



# Full Motion Player

## Application Programmer Interface

**REALmagic™** embedded API  
preliminary version 0.2.050.9  
subject to change without notice

***SIGMA DESIGNS®***

<mailto:support@sdesigns.com>

<http://www.sigmadesigns.com>

March 26, 2002

---

# Contents

<b>1</b>	<b>Command description</b>	<b>9</b>
1.1	Callback functions	9
1.1.1	Callback functions	10
1.2	CDDA functions	10
1.2.1	CDDAFastForward	10
1.2.2	CDDAGetNumberOfTracks	11
1.2.3	CDDAGetStatus	11
1.2.4	CDDAGetTrackDuration	12
1.2.5	CDDANextTrack	12
1.2.6	CDDAPlayTrack	12
1.2.7	CDDAPrevTrack	13
1.2.8	CDDARepeat	13
1.2.9	CDDARewind	14
1.3	Digital overlay	14
1.3.1	FMPSetVideoPortDimensions	14
1.4	Disc functions	15
1.4.1	AngleChange	15
1.4.2	AudioStreamChange	15
1.4.3	BackwardScan	16
1.4.4	ButtonActivate	16
1.4.5	ButtonSelectAndActivate	17
1.4.6	ChapterPlay	17
1.4.7	ChapterSearch	18
1.4.8	ClearRepeatAB	18
1.4.9	DefaultPGSearch	19
1.4.10	FastForward	19
1.4.11	ForwardScan	20
1.4.12	GoUp	20
1.4.13	KaraokeAudioPresentationModeChange	20
1.4.14	LeftButtonSelect	21
1.4.15	LowerButtonSelect	21
1.4.16	MenuCall	22
1.4.17	NextPGSearch	22
1.4.18	NumericSelections	23
1.4.19	PrevPGSearch	24
1.4.20	RepeatAB	24
1.4.21	RepeatChapter	25
1.4.22	RepeatTitle	25

1.4.23	Resume	25
1.4.24	ResumeBookmark	26
1.4.25	ReturnPGSearch	27
1.4.26	Rewind	27
1.4.27	RightButtonSelect	28
1.4.28	StillOff	28
1.4.29	StopForResume	28
1.4.30	StoreBookmark	29
1.4.31	SubPictureStreamChange	29
1.4.32	TimePlay	30
1.4.33	TimeSearch	31
1.4.34	TitlePlay	33
1.4.35	TopPGSearch	34
1.4.36	UpperButtonSelect	34
1.5	General functions	34
1.5.1	FMPAbout	34
1.5.2	FMPClose	35
1.5.3	FMPGet	35
1.5.4	FMPGetProfileWord	37
1.5.5	FMPOpenDiscPlayback	38
1.5.6	FMPPause	40
1.5.7	FMPPlay	41
1.5.8	FMPProperty	41
1.5.9	FMPSet	45
1.5.10	FMPStop	46
1.5.11	FMPTrayOperation	46
1.5.12	FMPWriteProfileWord	47
1.5.13	MPEGDriverEntry	48
1.5.14	MPEGDriverUnload	49
1.5.15	FMPQueryInterface	49
1.6	Picture placement	50
1.6.1	FMPAnalogOverlay	50
1.6.2	FMPSetDestination	51
1.6.3	FMPSetSource	52
1.6.4	FMPSetVisibleSource	52
1.7	Push model functions	53
1.7.1	FMPFlush	53
1.7.2	FMPGetBuffer	53
1.7.3	FMPOpen	54
1.7.4	FMPPush	55
2	Error messages	57
3	Events	59
3.1	FMPM_AUTOPAUSE	60
3.2	FMPM_CDDA_PSD_END	60
3.3	FMPM_DVD_ANGLE_CHANGE	60
3.4	FMPM_DVD_ANGLES_BLOCK	61
3.5	FMPM_DVD_AUDIO_STREAM_CHANGE	61
3.6	FMPM_DVD_BUTTON_CHANGE	61

3.7	FMPM_DVD_BUTTONS_CHANGE	61
3.8	FMPM_DVD_CHAPTER_CHANGE	62
3.9	FMPM_DVD_CURRENT_TIME	62
3.10	FMPM_DVD_DOMAIN_CHANGE	62
3.11	FMPM_DVD_END_PLAYBACK	62
3.12	FMPM_DVD_FATAL_ERROR	62
3.13	FMPM_DVD_MACROVISION_LEVEL	63
3.14	FMPM_DVD_NO_FP_PGC	63
3.15	FMPM_DVD_PARENTAL_CHECK	63
3.16	FMPM_DVD_PARENTAL_ERROR	63
3.17	FMPM_DVD_PARENTAL_LEVEL_CHANGE	64
3.18	FMPM_DVD_PLAYBACK_STOPPED	64
3.19	FMPM_DVD_PROGRAM_START	64
3.20	FMPM_DVD_STILL_OFF	64
3.21	FMPM_DVD_STILL_ON	64
3.22	FMPM_DVD_SUBPICTURE_STREAM_CHANGE	65
3.23	FMPM_DVD_TITLE_CHANGE	65
3.24	FMPM_DVD_VALID_UOPS_CHANGE	65
3.25	FMPM_DVDROM_ERR	65
3.26	FMPM_DVDROM_NOT_READY	66
3.27	FMPM_EOS	66
3.28	FMPM_ERROR	66
3.29	FMPM_REGION_MISMATCH	66
3.30	FMPM_STARVATION	66
3.31	FMPM_SVCD_PSD_END	66
3.32	FMPM_TRICK_MODE_CHANGE	67
<b>4</b>	<b>Installation</b>	<b>69</b>
4.1	Linux	69
4.2	VxWorks	69
4.3	Windows CE	69
<b>5</b>	<b>Language codes</b>	<b>71</b>
<b>6</b>	<b>Registry values</b>	<b>75</b>
6.1	Os - Linux	75
6.1.1	Analogoverlay	75
6.1.2	DVD - celesteapp	76
6.2	Os - WinCE	83
6.2.1	DVD registries	83
6.2.2	Hwlibrary	84
<b>7</b>	<b>Settings</b>	<b>89</b>
7.1	CDDA settings	89
7.1.1	CDDAI_TOC	89
7.2	DVD settings	91
7.2.1	DVDI_AST_ATR	91
7.2.2	DVDI_SPRM	93
7.2.3	DVDI_SPST_ATR	94
7.2.4	DVDI_TT_SRPTI	95

7.2.5	DVDI_VIDEO_MODE	95
7.2.6	DVDI_CURRENT_UOPS	96
7.3	DVD, SVCD, VCD settings	98
7.4	General settings	99
7.4.1	FMPI_STANDARDTV	100
7.5	Source and destination settings	100
8	Streaming video	101
9	Use guidelines	105
9.1	Programming notes	105
9.2	Technical support	105

# List of Figures

1.1	CDDAFastForward Operating Systems and Streaming Models . . . . .	10
1.2	CDDAGetNumberOfTracks Operating Systems and Streaming Models . . . . .	11
1.3	CDDAGetStatus Operating Systems and Streaming Models . . . . .	11
1.4	CDDAGetTrackDuration Operating Systems and Streaming Models . . . . .	12
1.5	CDDANextTrack Operating Systems and Streaming Models . . . . .	12
1.6	CDDAPlayTrack Operating Systems and Streaming Models . . . . .	13
1.7	CDDAPrevTrack Operating Systems and Streaming Models . . . . .	13
1.8	CDDARepeat Operating Systems and Streaming Models . . . . .	14
1.9	CDDARewind Operating Systems and Streaming Models . . . . .	14
1.10	AngleChange Operating Systems and Streaming Models . . . . .	15
1.11	AudioStreamChange Operating Systems and Streaming Models . . . . .	15
1.12	BackwardScan Operating Systems and Streaming Models . . . . .	16
1.13	ButtonActive Operating Systems and Streaming Models . . . . .	16
1.14	ButtonSelectAndActivate Operating Systems and Streaming Models . . . . .	17
1.15	ChapterPlay Operating Systems and Streaming Models . . . . .	17
1.16	ChapterSearch Operating Systems and Streaming Models . . . . .	18
1.17	ClearRepeatAB Operating Systems and Streaming Models . . . . .	18
1.18	DefaultPGSearch Operating Systems and Streaming Models . . . . .	19
1.19	FastForward Operating Systems and Streaming Models . . . . .	19
1.20	ForwardScan Operating Systems and Streaming Models . . . . .	20
1.21	GoUp Operating Systems and Streaming Models . . . . .	20
1.22	KaraokeAudioPresentationMode Operating Systems and Streaming Models . . . . .	21
1.23	LeftButtonSelect Operating Systems and Streaming Models . . . . .	21
1.24	LowerButtonSelect Operating Systems and Streaming Models . . . . .	21
1.25	MenuCall Operating Systems and Streaming Models . . . . .	22
1.26	NextPGSearch Operating Systems and Streaming Models . . . . .	23
1.27	NumericSelections Operating Systems and Streaming Models . . . . .	23
1.28	PrevPGSearch Operating Systems and Streaming Models . . . . .	24
1.29	RepeatAB Operating Systems and Streaming Models . . . . .	24
1.30	RepeatChapter Operating Systems and Streaming Models . . . . .	25
1.31	RepeatTitle Operating Systems and Streaming Models . . . . .	25
1.32	Resume Operating Systems and Streaming Models . . . . .	26
1.33	ResumeBookmark Operating Systems and Streaming Models . . . . .	26
1.34	ReturnPGSearch Operating Systems and Streaming Models . . . . .	27
1.35	Rewind Operating Systems and Streaming Models . . . . .	27
1.36	RightButtonSelect Operating Systems and Streaming Models . . . . .	28
1.37	StillOff Operating Systems and Streaming Models . . . . .	28
1.38	StopForResume Operating Systems and Streaming Models . . . . .	29
1.39	StoreBookmark Operating Systems and Streaming Models . . . . .	29

1.40 SubPictureStreamChange Operating Systems and Streaming Models . . . . .	30
1.41 TimePlay Operating Systems and Streaming Models . . . . .	31
1.42 TimeSearch Operating Systems and Streaming Models . . . . .	33
1.43 TitlePlay Operating Systems and Streaming Models . . . . .	33
1.44 TopPGSearch Operating Systems and Streaming Models . . . . .	34
1.45 UpperButtonSelect Operating Systems and Streaming Models . . . . .	34
1.46 FMPAbout Operating Systems and Streaming Models . . . . .	35
1.47 FMPClose Operating Systems and Streaming Models . . . . .	35
1.48 FMPGet Operating Systems and Streaming Models . . . . .	37
1.49 FMPGetProfileWord Operating Systems and Streaming Models . . . . .	38
1.50 FMPOpenDiscPlayback Operating Systems and Streaming Models . . . . .	40
1.51 FMPPause Operating Systems and Streaming Models . . . . .	41
1.52 FMPPPlay Operating Systems and Streaming Models . . . . .	41
1.53 FMPPProperty Operating Systems and Streaming Models . . . . .	45
1.54 FMPSet Operating Systems and Streaming Models . . . . .	46
1.55 FMPSStop Operating Systems and Streaming Models . . . . .	46
1.56 FMPTrayOperation Operating Systems and Streaming Models . . . . .	47
1.57 FMPWriteProfileWord Operating Systems and Streaming Models . . . . .	48
1.58 MPEGDriverEntry Operating Systems and Streaming Models . . . . .	48
1.59 MPEGDriverUnload Operating Systems and Streaming Models . . . . .	49
1.60 FMPQueryInterface Operating Systems and Streaming Models . . . . .	49
1.61 FMPFlush Operating Systems and Streaming Models . . . . .	53
1.62 FMPGetBuffer Operating Systems and Streaming Models . . . . .	54
1.63 FMPOpen Operating Systems and Streaming Models . . . . .	55
1.64 FMPPush Operating Systems and Streaming Models . . . . .	56
7.1 CDDA settings . . . . .	89
7.2 CDDAI_TOC Operating Systems and Streaming Models . . . . .	91
7.3 DVD settings . . . . .	91
7.4 DVDI_AST_ATR Operating Systems and Streaming Models . . . . .	93
7.5 System Parameters settings . . . . .	93
7.6 DVDI_SPRM Operating Systems and Streaming Models . . . . .	94
7.7 DVDI_SPST_ATR Operating Systems and Streaming Models . . . . .	95
7.8 DVDI_TT_SRPTI Operating Systems and Streaming Models . . . . .	95
7.9 DVDI_VIDEO_MODE Operating Systems and Streaming Models . . . . .	96
7.10 System Parameters settings . . . . .	97
7.11 DVDI_CURRENT_UOPS Operating Systems and Streaming Models . . . . .	97
7.12 General settings . . . . .	99
7.13 FMPI_STANDARDTV Operating Systems and Streaming Models . . . . .	100



# Chapter 1

## Command description

### *Push Model*

Following are the different commands you can use to access the driver. Here is a summary of the commands classed by function.

#### *To open and close the MPEG driver*

MPEGDriverEntry opens the MPEG driver  
MPEGDriverUnload closes the MPEG driver

#### *To open and close a stream*

FMPOpen opens a stream (Push Model)  
FMPClose closes a stream

#### *To play the stream*

FMPPPlay plays a stream  
FMPPPause pauses a stream  
FMPPStop stops a stream

#### *To set and get parameters*

FMPSet sets a parameter  
FMPPGet gets a parameter

#### *To send data to the driver*

FMPPGetBuffer gets a buffer from the driver  
FMPPush pushes data to the driver  
FMPPFlush flushes the internal buffers

## 1.1 Callback functions

A callback function is a function of your application called by the driver in some situations.

You use the FMPOpen command to define callback functions.

Depending on the flags specified when declaring your function, the return status is passed in DWORD Value

Your function should look like this:

```
DWORD MyCallBack(DWORD dwContext, DWORD dwMsg, DWORD dwValue);
```

You should return a zero value if successful. A return value of FMPE\_ERROR specifies that an error occurred while executing a command. The error in Value may be one of the following:

FMPE\_ERROR System error occurred while reading the stream

FMPM\_STARVATION Message issued when the video decoder is starving. Value is the total size of buffers in the driver in PTS units.

FMPM\_EOS Message issued when the video and audio decoders have finished decoding the stream.

### 1.1.1 Callback functions

A callback function is a function of your application called by the driver in some situations.

You use the FMPOpen command to define callback functions.

Depending on the flags specified when declaring your function, the return status is passed in DWORD Value

Your function should look like this:

```
DWORD MyCallBack(DWORD dwContext, DWORD dwMsg, DWORD dwValue);
```

You should return a zero value if successful. A return value of FMPE\_ERROR specifies that an error occurred while executing a command. The error in Value may be one of the following:

FMPE\_ERROR System error occurred while reading the stream

FMPM\_STARVATION Message issued when the video decoder is starving. Value is the total size of buffers in the driver in PTS units.

FMPM\_EOS Message issued when the video and audio decoders have finished decoding the stream.

## 1.2 CDDA functions

### 1.2.1 CDDAFastForward

Fast forward.

```
DWORD CDDAFastForward (DWORD dwSpeed);
```

*Return value*

FMPE\_OK successful

FMPE\_DRIVER\_NOT\_OPEN error - driver not opened

FMPE\_CDDA\_FAILURE failure

*Remarks*

The dwSpeed parameter indicated how many cdda sectors will be skipped inbetween each read. The number of sectors skipped is equal to: (dwSpeed-1)\*6. If dwSpeed is equal to 0 or 1, the playback will resume normally.

