line 2462 of file vvwtypes.h.

#define VVWINFODESC_szManufacturer "Manufacturer"

Definition at line 2639 of file vvwtypes.h.

#define VVWINFODESC_szModel "Model"

Definition at line 2470 of file vvwtypes.h.

#define VVWINFODESC_szNativeLocator "NativeLocator"

Definition at line 2162 of file vvwtypes.h.

#define VVWINFODESC_szOriginalArtist "OriginalArtist"

Definition at line 2478 of file vvwtypes.h.

#define VVWINFODESC_szOriginalFormat "OriginalFormat"

Definition at line 2486 of file vvwtypes.h.

#define VVWINFODESC_szOwner "Owner"

Definition at line 2280 of file vvwtypes.h.

#define VVWINFODESC_szPerformers "Performers"

Definition at line 2494 of file vvwtypes.h.

#define VVWINFODESC_szPictureIcon "PictureIcon"

Definition at line 2782 of file vvwtypes.h.

#define VVWINFODESC_szProduced "Produced"

Definition at line 2348 of file vvwtypes.h.

#define VVWINFODESC_szProducer "Producer"

Definition at line 2502 of file vvwtypes.h.

#define VVWINFODESC_szProduct "Product"

Definition at line 2510 of file vvwtypes.h.

#define VVWINFODESC_szProject "Project"

Definition at line 2315 of file vvwtypes.h.

#define VVWINFODESC_szProxyFile "ProxyFile"

Definition at line 2789 of file vvwtypes.h.

#define VVWINFODESC_szRatings "Ratings"

Definition at line 2254 of file vvwtypes.h.

#define VVWINFODESC_szReel "Reel"

Definition at line 2687 of file vvwtypes.h.

#define VVWINFODESC_szRevisionNumber "RevisionNumber"

Definition at line 2340 of file vvwtypes.h.

#define VVWINFODESC_szScene "Scene"

Definition at line 2733 of file vvwtypes.h.

#define VVWINFODESC_szShot "Shot"

Definition at line 2695 of file vvwtypes.h.

#define VVWINFODESC_szSlateInfo "SlateInfo"

Definition at line 2711 of file vvwtypes.h.

#define VVWINFODESC_szSoftware "Software"

Definition at line 2518 of file vvwtypes.h.

#define VVWINFODESC_szSoundRoll "SoundRoll"

Definition at line 2754 of file vvwtypes.h.

#define VVWINFODESC_szSource "Source"

Definition at line 2306 of file vvwtypes.h.

#define VVWINFODESC_szSourceLocator "SourceLocator"

Definition at line 2189 of file vvwtypes.h.

**#define VVWINFODESC_szSpecialPlaybackRequirements
"SpecialPlaybackRequirements"**

Definition at line 2526 of file vvwtypes.h.

#define VVWINFODESC_szStatus "Status"

Definition at line 2323 of file vvwtypes.h.

#define VVWINFODESC_szSubject "Subject"

Definition at line 2230 of file vvwtypes.h.

#define VVWINFODESC_szSupplier "Supplier"

Definition at line 2298 of file vvwtypes.h.

#define VVWINFODESC_szTake "Take"

Definition at line 2703 of file vvwtypes.h.

#define VVWINFODESC_szTitle "Title"

Definition at line 2222 of file vvwtypes.h.

#define VVWINFODESC_szTrack "Track"

Definition at line 2534 of file vvwtypes.h.

#define VVWINFODESC_szUniversalName "UniversalLocator"

Definition at line 2171 of file vvwtypes.h.

#define VVWINFODESC_szURLLink "URL"

Definition at line 2551 of file vvwtypes.h.

#define VVWINFODESC_szVersionString "VersionString"

Definition at line 2631 of file vvwtypes.h.

#define VVWINFODESC_szWarning "Warning"

Definition at line 2542 of file vvwtypes.h.

#define VVWINFODESC_UMID "UMID"

Definition at line 2817 of file vvwtypes.h.

#define VVWINFODESC_woAudioBits "AudioBits"

Definition at line 2028 of file vvwtypes.h.

#define VVWINFODESC_woAudioChannels "AudioChannels"

Definition at line 2018 of file vvwtypes.h.

#define VVWINFODESC_woAudioFrequency "AudioFrequency"

Definition at line 2023 of file vvwtypes.h.

#define VVWINFODESC_woAudioType "AudioType"

Definition at line 2013 of file vvwtypes.h.

#define VVWINFODESC_woVideoBitCount "BitCount"

Definition at line 1873 of file vvwtypes.h.

#define VVWINFODESC_woVideoCaps "Caps"

Definition at line 1928 of file vvwtypes.h.

#define VVWINFODESC_woVideoCIrImportant "CIrImportant"

Definition at line 1903 of file vvwtypes.h.

#define VVWINFODESC_woVideoClrUsed "ClrUsed"

Definition at line 1898 of file vvwtypes.h.

#define VVWINFODESC_woVideoCompression "Compression"

Definition at line 1878 of file vvwtypes.h.

#define VVWINFODESC_woVideoDrFlags "DrFlags"

Definition at line 1998 of file vvwtypes.h.

#define VVWINFODESC_woVideoEditCount "EditCount"

Definition at line 1983 of file vvwtypes.h.

#define VVWINFODESC_woVideoFccHandler "FccHandler"

Definition at line 1918 of file vvwtypes.h.

#define VVWINFODESC_woVideoFccType "FccType"

Definition at line 1913 of file vvwtypes.h.

#define VVWINFODESC_woVideoFileType "FileType"

Definition at line 2003 of file vvwtypes.h.

#define VVWINFODESC_woVideoFlags "Flags"

Definition at line 1923 of file vvwtypes.h.

#define VVWINFODESC_woVideoFormatChangeCount "FormatChangeCount"

Definition at line 1988 of file vvwtypes.h.

#define VVWINFODESC_woVideoHeight "Height"

Definition at line 1863 of file vvwtypes.h.

#define VVWINFODESC_woVideoInitialFrames "InitialFrames"

Definition at line 1963 of file vvwtypes.h.

#define VVWINFODESC_woVideoLanguage "Language"

Definition at line 1938 of file vvwtypes.h.

#define VVWINFODESC_woVideoLength "Length"

Definition at line 1958 of file vvwtypes.h.

#define VVWINFODESC_woVideoPitch "Pitch"

Definition at line 1993 of file vvwtypes.h.

#define VVWINFODESC_woVideoPlanes "Planes"

Definition at line 1868 of file vvwtypes.h.

#define VVWINFODESC_woVideoPriority "Priority"

Definition at line 1933 of file vvwtypes.h.

#define VVWINFODESC_woVideoQuality "Quality"

Definition at line 1973 of file vvwtypes.h.

#define VVWINFODESC_woVideoRate "Rate"

Definition at line 1948 of file vvwtypes.h.

#define VVWINFODESC_woVideoResDrastic "ResDrastic"

Definition at line 2008 of file vvwtypes.h.

#define VVWINFODESC_woVideoReserved "Reserved"

Definition at line 1908 of file vvwtypes.h.

#define VVWINFODESC_woVideoSampleSize "SampleSize"

Definition at line 1978 of file vvwtypes.h.

#define VVWINFODESC_woVideoScale "Scale"

Definition at line 1943 of file vvwtypes.h.

#define VVWINFODESC_woVideoSizelImage "SizelImage"

Definition at line 1883 of file vvwtypes.h.

#define VVWINFODESC_woVideoStart "Start"

Definition at line 1953 of file vvwtypes.h.

#define VVWINFODESC_woVideoSuggestedBufferSize "SuggestedBufferSize"

Definition at line 1968 of file vvwtypes.h.

#define VVWINFODESC_woVideoWidth "Width"

Definition at line 1858 of file vvwtypes.h.

#define VVWINFODESC_woVideoXPelsPerMeter "XPelsPerMeter"

Definition at line 1888 of file vvwtypes.h.

#define VVWINFODESC_woVideoYPelsPerMeter "YPelsPerMeter"

Definition at line 1893 of file vvwtypes.h.