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks					•
Windows CE					
Linux					•

Figure 1.1: CDDAFastForward Operating Systems and Streaming Models

### 1.2.2 CDDAGetNumberOfTracks

To get the number of tracks for audio CD.

```
DWORD CDDAGetNumberOfTracks( BYTE* pbTrackNumber );
```

#### Arguments

BYTE\* pbTrackNumber Pointer to get tracks.

#### Return value

FMPE\_OK successful

FMPE\_CDDA\_FAILURE failure

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks					•
Windows CE					
Linux					•

Figure 1.2: CDDAGetNumberOfTracks Operating Systems and Streaming Models

### 1.2.3 CDDAGetStatus

Gets the current status of the CDDA playback.

```
DWORD CDDAGetStatus (DWORD *pdwMode, BYTE *pbTrack, DWORD *pdwSeconds);
```

#### Arguments

pdwMode is a pointer to a DWORD that will receive the current repeat mode. The valid values are CDDA\_MODE\_REPEAT\_NONE, CDDA\_MODE\_REPEAT\_TRACK, and CDDA\_MODE\_REPEAT\_DISC, and are defined in FMP.H.

pbTrack is a pointer to a BYTE that will receive the current track.

pdwSeconds is a pointer to a DWORD that will receive the current position relative to the start of the current track. The units are SECONDS.

#### Return value

FMPE\_OK successful

FMPE\_DRIVER\_NOT\_OPEN error - driver not opened

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks					•
Windows CE					
Linux					•

Figure 1.3: CDDAGetStatus Operating Systems and Streaming Models

### 1.2.4 CDDAGetTrackDuration

To get the time for specified audio track.

```
DWORD CDDAGetTrackDuration(BYTE bTrackNumber, DWORD *pbSeconds);
```

#### Arguments

BYTE bTrackNumber- Track Number

DWORD\* pbSeconds - Pointer to a DWORD that will get the time for specified audio track. The units are SECONDS.

#### Return value

FMPE\_OK successful

FMPE\_CDDA\_FAILURE failure or end of disk reached

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks					•
Windows CE					
Linux					•

Figure 1.4: CDDAGetTrackDuration Operating Systems and Streaming Models

### 1.2.5 CDDANextTrack

Next audio track.

```
DWORD CDDANextTrack();
```

#### Return value

FMPE\_OK successful

FMPE\_CDDA\_FAILURE failure or end of disk reached

#### Remarks

If the last track is reached, this function will loop around to the first track.

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks					•
Windows CE					
Linux					•

Figure 1.5: CDDANextTrack Operating Systems and Streaming Models

### 1.2.6 CDDAPlayTrack

Plays a CDDA track from a specified position.

```
DWORD CDDAPlayTrack( DWORD dwTrack, DWORD dwMinutes, DWORD dwSeconds);
```

#### Return value

FMPE\_OK successful

FMPE\_DRIVER\_NOT\_OPEN error - driver not opened

FMPE\_CDDA\_FAILURE failure

FMPE\_INVALID\_PARAMETER invalid parameter.

#### Remarks

This function will cause the current playback to continue at the specified position. This function is only valid when the playback is normal (not fast forward or rewind). This function will return FMPE\_INVALID\_PARAMETER if the specified track is invalid.

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks					•
Windows CE					
Linux					•

Figure 1.6: CDDAPlayTrack Operating Systems and Streaming Models

### 1.2.7 CDDAPrevTrack

Previous audio track.

```
DWORD CDDAPrevTrack( );
```

#### Return value

FMPE\_OK successful

FMPE\_CDDA\_FAILURE failure or beginning of disk reached

#### Remarks

If the current track is the first track, this function will loop to the last track.

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks					•
Windows CE					
Linux					•

Figure 1.7: CDDAPrevTrack Operating Systems and Streaming Models

### 1.2.8 CDDARepeat

Change the CDDA repeat mode.

```
DWORD CDDARepeat (DWORD dwMode);
```

#### Return value

FMPE\_OK successful

FMPE\_DRIVER\_NOT\_OPEN error - driver not opened

FMPE\_CDDA\_FAILURE failure

*Remarks*

The valid modes are CDDA\_MODE\_REPEAT\_NONE, CDDA\_MODE\_REPEAT\_TRACK, CDDA\_MODE\_REPEAT\_DISC. These values are defined in FMP.H.

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks					•
Windows CE					
Linux					•

Figure 1.8: CDDARepeat Operating Systems and Streaming Models

**1.2.9 CDDARewind**

## CD-Audio Rewind

```
DWORD CDDARewind (DWORD dwSpeed);
```

*Return value*

FMPE\_OK successful

FMPE\_CDDA\_FAILURE failure or end of disk reached

*Remarks*

The dwSpeed parameter indicated how many cdda sectors will be skipped inbetween each read. The number of sectors skipped is equal to: (dwSpeed-1)\*6. If dwSpeed is equal to 0 or 1, the playback will resume normally.

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks					•
Windows CE					
Linux					•

Figure 1.9: CDDARewind Operating Systems and Streaming Models

**1.3 Digital overlay****1.3.1 FMPSetVideoPortDimensions**

This function allows the user to change the video port output dimensions.

```
DWORD FMPSetVideoPortDimensions(DWORD Width,DWORD Height);
```

*Arguments*

DWORD Width Horizontal length

DWORD Height Vertical length

*See also*

FMPSet with FMPI\_VIDEOPORT\_WINDOW ([1.5.9](#))

## 1.4 Disc functions

### 1.4.1 AngleChange

In Angle Block, change the Angle

```
DWORD AngleChange (bAngleNumber);
```

*Arguments*

BYTE bAngleNumber Angle Number. Must be set between 1 and 9

*Return value*

Zero if successful. Otherwise returns an error code.

DVDE\_ARG Arguments are not valid.

DVDE\_UNAUTHORIZED Unauthorized operation.

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks		•			
Windows CE		•			
Linux		•			

Figure 1.10: AngleChange Operating Systems and Streaming Models

### 1.4.2 AudioStreamChange

Set the Stream number of the Audio.

```
DWORD AudioStreamChange (bStreamNumber);
```

*Arguments*

BYTE bStreamNumber Audio Stream number (DVD: between 0 and 7, SVCD: 0 or 1)

*Return value*

DVD Zero if successful. Otherwise returns an error code.

DVDE\_ARG Arguments are not valid.

DVDE\_UNAUTHORIZED Unauthorized operation.

SVCD

FMPE\_OK successful

FMPE\_UNEXPECTED unknown errors

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks		•		•	
Windows CE		•		•	
Linux		•		•	

Figure 1.11: AudioStreamChange Operating Systems and Streaming Models

### 1.4.3 BackwardScan

The operation to scan play at the specified Speed. This user function includes scan play and slow playback for backward navigation at any speed (except normal speed).

`DWORD BackwardScan (wSpeed, bFast);`

#### Arguments

DWORD wSpeed Speed of the stream (2 for x2, 4 for x4, 6 for x6, 8 for x8)

BOOL bFast TRUE if Fast-Forward, FALSE if Slow motion

#### Return value

Zero if successful. Otherwise returns an error code.

DVDE\_ARG Arguments are not valid.

DVDE\_UNAUTHORIZED Unauthorized operation.

#### Remarks

Backward slow motion is implemented. See revived for VCD, SVCD and CDDA revived for CDDA.

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks		•			
Windows CE		•			
Linux		•			

Figure 1.12: BackwardScan Operating Systems and Streaming Models

### 1.4.4 ButtonActivate

Activate the current Highlighted Button or push the button.

`DWORD ButtonActivate();`

#### Return value

Zero if successful. Otherwise returns an error code.

DVDE\_UNAUTHORIZED Unauthorized operation.

See also

VCD:NumericSelections([1.4.18](#))

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks		•			
Windows CE		•			
Linux		•			

Figure 1.13: ButtonActive Operating Systems and Streaming Models



### 1.4.5 ButtonSelectAndActivate

Activate the specified Highlighted Button.

```
DWORD ButtonSelectAndActivate(BYTE bButton);
```

*Arguments*

BYTE bButton the button number (between 1 and 36)

*Return value*

Zero if successful. Otherwise returns an error code.

DVDE.UNAUTHORIZED Unauthorized operation

DVDE\_ARG Arguments are not valid

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks		•			
Windows CE		•			
Linux		•			

Figure 1.14: ButtonSelectAndActivate Operating Systems and Streaming Models

### 1.4.6 ChapterPlay

Play from the beginning of a chapter specified by the Title number and the chapter number

```
DWORD ChapterPlay(bTitleNumber, wChapterNumber);
```

*Arguments*

BYTE bTitleNumber Title Number to play. The number is set between 1 and 99

WORD bChapterNumber Chapter Number to play. The number is set between 1 and 999

*Return value*

Zero if successful. Otherwise, returns an error code.

DVDE\_ARG Argument is not valid.

DVDE.TITLENUMBER The Title number does not exist

DVDE.PTTNUMBER The PTT number does not exist

DVDE.UNAUTHORIZED Unauthorized operation.

*Remarks*

Branch to PTT specified by user. Execute Pre-Command.

*See also*

TitlePlay(1.4.34), DVDI.TT\_SRPTI (7)

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks		•			
Windows CE		•			
Linux		•			

Figure 1.15: ChapterPlay Operating Systems and Streaming Models

### 1.4.7 ChapterSearch

Stop the current presentation and start the presentation from the beginning of PTT (Part\_of\_Title) specified by PTT numbers within the same Title.

```
DWORD ChapterSearch(wChapterNumber);
```

#### Arguments

WORD wChapterNumber Chapter Number to play. The number is set between 1 and 999

#### Return value

Zero if successful. Otherwise, returns an error code.

DVDE\_ARG Argument is not valid.

DVDE\_PTTNUMBER The Chapter number does not exist

DVDE\_UNAUTHORIZED Unauthorized operation.

#### Remarks

Branch to nominated PTT specified by the user. Do not execute Pre-Command

#### See also

ChapterPlay(1.4.6)

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks		•			
Windows CE		•			
Linux		•			

Figure 1.16: ChapterSearch Operating Systems and Streaming Models

### 1.4.8 ClearRepeatAB

Turns off the RepeatAB function and resets all bookmarks.

```
DWORD ClearRepeatAB();
```

#### Return value

Zero if successful. Otherwise returns an error code.

DVDE\_UNAUTHORIZED Unauthorized operation.

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks		•	•	•	
Windows CE		•	•	•	
Linux		•	•	•	

Figure 1.17: ClearRepeatAB Operating Systems and Streaming Models

### 1.4.9 DefaultPGSearch

Stop the current play item and start from the list corresponding to the Default List Offset specified in the PSD (Play Sequence Descriptor).

`DWORD DefaultPGSearch ();`

*Return value*

Zero if successful. Otherwise returns an error code.

VCD/SVCD

FMPE\_OK successful  
 FMPE\_UNEXPECTED unknown errors  
 FMPE\_DISABLED the function is currently disabled  
 FMPE\_INVALIDARG the list offset requested is out of range  
 FMPE\_TIMERACTIVATED the timer has been activated  
 FMPE\_FAILURE

*See also*

PrevPGSearch(1.4.19), NextPGSearch(1.4.17)

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks			•	•	
Windows CE			•	•	
Linux			•	•	

Figure 1.18: DefaultPGSearch Operating Systems and Streaming Models

### 1.4.10 FastForward

This function is for playing fast forward and there is no slow motion for VCD, VCD2 and SVCD.

`DWORD FastForward(speed);`

*Arguments*

BYTE speed Fast forward speed. 3 speeds supported. Press fast forward key once for 2X, twice for 3X, and three times for 4X. Your CDRom or DVDROM must be able to support 2x for 2x, 3x for 3x and 4x for 4x.

*Return value*

Zero if successful. Otherwise returns an error code.

VCD/VCD2/SVCD

FMPE\_OK successful  
 FMPE\_UNEXPECTED unknown errors

*Remarks*

When PBC mode is off (PBCOff=1), FastForward function doesn't work and it will return error 0x12 FMPE\_DISABLED.

OS	Push Model	DVD	VCD	VCD2	SVCD	CD Audio
VxWorks			•	•	•	
Windows CE			•	•	•	
Linux			•	•	•	

Figure 1.19: FastForward Operating Systems and Streaming Models

### 1.4.11 ForwardScan

The operation to scan play at the specified Speed. This user function includes scan play and slow playback for forward navigation at any speed (except normal speed).

**DWORD** ForwardScan (wSpeed, bFast);

#### Arguments

WORD wSpeed Speed of the stream (2 for x2, 4 for x4, 6 for x6, 8 for x8)

BOOL bFast TRUE if Fast-Forward, FALSE if Slow motion

#### Return value

Zero if successful. Otherwise returns an error code.

DVDE\_ARG Arguments are not valid.

DVDE\_UNAUTHORIZED Unauthorized operation.

#### Remarks

Slow motion is implemented.

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks		•			
Windows CE		•			
Linux		•			

Figure 1.20: ForwardScan Operating Systems and Streaming Models

### 1.4.12 GoUp

Stop the execution of the current Program Chain and play the GoUp Program Chain (specified in the current Program Chain). In the menu, you go one level up.

**DWORD** GoUp();

#### Return value

DVDE\_UNAUTHORIZED Unauthorized operation.

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks		•			
Windows CE		•			
Linux		•			

Figure 1.21: GoUp Operating Systems and Streaming Models

### 1.4.13 KaraokeAudioPresentationModeChange

Change the mode of Audio Mixing mode for Karaoke. DVD with Karaoke function cannot support because of hardware limitation.

**DWORD** KaraokeAudioPresentationMode (dwMode);

*Arguments*

DWORD dwMode Mode for the Audio Mixing mode for Karaoke

*Return value*

Zero if successful. Otherwise returns an error code.

DVDE\_ARG Arguments are not valid.

DVDE\_UNAUTHORIZED Unauthorized operation.

*Remarks*

Not implemented for DVD

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks					
Windows CE					
Linux					

Figure 1.22: KaraokeAudioPresentationMode Operating Systems and Streaming Models

#### 1.4.14 LeftButtonSelect

Select the Buttons on the Menu Screen.

```
DWORD LeftButtonSelect();
```

*Return value*

Zero if successful. Otherwise returns an error code.

DVDE\_UNAUTHORIZED Unauthorized operation.

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks		•			
Windows CE		•			
Linux		•			

Figure 1.23: LeftButtonSelect Operating Systems and Streaming Models

#### 1.4.15 LowerButtonSelect

Select the Buttons on the Menu Screen.

```
DWORD LowerButtonSelect();
```

*Return value*

Zero if successful. Otherwise returns an error code.

DVDE\_UNAUTHORIZED Unauthorized operation.

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks		•			
Windows CE		•			
Linux		•			

Figure 1.24: LowerButtonSelect Operating Systems and Streaming Models

1.4.16 MenuCall

Call the Menu in the Program Chain (PGC).

```
DWORD MenuCall (bMenuID);
```

Arguments

BYTE bMenuID Specifies the Menu ID to be called

There are six type of Menu ID:

- DVD\_TITLE\_ID Call Title Menu
- DVD\_ROOT\_ID Call Root Menu
- DVD\_AUDIO\_ID Call Audio Menu
- DVD\_SUBPICTURE\_ID Call Sub-picture Menu
- DVD\_ANGLE\_ID Call Angle Menu
- DVD\_PTT\_ID Call PTT Menu

Return value

Zero if successful. Otherwise returns an error code.

- DVDE\_ARG Arguments are not valid.
- DVDE\_UNAUTHORIZED Unauthorized operation.