#define VVWINFOTAG_dwAFrame "A-Frame"

XML tag name for [VVWINFO::dwAFrame](#).

Definition at line 2990 of file vvwtypes.h.

#define VVWINFOTAG_dwAspectRatio "AspectRatio"

XML tag name for [VVWINFO::dwAspectRatio](#).

Definition at line 2998 of file vvwtypes.h.

#define VVWINFOTAG_dwBlackGain "BlackGain"

XML tag name for [VVWINFO::dwBlackGain](#).

Definition at line 3120 of file vvwtypes.h.

#define VVWINFOTAG_dwBlackPoint "BlackPoint"

XML tag name for [VVWINFO::dwBlackPoint](#).

Definition at line 3112 of file vvwtypes.h.

#define VVWINFO_TAG_dwBreakPoint "BreakPoint"

XML tag name for [VVWINFO::dwBreakPoint](#).

Definition at line 3128 of file vvwtypes.h.

#define VVWINFO_TAG_dwConversions "TotalConversions"

XML tag name for VVWINFO::dwTotalConversions.

Definition at line 3022 of file vvwtypes.h.

#define VVWINFO_TAG_dwCustomMetadataBlockSize "CustomMetadataBlockSize"

XML tag name for [VVWINFO::dwCustomMetadataBlockSize](#).

Definition at line 3203 of file vvwtypes.h.

#define VVWINFO_TAG_dwCustomMetadataBlockType "CustomMetadataBlockType"

XML tag name for [VVWINFO::dwCustomMetadataBlockType](#).

Definition at line 3196 of file vvwtypes.h.

#define VVWINFO_TAG_dwFileDate "FileDate"

XML tag name for [VVWINFO::dwFileDate](#).

Definition at line 3047 of file vvwtypes.h.

#define VVWINFO_TAG_dwFileSize "FileSize"

XML tag name for [VVWINFO::dwFileSize](#).

Definition at line 3038 of file vvwtypes.h.

#define VVWINFO_TAG_dwFileTime "FileTime"

XML tag name for [VVWINFO::dwFileTime](#).

Definition at line 3056 of file vvwtypes.h.

#define VVWINFO_TAG_dwFilmManufacturerCode "FilmManufacturerCode"

XML tag name for [VVWINFO::dwFilmManufacturerCode](#).

Definition at line 3088 of file vvwtypes.h.

#define VVWINFO_TAG_dwFilmTypeCode "FilmTypeCode"

XML tag name for [VWINFO::dwFilmTypeCode](#).

Definition at line 3096 of file vwtypes.h.

#define VWINFO_TAG_dwFlags "Flags"

XML tag name for [VWINFO::dwFlags](#).

Definition at line 3150 of file vwtypes.h.

#define VWINFO_TAG_dwGamma1000 "Gamma1000"

XML tag name for [VWINFO::dwGamma1000](#).

Definition at line 3136 of file vwtypes.h.

#define VWINFO_TAG_dwLatitude "Latitude"

XML tag name for [VWINFO::dwLatitude](#).

Definition at line 3217 of file vwtypes.h.

#define VWINFO_TAG_dwLongitude "Longitude"

XML tag name for [VWINFO::dwLongitude](#).

Definition at line 3224 of file vwtypes.h.

#define VWINFO_TAG_dwLTCTimeCodeType "LTCTimeCodeType"

XML tag name for [VWINFO::dwLTCTimeCodeType](#).

Definition at line 3166 of file vwtypes.h.

#define VWINFO_TAG_dwNorthSouthEastWest "NorthSouthEastWest"

XML tag name for [VWINFO::dwNorthSouthEastWest](#).

Definition at line 3210 of file vwtypes.h.

#define VWINFO_TAG_dwNU_38 "NumericNotUsed_38"

XML tag name for [VWINFO::dwNU_38](#).

Definition at line 3231 of file vwtypes.h.

#define VWINFO_TAG_dwNU_39 "NumericNotUsed_39"

XML tag name for [VWINFO::dwNU_39](#).

Definition at line 3238 of file vwtypes.h.

#define VVWINFOTAG_dwNU_40 "NumericNotUsed_40"

XML tag name for [VVWINFO::dwNU_40](#).

Definition at line 3245 of file vvwtypes.h.

#define VVWINFOTAG_dwNU_41 "NumericNotUsed_41"

XML tag name for [VVWINFO::dwNU_41](#).

Definition at line 3252 of file vvwtypes.h.

#define VVWINFOTAG_dwNU_42 "NumericNotUsed_42"

XML tag name for [VVWINFO::dwNU_42](#).

Definition at line 3259 of file vvwtypes.h.

#define VVWINFOTAG_dwNU_43 "NumericNotUsed_43"

XML tag name for [VVWINFO::dwNU_43](#).

Definition at line 3266 of file vvwtypes.h.

#define VVWINFOTAG_dwNU_44 "NumericNotUsed_44"

XML tag name for [VVWINFO::dwNU_44](#).

Definition at line 3273 of file vvwtypes.h.

#define VVWINFOTAG_dwNU_45 "NumericNotUsed_45"

XML tag name for [VVWINFO::dwNU_45](#).

Definition at line 3280 of file vvwtypes.h.

#define VVWINFOTAG_dwNU_46 "NumericNotUsed_46"

XML tag name for [VVWINFO::dwNU_46](#).

Definition at line 3287 of file vvwtypes.h.

#define VVWINFOTAG_dwNU_47 "NumericNotUsed_47"

XML tag name for [VVWINFO::dwNU_47](#).

Definition at line 3294 of file vvwtypes.h.

#define VVWINFOTAG_dwNU_48 "NumericNotUsed_48"

XML tag name for [VWINFO::dwNU_48](#).

Definition at line 3301 of file vwtypes.h.

#define VWINFO_TAG_dwNU_49 "NumericNotUsed_49"

XML tag name for [VWINFO::dwNU_49](#).

Definition at line 3308 of file vwtypes.h.

#define VWINFO_TAG_dwOriginalRate "OriginalRate"

XML tag name for [VWINFO::dwOriginalRate](#).

Definition at line 3006 of file vwtypes.h.

#define VWINFO_TAG_dwOriginalScale "OriginalScale"

XML tag name for [VWINFO::dwOriginalScale](#).

Definition at line 3014 of file vwtypes.h.

#define VWINFO_TAG_dwPosterFrame "PosterFrame"

XML tag name for [VWINFO::dwPosterFrame](#).

Definition at line 2982 of file vwtypes.h.

#define VWINFO_TAG_dwProdDate "ProdDate"

XML tag name for [VWINFO::dwProdDate](#).

Definition at line 3181 of file vwtypes.h.

#define VWINFO_TAG_dwSequenceNumber "SequenceNumber"

XML tag name for [VWINFO::dwSequenceNumber](#).

Definition at line 3064 of file vwtypes.h.

#define VWINFO_TAG_dwTagNumber "TagNumber"

XML tag name for [VWINFO::dwTagNumber](#).

Definition at line 3143 of file vwtypes.h.

#define VWINFO_TAG_dwTimeCode "TimeCode"

XML tag name for [VWINFO::dwTimeCode](#).

Definition at line 2942 of file vwtypes.h.

#define VVWINFOTAG_dwTimeCodeType "TimeCodeType"

XML tag name for [VVWINFO::dwTimeCodeType](#).

Definition at line 3158 of file vvwtypes.h.

#define VVWINFOTAG_dwTotalLength "TotalLength"

XML tag name for [VVWINFO::dwTotalLength](#).

Definition at line 3080 of file vvwtypes.h.

#define VVWINFOTAG_dwTotalStreams "TotalStreams"

XML tag name for [VVWINFO::dwTotalStreams](#).

Definition at line 3072 of file vvwtypes.h.

#define VVWINFOTAG_dwUniqueID "UniqueID"

XML tag name for [VVWINFO::dwUniqueID](#).

Definition at line 3189 of file vvwtypes.h.

#define VVWINFOTAG_dwUserBits "UserBits"

XML tag name for [VVWINFO::dwUserBits](#).