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks		•			
Windows CE		•			
Linux		•			

Figure 1.25: MenuCall Operating Systems and Streaming Models

1.4.17 NextPGSearch

- DVD Stop the current presentation and start the presentation from the beginning of the next Program within the same PGC.
- VCD2/SVCD Stop the current play item and start from the list corresponding to the Next List Offset specified in the PSD (Play Sequence Descriptor)

```
DWORD NextPGSearch();
```

Return value

- DVD Zero if successful. Otherwise returns an error code.
  - DVDE\_ARG Arguments are not valid.
  - DVDE\_UNAUTHORIZED Unauthorized operation.
- VCD2/SVCD
  - FMPE\_OK successful
  - FMPE\_UNEXPECTED unknow errors

FMPE\_DISABLED the function is currently disabled  
 FMPE\_INVALIDARG the specified time is out of range  
 FMPE\_TIMERACTIVATED the timer has been activated  
 FMPE\_FAILURE vcd2/svcd is not currently playing

#### Remarks

When PBC mode is off (PBCOff=1), NextPGSearch function doesn't work and it will return error 0x12 FMPE\_DISABLED.

#### See also

DVD: GoUp(1.4.12), TopPGSearch(1.4.35), PrevPGSearch(1.4.19)

VCD2/SVCD: PrevPGSearch(1.4.19), ReturnPGSearch(1.4.25), DefaultPGSearch(??)

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks		•	•	•	
Windows CE		•	•	•	
Linux		•	•	•	

Figure 1.26: NextPGSearch Operating Systems and Streaming Models

### 1.4.18 NumericSelections

Implement numeric functions of the Selection List for VCD and SVCD.

**DWORD NumericSelections (BYTE sel)**

#### Arguments

BYTE sel Selection number selected by user (between 0 and 99)

#### Return value

Zero if successful. Otherwise returns an error code.

VCD/SVCD

FMPE\_OK successful

FMPE\_UNEXPECTED unknown errors

FMPE\_DISABLED the function is currently disabled

FMPE\_INVALIDARG

- BSN (base selection number) is zero.
- selection is out of range.
- the list offset requested is out of range.

FMPE\_TIMERACTIVATED the timer has been activated

FMPE\_FAILURE

#### See also

CDDAPlayTrack(1.2.6)

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks			•	•	
Windows CE			•	•	
Linux			•	•	

Figure 1.27: NumericSelections Operating Systems and Streaming Models

### 1.4.19 PrevPGSearch

DVD Stop the current presentation and start the presentation from the beginning of previous Program within the same Program Chain (PGC). It is not a replay.

VCD2/SVCD Stop the current play item and start from the list corresponding to the Previous List Offset specified in the PSD (Play Sequence Descriptor).

**DWORD PrevPGSearch();**

*Return value*

DVD Zero if successful. Otherwise returns an error code.

DVDE\_ARG Arguments are not valid.

DVDE\_UNAUTHORIZED Unauthorized operation.

VCD/SVCD

FMPE\_OK successful

FMPE\_UNEXPECTED unknow errors

FMPE\_DISABLED the function is currently disabled

FMPE\_INVALIDARG the specified time is out of range

FMPE\_TIMERACTIVATED the timer has been activated

FMPE\_FAILURE vcd/svcd is not currently playing

*Remarks*

When PBC mode is off (PBCOff=1), PrevPGSearch function doesn't work and it will return error 0x12 FMPE\_DISABLED.

*See also*

DVD: GoUp(1.4.12), TopPGSearch(1.4.35), NextPGSearch(1.4.17)

VCD2/SVCD: NextPGSearch(1.4.17), ReturnPGSearch(1.4.25), DefaultPGSearch(??)

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks		•	•	•	
Windows CE		•	•	•	
Linux		•	•	•	

Figure 1.28: PrevPGSearch Operating Systems and Streaming Models

### 1.4.20 RepeatAB

Turn on A-B Repeat function. Call once to set Bookmark A. Call second time to set Bookmark B and start the looping playback. Call a third time to turn off.

**DWORD RepeatAB();**

*Return value*

Zero if successful. Otherwise returns an error code.

DVDE\_UNAUTHORIZED Unauthorized operation.

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks		•	•	•	
Windows CE		•	•	•	
Linux		•	•	•	

Figure 1.29: RepeatAB Operating Systems and Streaming Models



### 1.4.21 RepeatChapter

Toggle RepeatChapter function on and off. For VCD and SVCD, this function does repeat current title and will not work with PBC off.

`DWORD RepeatChapter();`

*Return value*

Zero if successful. Otherwise returns an error code.

DVDE\_UNAUTHORIZED Unauthorized operation.

*see also*

CDDARepeat(1.2.8)

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks		•	•	•	
Windows CE		•	•	•	
Linux		•	•	•	

Figure 1.30: RepeatChapter Operating Systems and Streaming Models

### 1.4.22 RepeatTitle

Toggle RepeatTitle function on and off.

`DWORD RepeatTitle();`

*Return value*

Zero if successful. Otherwise returns an error code.

DVDE\_UNAUTHORIZED Unauthorized operation.

*see also*

CDDARepeat(1.2.8)

VCD/SVCD:RepeatChapter(1.4.21)

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks		•			
Windows CE		•			
Linux		•			

Figure 1.31: RepeatTitle Operating Systems and Streaming Models

### 1.4.23 Resume

For DVD, this operation resumes the playing in the middle of a movie if DVD allows it. For VCD and SVCD it gets you out of the repeat mode or after StopForResume mode.

`DWORD Resume();`

*Return value*

DVD Zero if successful. Otherwise returns an error code.

DVDE\_NORESUME Nothing to resume

DVDE\_UNAUTHORIZED Unauthorized operation.

VCD/SVCD

FMPE\_OK successful.

FMPE\_UNEXPECTED unknown errors.

FMPE\_DISABLED its not in pausing.

FMPE\_FAILURE

*see also*

StopForResume([1.4.29](#))

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks		•	•	•	
Windows CE		•	•	•	
Linux		•	•	•	

Figure 1.32: Resume Operating Systems and Streaming Models

## 1.4.24 ResumeBookmark

Resumes playback from last saved bookmark.

**DWORD** ResumeBookmark(pBookmark);

*Arguments*

void\* pBookmark Pointer to bookmark to resume from. This pointer must point to a CRMBookmark structure.

*Return value*

Zero if successful. Otherwise, returns an error code.

DVDE\_UNAUTHORIZED Unauthorized operation.

*Remarks*

*See also*

StoreBookmark([1.4.30](#))

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks		•			
Windows CE		•			
Linux		•			

Figure 1.33: ResumeBookmark Operating Systems and Streaming Models

### 1.4.25 ReturnPGSearch

Stop the current play item and start from the list corresponding to the Return List Offset specified in the PSD (Play Sequence Descriptor).

`DWORD ReturnPGSearch ();`

*Return value*

SVCD

FMPE\_OK successful  
 FMPE\_UNEXPECTED unknown errors  
 FMPE\_DISABLED the function is currently disabled  
 FMPE\_INVALIDARG the list offset requested is out of range  
 FMPE\_TIMERACTIVATED the timer has been activated  
 FMPE\_FAILURE

*See also*

PrevPGSearch(1.4.19), NextPGSearch(1.4.17), DefaultPGSearch(??)

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks				•	
Windows CE				•	
Linux				•	

Figure 1.34: ReturnPGSearch Operating Systems and Streaming Models

### 1.4.26 Rewind

This function is for playing rewind and it is not implemented for slow motion.

`DWORD Rewind(speed)`

*Arguments*

BYTE speed Rewind speed. 3 speeds supported. Press rewind key once for 2X, twice for 3X, and three times for 4X.

*Return value*

VCD/VCD2/SVCD

FMPE\_OK successful  
 FMPE\_UNEXPECTED unknown errors

*Remarks*

When PBC mode is off(PBCOff=1), Rewind function doesn't work and it will return error 0x12 FMPE\_DISABLED.

*see also*

FastForward(1.4.10), CDDARewind1.2.9)

OS	Push Model	DVD	VCD	VCD2	SVCD	CD Audio
VxWorks			•	•	•	
Windows CE			•	•	•	
Linux			•	•	•	

Figure 1.35: Rewind Operating Systems and Streaming Models

### 1.4.27 RightButtonSelect

Select the Buttons on the Menu Screen.

```
DWORD RightButtonSelect();
```

*Return value*

Zero if successful. Otherwise returns an error code.

DVDE\_UNAUTHORIZED Unauthorized operation.

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks		•			
Windows CE		•			
Linux		•			

Figure 1.36: RightButtonSelect Operating Systems and Streaming Models

### 1.4.28 StillOff

Operation to release Still.(Still picture)

```
DWORD StillOff();
```

*Return value*

Zero if successful. Otherwise returns an error code.

DVDE\_UNAUTHORIZED Unauthorized operation.

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks		•			
Windows CE		•			
Linux		•			

Figure 1.37: StillOff Operating Systems and Streaming Models

### 1.4.29 StopForResume

This operation stops playback and prepares to resume later.( needs to be allowed by DVD)

```
DWORD StopForResume();
```

*Return value*

DVD Zero if successful. Otherwise returns an error code.

VCD/SVCD

FMPE\_OK successful.

FMPE\_UNEXPECTED unknown errors.

FMPE\_DISABLED already pausing

*Remarks*

When PBC mode is off (PBCOff=1), StopForResume function doesn't work and it will return error 0x12 FMPE\_DISABLED.

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks		•	•	•	
Windows CE		•	•	•	
Linux		•	•	•	

Figure 1.38: StopForResume Operating Systems and Streaming Models

### 1.4.30 StoreBookmark

Stop the current playback, and saves a bookmark that can be used to resume playback later.(needs to be allowed by DVD)

```
DWORD StoreBookmark(pBookmark);
```

#### Arguments

void\* pBookmark Pointer to save bookmark to. This pointer must have at least a size of CRMBookmark (at least 10K bytes) allocated.

#### Return value

Zero if successful. Otherwise, returns an error code.

DVDE\_UNAUTHORIZED Unauthorized operation. Bookmark cannot be saved.

#### Remarks

When function returns pBookmark will contain the current Bookmark information. Also note that sometimes Bookmark cannot be saved.

#### See also

ResumeBookmark([1.4.24](#))

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks		•			
Windows CE		•			
Linux		•			

Figure 1.39: StoreBookmark Operating Systems and Streaming Models

### 1.4.31 SubPictureStreamChange

Set the stream number of the Sub-Picture and whether the sub-picture is displayed or not.

```
DWORD SubPictureStreamChange (bStreamNumber, bDisplayFlag);
```

#### Arguments

BYTE StreamNumber Subpicture Stream number (between 0 and 31) (62: No Subpicture or Subpicture Stream not selected, 63: Dummy Stream)

BOOL DisplayFlag Display Sub-picture or not

#### Return value

Zero if successful. Otherwise returns an error code.

DVDE\_ARG Arguments are not valid.

DVDE\_UNAUTHORIZED Unauthorized operation.

See also

Settings(7)

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks		•			
Windows CE		•			
Linux		•			

Figure 1.40: SubPictureStreamChange Operating Systems and Streaming Models

### 1.4.32 TimePlay

Play from the specified position of the Title by the Title number and Time

```
DWORD TimePlay(bTitleNumber, dvdTimeCode);
```

Arguments

BYTE bTitleNumber Title Number to play(The number is set between 1 and 99)

ULONG dvdTimeCode See description below

For DVD:

dvdTimeCode is the value defining the time to start playback  
in the form of the DVD\_TIMECODE structure as defined by the Microsoft DirectX API.

```
typedef struct tagDVD_TIMECODE"
```

```
{
```

```
ULONG Hours1 :4;
```

```
ULONG Hours10 :4;
```

```
ULONG Minutes1 :4;
```

```
ULONG Minutes10:4;
```

```
ULONG Seconds1 :4;
```

```
ULONG Seconds10:4;
```

```
ULONG Frames1 :4;
```

```
ULONG Frames10 :2;
```

```
ULONG FrameRateCode:2;
```

```
} DVD_TIMECODE;
```

DVD timecode is binary coded decimal (BCD) encoded in the format 0xHhMmSsFf, where:

H is tens of hours

h is hours

M is tens of minutes

m is minutes

```

S is tens of seconds
s is seconds
F is tens of frames
f is frames
To use a BCD, first create the BCD.

ULONG timeCode = 0; // create the BCD
//create a pointer to a DVD_TIMECODE structure at the address of the BCD
DVD_TIMECODE * dvdTimeCode = ( DVD_TIMECODE * ) &timeCode;

Then, dvdTimeCode can be used as a DVD_TIMECODE structure.

dvdTimeCode->Hours10 = 0;
dvdTimeCode->Hours1 = 2;

```

#### Return value

Zero if successful. Otherwise returns an error code. DVDE\_ARG Arguments are not valid. DVDE\_TITLENUMBER The Title number does not exist DVDE\_UNAUTHORIZED Unauthorized operation.

#### Remarks

The Player accesses to the specified frame (time code) and starts playing. This function is only allowed for a One\_Sequential\_PGC\_Title. This function executes the Pre-Command of the Program Chain.

#### See also

TimeSearch(1.4.33), CDDAPlayTrack(1.2.6)

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks		•			
Windows CE		•			
Linux		•			

Figure 1.41: TimePlay Operating Systems and Streaming Models

### 1.4.33 TimeSearch

This command stops the current presentation and starts presentation from the specified position of the title by Time within the same title.

```
DWORD TimeSearch( dvdTimeCode);
```

#### Arguments

ULONG dvdTimeCode See details below

```

For DVD:
dvdTimeCode is a value defining the time to start playback in the
form of the DVD_TIMECODE structure as defined by the Microsoft DirectX API
typedef struct tagDVD_TIMECODE)
{
    ULONG Hours1    :4;

```

```

ULONG Hours10 :4;

ULONG Minutes1 :4;
ULONG Minutes10:4;

ULONG Seconds1 :4;
ULONG Seconds10:4;

ULONG Frames1 :4;
ULONG Frames10 :2;

ULONG FrameRateCode:2;
} DVD_TIMECODE;

```

DVD timecode is binary coded decimal (BCD) encoded in the format 0xHhMmSsFf, where:

```

H is tens of hours
h is hours
M is tens of minutes
m is minutes
S is tens of seconds
s is seconds
F is tens of frames
f is frames
To use a BCD, first create the BCD.

```

```

ULONG timeCode = 0; // create the BCD
//create a pointer to a DVD_TIMECODE structure at the address of the BCD
DVD_TIMECODE * dvdTimeCode = ( DVD_TIMECODE * ) &timeCode;

```

Then, dvdTimeCode can be used as a DVD\_TIMECODE structure.

```

dvdTimeCode->Hours10 = 0;
dvdTimeCode->Hours1 = 2;

```

For SVCD:

dvdTimeCode is an integer value which is the time to start playback expressed in number of seconds.  
For example inputing 600 means to start playback at 10 minutes.

#### Return value

##### DVD

Zero if successful. Otherwise returns an error code.

DVDE\_ARG Arguments are not valid.

DVDE\_UNAUTHORIZED Unauthorized operation.

##### SVCD

FMPE\_OK successful

FMPE\_NOINTERFACE this interface is not supported in this case

FMPE\_UNEXPECTED unknow errors



FMPE\_INVALIDARG the specified time is out of range

FMPE\_TIMERACTIVATED the timer has been activated

FMPE\_FAILURE SVCD is not currently playing

#### Remarks

The Player accesses to the specified frame (time code) and starts playing. This function is only allowed for a One\_Sequential\_PGC. This function does not execute the Pre-Command of the Program Chain.

#### See also

TimePlay(1.4.32), CDDAPlayTrack(1.2.6)

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks		•		•	
Windows CE		•		•	
Linux		•		•	

Figure 1.42: TimeSearch Operating Systems and Streaming Models

### 1.4.34 TitlePlay

Play the Title by the Title number for DVD and the track for VCD and SVCD.

```
DWORD TitlePlay (bTitleNumber);
```

#### Arguments

BYTE bTitleNumber Title Number to play The number is set between 1 and 99

#### Return value

Zero if successful or returns an error code

FMPGetLastError may be used to retrieve the last error code. The following are valid error codes from FMPGetLastError:

DVDE\_ARG Argument is not valid.

DVDE\_TITLENUMBER The Title number does not exist

DVDE\_UNAUTHORIZED Unauthorized operation.

#### Remarks

Jump to the specific Title number. Execute Pre-Command (for DVD)

#### See also

ChapterPlay(1.4.6), DVDI\_TT\_SRPTI(7)

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks		•	•	•	
Windows CE		•	•	•	
Linux		•	•	•	

Figure 1.43: TitlePlay Operating Systems and Streaming Models

### 1.4.35 TopPGSearch

Stop the current presentation and start the presentation from the beginning of the current Program within the same PGC. This relates to hierarchy of the program.

```
DWORD TopPGSearch ();
```

*Return value*

Zero if successful. Otherwise, returns an error code.  
DVDE.UNAUTHORIZED Unauthorized operation.

*See also*

GoUp(1.4.12), PrevPGSearch(1.4.19), NextPGSearch(1.4.17)

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks		•			
Windows CE		•			
Linux		•			

Figure 1.44: TopPGSearch Operating Systems and Streaming Models

### 1.4.36 UpperButtonSelect

Select the Buttons on the Menu Screen.

```
DWORD UpperButtonSelect();
```

*Return value*

Zero if successful. Otherwise returns an error code.  
DVDE.UNAUTHORIZED Unauthorized operation.

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks		•			
Windows CE		•			
Linux		•			

Figure 1.45: UpperButtonSelect Operating Systems and Streaming Models

## 1.5 General functions

### 1.5.1 FMPAbout

This function returns a version number, the date of compilation and a Copyright notice.

```
DWORD FMPAbout( TCHAR* sVersion, TCHAR* sDate, TCHAR *sCopyright);
```

*Remarks*

The version number returned is of the following format : *a.b.c.d*, where *a, b, c, d* are respectively the version, revision, build and release numbers.

The date of compilation returned is of the following format: *mm dd yyyy*. As an example: Dec 9 1999.

sVersion should point to a string 16-TCHAR-or-more long.

sDdate should point to a 12-TCHAR-or-more long one.

sCopyright should point to a 64-TCHAR-or-more long one.

No string-overflow checking is performed within the function (the string have to be allocated by the application).

Any of those three pointers can have a NULL value, if the information it provides is unwanted.

#### *ReturnValue*

Returns a 32 bit unsigned value. Typical return values might include one of the following :

FMPE_OK	No error
---------	----------

#### *Operating Systems and Streaming Models*

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks	•	•	•	•	•
Windows CE	•	•	•	•	•
Linux	•	•	•	•	•

Figure 1.46: FMPEAbout Operating Systems and Streaming Models

### 1.5.2 FMPClose

This function closes the driver

```
DWORD FMPClose ();
```

#### *ReturnValue*

Returns a 32 bit unsigned value. Typical return values might include one of the following :

FMPE_OK	No error
FMPE_DRIVER_NOT_OPEN	The driver is not open

See also

FMPOpen(1.7.3), FMPOpenDiscPlayback(1.5.5), MPEGDriverUnload(1.5.14)

#### *Operating Systems and Streaming Models*

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks	•	•	•	•	•
Windows CE	•	•	•	•	•
Linux	•	•	•	•	•

Figure 1.47: FMPClose Operating Systems and Streaming Models

### 1.5.3 FMPPGet

This command returns a value of specified setting for specified stream

```
DWORD FMPPGet (index);
```

*Arguments*

DWORD index Index of the value to get

FMPGet Indexs are:

00000002 FMPI\_STC  
00000003 FMPI\_PT  
00000004 FMPI\_AUDIO\_COUNT  
00000005 FMPI\_AUDIO\_SELECT  
00000006 FMPI\_NPT\_REFERENCE  
00000007 FMPI\_STC\_REFERENCE  
00000008 FMPI\_BRIGHTNESS  
00000009 FMPI\_SATURATION  
0000000A FMPI\_CONTRAST  
0000000B FMPI\_LEFT\_VOLUME  
0000000C FMPI\_RIGHT\_VOLUME  
0000000D FMPI\_MUTE  
0000000E FMPI\_AUDIO\_OUTPUT  
0000000F FMPI\_AUDIO\_MODE  
00000010 FMPI\_AUDIO\_VCX0  
00000011 FMPI\_VIDEOOUT  
00000012 FMPI\_SOURCE\_WINDOW  
00000013 FMPI\_VISIBLE\_SOURCE\_WINDOW  
00000014 FMPI\_DESTINATION\_WINDOW  
00000015 FMPI\_OVERLAY\_FLAGS  
00000016 FMPI\_VIDEO\_SPEED  
00000017 FMPI\_VIDEOPORT\_WINDOW  
00000018 FMPI\_VIDEOASPECT  
00000019 FMPI\_AUDIO\_SPEED  
00001001 FMPI\_DISC\_TYPE  
00001002 FMPI\_ANGLES\_AVAILABLE  
00001003 FMPI\_AUDIO\_STREAMS\_AVAILABLE  
00001004 FMPI\_BUTTONS\_AVAILABLE  
00001005 FMPI\_CAN\_SCAN  
00001006 FMPI\_CAN\_SEEK  
00001007 FMPI\_CURRENT\_ANGLE  
00001008 FMPI\_CURRENT\_AUDIO\_STREAM  
00001009 FMPI\_CURRENT\_BUTTON  
0000100A FMPI\_CURRENT\_CHAPTER  
0000100B FMPI\_CURRENT\_DOMAIN  
0000100C FMPI\_CURRENT\_POSITION  
0000100D FMPI\_CURRENT\_SUBPICT\_STREAM  
0000100E FMPI\_CURRENT\_TIME

```

0000100F FMPI_CURRENT_TITLE
00001010 FMPI_SUBPICTURE_ON
00001011 FMPI_SUBPICTURE_STREAM_AVAILABLE
00001012 FMPI_TITLES_AVAILABLE
00001013 FMPI_TOTAL_TITLE_TIME
00001014 FMPI_CURRENT_SPEED
00002000 FMPI_SVCD_TRICK_MODE
00002001 FMPI_CURRENT_TRACK
00002002 FMPI_SVCD_DISCONTINUITY
00002003 FMPI_SVCD_CURRENT_VOLUME
00002004 FMPI_SVCD_VOLUMES_AVAILABLE
00002005 FMPI_TRACKS_AVAILABLE
00002006 FMPI_TOTAL_TRACK_TIME

```

*Return value*

Value of the setting. Or 0xFFFFFFFF in case of error.  
 FMPLastError can be used to obtain error information about the command.

*Remarks*

The FMPLastError command allows you to get a stream setting. A NULL handle allows retrieval of a non-stream setting. The driver settings include information, status and default settings.

*See also*

Settings(7), FMPLastError(1.5.9)

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks	•	•	•	•	•
Windows CE	•	•	•	•	•
Linux	•	•	•	•	•

Figure 1.48: FMPLastError Operating Systems and Streaming Models

### 1.5.4 FMPLastErrorProfileWord

This function returns a value of specified setting for specified stream of local storage and dealing with registries.

```
DWORD FMPLastErrorProfileWord ( DWORD dwKeyIndex );
```

*Arguments*

DWORD dwKeyIndex dwKeyIndex of the value to get  
 dwKeyIndex are:

```

00 RMREG_LANGUAGECODE
01 RMREG_MENU_LANGUAGECODE
02 RMREG_AUDIO_LANGUAGECODE
03 RMREG_SUBPICTURE_LANGUAGECODE

```

04 RMREG\_SUBPICTURE\_DISPLAY\_MODE  
 05 RMREG\_USER\_ASPECT\_RATIO  
 06 RMREG\_USER\_VIDEO\_OUTPUT\_MODE  
 07 RMREG\_COUNTRYCODE  
 08 RMREG\_PARENTALCONTROL  
 09 RMREG\_PASSW\_LOW\_WORD  
 0A RMREG\_PASSW\_HIGH\_WORD  
 0B RMREG\_PASSW\_STATE  
 0c RMREG\_AUTOPLAY  
 0D RMREG\_PBC\_OFF  
 0E RMREG\_MACROVISION  
 0F RMREG\_REGION\_CODE  
 10 RMREG\_AUDIO\_TYPE  
 11 RMREG\_OUTPUTMODE  
 12 RMREG\_HDTV\_RES  
 13 RMREG\_MAX

*Return value*

Value of the setting. Or 0xFFFFFFFF in case of error.

FMPGetLastError can be used to obtain error information about the command.

*Remarks*

The FMPGetProfileWord command allows you to get a stream setting.

*See also*

Settings(7), FMPWriteProfileWord(1.5.12)

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks	•	•	•	•	•
Windows CE	•	•	•	•	
Linux	•	•	•	•	•

Figure 1.49: FMPGetProfileWord Operating Systems and Streaming Models

### 1.5.5 FMPOpenDiscPlayback

This command opens the MPEG driver in pull mode for Disc Playback.

```
DWORD FMPOpenDiscPlayback (PFMP_OPENSTRUCT pFMPOpenStruct);
```

*Arguments*

PFMP\_OPENSTRUCT pFMPOpenStruct Pointer to FMP\_OPENSTRUCT structure

```
typedef struct tagFMP_OPENSTRUCT {
DWORD   dwStructSize;    // Size of the structure
DWORD   dwFlags;         // Open Flag
DWORD   dwSize;          // Size of buffers
DWORD   dwCount;         // Number of buffers
PFMPCALLBACK pCallback;  // FMP Callback -see under variable types
DWORD   dwContext;       // Callback context
BYTE    bDriverNumber;   // Driver Number (not used : MPEGDriverEntry defines the driver number)
TCHAR*  sFileName;       // File Name (not used)
BYTE    bFileSystem;     // Disk data access method
BYTE    bDiscContent;    // Disc Content (returned by the driver)
} FMP_OPENSTRUCT, *PFMP_OPENSTRUCT;
```

Open Flag is:

- FMPF\_TRANSPORT Transport stream demux
- FMPF\_SYSTEM System stream demux
- FMPF\_PROGRAM Program stream demux
- FMPF\_VIDEO Video stream (MPEG1/2)
- FMPF\_MPEG\_AUDIO MPEG Audio stream
- FMPF\_AC3 AC3 Audio stream
- FMPF\_DVD DVD demux
- FMPF\_SVCD VCD/SVCD demux

Disk data access method is:

- SYS\_ACCESS using the OS function

Disc content is:

- FMPC\_DVD\_VIDEO DVD Video Disc
- FMPC\_VCD Video CD Disc
- FMPC\_SVCD Super Video CD Disc
- FMPC\_CDDA CD audio
- FMPC\_UNSUPPORTED Namely, CDI and HQ-VCD
- FMPC\_UNKNOWN Failure to recognize a known type of disc

Remarks

When FMPOpenDiscPlayback returned the content as FMPC\_VCD or FMPC\_SVCD  
FMPGet (FMPI\_DISC\_TYPE) will return

- 1 = VCD 1.1
- 2 = VCD 2.0
- 3 = SVCD

ReturnValue

Returns a 32 bit unsigned value. Typical return values might include one of the following :

FMPE_OK	No error
FMPE_DRIVER_OPEN	The driver is already open
FMPE_CANNOT_OPEN_DRIVER	Cannot open the MPEG driver
FMPE_NOT_ENOUGH_MEMORY	No more memory available

See also

FMPClose(1.5.2)

Operating Systems and Streaming Models

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks		•	•	•	•
Windows CE		•	•	•	
Linux		•	•	•	•

Figure 1.50: FMPOpenDiscPlayback Operating Systems and Streaming Models

### variable types

Here is Callback definition.

```
typedef DWORD (*PFMPCALLBACK) (DWORD dwContext, DWORD dwMsg, DWORD dwValue);
```

Here is FMP Message Structure:

```
typedef struct tagFMPmsg{
    DWORD dwCommand;
    DWORD dwArg1;
    DWORD dwArg2;
    DWORD dwArg3;
    DWORD dwArg4;
    DWORD dwArg5;
    BYTE pBuffer[sizeof (FMP_OPENSTRUCT) + _MAX_STRING_];
} FMP_MSG, *PFMP_MSG;
```

### 1.5.6 FMPPause

This command will PAUSE the video and audio decoders. The source can still send data before pause, but this data will not be decoded. This function can be used before a PLAY to prefill the driver.

```
DWORD FMPPause();
```

Remarks

This function is synchronous: when the call returns, the driver will be in paused mode.

ReturnValue

Returns a 32 bit unsigned value. Typical return values might include one of the following :

FMPE_OK	No error
FMPE_DRIVER_NOT_OPEN	The driver is not open

SeeAlso

FMPStop(1.5.10), FMPPplay(1.5.7)

Operating Systems and Streaming Models



OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks	•	•	•	•	•
Windows CE	•	•	•	•	
Linux	•	•	•	•	•

Figure 1.51: FMPPause Operating Systems and Streaming Models

### 1.5.7 FMPPPlay

This command starts the playback.

```
DWORD FMPPPlay ();
```

#### Remarks

This function is synchronous: when the call returns, the driver will be in play mode.

#### ReturnValue

Returns a 32 bit unsigned value. Typical return values might include one of the following :

FMPE_OK	No error
FMPE_DRIVER_NOT_OPEN	The driver is not open

#### SeeAlso

FMPPPause(1.5.6), FMPPStop(1.5.10)

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks	•	•	•	•	•
Windows CE	•	•	•	•	
Linux	•	•	•	•	•

Figure 1.52: FMPPPlay Operating Systems and Streaming Models

### 1.5.8 FMPPProperty

This function allows the user to get hardware-specific settings - see Realmagic HWLrc0.

```
DWORD FMPPProperty(DWORD dwFlags, DWORD dwPropSet, DWORD dwPropId, DWORD dwPropFlags, void* pPropData, DWORD dwPropSizeIn, DWORD* pdwPropSizeOut);
```

#### Arguments

DWORD dwFlags - flags to specify set or get function - see rm84cmn.h

DWORD dwPropSet - property set - see PROPERTY\_SETS enumeration in rm84cmn.h

DWORD dwPropId - property identifier - see enumerations in rm84cmn.h

DWORD dwPropFlags - reserved for future use

void\* pPropData - pointer to buffer data used to pass the information. Usually the buffer is a DWORD but there are few exceptions - see rm84cmn.h.

DWORD dwPropSizeIn - size of buffer in bytes - see rm84cmn.h

DWORD\* pdwPropSizeOut - pointer to DWORD. It can be NULL. If it is not NULL, receives the necessary buffer size for the specific property.