Definition at line 2950 of file vvwtypes.h.

#define VVWINFOTAG_dwVersionNumber "VersionNumber"

XML tag name for [VVWINFO::dwVersionNumber](#).

Definition at line 3030 of file vvwtypes.h.

#define VVWINFOTAG_dwVITCLine3 "VITCExtraData"

XML tag name for [VVWINFO::dwVICTLine3](#).

Definition at line 2974 of file vvwtypes.h.

#define VVWINFOTAG_dwVITCTimeCode "VITCTimeCode"

XML tag name for [VVWINFO::dwVITCTimeCode](#).

Definition at line 2958 of file vvwtypes.h.

#define VVWINFOTAG_dwVITCTimeCodeType "VITCTimeCodeType"

XML tag name for [VWINFO::dwVITCTimeCodeType](#).

Definition at line 3174 of file vwtypes.h.

#define VWINFO_TAG_dwVITCUserBits "VITCUserBits"

XML tag name for [VWINFO::dwVITCUserBits](#).

Definition at line 2966 of file vwtypes.h.

#define VWINFO_TAG_dwWhitePoint "WhitePoint"

XML tag name for [VWINFO::dwWhitePoint](#).

Definition at line 3104 of file vwtypes.h.

#define VWINFO_TAG_szAlbum "Album"

XML tag name for [VWINFO::szAlbum](#).

Definition at line 2355 of file vwtypes.h.

#define VWINFO_TAG_szArtist "Artist"

XML tag name for [VWINFO::szArtist](#).

Definition at line 2363 of file vwtypes.h.

#define VWINFO_TAG_szAuthor "Author"

XML tag name for [VWINFO::szAuthor](#).

Definition at line 2331 of file vwtypes.h.

#define VWINFO_TAG_szCamRoll "CamRoll"

XML tag name for [VWINFO::szCamRoll](#).

Definition at line 2746 of file vwtypes.h.

#define VWINFO_TAG_szCategory "Category"

XML tag name for [VWINFO::szCategory](#).

Definition at line 2237 of file vwtypes.h.

#define VWINFO_TAG_szChannel "ChannelIdentifier"

XML tag name for [VWINFO::szChannelIdentifier](#).

Definition at line 2197 of file vwtypes.h.

#define VVWINFO::szChannelDescription "ChannelDescription"

XML tag name for [VVWINFO::szChannelDescription](#).

Definition at line 2213 of file vvwtypes.h.

#define VVWINFO::szChannelName "ChannelName"

XML tag name for [VVWINFO::szChannelName](#).

Definition at line 2205 of file vvwtypes.h.

#define VVWINFO::szComments "Comments"

XML tag name for [VVWINFO::szComments](#).

Definition at line 2261 of file vvwtypes.h.

#define VVWINFO::szComposer "Composer"

XML tag name for [VVWINFO::szComposer](#).

Definition at line 2371 of file vvwtypes.h.

#define VVWINFO::szCopyright "Copyright"

XML tag name for [VVWINFO::szCopyright](#).

Definition at line 2380 of file vvwtypes.h.

#define VVWINFO::szCreationData "CreationData"

XML tag name for [VVWINFO::szCreationData](#).

Definition at line 2388 of file vvwtypes.h.

#define VVWINFO::szCustomMetadataBlockPointer "CustomMetadataBlockPointer"

XML tag name for [VVWINFO::szCustomMetadataBlockPointer](#).

Definition at line 2795 of file vvwtypes.h.

#define VVWINFO::szDailyRoll "DailyRoll"

XML tag name for [VVWINFO::szDailyRoll](#).

Definition at line 2739 of file vvwtypes.h.

#define VVWINFO::szDescription "Description"

XML tag name for [VVWINFO::szDescription](#).

Definition at line 2396 of file vvwtypes.h.

#define VVWINFOTAG_szDeviceModelNum "DeviceModelNum"

XML tag name for [VVWINFO::szDeviceModelNum](#).

Definition at line 2670 of file vvwtypes.h.

#define VVWINFOTAG_szDeviceSerialNum "DeviceSerialNum"

XML tag name for [VVWINFO::szDeviceSerialNum](#).

Definition at line 2678 of file vvwtypes.h.

#define VVWINFOTAG_szDirector "Director"

XML tag name for [VVWINFO::szDirector](#).

Definition at line 2404 of file vvwtypes.h.

#define VVWINFOTAG_szDisclaimer "Disclaimer"

XML tag name for [VVWINFO::szDisclaimer](#).

Definition at line 2412 of file vvwtypes.h.

#define VVWINFOTAG_szDoNotUse "DoNotUse"

XML tag name for [VVWINFO::szDoNotUse](#).

Definition at line 2270 of file vvwtypes.h.

#define VVWINFOTAG_szEditData1 "EditData1"

XML tag name for [VVWINFO::szEditData1](#).

Definition at line 2558 of file vvwtypes.h.

#define VVWINFOTAG_szEditData2 "EditData2"

XML tag name for [VVWINFO::szEditData2](#).

Definition at line 2566 of file vvwtypes.h.

#define VVWINFOTAG_szEditData3 "EditData3"

XML tag name for [VVWINFO::szEditData3](#).

Definition at line 2574 of file vvwtypes.h.

#define VVWINFOTAG_szEditData4 "EditData4"

XML tag name for [VVWINFO::szEditData4](#).

Definition at line 2582 of file vvwtypes.h.

#define VVWINFOTAG_szEditData5 "EditData5"

XML tag name for [VVWINFO::szEditData5](#).

Definition at line 2590 of file vvwtypes.h.

#define VVWINFOTAG_szEditData6 "EditData6"

XML tag name for [VVWINFO::szEditData6](#).

Definition at line 2598 of file vvwtypes.h.

#define VVWINFOTAG_szEditData7 "EditData7"

XML tag name for [VVWINFO::szEditData7](#).

Definition at line 2606 of file vvwtypes.h.

#define VVWINFOTAG_szEditData8 "EditData8"

XML tag name for [VVWINFO::szEditData8](#).

Definition at line 2614 of file vvwtypes.h.

#define VVWINFOTAG_szEditData9 "EditData9"

XML tag name for [VVWINFO::szEditData9](#).

Definition at line 2622 of file vvwtypes.h.

#define VVWINFOTAG_szEditor "Editor"

XML tag name for [VVWINFO::szEditor](#).

Definition at line 2288 of file vvwtypes.h.

#define VVWINFOTAG_szEncodedBy "EncodedBy"

XML tag name for [VVWINFO::szEncodedBy](#).

Definition at line 2421 of file vvwtypes.h.

#define VVWINFOTAG_szEpisode "Episode"

XML tag name for [VWINFO::szEpisode](#).

Definition at line 2725 of file vwtypes.h.

#define VWINFOTAG_szFileName "FileName"

XML tag name for [VWINFO::szFileName](#).

Definition at line 2151 of file vwtypes.h.

#define VWINFOTAG_szFormat "Format"

XML tag name for [VWINFO::szFormat](#).

Definition at line 2654 of file vwtypes.h.

#define VWINFOTAG_szFrameAttribute "FrameAttribute"

XML tag name for [VWINFO::szFrameAttribute](#).

Definition at line 2718 of file vwtypes.h.

#define VWINFOTAG_szFullName "FullName"

XML tag name for [VWINFO::szFullName](#).

Definition at line 2429 of file vwtypes.h.

#define VWINFOTAG_szGenre "Genre"

XML tag name for [VWINFO::szGenre](#).

Definition at line 2437 of file vwtypes.h.

#define VWINFOTAG_szHostComputer "HostComputer"

XML tag name for [VWINFO::szHostComputer](#).

Definition at line 2445 of file vwtypes.h.

#define VWINFOTAG_szImageInfo "ImageInfo"

XML tag name for [VWINFO::szImageInfo](#).

Definition at line 2802 of file vwtypes.h.

#define VWINFOTAG_szInformation "Information"

XML tag name for [VWINFO::szInformation](#).

Definition at line 2453 of file vwtypes.h.

#define VVWINFOTAG_szInkNumberPrefix "InkNumberPrefix"

XML tag name for [VVWINFO::szInkNumberPrefix](#).

Definition at line 2774 of file vvwtypes.h.

#define VVWINFOTAG_szInputDevice "InputDevice"

XML tag name for [VVWINFO::szInputDevice](#).

Definition at line 2662 of file vvwtypes.h.

#define VVWINFOTAG_szIP "TCP-IPAddress"

XML tag name for [VVWINFO::szIP](#).

Definition at line 2179 of file vvwtypes.h.

#define VVWINFOTAG_szKeyNumberPrefix "KeyNumberPrefix"

XML tag name for [VVWINFO::szKeyNumberPrefix](#).

Definition at line 2767 of file vvwtypes.h.

#define VVWINFOTAG_szKeywords "Keywords"

XML tag name for [VVWINFO::szKeywords](#).

Definition at line 2245 of file vvwtypes.h.

#define VVWINFOTAG_szLabRoll "LabRoll"

XML tag name for [VVWINFO::szLabRoll](#).

Definition at line 2760 of file vvwtypes.h.

#define VVWINFOTAG_szLanguage "Language"

XML tag name for [VVWINFO::szLanguage](#).

Definition at line 2646 of file vvwtypes.h.

#define VVWINFOTAG_szMake "Make"

XML tag name for [VVWINFO::szMake](#).

Definition at line 2461 of file vvwtypes.h.

#define VVWINFOTAG_szManufacturer "Manufacturer"

XML tag name for [VWINFO::szManufacturer](#).

Definition at line 2638 of file vwtypes.h.

#define VWINFO_TAG_szModel "Model"

XML tag name for [VWINFO::szModel](#).

Definition at line 2469 of file vwtypes.h.

#define VWINFO_TAG_szNativeLocator "NativeLocator"

XML tag name for [VWINFO::szNativeLocator](#).

Definition at line 2161 of file vwtypes.h.

#define VWINFO_TAG_szNU_84 "NotUsed_84"

XML tag name for [VWINFO::szNU_84](#).

Definition at line 2823 of file vwtypes.h.

#define VWINFO_TAG_szNU_85 "NotUsed_85"

XML tag name for [VWINFO::szNU_85](#).

Definition at line 2830 of file vwtypes.h.

#define VWINFO_TAG_szNU_86 "NotUsed_86"

XML tag name for [VWINFO::szNU_86](#).

Definition at line 2837 of file vwtypes.h.

#define VWINFO_TAG_szNU_87 "NotUsed_87"

XML tag name for [VWINFO::szNU_87](#).

Definition at line 2844 of file vwtypes.h.

#define VWINFO_TAG_szNU_88 "NotUsed_88"

XML tag name for [VWINFO::szNU_88](#).

Definition at line 2851 of file vwtypes.h.

#define VWINFO_TAG_szNU_89 "NotUsed_89"

XML tag name for [VWINFO::szNU_89](#).

Definition at line 2858 of file vwtypes.h.

#define VVWINFOTAG_szNU_90 "NotUsed_90"

XML tag name for [VVWINFO::szNU_90](#).

Definition at line 2865 of file vvwtypes.h.

#define VVWINFOTAG_szNU_91 "NotUsed_91"

XML tag name for [VVWINFO::szNU_91](#).

Definition at line 2872 of file vvwtypes.h.

#define VVWINFOTAG_szNU_92 "NotUsed_92"

XML tag name for [VVWINFO::szNU_92](#).

Definition at line 2879 of file vvwtypes.h.

#define VVWINFOTAG_szNU_93 "NotUsed_93"

XML tag name for [VVWINFO::szNU_93](#).

Definition at line 2886 of file vvwtypes.h.

#define VVWINFOTAG_szNU_94 "NotUsed_94"

XML tag name for [VVWINFO::szNU_94](#).

Definition at line 2893 of file vvwtypes.h.

#define VVWINFOTAG_szNU_95 "NotUsed_95"

XML tag name for [VVWINFO::szNU_95](#).

Definition at line 2900 of file vvwtypes.h.

#define VVWINFOTAG_szNU_96 "NotUsed_96"

XML tag name for [VVWINFO::szNU_96](#).

Definition at line 2907 of file vvwtypes.h.

#define VVWINFOTAG_szNU_97 "NotUsed_97"

XML tag name for [VVWINFO::szNU_97](#).

Definition at line 2914 of file vvwtypes.h.

#define VVWINFOTAG_szNU_98 "NotUsed_98"

XML tag name for [VWINFO::szNU_98](#).

Definition at line 2921 of file vwtypes.h.

#define VWINFOTAG_szNU_99 "NotUsed_99"

XML tag name for [VWINFO::szNU_99](#).

Definition at line 2928 of file vwtypes.h.

#define VWINFOTAG_szOriginalArtist "OriginalArtist"

XML tag name for [VWINFO::szOriginalArtist](#).

Definition at line 2477 of file vwtypes.h.

#define VWINFOTAG_szOriginalFormat "OriginalFormat"

XML tag name for [VWINFO::szOriginalFormat](#).

Definition at line 2485 of file vwtypes.h.

#define VWINFOTAG_szOwner "Owner"

XML tag name for [VWINFO::szOwner](#).

Definition at line 2279 of file vwtypes.h.

#define VWINFOTAG_szPerformers "Performers"

XML tag name for [VWINFO::szPerformers](#).

Definition at line 2493 of file vwtypes.h.

#define VWINFOTAG_szPictureIcon "PictureIcon"

XML tag name for [VWINFO::szPictureIcon](#).

Definition at line 2781 of file vwtypes.h.

#define VWINFOTAG_szProduced "Produced"

XML tag name for [VWINFO::szProduced](#).

Definition at line 2347 of file vwtypes.h.

#define VWINFOTAG_szProducer "Producer"

XML tag name for [VWINFO::szProducer](#).

Definition at line 2501 of file vwtypes.h.

#define VVWINFOTAG_szProduct "Product"

XML tag name for [VVWINFO::szProduct](#).

Definition at line 2509 of file vvwtypes.h.

#define VVWINFOTAG_szProject "Project"

XML tag name for [VVWINFO::szProject](#).

Definition at line 2314 of file vvwtypes.h.

#define VVWINFOTAG_szProxyFile "ProxyFile"

XML tag name for [VVWINFO::szProxyFile](#).

Definition at line 2788 of file vvwtypes.h.

#define VVWINFOTAG_szRatings "Ratings"

XML tag name for [VVWINFO::szRatings](#).

Definition at line 2253 of file vvwtypes.h.

#define VVWINFOTAG_szReel "Reel"

XML tag name for [VVWINFO::szReel](#).

Definition at line 2686 of file vvwtypes.h.

#define VVWINFOTAG_szRevisionNumber "RevisionNumber"

XML tag name for [VVWINFO::szRevisionNumber](#).

Definition at line 2339 of file vvwtypes.h.

#define VVWINFOTAG_szScene "Scene"

XML tag name for [VVWINFO::szScene](#).

Definition at line 2732 of file vvwtypes.h.

#define VVWINFOTAG_szShot "Shot"

XML tag name for [VVWINFO::szShot](#).

Definition at line 2694 of file vvwtypes.h.

#define VVWINFOTAG_szSlateInfo "SlateInfo"

XML tag name for [VWINFO::szSlateInfo](#).

Definition at line 2710 of file vwtypes.h.

#define VWINFOTAG_szSoftware "Software"

XML tag name for [VWINFO::szSoftware](#).

Definition at line 2517 of file vwtypes.h.

#define VWINFOTAG_szSoundRoll "SoundRoll"

XML tag name for [VWINFO::szSoundRoll](#).

Definition at line 2753 of file vwtypes.h.

#define VWINFOTAG_szSource "Source"

XML tag name for [VWINFO::szSource](#).

Definition at line 2305 of file vwtypes.h.

#define VWINFOTAG_szSourceLocator "SourceLocator"

XML tag name for [VWINFO::szSourceLocator](#).