FMPPProperty dwFlags are:

KSPROPERTY\_TYPE\_GET

KSPROPERTY\_TYPE\_SET

Different types of property sets are:

```
typedef enum{
    REGISTRY_COMMON_SET=1, // registry set common for all board versions
    REGISTRY_HDTV_SET,      // registry set used only for hdtv mode
    REGISTRY_VGAENDOR_SET, // registry set used mostly for digital overlay
    SCANCONVERTER_SET,      // property set to access the scan converter
    EEPROM_SET,             // property set to access the scan converter
    BOARDINFO_SET,          // property set to access general properties of the board
    VIDEO_SET,              // property set to access video properties of the board
    AUDIO_SET,              // property set to access audio properties of the board
    PROPERTY_MAX_SET        // property set to maximum
}PROPERTY_SETS;
```

REGISTRY\_COMMON\_SET use DWORD. ID of property sets are:

```
typedef enum{
    eTvOut,
    eTvMRO,
    eDoHwReset,
    eDisableSpdifOutputInReset,
    eActiveVideoWidthNtsc,
    eActiveVideoWidthPal,
    eForcedProgressiveSourceOff,
    eForcedProgressiveAlways,
    eNtscPalFrameDrop,
    eVGAForcedInterlaced,
    eAudioOutput,
    eVolumeRight,
    eVolumeLeft,
    eDacType,
    eAudioDacBitsPerSample,
    eDoAudioLater,
    eBrightness,
    eContrast,
    eSaturation,
    eTvBrightness,
    eTvContrast,
    eTvSaturation,
    eDResBitsPerPixel,
    eDResScreenWidth,
    eDResScreenHeight,
    ePreviousHFreq,
    eTotalPixelsPerLine,
    eAcpiEnable,
    eDecoderIsSlave,
    eZoomEnable,
```

```
eWindowTvEnable,
eWindowHdtvEnable,
eOsdVideoIndependent,
eMaximumDvclk,
eCommonRegMax
}REGISTRY_COMMON;
```

REGISTRY\_HDTV set uses DWORD

```
typedef enum{
eHdtvHFreq,
eHdtvVFreq,
eHdtvVideoWidth,
eHdtvVideoHeight,
eHdtvHSyncTotal,
eHdtvPreHSync,
eHdtvHSyncActive,
eHdtvPostHSync,
eHdtvVSyncTotal,
eHdtvPreVSync,
eHdtvVSyncActive,
eHdtvPostVSync,
eHdtvPixelFreq,
eHdtvInterlaced,
eHdtvRegMax
}REGISTRY_HDTV;
```

REGISTRY\_VGA\_VENDOR set uses DWORD

```
typedef enum{
eInvertField,
eVmi_16bits,
eCcir_656,
eSyncEnable,
eVip20,
eVgaVendorRegMax
}REGISTRY_VGA_VENDOR;
```

SCANCONVERTER set uses DWORD, few exceptions apply.

```
typedef enum{
escAccessRegister,      // uses RM_WRITE
escResetDefault,
escBrightness,
escContrast,
escSaturation,
escSharpness,
escFlicker,
escChromaFilter,
escLumaFilter,
escNtscPedestal,
escMacrovision,
escTvStandard,
escOutputFormat,
escScart,
escHwOutputHorzOffsShadow,
```

```

escHwOutputVertOffsShadow,
escTVPixels,
escTVLines,
escHorizontalPositionOffset,
escVerticalPositionOffset,
escHorizontalScaleStep,
escVerticalScaleStep,
escHorizontalPanPosition,
escVerticalPanPosition,
escZoom,
escScanConverterMax
}SCANCONVERTER;

```

EEPROM\_SET use DWORD, few exceptions apply.

```

typedef enum{
eEepromAccess, // uses RM_WRITE
eEepromMax
}EEPROM;

```

BOARDINFO\_SET uses DWORD.

```

typedef enum{
ebiDeviceId,
ebiSubId,
ebiVersion,
ebiAPMState,
ebiPIOAccess, //access to PIO - danger - GET changes PIO in input, SET changes PIO in output
eBoardInfoMax
}BOADDINFO;

```

VIDEO\_SET uses DWORD

```

typedef enum{
evMax
}VIDEO_ENUM;

```

AUDIO\_SET uses DWORD

```

typedef enum{
eAudioFineVcxo,
eAudioPllCoef,
eaMax
}AUDIO_ENUM;

```

*Return value*

Zero if successful. Otherwise returns an error code.

*Remarks*

Here is a small sample code for usage:

```

DWORD get_TvBrightness(short *pVal)
{
    DWORD sizeOut, fVal;

```

```

    if (pVal == NULL)
        return NOTGOOD;

    //Get the focus TV output to new standard

    if (FMPPProperty (KSPROPERTY_TYPE_GET, SCANCONVERTER_SET, escBrightness, 0, &fVal,
        sizeof(fVal), &sizeOut) == OK){
        if(sizeOut != sizeof(fVal))
            return NOTGOOD;
        *pVal = (short) fVal;
    }
    else{
        *pVal = BRT_RESET;
        return NOTGOOD;
    }
    return OK;
}

```

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks	•	•	•	•	•
Windows CE	•	•	•	•	
Linux	•	•	•	•	•

Figure 1.53: FMPPProperty Operating Systems and Streaming Models

### 1.5.9 FMPSet

This command will initialize the stream or the driver with specified setting.

```
DWORD FMPSet (index, value);
```

#### Arguments

DWORD index Index of the value to set or zero for nothing that can be combined with a flag

DWORD value New value for the setting

#### Return value

Returns the previous value of the given setting if successful otherwise returns 0xFFFFFFFF in case of error.

#### Remarks

The FMPSet command allows you to set a parameter of a stream or the driver if you specify a null handle. The driver settings include information, status and default settings.

You can specify audio or video settings for the driver. In this case, the values set in the driver will be the default values, for any future opened streams.

For FMPSet Index please see under FMPPGet Index.

#### See also

FMPPGet(1.5.3), Settings7

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks	•	•	•	•	•
Windows CE	•	•	•	•	•
Linux	•	•	•	•	•

Figure 1.54: FMPSet Operating Systems and Streaming Models

### 1.5.10 FMPSStop

This command stops the playback.

```
DWORD FMPSStop ();
```

#### Remarks

This function is synchronous: when the call returns, the driver will be in stop mode.

Using the push model (FMPSOpen), it is possible to resume playback after a FMPSStop (using FMPSPlay).

Using the disc playback model (FMPSOpenDiscPlayback), it is not possible to resume playback after a FMPSStop. You have to call FMPSClose after using FMPSStop.

#### ReturnValue

Returns a 32 bit unsigned value. Typical return values might include one of the following :

FMPE_OK	No error
FMPE_DRIVER_NOT_OPEN	The driver is not open

#### See also

FMPSPause(1.5.6), FMPSPlay(1.5.7)

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks	•	•	•	•	•
Windows CE	•	•	•	•	•
Linux	•	•	•	•	•

Figure 1.55: FMPSStop Operating Systems and Streaming Models

### 1.5.11 FMPSTrayOperation

This command allows you to change the status of DVDROM tray operation.

```
DWORD FMPSTrayOperation ( DWORD dwFMPSAction);
```

#### Arguments

DWORD dwFMPSAction Tray actions can put

Tray actions are:

- 10 FMPS\_TRAYCHECK
- 11 FMPS\_TRAYCHANGE
- 1 FMPS\_TRAYCLOSE
- 0 FMPS\_TRAYOPEN

*ReturnValue*

Returns a 32 bit unsigned value. Typical return values might include one of the following :

FMPE_OK	No error
FMPE_DRIVER_OPEN	The driver is already open

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks		•	•	•	•
Windows CE		•	•	•	
Linux		•	•	•	•

Figure 1.56: FMPTrayOperation Operating Systems and Streaming Models

### 1.5.12 FMPWriteProfileWord

This function will initialize the stream or the driver with specified setting of local storage and dealing with registries.

```
DWORD FMPWriteProfileWord(DWORD dwKeyIndex, DWORD wValue);
```

*Arguments*

DWORD dwKeyIndex dwKeyIndex of the value to write

DWORD wValue New value for the setting

dwKeyIndex are:

- 00 RMREG\_LANGUAGECODE
- 01 RMREG\_MENU\_LANGUAGECODE
- 02 RMREG\_AUDIO\_LANGUAGECODE
- 03 RMREG\_SUBPICTURE\_LANGUAGECODE
- 04 RMREG\_SUBPICTURE\_DISPLAY\_MODE
- 05 RMREG\_USER\_ASPECT\_RATIO
- 06 RMREG\_USER\_VIDEO\_OUTPUT\_MODE
- 07 RMREG\_COUNTRYCODE
- 08 RMREG\_PARENTALCONTROL
- 09 RMREG\_PASSW\_LOW\_WORD
- 0A RMREG\_PASSW\_HIGH\_WORD
- 0B RMREG\_PASSW\_STATE
- 0c RMREG\_AUTOPLAY
- 0D RMREG\_PBC\_OFF
- 0E RMREG\_MACROVISION
- 0F RMREG\_REGION\_CODE
- 10 RMREG\_AUDIO\_TYPE
- 11 RMREG\_OUTPUTMODE
- 12 RMREG\_HDTV\_RES
- 13 RMREG\_MAX

*Return value*

Value of the setting. Or 0xFFFFFFFF in case of error.

FMPGetLastError can be used to obtain error information about the command.

*Remarks*

The FMPWriteProfileWord command allows you to write a parameter of a stream or the driver. Need to restart the application to take affect the new setting.

*See also*

Settings(7),FMPGetProfileWord(1.5.4)

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks	•	•	•	•	•
Windows CE	•	•	•	•	•
Linux	•	•	•	•	•

Figure 1.57: FMPWriteProfileWord Operating Systems and Streaming Models

### 1.5.13 MPEGDriverEntry

Initialize the MPEG driver. This function has to be called before FMPOpen.

```
DWORD MPEGDriverEntry(BYTE bDriveNumber);
```

*Arguments*

BYTE bDriveNumber is the DVD Drive Number:

- 0 Primary master
- 2 Secondary master
- NO\_DRIVE Do not attempt DVD or CDRom drive access

*Remarks*

NO\_DRIVE parameter is useful for instance when playing files from hard disk.

*ReturnValue*

Returns a 32 bit unsigned value. Typical return values might include one of the following :

FMPE_OK	No error
FMPE_ENTRY_MOD_INIT_FAILED	Module loader failed to initialize
FMPE_ENTRY_HWL_INIT_FAILED	Failed to create MPEG Hardware driver
FMPE_ENTRY_DVDDEV_INIT_FAILED	Failed to create DVD-ROM device driver

*Operating Systems and Streaming Models*

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks	•	•	•	•	•
Windows CE	•	•	•	•	•
Linux	•	•	•	•	•

Figure 1.58: MPEGDriverEntry Operating Systems and Streaming Models



### 1.5.14 MPEGDriverUnload

Prepare the MPEG driver to be unloaded. This function has to be called after FMPClose.

```
DWORD MPEGDriverUnload ();
```

*ReturnValue*

Returns a 32 bit unsigned value. Typical return values might include one of the following :

FMPE_OK	No error
---------	----------

*Operating Systems and Streaming Models*

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks	•	•	•	•	•
Windows CE	•	•	•	•	•
Linux	•	•	•	•	•

Figure 1.59: MPEGDriverUnload Operating Systems and Streaming Models

### 1.5.15 FMPQueryInterface

Use this function to get a pointer on the OSD (ON-SCREEN\_DISPLAY) Interface.

```
DWORD FMPQueryInterface (DWORD InterfaceId, void **ppv);
```

*Arguments*

DWORD InterfaceId –Identifier of the interface

void \*\*ppv –Address of pointer variable that receives the interface pointer requested in InterfaceId

Identifier of the interface:

```
IID_IOSD      0x00000000
IID_IHDTV     0x00000001
IID_IUserI2C  0x00000002
IID_IGPIO     0x00000003
```

*ReturnValue*

Returns a 32 bit unsigned value. Typical return values might include one of the following :

FMPE_OK	No error
FMPE_DRIVER_NOT_OPEN	The driver is not open

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks	•	•	•	•	•
Windows CE	•	•	•	•	•
Linux	•	•	•	•	•

Figure 1.60: FMPQueryInterface Operating Systems and Streaming Models

## 1.6 Picture placement

### 1.6.1 FMPAnalogOverlay

This function allows the user to put and set the video image onto VGA monitor.(Linux only)

```
DWORD FMPAnalogOverlay((DWORD SubCommand, DWORD dwArg1, DWORD dwArg2, DWORD dwArg3, DWORD dwArg4,
DWORD dwArg5);
```

#### Arguments

DWORD SubCommand - AnalogOverlay SubCommand

DWORD dwArg1 - Argument

DWORD dwArg2 - Argument

DWORD dwArg3 - Argument

DWORD dwArg4 - Argument

DWORD dwArg5 - Argument

FMPAnalogOverlay SubCommand are:

1000 FMP\_ANALOG\_OVERLAY\_ACCESS

dwArg1=0 pseudo-close

dwArg1=1 pseudo-open

1001 FMP\_ANALOG\_OVERLAY\_GET

1002 FMP\_ANALOG\_OVERLAY\_SET

1003 FMP\_ANALOG\_OVERLAY\_AUTOCALIBRATION

dwArg2=0x88 YOffset autocalibration (horizontal at top pattern)

dwArg2=0x89 XOffset autocalibration (vertical at left pattern)

dwArg2=0x8a Correction autocalibration (vertical offset right pattern)

dwArg2=0x80 Color (WHITE) autocalibration (full white vertical bar pattern)

dwArg2=0x82 Color (GREY) autocalibration (half white vertical bar pattern)

FMPAnalogOverlay General commands for dwArg1 are:

202 FMP\_ANALOG\_OVERLAY\_ATTRIBUTE\_VGAKEY

vgakey selection (dwArg2=Rii16+Gii8+B. indexed mode is not supported)

203 FMP\_ANALOG\_OVERLAY\_ATTRIBUTE\_DISPLAYRESOLUTION

send dwArg2=depth (8,16,24,32), dwArg3=physical display width, dwArg4=physical display height

204 FMP\_ANALOG\_OVERLAY\_ATTRIBUTE\_MODE

mode selection (dwArg2=0(none),1(rectangle),2(overlay))

205 FMP\_ANALOG\_OVERLAY\_ATTRIBUTE\_OVERLAY\_CABLE\_PRESENT

get only: 1=yes, 0=no

FMPAnalogOverlay fine-tuning commands for dwArg1 are:

301 FMP\_ANALOG\_OVERLAY\_ATTRIBUTE\_XOFFSET

302 FMP\_ANALOG\_OVERLAY\_ATTRIBUTE\_YOFFSET

303 FMP\_ANALOG\_OVERLAY\_ATTRIBUTE\_VGACORRECTION  
 304 FMP\_ANALOG\_OVERLAY\_ATTRIBUTE\_VGARUPPER  
 305 FMP\_ANALOG\_OVERLAY\_ATTRIBUTE\_VGARLOWER  
 306 FMP\_ANALOG\_OVERLAY\_ATTRIBUTE\_VGAGUPPER  
 307 FMP\_ANALOG\_OVERLAY\_ATTRIBUTE\_VGAGLOWER  
 308 FMP\_ANALOG\_OVERLAY\_ATTRIBUTE\_VGABUPPER  
 309 FMP\_ANALOG\_OVERLAY\_ATTRIBUTE\_VGABLOWER  
 30a FMP\_ANALOG\_OVERLAY\_ATTRIBUTE\_FINEADJUSTMENT  
 30b FMP\_ANALOG\_OVERLAY\_ATTRIBUTE\_HFREQUENCY  
 30c FMP\_ANALOG\_OVERLAY\_ATTRIBUTE\_JITTERADJUSTMENT

#### Remarks

This function is not implemented for Windows CE. Here is a example code for usage:

For open connection

```
FMPAnalogOverlay(FMP_ANALOG_OVERLAY_ACCESS,1,0,0,0,0);
```

For close the connection

```
FMPAnalogOverlay(FMP_ANALOG_OVERLAY_ACCESS,0,0,0,0,0);
```

For set mode to overlay

```
FMPAnalogOverlay(FMP_ANALOG_OVERLAY_SET,FMP_ANALOG_OVERLAY_ATTRIBUTE_MODE,2,0,0,0);
```

For set vgacorrection to 1000

```
FMPAnalogOverlay(FMP_ANALOG_OVERLAY_SET,FMP_ANALOG_OVERLAY_ATTRIBUTE_VGACORRECTION,1000,0,0,0);
```

For save XOffset

DWORD param;

DWORD FMPreturn;

```
param=FMPAnalogOverlay(FMP_ANALOG_OVERLAY_GET,FMP_ANALOG_OVERLAY_ATTRIBUTE_XOFFSET,0,0,0,0);
```

```
FMPreturn=FMPWriteProfileWord(RMREG_XOFFSET,param);
```

### 1.6.2 FMPSetDestination

Allow choosing coordinates of destination image if possible (depends on video output choice)  
(Linux, Windows CE and VxWorks)

```
FMPSetDestination(DWORD x,DWORD y,DWORD w,DWORD h)
```

#### Arguments

DWORD x Left horizontal coordinate

DWORD y Top vertical coordinate

DWORD w Horizontal length

DWORD h Vertical length

*Return value*

Zero if successful. Otherwise returns an error code.