Definition at line 2188 of file vwtypes.h.

#define VWINFOTAG_szSpecialPlaybackRequirements "SpecialPlaybackRequirements"

XML tag name for [VWINFO::szSpecialPlaybackRequirements](#).

Definition at line 2525 of file vwtypes.h.

#define VWINFOTAG_szStatus "Status"

XML tag name for [VWINFO::szStatus](#).

Definition at line 2322 of file vwtypes.h.

#define VWINFOTAG_szSubject "Subject"

XML tag name for [VWINFO::szSubject](#).

Definition at line 2229 of file vwtypes.h.

#define VWINFOTAG_szSupplier "Supplier"

XML tag name for [VWINFO::szSupplier](#).

Definition at line 2297 of file vwtypes.h.

#define VVWINFOTAG_szTake "Take"

XML tag name for [VVWINFO::szTake](#).

Definition at line 2702 of file vvwtypes.h.

#define VVWINFOTAG_szTitle "Title"

XML tag name for [VVWINFO::szTitle](#).

Definition at line 2221 of file vvwtypes.h.

#define VVWINFOTAG_szTrack "Track"

XML tag name for [VVWINFO::szTrack](#).

Definition at line 2533 of file vvwtypes.h.

#define VVWINFOTAG_szUMID "UMID"

XML tag name for [VVWINFO::szUMID](#).

Definition at line 2816 of file vvwtypes.h.

#define VVWINFOTAG_szUniversalName "UniversalLocator"

XML tag name for [VVWINFO::szUniversalLocator](#).

Definition at line 2170 of file vvwtypes.h.

#define VVWINFOTAG_szURLLink "URL"

XML tag name for [VVWINFO::szURLLink](#).

Definition at line 2550 of file vvwtypes.h.

#define VVWINFOTAG_szVersionString "VersionString"

XML tag name for [VVWINFO::szVersionString](#).

Definition at line 2630 of file vvwtypes.h.

#define VVWINFOTAG_szWarning "Warning"

XML tag name for [VVWINFO::szWarning](#).

Definition at line 2541 of file vvwtypes.h.

#define VVWINFOTAG_woAudioBits "AudioBits"

XML tag.

Definition at line 2027 of file vvwtypes.h.

#define VVWINFOTAG_woAudioChannels "AudioChannels"

XML tag.

Definition at line 2017 of file vvwtypes.h.

#define VVWINFOTAG_woAudioFrequency "AudioFrequency"

XML tag.

Definition at line 2022 of file vvwtypes.h.

#define VVWINFOTAG_woAudioType "AudioType"

XML tag.

Definition at line 2012 of file vvwtypes.h.

#define VVWINFOTAG_woVideoBitCount "BitCount"

XML tag name for bit count.

Definition at line 1872 of file vvwtypes.h.

#define VVWINFOTAG_woVideoCaps "Caps"

XML tag name for capabilities.

Definition at line 1927 of file vvwtypes.h.

#define VVWINFOTAG_woVideoClrImportant "ClrImportant"

XML tag name for.

Definition at line 1902 of file vvwtypes.h.

#define VVWINFOTAG_woVideoClrUsed "ClrUsed"

XML tag name for color elements used.

Definition at line 1897 of file vvwtypes.h.

#define VVWINFOTAG_woVideoCompression "Compression"

XML tag name for compression (fourcc)

Definition at line 1877 of file vvwtypes.h.

#define VVWINFOTAG_woVideoDrFlags "DrFlags"

XML tag name for internal drastic flags.

Definition at line 1997 of file vvwtypes.h.

#define VVWINFOTAG_woVideoEditCount "EditCount"

XML tag name for number of edits done on this file.

Definition at line 1982 of file vvwtypes.h.

#define VVWINFOTAG_woVideoFccHandler "FccHandler"

XML tag name for four cc handler.

Definition at line 1917 of file vvwtypes.h.

#define VVWINFOTAG_woVideoFccType "FccType"

XML tag name for four cc type (video/audio)

Definition at line 1912 of file vvwtypes.h.

#define VVWINFOTAG_woVideoFileType "FileType"

XML tag name for drastic 'mft' file type.

Definition at line 2002 of file vvwtypes.h.

#define VVWINFOTAG_woVideoFlags "Flags"

XML tag name for flags.

Definition at line 1922 of file vvwtypes.h.

#define VVWINFOTAG_woVideoFormatChangeCount "FormatChangeCount"

XML tag name for number of format changes.

Definition at line 1987 of file vvwtypes.h.

#define VVWINFOTAG_woVideoHeight "Height"

XML tag name for height.

Definition at line 1862 of file vvwtypes.h.

#define VVWINFOTAG_woVideoInitialFrames "InitialFrames"

XML tag name for number of initial frames to load.

Definition at line 1962 of file vvwtypes.h.

#define VVWINFOTAG_woVideoLanguage "Language"

XML tag name for language.

Definition at line 1937 of file vvwtypes.h.

#define VVWINFOTAG_woVideoLength "Length"

XML tag name for the length in frames.

Definition at line 1957 of file vvwtypes.h.

#define VVWINFOTAG_woVideoPitch "Pitch"

XML tag name for video line pitch.

Definition at line 1992 of file vvwtypes.h.

#define VVWINFOTAG_woVideoPlanes "Planes"

XML tag name for planes.

Definition at line 1867 of file vvwtypes.h.

#define VVWINFOTAG_woVideoPriority "Priority"

XML tag name for priority.

Definition at line 1932 of file vvwtypes.h.

#define VVWINFOTAG_woVideoQuality "Quality"

XML tag name for quality.

Definition at line 1972 of file vvwtypes.h.

#define VVWINFOTAG_woVideoRate "Rate"

XML tag name for rate (fps = rate / scale)

Definition at line 1947 of file vvwtypes.h.

#define VVWINFOTAG_woVideoResDrastic "ResDrastic"

XML tag name for reserved drastic array of DWORDs.

Definition at line 2007 of file vvwtypes.h.

#define VVWINFOTAG_woVideoReserved "Reserved"

XML tag name for reserved array.

Definition at line 1907 of file vvwtypes.h.

#define VVWINFOTAG_woVideoSampleSize "SampleSize"

XML tag name for recommended sample size.

Definition at line 1977 of file vvwtypes.h.

#define VVWINFOTAG_woVideoScale "Scale"

XML tag name for scale (fps = rate / scale)

Definition at line 1942 of file vvwtypes.h.

#define VVWINFOTAG_woVideoSizeImage "SizeImage"

XML tag name for size of the image in unsigned chars.

Definition at line 1882 of file vvwtypes.h.

#define VVWINFOTAG_woVideoStart "Start"

XML tag name for start frame.

Definition at line 1952 of file vvwtypes.h.

#define VVWINFOTAG_woVideoSuggestedBufferSize "SuggestedBufferSize"

XML tag name for suggested maximum buffer size.

Definition at line 1967 of file vvwtypes.h.

#define VVWINFOTAG_woVideoWidth "Width"

XML tag name for width.

Definition at line 1857 of file vvwtypes.h.

#define VVWINFOTAG_woVideoXPelsPerMeter "XPelsPerMeter"

XML tag name for X pels per meter.

Definition at line 1887 of file vvwtypes.h.

#define VVWINFOTAG_woVideoYPelsPerMeter "YPelsPerMeter"

XML tag name for Y pels per meter.

Definition at line 1892 of file vvwtypes.h.

```
#define VVWSYS_CLR( __pvwsys_)
```

```
Value: {
    \
    (__pvwsys_)->dwFlags = 0;          \
    (__pvwsys_)->dwInitialFrames = 0;  \
    (__pvwsys_)->dwReserved[0] = 0;    \
    (__pvwsys_)->dwReserved[1] = 0;    \
    (__pvwsys_)->dwReserved[2] = 0;    \
    (__pvwsys_)->dwReserved[3] = 0;    \
    (__pvwsys_)->dwCaps = 0;          \
    (__pvwsys_)->dwEditCount = 0;     \
    (__pvwsys_)->dwType = 0;          \
    (__pvwsys_)->dwMfCaps = 0;        \
    (__pvwsys_)->dwVidStandard = 0;    \
    (__pvwsys_)->dwDrFlags = 0;       \
    (__pvwsys_)->dwResDrastic = 0;    \
}
```

Clear a important 0 of a [VWVIDEO](#) structure pointer.

Definition at line 905 of file vvwtypes.h.

```
#define VVWSYS_SET( __pvwsys_, __pvwvid_, __pvwaud_, _granularity, _frames)
```

```
Value: {
    \
    (__pvwsys_)->dwMicroSecPerFrame = (DWORD)(1000000.0 / ((double)(__pvwvid_)-
>dwRate / (double)(__pvwvid_)->dwScale)); \
    (__pvwsys_)->dwMaxBytesPerSec = ((__pvwvid_)->biSizeImage * (__pvwvid_)-
>dwScale / (__pvwvid_)->dwRate) + (__pvwaud_)->nAvgBytesPerSec; \
    (__pvwsys_)->dwPaddingGranularity = _granularity; \
    (__pvwsys_)->dwTotalFrames = _frames; \
    (__pvwsys_)->dwSuggestedBufferSize = (__pvwvid_)->dwSuggestedBufferSize +
(__pvwaud_)->dwSuggestedBufferSize; \
    (__pvwsys_)->dwWidth = (__pvwvid_)->biWidth; \
    if((__pvwvid_)->biHeight < 0) (__pvwsys_)->dwHeight = -(LONG)(__pvwvid_)-
>biHeight; else (__pvwsys_)->dwHeight = (__pvwvid_)->biHeight; \
    (__pvwsys_)->dwScale = (__pvwvid_)->dwScale; \
    (__pvwsys_)->dwRate = (__pvwvid_)->dwRate; \
    (__pvwsys_)->dwLength= (__pvwvid_)->dwLength; \
}
```

Set a [VWSYSTEM](#) structure pointer from [VWVIDEO](#) pointer, [VWAUDIO](#) pointer, a granularity size and number of frames.

Definition at line 866 of file vvwtypes.h.

```
#define VVWSYS_SETAUDONLY( __pvwsys_, __pvwaud_, _granularity, _frames)
```

```
Value: {
    \
    (__pvwsys_)->dwMicroSecPerFrame = (DWORD)(1000000.0 / ((double)(__pvwaud_)-
>dwRate / (double)(__pvwaud_)->dwScale)); \
    (__pvwsys_)->dwMaxBytesPerSec = (__pvwaud_)->nAvgBytesPerSec; \
    (__pvwsys_)->dwPaddingGranularity = _granularity; \
    (__pvwsys_)->dwTotalFrames = _frames; \
    (__pvwsys_)->dwSuggestedBufferSize = (__pvwaud_)->dwSuggestedBufferSize;
\
    (__pvwsys_)->dwWidth = 0;          \
    (__pvwsys_)->dwHeight = 0;         \
    (__pvwsys_)->dwScale = (__pvwaud_)->dwScale; \
    (__pvwsys_)->dwRate = (__pvwaud_)->dwRate; \
    (__pvwsys_)->dwLength= (__pvwaud_)->dwLength; \
}
```

Set a [VWSYSTEM](#) structure pointer from [VWAUDIO](#) pointer, a granularity size and number of frames.

Definition at line 892 of file vvwtypes.h.

```

#define VVWSYS_SETVIDONLY( __pvvwsys_, __pvvwvid_, _granularity, _frames)
    Value: {
        \
        (__pvvwsys_)->dwMicroSecPerFrame = (DWORD)(1000000.0 / ((double)(__pvvwvid_)-
>dwRate / (double)(__pvvwvid_)->dwScale)); \
        (__pvvwsys_)->dwMaxBytesPerSec = ((__pvvwvid_)->biSizeImage * (__pvvwvid_)-
>dwScale / (__pvvwvid_)->dwRate); \
        (__pvvwsys_)->dwPaddingGranularity = _granularity; \
        (__pvvwsys_)->dwTotalFrames = _frames; \
        (__pvvwsys_)->dwSuggestedBufferSize = (__pvvwvid_)->dwSuggestedBufferSize;
    \
        (__pvvwsys_)->dwWidth = (__pvvwvid_)->biWidth; \
        if((__pvvwvid_)->biHeight < 0) (__pvvwsys_)->dwHeight = -(LONG)(__pvvwvid_)-
>biHeight; else (__pvvwsys_)->dwHeight = (__pvvwvid_)->biHeight; \
        (__pvvwsys_)->dwScale = (__pvvwvid_)->dwScale; \
        (__pvvwsys_)->dwRate = (__pvvwvid_)->dwRate; \
        (__pvvwsys_)->dwLength= (__pvvwvid_)->dwLength; \
    }

```

Set a [VVWSYSTEM](#) structure pointer from [VWVIDEO](#) pointer, a granularity size and number of frames.

Definition at line 879 of file vvwtypes.h.

```

#define VVWVID_PITCHALIGN1( __pvvwvid_ ) { (__pvvwvid_)->biPitch = (((__pvvwvid_)-
>biWidth * (__pvvwvid_)->biBitCount+ 7) / 8); }

```

Set [VWVIDEO::biPitch](#) member to 1 byte alignment based on [VWVIDEO::biWidth](#) and [VWVIDEO::biBitCount](#).

Definition at line 1207 of file vvwtypes.h.

```

#define VVWVID_PITCHALIGN16( __pvvwvid_ ) { (__pvvwvid_)->biPitch = (((__pvvwvid_)-
>biWidth * (__pvvwvid_)->biBitCount+127) / 128) * 16; }

```

Set [VWVIDEO::biPitch](#) member to 16 byte alignment based on [VWVIDEO::biWidth](#) and [VWVIDEO::biBitCount](#).

Definition at line 1213 of file vvwtypes.h.

```

#define VVWVID_PITCHALIGN4( __pvvwvid_ ) { (__pvvwvid_)->biPitch = (((__pvvwvid_)-
>biWidth * (__pvvwvid_)->biBitCount+ 31) / 32) * 4; }

```

Set [VWVIDEO::biPitch](#) member to 4 byte (DWORD) alignment based on [VWVIDEO::biWidth](#) and [VWVIDEO::biBitCount](#).

Definition at line 1209 of file vvwtypes.h.

```

#define VVWVID_PITCHALIGN8( __pvvwvid_ ) { (__pvvwvid_)->biPitch = (((__pvvwvid_)-
>biWidth * (__pvvwvid_)->biBitCount+ 63) / 64) * 8; }

```

Set [VWVIDEO::biPitch](#) member to 8 byte alignment based on [VWVIDEO::biWidth](#) and [VWVIDEO::biBitCount](#).

Definition at line 1211 of file vvwtypes.h.

```
#define VVWVID_PITCHALIGNANY( __pvwvid_, _align_1_4_8_16) { (__pvwvid_)-
>biPitch = (((__pvwvid_)->biWidth * (__pvwvid_)->biBitCount+((_align_1_4_8_16 * 8) -
1)) / (_align_1_4_8_16 * 8)) * _align_1_4_8_16; }
```

Set [VVWVIDEO::biPitch](#) member to alignment specified based on [VVWVIDEO::biWidth](#) and [VVWVIDEO::biBitCount](#).

Definition at line 1215 of file vvwtypes.h.

```
#define VVWVID_PITCHMODULO( __pvwvid_) ((__pvwvid_)->biPitch - ((__pvwvid_)-
>biWidth * (__pvwvid_)->biBitCount + 7) / 8)
```

Set pitch to 1-0 alignment ?