*See also*

FMPSet with FMPI\_DESTINATION\_WINDOW (1.5.9)

### 1.6.3 FMPSetSource

This function allows the user to set a small rectangle area inside the source area as the play back source area. The coordinates are relative to the source window (usually 720\*480 for MPEG2 NTSC).  
(Linux, Windows CE and VxWorks)

```
DWORD FMPSetSource(DWORD dwX, DWORD dwY, DWORD dwcX, DWORD dwcY);
```

*Arguments*

DWORD dwX- Left horizontal coordinate

DWORD dwY- Top vertical coordinate

DWORD dwcX- Horizontal length

DWORD dwcY- Vertical length

*Remarks*

Usually dwX, dwY are ignored. They are considered always 0. dwcX, dwcY should be the width, height of Mpeg clip decoded. The settings are used only for zoom feature. By default the source size is (dwX, dwY, dwcX, dwcY) = (0, 0, 720, 480), for Mpeg2 Ntsc.

*See also*

FMPSet with FMPI\_SOURCE\_WINDOW (1.5.9)

### 1.6.4 FMPSetVisibleSource

This function defines a rectangle inside the video source (smaller or equal), that will be zoomed (up-scaled) to the entire destination window. ( Linux, Windows CE and VxWorks)

```
DWORD FMPSetVisibleSource(DWORD dwX, DWORD dwY, DWORD dwcX, DWORD dwcY);
```

*Arguments*

DWORD dwX- Left horizontal coordinate

DWORD dwY- Top vertical coordinate

DWORD dwcX- Horizontal length

DWORD dwcY- Vertical length

*Remarks*

This function is used to zoom the video, if the registry entry permits ( "ZoomEnable = 1" ). The following feature is to call when another way to enable/disable zoom and tvdestination.

FMPSet(FMPI\_OVERLAY\_FLAGS, flags);, where flags is a combination of:

VIDEO\_ZOOM\_ENABLE 0x0001  
 VIDEO\_TV\_DEST\_ENABLE 0x0002  
 VIDEO\_HDTV\_DEST\_ENABLE 0x0004  
 OSD\_VIDEO\_INDEPENDENT\_DEST 0x0008

*See also*

FMPSet with FMPI\_VISIBLE\_SOURCE\_WINDOW (1.5.9)

## 1.7 Push model functions

### 1.7.1 FMPFlush

This function flushes internal Fifos of the driver

`DWORD FMPFlush (BOOL bGDF);`

*Arguments*

BOOL bGDF If TRUE, discard any incoming data till the next section

*Return value*

Zero if successful. Otherwise returns an error code.

*Remarks*

This function can be used to accelerate seek time (after a seek you can flush the internal buffers of the driver to accelerate the display time).

*See also*

FMPPush(1.7.4)

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks	•				
Windows CE	•				
Linux	•				

Figure 1.61: FMPFlush Operating Systems and Streaming Models

### 1.7.2 FMPGetBuffer

This function gets a buffer for the memory Manager of the driver

`DWORD FMPGetBuffer (PFMP_BUFFER pBuffer, BOOL bBlockingCall);`

*Arguments*

PFMP\_BUFFER pBuffer Pointer to a FMP\_BUFFER structure:

BOOL bBlockingCall If set to TRUE and no more memory is available in the Memory Manager, this call will block till some buffer gets freed up.

If set to FALSE, and no more memory is available in the Memory Manager, this call will return an error code.

#### ReturnValue

Returns a 32 bit unsigned value. Typical return values might include one of the following :

FMPE_OK	No error
FMPE_DRIVER_NOT_OPEN	The driver is not open
FMPE_NOT_ENOUGH_MEMORY	All the memory of the memory manager is used

#### See also

FMPPush for FMP\_BUFFER declaration(1.7.4), Streaming Video.

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks	•				
Windows CE	•				
Linux	•				

Figure 1.62: FMPGetBuffer Operating Systems and Streaming Models

### 1.7.3 FMPOpen

This command opens the MPEG driver in push mode.

DWORD FMPOpen (dwflags, dwSize, dwCount, pCallback, dwContext);

#### Arguments

DWORD flags Specifies initial demux and stream playback mode on Open

FMPF\_TRANSPORT Transport stream demux (push model)

FMPF\_SYSTEM System stream demux (push model)

FMPF\_SYSTEM Program stream demux (push model)

FMPF\_VIDEO Video stream (MPEG1/MPEG2) (push model)

FMPF\_MPEG\_AUDIO MPEG Audio stream (push model)

FMPF\_AC3 AC3 Audio stream (push model)

FMPF\_DVD DVD Demux (push model)

FMPF\_SVCD SVCD/VCD Demux (push model)

FMPF\_CDDA CDDA demux (push model)

FMPF\_PES pes stream support (push model)

DWORD dwSize Size of buffers (Memory Manager)

DWORD dwCount Number of buffers to allocate. The total size of the Memory Manager is dwSize \* dwCount.

FUNCPTR pCallback User Callback to receive driver notifications.

DWORD dwContext Callback Context (reported when the callback is called)

#### Remarks

This command open the driver in push mode. The application has to feed the driver with data. Please refer to the streaming video section and the sample code for the details on the push model.

#### ReturnValue

Returns a 32 bit unsigned value. Typical return values might include one of the following :

FMPE.OK	No error
FMPE.DRIVER_OPEN	The driver is already open
FMPE.CANNOT_OPEN_DRIVER	Cannot open the MPEG driver
FMPE.NOT_ENOUGH_MEMORY	No more memory available

#### See also

FMPClose(1.5.2)

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks	•				
Windows CE	•				
Linux	•				

Figure 1.63: FMPOpen Operating Systems and Streaming Models

### 1.7.4 FMPPush

This function pushes data into the driver to be decoded

`DWORD FMPPush (PFMP_BUFFER pBuffer);`

#### Arguments

PFMP\_BUFFER pBuffer Pointer to a FMP\_BUFFER structure

```
typedef struct {
void *pBuffer;           // Address of the buffer
DWORD dwBufferSize;      // Size of the buffer
DWORD dwDataSize;        // Size of valid data in the buffer
DWORD dwFlags;           // Flags
DWORD dwFlagsEx;         // Extended flags (used for PES support)
DWORD pReserved[8];      // Reserved
} FMP_BUFFER, *PFMP_BUFFER;
```

Flags can be :

```
FMP_DATADISCONTINUITY = 0x1, // Data discontinuity : data loss
FMP_TIMEDISCONTINUITY = 0x2, // PTS discontinuity
FMP_TRICKMODE_START   = 0x4, // Trick Mode Starts
FMP_TYPECHANGED       = 0x8, // Type of data has changed
FMP_TIMEVALID         = 0x10, // PTS is valid
FMP_ENDOFSTREAM       = 0x20, // End of stream
FMP_TRICKMODE_END     = 0x40, // Trick Mode Ends
FMP_TRICKMODE_SAMPLE = 0x80, // A Trick Mode Sample
FMP_SVCD_TM_2X        = 0x100, // SVCD trick mode speed 2x
FMP_SVCD_TM_3X        = 0x200, // SVCD trick mode speed 3x
FMP_SVCD_TM_4X        = 0x400, // SVCD trick mode speed 4x
```

```

FMP_SVCD_FR = 0x1000,
FMP_SVCD_AUTOPAUSE = 0x2000,
FMP_SVCD_SEQ_HDR = 0x4000,
FMP_SVCD_UPDATE_PTS_REF = 0x8000,

FMP_TRICKMODE_DVD = 0x10000,
FMP_SEEKINGOP_DVD = 0x20000,
FMP_SVCD_CDDA_TRACK = 0x40000, // CDDA track on the VCD disk

```

Extended flags are:

```

FMP_VIDEO_PES = 0x1,
FMP_AUDIO_PES = 0x2,
FMP_DVD_AUDIO_PES = 0x4,
//Please see details PES Support under Chapter 8 (Streaming video).

```

#### Remarks

You can push data only in Paused and Play mode. You cannot push data in stop mode. The Buffer you push must be allocated by the Memory Manager of the driver (you must get it using FMPPush). This call is supposed to be asynchronous: it should return very quickly.

#### ReturnValue

Returns a 32 bit unsigned value. Typical return values might include one of the following :

FMPE_OK	No error
FMPE_DRIVER_NOT_OPEN	The driver is not open
FMPE_PUSH_WHILE_STOPPED	Pushing data in stopped state

#### See also

FMPPush(1.7.2)

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks	•				
Windows CE	•				
Linux	•				

Figure 1.64: FMPPush Operating Systems and Streaming Models



## Chapter 2

# Error messages

Name	Description	Codes
FMPE_OK	No error	0x00000000
FMPE_ERROR	Error: use getLastError to get the Error code	0xFFFFFFFF
FMPE_DRIVER_OPEN	Driver is already open.	0x00000001
FMPE_DRIVER_NOT_OPEN	Driver is not open.	0x00000002
FMPE_CANNOT_OPEN_DRIVER	Cannot open the MPEG driver.	0x00000003
FMPE_NOT_ENOUGH_MEMORY	There is no more memory available.	0x00000004
FMPE_NOT_IMPLEMENTED	This function is not implemented.	0x00000005
FMPE_INVALID_INDEX	The setting is invalid index.	0x00000006
FMPE_READ_ONLY	The setting is only for reading.	0x00000007
FMPE_PUSH_WHILE_STOPPED	Pushing data in stopped state.	0x00000008
FMPE_INVALID_PARAMETER	Indicates that using invalid parameter.	0x00000009
FMPE_NO_MPEG_SERVER	There is no MPEG Server running.	0x0000000A
FMPE_NO_ANALOG_OVERLAY_CAPABILITY	There is no analog overlay.	0x0000000B
FMPE_INVALID_STREAM_TYPE	This function does not support the current stream type.	0x0000000C
FMPE_ENTRY_MOD_INIT_FAILED	MPEGDriverEntry failed because module loader failed to initialize.	0x0000000D
FMPE_ENTRY_HWL_INIT_FAILED	MPEGDriverEntry Failed because failed to create MPEG Hardware driver.	0x0000000E
FMPE_ENTRY_DVDDEV_INIT_FAILED	MPEGDriverEntry Failed because failed to create DVD-ROM device driver.	0x0000000F
FMPE_CDDA_FAILURE	CDDANextTrack failure or end of disk or beginning of disk has been reached.	0x00000010
FMPE_UNEXPECTED	Unexpected unknown errors for VCD and SVCD.	0x00000011
FMPE_DISABLED	The function is currently disabled.	0x00000012
FMPE_INVALIDARG	The list offset requested is out of range.	0x00000013
FMPE_TIMERACTIVATED	The timer has been activated.	0x00000014
FMPE_FAILURE	Indicates for VCD and SVCD failure.	0x00000015



## Chapter 3

### Events

DVD event notification code	Description	Status
FMPM_DVD_ANGLE_CHANGE	Signals that either the number of available angles changed or that the current user angle number changed	Implemented
FMPM_DVD_ANGLES_BLOCK	Indicates whether an angle block is being played and angle changes can be performed	Implemented
FMPM_DVD_AUDIO_STREAM_CHANGE	Signals that the current user audio stream number changed	Implemented
FMPM_DVD_BUTTONS_CHANGE	Signals that number of available buttons changed	Implemented
FMPM_DVD_BUTTON_CHANGE	Signals that the currently selected button number changed	Implemented
FMPM_DVD_PROGRAM_START	Signals that the DVD player started playback of a new program	Implemented
FMPM_DVD_CHAPTER_CHANGE	Signals that a new chapter is being played	Implemented
FMPM_DVD_CURRENT_TIME	Signals the beginning of every video object unit (VOBU), which occurs every 0.4 to 1.0 seconds	Implemented
FMPM_DVD_DOMAIN_CHANGE	Indicates the DVD player's new domain	Implemented
FMPM_DVD_NO_FP_PGC	Indicates that the DVD disc does not have a FP_PGC (First Play Program Chain)	Implemented
FMPM_DVD_MACROVISION_LEVEL	Indicates a new Macrovision level has been set	Implemented
FMPM_DVD_PARENTAL_LEVEL_CHANGE	Signals that the authored content has changed the parental level setting in the player	Implemented
FMPM_DVD_PARENTAL_CHECK	Signals that the player should query the user for the parental check password	Implemented
FMPM_DVD_PARENTAL_ERROR	Signals that a parental level error has been detected	Implemented
FMPM_DVD_PLAYBACK_STOPPED	Indicates that playback has been stopped. The DVD Navigator has completed playback of the PGC and did not find any other branching instruction for subsequent playback	Implemented
FMPM_DVD_STILL_OFF	Signals the end of any still (PGC, Cell, or VOB)	Implemented
FMPM_DVD_STILL_ON	Signals the beginning of any still (PGC, Cell, or VOB)	Implemented
FMPM_DVD_SUBPICTURE_STREAM_CHANGE	Signals that the current user subpicture stream number changed for the main title	Implemented
FMPM_DVD_TITLE_CHANGE	Indicates when the current title number changes	Implemented
		to be cr'd. . .

DVD event notification code	Description (ct'd)	Status
FMPM_DVD_VALID_UOPS_CHANGE	Signals that the available set of UOPS has changed	Implemented
FMPM_ERROR	Specifies that an error occurred while executing a command.	Implemented
FMPM_STARVATION	Message issued when the video decoder is starving.	Implemented
FMPM_EOS	Signals that the video and audio decoders have finished decoding the stream.	Implemented
FMPM_DVDROM_NOT_READY	Signals that DVDROM Drive is open or is not ready to play the play back.	Implemented
FMPM_AUTOPAUSE	Indicates that decoder enter into auto pause mode due to SVCD Navigation State.	Implemented
FMPM_REGION_MISMATCH	Signals that DVD has invalid region code.	Implemented
FMPM_TRICK_MODE_CHANGE	Signals that finished scanning or scanning is stopped.	Implemented
FMPM_DVD_END_PLAYBACK	Indicates that DVD play back reaches to the end.	Implemented
FMPM_DVDROM_ERR	Indicates that DVDROM has problem for reading or playing DVD.	Implemented
FMPM_DVD_FATAL_ERROR	Indicates that DVD cannot read the disc information.	Implemented
FMPM_SVCD_PSD_END	Indicates that SVCD reaches to the end of the playback.	Implemented
FMPM_CDDA_PSD_END	Indicates that AUDIO CD reaches to the end of the playback.	Implemented

### 3.1 FMPM\_AUTOPAUSE

Indicates that decoder enter into auto pause mode due to SVCD Navigation State.

#### Remarks

All domains raise this event.

### 3.2 FMPM\_CDDA\_PSD\_END

Indicates that AUDIO CD reaches to the end of the playback.

#### Remarks

All domains raise this event.

### 3.3 FMPM\_DVD\_ANGLE\_CHANGE

Signals that the current user angle number has changed.

#### Arguments

DWORD dwValue Value indicating the current user angle number

#### Remarks

Angle numbers range from 1 to 9. The current angle number can change automatically with a navigation command authored on the disc as well as through application control by using the DVDAngleChange function. All domains raise this event.

### 3.4 FMPPM\_DVD\_ANGLES\_BLOCK

Indicates whether an angle block is being played and angle changes can be performed.

#### *Arguments*

DWORD dwValue Boolean (BOOL) value that indicates if an angle block is being played back. Zero (0) indicates that playback is not in an angle block and angles are not available, One (1) indicates that an angle block is being played back and angle changes can be performed.

#### *Remarks*

Angle changes are not restricted to angle blocks and the manifestation of the angle change can be seen only in an angle block.

### 3.5 FMPPM\_DVD\_AUDIO\_STREAM\_CHANGE

Signals that the current user audio stream number changed for the main title.

#### *Arguments*

DWORD dwValue Value indicating the new user audio stream number. Audio stream numbers range from 0 to 7. Stream 0xFFFFFFFF indicates that no stream is selected.

#### *Remark*

The current user audio stream can change automatically with a navigation command authored on the disc as well as through application control by using the DVDSelectAudioStream function. All domains raise this event.

### 3.6 FMPPM\_DVD\_BUTTON\_CHANGE

Signals that the currently selected button number has changed.

#### *Arguments*

DWORD dwValue Value indicating the new button number.

#### *Remarks*

Button numbers range from 1 to 36.

The currently selected button can change automatically with a navigation command authored on the disc as well as through application control by using DVDSelectButton function. All domains raise this event.

### 3.7 FMPPM\_DVD\_BUTTONS\_CHANGE

Signals that the number of available buttons changed.

#### *Arguments*

DWORD dwValue Value indicating the number of available buttons

#### *Remarks*

Button numbers range from 1 to 36.

All domains raise this event.

### 3.8 FMPPM\_DVD\_CHAPTER\_CHANGE

Signals that the current chapter has changed.

#### Arguments

DWORD dwValue Value indicating the new chapter number.

#### Remarks

All domains raise this event.

### 3.9 FMPPM\_DVD\_CURRENT\_TIME

Signals the beginning of every video object unit (VOBU), which occurs every 0.4 to 1.0 seconds.

#### Arguments

DWORD dwValue Value indicating the current playback timecode in a binary coded decimal (BCD) hours, minutes, seconds, frames (HH:MM:SS:FF) format

DWORD dwValue Value indicating the current playback timecode in a binary coded decimal (BCD) hours, minutes, seconds, frames (HH:MM:SS:FF) format.

#### Remarks

Only simple linear movies signal this event.

The DVD\_TITLE\_DOMAIN domain raises this event.

### 3.10 FMPPM\_DVD\_DOMAIN\_CHANGE

Indicates the DVD player's new domain.

#### Arguments

DWORD dwValue Value indicating the new domain.

#### Remarks

The DVD player signals this event whenever it changes domains. All domains raise this event.

### 3.11 FMPPM\_DVD\_END\_PLAYBACK

Indicates that DVD play back reaches to the end.

#### Remarks

All domains raise this event.

### 3.12 FMPPM\_DVD\_FATAL\_ERROR

Indicates that DVD cannot read the disc information.

#### Remarks

All domains raise this event.

### 3.13 FMPM\_DVD\_MACROVISION\_LEVEL

Signals that a new Macrovision level has been set.

#### *Arguments*

DWORD dwValue Value representing the Macrovision level that has been set.

#### *Remarks*

This event indicates that a new Macrovision level for the movie has been set. All domains raise this event.

### 3.14 FMPM\_DVD\_NO\_FP\_PGC

Signals that the DVD disc does not have a FP\_PGC (First Play Program Chain) and that the DVD Navigator will not automatically load any PGC and start playback.

#### *Remarks*

The FP\_DOM domain raises this event.

### 3.15 FMPM\_DVD\_PARENTAL\_CHECK

A callback signalling that the DVD is requesting for a parental level check.

#### *Arguments*

DWORD dwValue Value representing the new parental level that is required.

#### *Remarks*

This is a callback indicating that the current parental level of the player does not match the one required by the authored content. The player should return this callback with either TRUE or FALSE, depending on whether the user is authorized to change the parental level.

Usually this is done by opening a dialog box so the user can enter the password to change the parental level. If the user enters the correct password, return TRUE, otherwise return FALSE.

### 3.16 FMPM\_DVD\_PARENTAL\_ERROR

Signals that the current parental level is incompatible with the authored content.

#### *Arguments*

DWORD dwValue Value representing the parental ID required.

#### *Remarks*

This event indicates that the current parental level of the player is incompatible with that required by the authored content and playback cannot continue.

### 3.17 FMPM\_DVD\_PARENTAL\_LEVEL\_CHANGE

Signals that the authored content has changed the parental level setting in the player.

#### Arguments

DWORD dwValue Value representing the new parental level set in the player.

#### Remarks

This event indicates that the authored content has changed the parental level setting. The new parental level indicated by lParam1 can be used to validate the current user.

### 3.18 FMPM\_DVD\_PLAYBACK\_STOPPED

Indicates that playback has been stopped. The DVD Navigator has completed playback of the PGC and did not find any other branching instruction for subsequent playback.

#### Remarks

All domains (DVD\_DOMAIN) raise this event. This event is not raised on user initiated stop, only on completion of the playback of the PGC.

### 3.19 FMPM\_DVD\_PROGRAM\_START

Signals that the DVD player started playback of a new program.

#### Arguments

DWORD dwValue Value indicating the new chapter (program) number

#### Remarks

The DVD\_TITLE\_DOMAIN domain raises this event.

### 3.20 FMPM\_DVD\_STILL\_OFF

Signals the end of any still (PGC, Cell, or VOB).

This event indicates that any currently active still has been released. All domains raise this event.

### 3.21 FMPM\_DVD\_STILL\_ON

Signals the beginning of any still (PGC, Cell, or VOB).

#### Arguments

DWORD dwValue Value indicating the number of seconds the still will last.

0xFFFFFFFF indicates an infinite still, meaning wait until the user presses a button or until the application calls DVDStillOff.

#### Remark

All combinations of buttons and still are possible (buttons on with still on, buttons on with still off, button off with still on, button off with still off).

All domains raise this event.



### 3.22 FMPM\_DVD\_SUBPICTURE\_STREAM\_CHANGE

Signals that the current user subpicture stream number changed for the main title.

#### *Arguments*

DWORD dwValue Value indicating the new user subpicture stream number.

Subpicture stream numbers range from 0 to 31. Stream 0xFFFFFFFF indicates that no stream is selected.

#### *Remark*

The subpicture can change automatically with a navigation command authored on disc as well as through application control using DVDSubpictureSelectStream.

All domains raise this event.

### 3.23 FMPM\_DVD\_TITLE\_CHANGE

Indicates when the current title number changes.

DWORD dwValue Value indicating the new title number.

#### *Remarks*

Title numbers range from 1 to 99. This number indicates the TTN, which is the title number with respect to the whole disc, not the VTS-TTN which is the title number with respect to just a current VTS.

The DVD\_TITLE\_DOMAIN domain raises this event.

### 3.24 FMPM\_DVD\_VALID\_UOPS\_CHANGE

Signals that the available authorized commands has changed.

#### *Arguments*

DWORD dwValue Value representing the VALID\_UOP\_SOMTHING\_OR\_OTHER bit-field structure that indicates which IDvdControl commands the DVD disc explicitly disabled.

#### *Remarks*

This event indicates only which operations are explicitly disabled by the content on the DVD disc, and does not guarantee that it is valid to call methods that are not disabled. For example, if no buttons are present, the DVDButtonActivate function won't work, even though the method is not explicitly disabled.

All domains raise this event.

### 3.25 FMPM\_DVDROM\_ERR

Indicates that DVDROM has problem for reading or playing DVD.

#### *Remarks*

All domains raise this event.

### 3.26 FMPM\_DVDROM\_NOT\_READY

Signals that DVDROM Drive is open or is not ready to play the play back.

*Remarks*

All domains raise this event.

### 3.27 FMPM\_EOS

Message issued when the video and audio decoders have finished decoding the stream.

*Remarks*

All domains raise this event.

### 3.28 FMPM\_ERROR

Specifies that an error occurred while executing a command.

*Remarks*

All domains raise this event.

### 3.29 FMPM\_REGION\_MISMATCH

Signals that DVD has invalid region code.

*Remarks*

All domains raise this event.

### 3.30 FMPM\_STARVATION

Message issued when the video decoder is starving.

*Remarks*

All domains raise this event.

### 3.31 FMPM\_SVCD\_PSD\_END

Indicates that SVCD reaches to the end of the playback.

*Remarks*

All domains raise this event.

### 3.32 FMPM\_TRICK\_MODE\_CHANGE

Signals that finished scanning or scanning is stopped.

#### *Arguments*

DWORD dwValue Value representing current speed of scanning.

#### *Remarks*

All domains raise this event.



# Chapter 4

## Installation

### 4.1 Linux

See the README file in archive

### 4.2 VxWorks

Under VxWorks, the driver is delivered as an object file (MPEGDrv.o).

The driver can be linked directly to VxWorks or loaded dynamically. Once loaded, all the FMP functions are directly accessible.

### 4.3 Windows CE

The driver is delivered in DLL form.

The FMP functions are exported from the RMMOD.DLL module.

The DLLs are:

RMGRAPH.DLL API implementation.

RMPROXY.DLL Low level driver.

RMDEMUX.DLL Demux and memory management

RMDVD.DLL DVD Navigation

RMSVCD.DLL SVCD/VCD Navigation

RMCC.DLL Closed Caption

RMSOURCE.DLL File systems

RMDVDDEV.DLL DVD-ROM Atapi driver.

RMCDDA.DLL Audio CD driver.

For some adaptations kits (like Venus), all the DLLs are combined into RMMOD.DLL



## Chapter 5

# Language codes

Language name	code	Language family
ABKHAZIAN	AB	IBERO-CAUCASIAN
AFAN (OROMO)	OM	HAMITIC
AFAR	AA	HAMITIC
AFRIKAANS	AF	GERMANIC
ALBANIAN	SQ	INDO-EUROPEAN (OTHER)
AMHARIC	AM	SEMITIC
ARABIC	AR	SEMITIC
ARMENIAN	HY	INDO-EUROPEAN (OTHER)
ASSAMESE	AS	INDIAN
AYMARA	AY	AMERINDIAN
AZERBAIJANI	AZ	TURKIC/ALTAIC
BASHKIR	BA	TURKIC/ALTAIC
BASQUE	EU	BASQUE
BENGALI;BANGLA	BN	INDIAN
BHUTANI	DZ	ASIAN
BIHARI	BH	INDIAN
BISLAMA	BI	[not given]
BRETON	BR	CELTIC
BULGARIAN	BG	SLAVIC
BURMESE	MY	ASIAN
BYELORUSSIAN	BE	SLAVIC
CAMBODIAN	KM	ASIAN
CATALAN	CA	ROMANCE
CHINESE	ZH	ASIAN
CORSICAN	CO	ROMANCE
CROATIAN	HR	SLAVIC
CZECH	CS	SLAVIC
DANISH	DA	GERMANIC
DUTCH	NL	GERMANIC
ENGLISH	EN	GERMANIC
ESPERANTO	EO	INTERNATIONAL AUX.
ESTONIAN	ET	FINNO-UGRIC
FAROESE	FO	GERMANIC
FIJI	FJ	OCEANIC/INDONESIAN
FINNISH	FI	FINNO-UGRIC
FRENCH	FR	ROMANCE
to be ct'd. . .		

Language name (ct'd)	code	Language family (ct'd)
FRISIAN	FY	GERMANIC
GALICIAN	GL	ROMANCE
GEORGIAN	KA	IBERO-CAUCASIAN
GERMAN	DE	GERMANIC
GREEK	EL	LATIN/GREEK
GREENLANDIC	KL	ESKIMO
GUARANI	GN	AMERINDIAN
GUJARATI	GU	INDIAN
HAUSA	HA	NEGRO-AFRICAN
HEBREW	HE	SEMITIC
HINDI	HI	INDIAN
HUNGARIAN	HU	FINNO-UGRIC
ICELANDIC	IS	GERMANIC
INDONESIAN	ID	OCEANIC/INDONESIAN
INTERLINGUA	IA	INTERNATIONAL AUX.
INTERLINGUE	IE	INTERNATIONAL AUX.
INUKTITUT	IU	[ ]
INUPIAK	IK	ESKIMO
IRISH	GA	CELTIC
ITALIAN	IT	ROMANCE
JAPANESE	JA	ASIAN
JAVANESE	JV	OCEANIC/INDONESIAN
KANNADA	KN	DRAVIDIAN
KASHMIRI	KS	INDIAN
KAZAKH	KK	TURKIC/ALTAIC
KINYARWANDA	RW	NEGRO-AFRICAN
KIRGHIZ	KY	TURKIC/ALTAIC
KURUNDI	RN	NEGRO-AFRICAN
KOREAN	KO	ASIAN
KURDISH	KU	IRANIAN
LAOTHIAN	LO	ASIAN
LATIN	LA	LATIN/GREEK
LATVIAN;LETTISH	LV	BALTIC
LINGALA	LN	NEGRO-AFRICAN
LITHUANIAN	LT	BALTIC
MACEDONIAN	MK	SLAVIC
MALAGASY	MG	OCEANIC/INDONESIAN
MALAY	MS	OCEANIC/INDONESIAN
MALAYALAM	ML	DRAVIDIAN
MALTESE	MT	SEMITIC
MAORI	MI	OCEANIC/INDONESIAN
MARATHI	MR	INDIAN
MOLDAVIAN	MO	ROMANCE
MONGOLIAN	MN	[not given]
NAURU	NA	[not given]
NEPALI	NE	INDIAN
NORWEGIAN	NO	GERMANIC
OCCITAN	OC	ROMANCE
ORIYA	OR	INDIAN
PASHTO;PUSHTO	PS	IRANIAN
PERSIAN (farsi)	FA	IRANIAN
POLISH	PL	SLAVIC

to be ct'd...