Definition at line 1217 of file vvwtypes.h.

```
#define VVWVID_RATESCALE_FILM( __pvwvid_) { (__pvwvid_)->dwScale = 1;
(__pvwvid_)->dwRate = 24; }
```

Set the [VVWVIDEO::dwRate](#) and [VVWVIDEO::dwScale](#) for 24/1 (FILM)

Definition at line 1229 of file vvwtypes.h.

```
#define VVWVID_RATESCALE_NTSC( __pvwvid_) { (__pvwvid_)->dwScale = 100;
(__pvwvid_)->dwRate = 2997; }
```

Set the [VVWVIDEO::dwRate](#) and [VVWVIDEO::dwScale](#) for 2997/100 (NTSC)

Definition at line 1225 of file vvwtypes.h.

```
#define VVWVID_RATESCALE_NTSC_AVI( __pvwvid_) { (__pvwvid_)->dwScale = 1001;
(__pvwvid_)->dwRate = 30000; }
```

Set the [VVWVIDEO::dwRate](#) and [VVWVIDEO::dwScale](#) for 30000/1001 (NTSC)

Definition at line 1223 of file vvwtypes.h.

```
#define VVWVID_RATESCALE_PAL( __pvwvid_) { (__pvwvid_)->dwScale = 1;
(__pvwvid_)->dwRate = 25; }
```

Set the [VVWVIDEO::dwRate](#) and [VVWVIDEO::dwScale](#) for 25/1 (PAL)

Definition at line 1227 of file vvwtypes.h.

```
#define VVWVID_SET( __pvwvid_, _bitcount, _width, _height, _pitch_align_1_4_8_16,
_fcccodec, _scale, _rate, _frames)
```

```
Value: {
    (__pvwvid_)->biBitCount = _bitcount; \
    if((LONG)_height < 0) { (__pvwvid_)->biHeight = -(LONG)_height; } else
{ (__pvwvid_)->biHeight = (LONG)_height; } \
    (__pvwvid_)->biWidth = _width; \
    (__pvwvid_)->biCompression = _fcccodec; \
    VVWVID_PITCHALIGNANY(__pvwvid_, _pitch_align_1_4_8_16); \
    VVWVID_SIZEIMAGE((__pvwvid_)); \
```

```

VWVVID_SUGGESTEDBUFFERSIZE((__pvvwwid_); \
(__pvvwwid_)->dwScale = _scale; (__pvvwwid_)->dwRate = _rate; \
VWVWXXX_SETSAMPLETOLENGTH((__pvvwwid_), _frames); \
(__pvvwwid_)->fccHandler = (__pvvwwid_)->biCompression; \
(__pvvwwid_)->rcFrame.top = 0; \
(__pvvwwid_)->rcFrame.bottom = abs((__pvvwwid_)->biHeight); \
(__pvvwwid_)->rcFrame.left = 0; \
(__pvvwwid_)->rcFrame.right = (__pvvwwid_)->biWidth; \
}

```

Basic setup of a [VWVVIDEO](#) pointer from width, height, align, codec, scale, rate and length in frames.

Definition at line 1231 of file vvwtypes.h.

```

#define VWVVID_SETCLR( __pvvwwid_, _bitcount, _width, _height,
_pitch_align_1_4_8_16, _fcccodec, _scale, _rate, _frames)

```

```

Value:{ \
    VWVVID_SET((__pvvwwid_), _bitcount, _width, _height, _pitch_align_1_4_8_16, \
    _fcccodec, _scale, _rate, _frames); \
    (__pvvwwid_)->biSize = sizeof(BITMAPINFOHEADER); \
    (__pvvwwid_)->biPlanes = 1; \
    (__pvvwwid_)->biXPelsPerMeter = 0; \
    (__pvvwwid_)->biYPelsPerMeter = 0; \
    (__pvvwwid_)->biClrUsed = 0; \
    (__pvvwwid_)->biClrImportant = 0; \
    ZeroMemory((__pvvwwid_)->dwReserved, VWVWXXX\_RESERVED\_SIZE); \
    (__pvvwwid_)->fccType = dtstreamtypeVIDEO; \
    (__pvvwwid_)->dwFlags = 0; \
    (__pvvwwid_)->dwCaps = 0; \
    (__pvvwwid_)->wPriority = 0; \
    (__pvvwwid_)->wLanguage = 0; \
    (__pvvwwid_)->dwStart = 0; \
    (__pvvwwid_)->dwInitialFrames = 0; \
    (__pvvwwid_)->dwQuality = 0xFFFFFFFF; \
    (__pvvwwid_)->dwSampleSize = 0; \
    (__pvvwwid_)->dwEditCount = 0; \
    (__pvvwwid_)->dwFormatChangeCount = 0; \
    ZeroMemory((__pvvwwid_)->szName, VWVWXXX\_NAME\_SIZE); \
    (__pvvwwid_)->dwDrFlags = 0; \
    (__pvvwwid_)->dwResDrastic = 0; \
}

```

Clean and do basic setup of a [VWVVIDEO](#) pointer from width, height, align, codec, scale, rate and length in frames.

Definition at line 1248 of file vvwtypes.h.

```

#define VWVVID_SIZEIMAGE( __pvvwwid_) { if(!(__pvvwwid_)->biPitch)
VWVVID_PITCHALIGN1((__pvvwwid_)); (__pvvwwid_)->biSizeImage = (__pvvwwid_)-
>biPitch * abs((__pvvwwid_)->biHeight); }

```

Set the [VWVVIDEO::biSizeImage](#) based on [VWVVIDEO::biWidth](#), [VWVVIDEO::biHeight](#) and [VWVVIDEO::biBitCount](#).

Definition at line 1219 of file vvwtypes.h.

```

#define VWVVID_SUGGESTEDBUFFERSIZE( __pvvwwid_) { if(!(__pvvwwid_)->biPitch)
VWVVID_PITCHALIGN1((__pvvwwid_)); (__pvvwwid_)->dwSuggestedBufferSize =
(__pvvwwid_)->biPitch * (abs((__pvvwwid_)->biHeight) + (abs((__pvvwwid_)->biHeight) >>
2)); }

```

Set the [VWVVIDEO::dwSuggestedBufferSize](#) based on [VWVVIDEO::biWidth](#),

[VWVIDEO::biHeight](#) and [VWVIDEO::biBitCount](#).

Definition at line 1221 of file vvwtypes.h.

```
#define VWVIDEO_2048_YCBCR10 5504
```

Definition at line 1132 of file vvwtypes.h.

```
#define VWVIDEO_720P_YCBCR10 3456
```

Definition at line 1131 of file vvwtypes.h.

```
#define VWXXX_GETSAMPLEFROMLENGTH( __pvw_ ) ( (__pvw_)->dwLength /  
( __pvw_)->dwScale)
```

This macro is incorrect.

Todo:

Find and remove
Definition at line 925 of file vvwtypes.h.

```
#define VWXXX_SETSAMPLETOLENGTH( __pvw_, _length ) { ( __pvw_)->dwLength =  
_length * ( __pvw_)->dwScale; }
```

This macro is incorrect.

Todo:

Find and remove
Definition at line 923 of file vvwtypes.h.

Typedef Documentation

typedef struct [DCLIP](#) * [PDCLIP](#)

typedef struct [DFRAME](#) * [PDFRAME](#)

typedef struct [DPOSSIZENAME](#) * [PDPOSSIZENAME](#)

typedef struct [DRASTIC_CHANNEL](#) * [pDRASTIC_CHANNEL](#)

typedef struct [DTDIRECT_WAVEHDR](#) * [pDTDIRECT_WAVEHDR](#)

typedef struct [FRAME_INFO](#) * [pFRAME_INFO](#)

typedef struct [VVWAUDIO](#) * [pVVWAUDIO](#)

typedef struct [vwwInfilmageInfo](#) * [pvwwInfilmageInfo](#)

typedef struct [VWINFO](#) * [pVWINFO](#)

typedef struct [VWSYSTEM](#) * [pVWSYSTEM](#)

typedef struct [VWVIDEO](#) * [pVWVIDEO](#)

Enumeration Type Documentation

enum [vwwInfoMetaTypes](#)

Numeric values for all the metadata information types available in MR and VVW

Enumerator:

vwwiFileName see [VWINFO::szFileName](#)

vwwiNativeLocator see [VWINFO::szNativeLocator](#)

vwwiUniversalName see [VWINFO::szUniversalName](#)

vwwiIP see [VWINFO::szIP](#)

vwwiSourceLocator see [VWINFO::szSourceLocator](#)

vwwiChannel see [VWINFO::szChannel](#)

vwwiChannelName see [VWINFO::szChannelName](#)

vwwiChannelDescription see [VWINFO::szChannelDescription](#)

vwwiTitle see [VWINFO::szTitle](#)

vwwiSubject see [VWINFO::szSubject](#)

vwwiCategory see [VWINFO::szCategory](#)

vwwiKeywords see [VWINFO::szKeywords](#)

vwwiRatings see [VWINFO::szRatings](#)

vvwiComments see [VVWINFO::szComments](#)
vvwiOwner see [VVWINFO::szOwner](#)
vvwiEditor see [VVWINFO::szEditor](#)
vvwiSupplier see [VVWINFO::szSupplier](#)
vvwiSource see [VVWINFO::szSource](#)
vvwiProject see [VVWINFO::szProject](#)
vvwiStatus see [VVWINFO::szStatus](#)
vvwiAuthor see [VVWINFO::szAuthor](#)
vvwiRevisionNumber see [VVWINFO::szRevisionNumber](#)
vvwiProduced see [VVWINFO::szProduced](#)
vvwiAlbum see [VVWINFO::szAlbum](#)
vvwiArtist see [VVWINFO::szArtist](#)
vvwiComposer see [VVWINFO::szComposer](#)
vvwiCopyright see [VVWINFO::szCopyright](#)
vvwiCreationData see [VVWINFO::szCreationData](#)
vvwiDescription see [VVWINFO::szDescription](#)
vvwiDirector see [VVWINFO::szDirector](#)
vvwiDisclaimer see [VVWINFO::szDisclaimer](#)
vvwiEncodedBy see [VVWINFO::szEncodedBy](#)
vvwiFullName see [VVWINFO::szFullName](#)
vvwiGenre see [VVWINFO::szGenre](#)
vvwiHostComputer see [VVWINFO::szHostComputer](#)
vvwiInformation see [VVWINFO::szInformation](#)
vvwiMake see [VVWINFO::szMake](#)
vvwiModel see [VVWINFO::szModel](#)
vvwiOriginalArtist see [VVWINFO::szOriginalArtist](#)
vvwiOriginalFormat see [VVWINFO::szOriginalFormat](#)
vvwiPerformers see [VVWINFO::szPerformers](#)
vvwiProducer see [VVWINFO::szProducer](#)
vvwiProduct see [VVWINFO::szProduct](#)
vvwiSoftware see [VVWINFO::szSoftware](#)
vvwiSpecialPlaybackRequirements see [VVWINFO::szSpecialPlaybackRequirements](#)
vvwiTrack see [VVWINFO::szTrack](#)
vvwiWarning see [VVWINFO::szWarning](#)
vvwiURLLink see [VVWINFO::szURLLink](#)
vvwiEditData1 see [VVWINFO::szEditData1](#)
vvwiEditData2 see [VVWINFO::szEditData2](#)
vvwiEditData3 see [VVWINFO::szEditData3](#)

vvwiEditData4 see [VVWINFO::szEditData4](#)
vvwiEditData5 see [VVWINFO::szEditData5](#)
vvwiEditData6 see [VVWINFO::szEditData6](#)
vvwiEditData7 see [VVWINFO::szEditData7](#)
vvwiEditData8 see [VVWINFO::szEditData8](#)
vvwiEditData9 see [VVWINFO::szEditData9](#)
vvwiVersionString see [VVWINFO::szVersionString](#)
vvwiManufacturer see [VVWINFO::szManufacturer](#)
vvwiLanguage see [VVWINFO::szLanguage](#)
vvwiFormat see [VVWINFO::szFormat](#)
vvwiInputDevice see [VVWINFO::szInputDevice](#)
vvwiDeviceModelNum see [VVWINFO::szDeviceModelNum](#)
vvwiDeviceSerialNum see [VVWINFO::szDeviceSerialNum](#)
vvwiReel see [VVWINFO::szReel](#)
vvwiShot see [VVWINFO::szShot](#)
vvwiTake see [VVWINFO::szTake](#)
vvwiSlateInfo see [VVWINFO::szSlateInfo](#)
vvwiFrameAttribute see [VVWINFO::szFrameAttribute](#)
vvwiEpisode see [VVWINFO::szEpisode](#)
vvwiScene see [VVWINFO::szScene](#)
vvwiDailyRoll see [VVWINFO::szDailyRoll](#)
vvwiCamRoll see [VVWINFO::szCamRoll](#)
vvwiSoundRoll see [VVWINFO::szSoundRoll](#)
vvwiLabRoll see [VVWINFO::szLabRoll](#)
vvwiKeyNumberPrefix see [VVWINFO::szKeyNumberPrefix](#)
vvwiInkNumberPrefix see [VVWINFO::szInkNumberPrefix](#)
vvwiPictureIcon see [VVWINFO::szPictureIcon](#)
vvwiProxyFile see [VVWINFO::szProxyFile](#)
vvwiCustomMetadataBlockPointer
vvwiImageInfo
vvwiUMID
vvwiEND_OF_STRINGS
vvwiNumericStart
vvwiTimeCode see [VVWINFO::dwTimeCode](#)
vvwiUserBits see [VVWINFO::dwUserBits](#)
vvwiVITCTimeCode see [VVWINFO::dwVITCTimeCode](#)
vvwiVITCUserBits see [VVWINFO::dwVITCUserBits](#)
vvwiVITCLine3 see [VVWINFO::dwVITCLine3](#)
vvwiPosterFrame see [VVWINFO::dwPosterFrame](#)

vvwiAFrame see [VVWINFO::dwAFrame](#)
vvwiAspectRatio see [VVWINFO::dwAspectRatio](#)
vvwiOriginalRate see [VVWINFO::dwOriginalRate](#)
vvwiOriginalScale see [VVWINFO::dwOriginalScale](#)
vvwiConversions see [VVWINFO::dwConversions](#)
vvwiVersionNumber see [VVWINFO::dwVersionNumber](#)
vvwiFileSize see [VVWINFO::dwFileSize](#)
vvwiFileDate see [VVWINFO::dwFileDate](#)
vvwiFileTime see [VVWINFO::dwFileTime](#)
vvwiSequenceNumber see [VVWINFO::dwSequenceNumber](#)
vvwiTotalStreams see [VVWINFO::dwTotalStreams](#)
vvwiTotalLength see [VVWINFO::dwTotalLength](#)
vvwiFilmManufacturerCode see [VVWINFO::dwFilmManufacturerCode](#)
vvwiFilmTypeCode see [VVWINFO::dwFilmTypeCode](#)
vvwiWhitePoint see [VVWINFO::dwWhitePoint](#)
vvwiBlackPoint see [VVWINFO::dwBlackPoint](#)
vvwiBlackGain see [VVWINFO::dwBlackGain](#)
vvwiBreakPoint see [VVWINFO::dwBreakPoint](#)
vvwiGamma1000 see [VVWINFO::dwGamma1000](#)
vvwiTagNumber see [VVWINFO::dwTagNumber](#)
vvwiFlags see [VVWINFO::dwFlags](#)
vvwiTimeCodeType see [VVWINFO::dwTimeCodeType](#)
vvwiLTCTimeCodeType see [VVWINFO::dwLTCTimeCodeType](#)
vvwiVITCTimeCodeType see [VVWINFO::dwVITCTimeCodeType](#)
vvwiProdDate see [VVWINFO::dwProdDate](#)
vvwiUniqueID see [VVWINFO::dwUniqueID](#)
vvwiCustomMetadataBlockType
vvwiCustomMetadataBlockSize
vvwiNorthSouthEastWest
vvwiLatitude
vvwiLongitude
vvwiEND_OF_DWORD_V2
vvwiVideoWidth INTERNAL: Auto generated for XML output from [VVWVIDEO/VVWAUDIO](#).

Definition at line 1609 of file vvwtypes.h.

Index

INDEX