Language name (ct'd)	code	Language family (ct'd)
PORTUGUESE	PT	ROMANCE
PUNJABI	PA	INDIAN
QUECHUA	QU	AMERINDIAN
RHAETO-ROMANCE	RM	ROMANCE
ROMANIAN	RO	ROMANCE
RUSSIAN	RU	SLAVIC
SAMOAN	SM	OCEANIC/INDONESIAN
SANGHO	SG	NEGRO-AFRICAN
SANSKRIT	SA	INDIAN
SCOTS GAELIC	GD	CELTIC
SERBIAN	SR	SLAVIC
SERBO-CROATIAN	SH	SLAVIC
SESOTHO	ST	NEGRO-AFRICAN
SETSWANA	TN	NEGRO-AFRICAN
SHONA	SN	NEGRO-AFRICAN
SINDHI	SD	INDIAN
SINGHALESE	SI	INDIAN
SISWATI	SS	NEGRO-AFRICAN
SLOVAK	SK	SLAVIC
SLOVENIAN	SL	SLAVIC
SOMALI	SO	HAMITIC
SPANISH	ES	ROMANCE
SUNDANESE	SU	OCEANIC/INDONESIAN
SWAHILI	SW	NEGRO-AFRICAN
SWEDISH	SV	GERMANIC
TAGALOG	TL	OCEANIC/INDONESIAN
TAJIK	TG	IRANIAN
TAMIL	TA	DRAVIDIAN
TATAR	TT	TURKIC/ALTAIC
TELUGU	TE	DRAVIDIAN
THAI	TH	ASIAN
TIBETAN	BO	ASIAN
TIGRINYA	TI	SEMITIC
TONGA	TO	OCEANIC/INDONESIAN
TSONGA	TS	NEGRO-AFRICAN
TURKISH	TR	TURKIC/ALTAIC
TURKMEN	TK	TURKIC/ALTAIC
TWI	TW	NEGRO-AFRICAN
UIGUR	UG	[ ]
UKRAINIAN	UK	SLAVIC
URDU	UR	INDIAN
UZBEK	UZ	TURKIC/ALTAIC
VIETNAMESE	VI	ASIAN
VOLAPUK	VO	INTERNATIONAL AUX.
WELSH	CY	CELTIC
WOLOF	WO	NEGRO-AFRICAN
XHOSA	XH	NEGRO-AFRICAN
YIDDISH	YI	GERMANIC
YORUBA	YO	NEGRO-AFRICAN
ZHUANG	ZA	[ ]
ZULU	ZU	NEGRO-AFRICAN



## Chapter 6

# Registry values

### 6.1 Os - Linux

.realmagicHWLrc0 is the Hardware Library settings for board 0 in the system.

.realmagicrc0 is the DVD settings for board 0 in the system.

.realmagicanalogoverlayrc0 is Analogoverlay settings for board 0 when you use the calibration application.

#### 6.1.1 Analogoverlay

Here is sample setting for .realmagicanalogoverlayrc0 which stores the analog overlay calibration parameters and is created after calibration of Analogoverlay settings. You can find it under \$HOME/ Directory.

```
#####
# Width resolution x Height resolution x Color depth x Refresh frequency
[ 1024 x 768 x 16 x 84 ]
#####
# Numbers of pixels
XOFFSET = 253
#####
# Numbers of lines
YOFFSET = 39
#####
# Ratio between VGA pixels frequency and MPEG pixels frequency
# RM_VGACORRECTION_DEFAULT          1000
VGACORRECTION = 1180
#####
# Chroma key upper limit for Red.
# RM_VGARUPPER_DEFAULT              14
VGARUPPER = 16
#####
# Chroma key lower limit for Red.
# RM_VGARLOWER_DEFAULT              0
VGARLOWER = 0
#####
# Chroma key upper limit for Green.
# RM_VGAGUPPER_DEFAULT              56
VGAGUPPER = 16
#####
# Chroma key lower limit for Green.
# RM_VGAGLOWER_DEFAULT              29
```

```

VGABLOWER = 0
#####
# Chroma key upper limit for Blue.
# RM_VGABUPPER_DEFAULT          0
VGABUPPER = 16
#####
# Chroma key lower limit for Blue.
# RM_VGABLOWER_DEFAULT         14
VGABLOWER = 0
#####
# Values to adjust stability of jumping pixels on MPEG video
# RM_JITTERADJUSTMENT_DEFAULT   0
STABILITY = 1

```

## 6.1.2 DVD - celesteapp

DVD settings are:

```

#####
# Language code : default language code
# RM_LANGUAGECODE_ENGLISH 0x656e // TEXT ("en")
# RM_LANGUAGECODE_FRENCH 0x6672 // TEXT ("fr")
# RM_LANGUAGECODE_SPANISH 0x7370 // TEXT ("sp")
# RM_LANGUAGECODE_ITALIAN 0x6974 // TEXT ("it")
# RM_LANGUAGECODE_GERMAN 0x6765 // TEXT ("ge")
# RM_LANGUAGECODE_DUTCH 0x6475 // TEXT ("du")
# RM_LANGUAGECODE_PORTUGUESE 0x706f // TEXT ("po")
# RM_LANGUAGECODE_CHINESE 0x6368 // TEXT ("ch")
# RM_LANGUAGECODE_JAPANESE 0x6a61 // TEXT ("ja")
# RM_LANGUAGECODE_ORIGINAL 0x6f72 // TEXT ("or")
# RM_LANGUAGECODE_OTHERS 0x6f74 // TEXT ("ot")
# Default:
LanguageCode = 25966
MenuLanguageCode = 25966
AudioLanguageCode = 28530
#####
# RM_LANGUAGECODE_AUDIOFOLLOW 0x6166 // TEXT ("af")
# Default:
SubpictureLanguageCode = 24934
#####
# RM_SUBPICTURE_DISPLAY_MODE_DEFAULT 0x00
SubpictureDisplayMode = 0
#####
# Aspect ratio of the screen
# ASPECT_RATIO_4_3 0x0000
# ASPECT_RATIO_16_9 0x0003
# Default:
UserAspectRatio = 0
#####
# Output mode for sources with a different aspect ratio
# DISPLAY_MODE_NORMAL_OR_WIDE 0x0000
# DISPLAY_MODE_PAN_SCAN 0x0001
# DISPLAY_MODE_LETTERBOX 0x0002

```

```

# DISPLAY_MODE_ZOOM_ON 0x0003
# Default:
UserVideoOutputMode = 2
#####
# Country code for parental control
# RM_COUNTRYCODE_US 0x5553 // TEXT ("US")
# RM_COUNTRYCODE_OTHERS 0x4f54 // TEXT ("OT")
# Default:
CountryCode = 21843
#####
# RM_PARENTALCONTROL_OFF 0x000f
# RM_PARENTALCONTROL_LEVEL1 0x0001
# RM_PARENTALCONTROL_LEVEL2 0x0002
# RM_PARENTALCONTROL_LEVEL3 0x0003
# RM_PARENTALCONTROL_LEVEL4 0x0004
# RM_PARENTALCONTROL_LEVEL5 0x0005
# RM_PARENTALCONTROL_LEVEL6 0x0006
# RM_PARENTALCONTROL_LEVEL7 0x0007
# RM_PARENTALCONTROL_LEVEL8 0x0008
# Default:
ParentalControl = 15
#####
# TV_NTSC                                0x0000
# TV_PAL                                0x0002
# TV_PAL60                              0x0008
# TV_PALN                                0x000a
# RM_OUTPUTMODE_DEFAULT TV_NTSC | 0x01
OutputMode = 17
#####
# Brightness from 0 to 1000
Brightness = 255
#####
# Saturation from 0 to 1000
Saturation = 255
#####
# Contrast from 0 to 1000
Contrast = 255
#####
# RM_PASSWORD_OFF 0x0000
# RM_PASSWORD_ON 0x0001
# RM_NO_PASSWORD 0xffff
# Default:
PasswordLowWord = 65535
PasswordHighWord = 65535
PasswordState = 0
#####
# RM_MACROVISION_ENABLED 0x0001
# RM_MACROVISION_DISABLED 0x0000
Macrovision = 1
#####
# set your region code if the DVDRom device is not RPC2
# valid numbers are 1-6
# Default:
RegionCode = 1

```

```
#####
# RM_AUDIO_TYPE_ANALOG 0x0000
# RM_AUDIO_TYPE_SPDIF 0x0001
AudioType = 0
#####
# RM_DAC_TYPE_1720 1720
# RM_DAC_TYPE_1716 1716
DacType = 1720
#####
# RM_BALANCE_DEFAULT 100
Balance = 100
#####
# RM_VOLUME_DEFAULT 75
# valid numbers are 0 - 1000
Volume = 75
#####
# RM_AUTOPLAY_OFF 0x0000
# RM_AUTOPLAY_ON 0x0001
Autoplay = 1
#####
# HDTV_480P_60FPS                0x0200
# HDTV_480P_72FPS                0x0201
# HDTV_480P_96FPS                0x0202
# HDTV_480P_120FPS               0x0203
# HDTV_720P_60FPS                0x0204
# HDTV_720P_72FPS                0x0205
# HDTV_720P_96FPS                0x0206
# HDTV_960P_60FPS                0x0207
# HDTV_960P_72FPS                0x0208
# Default:
HDTVResolution = 512
```

[ General ]

```
#####
# TvOut byte:
# Bit7,Bit6 = YcYuvRgb          - one of COMPOSITE, COMPONENT_YUV, COMPONENT_RGB
# Bit4      = TvAsSource         - one of SET_TV_AS_SOURCE, SET_TV_AS_USER
# Bit2      = TvNoScaling        - one of SET_ONETOONE, SET_SCALE
# Bit3,Bit1 = Standard_TvOut     - one of SET_NTSC, SET_PAL, SET_PAL60, SET_PALM
# Bit5,Bit0 = TvOut             - one of SET_VGA, SET_TV, SET_HDTV
#define COMPONENT_MASK 0x00C0
#define COMPOSITE 0x0000
#define COMPONENT_YUV 0x0080
#define COMPONENT_RGB 0x00C0

#define VIDEOOUT_MASK 0x0021
#define SET_VGA 0x0000
#define SET_TV 0x0001
#define SET_HDTV 0x0020

#define STANDARDTV_MASK 0x000A
#define SET_NTSC 0x0000
#define SET_PAL 0x0002
```

```

#define SET_PAL60 0x0008
#define SET_PALM 0x000A

#define SET_ONETOONE 0x0000
#define SET_SCALE 0x0004

#define SET_TV_AS_SOURCE 0x0010
#define SET_TV_AS_USER 0x0000

# Example :
# TvOut = 0 == SET_VGA | SET_NTSC | SET_ONETOONE | SET_TV_AS_USER will set:
# -the analog overlay boards to play on Vga monitor
# -the digital overlay boards to play in same time on vga and Tv , Ntsc
# TvOut = 1 == SET_TV | SET_NTSC | SET_ONETOONE | SET_TV_AS_USER will set:
# -the analog overlay boards to play on Tv Ntsc
# -the digital overlay boards to play on 480P connector (only Ventura2k has this connector)
# TvOut = 32 (0x20) == SET_HDTV | SET_NTSC | SET_ONETOONE | SET_TV_AS_USER will set:
# -the analog overlay boards to play on Hdtv
# -not used for the digital overlay boards
# Default :
TvOut = 0
#####
# TvMRO ;          - 0x10 for AnalogDevice TvEncoders (extended NTSC filters)
# Default :
TvMRO = 16 (0x10)
#####
# DoHwReset;      - 0 IDecoderBoard_HwReset returns without resetting the hardware
# - 1 IDecoderBoard_HwReset resets the hardware
# Default :
DoHwReset = 1
#####
# DisableSpdifOutputInReset; - 0 IDecoderBoard_HwReset leaves the Spdif output enabled
*   - 1 IDecoderBoard_HwReset disables the Spdif output
# Default :
DisableSpdifOutputInReset = 1
#####
# ActiveVideoWidthNtsc
# Number of active video pixels for NTSC resolution
# Default :
ActiveVideoWidthNtsc = 720
#####
# ActiveVideoWidthPal
# Number of active video pixels for PAL resolution
# Default :
ActiveVideoWidthPal = 720
# The above two values are used when playing on TV to set the width of the active window.
# They show the number of active pixels displayed on TV (any value between 0...720) for
# Ntsc / Pal. They will not affect the destination window of the video or the scaling factors.
#####
#ForcedProgressiveSourceOff;
#           - 1 for FORCED_PROGRESSIVE_OFF - displays progressive or interlaced like in th
#           - 0 for FORCED_PROGRESSIVE_ON - displays only progressive if the stream switch
#           fast from interlaced to progressive
# Default :

```

```

ForcedProgressiveSourceOff = 1 (FORCED_PROGRESSIVE_OFF)
#####
#ForcedProgressiveAlways;
#
#           - 1 for PROGRESSIVE_ALWAYS - displays progressive for any video stream
#           - 0 for PROGRESSIVE_MOVIES - displays progressive only for movies
# Default :
ForcedProgressiveAlways = 0 (PROGRESSIVE_MOVIES)
#####
#NtscPalFrameDrop;
#
#           - 1 for FRAME_DROP - it drops frames when it converts Ntsc clip to PAL TV.
#           - 0 for FIELD_DROP - it drops fields when it converts Ntsc clip to PAL TV.
# Default :
NtscPalFrameDrop = 0 (FIELD_DROP)
#####
#VGAForcedInterlaced;
#
#           - 1 for VGA_INTERLACED - displays only BOB on VGA
#           - 0 VGA_PROGRESSIVE - displays BOB or WEAVE on VGA, depending on stream
# Default :
VGAForcedInterlaced = 0 (VGA_PROGRESSIVE)
#
#####
#EnableSpdif;
# - 0 for AUDIO_OUTPUT_STEREO
# - 1 for AUDIO_OUTPUT_AC3DTS
# Default :
EnableSpdif = 0 (AUDIO_OUTPUT_STEREO)
#####
#VolumeRight;    - audio volume right from 0 to 100
#The default value is 100.
VolumeRight = 100
#####
#VolumeLeft;     - audio volume left from 0 to 100
#The default value is 100.
VolumeLeft = 100
#####
#DWORD DoAudioLater;
#This dword can be used for testing audio / video synchronization.
# DoAudioLater is the number of PTS units that will be added to the audio PTS
# from file send to the hardware. The HwLib will typecast DoAudioLater
# to LONG - this means that negative values can be programmed.
# The PTS unit is 11.(1) microseconds ( 90 kHz ).
#The default value is 0.
DoAudioLater = 0
#####
#DacType;
# Used only for a specific hardware design (STPC - to select Pcm1716 or Pcm1720)
#The default value is 1720.
#####
#AudioDacBitsPerSample;
# - 0 the audio I2S 16 or 24 bit will be selected by HwLib
# - 16 for audio I2S 16 bit
# - 24 for audio I2S 24 bit
#The default value is 0.
AudioDacBitsPerSample = 0

```



```
#####
#Brightness;      - brightness from 0 to 1000, used when SET_VGA is selected
# Default :
Brightness = 500
#####
#Contrast;        - brightness from 0 to 1000, used when SET_VGA is selected
# Default :
Contrast = 500
#####
#Saturation;      - brightness from 0 to 1000, used when SET_VGA is selected
# Default :
Saturation = 500
#####
#TvBrightness;    - brightness from 0 to 1000, used when SET_TV is selected
# Default :
TvBrightness = 500
#####
#TvContrast;      - brightness from 0 to 1000, used when SET_TV is selected
# Default :
TvContrast = 500
#####
#TvSaturation;    - brightness from 0 to 1000, used when SET_VGA is selected
# Default :
TvSaturation = 500
#####
#DecoderIsSlave;
#               - 0 for Ventura2000 when EM8400 is master - normal situation
#               - 1 for Ventura2000 when EM8400 is slave
# Default :
DecoderIsSlave = 0
#####
#BitsPerPixel;
#ScreenWidth;
#ScreenHeight;
# These values should be set according to VGA mode selected.
#The default values are:
BitsPerPixel = 8
ScreenWidth = 1024
ScreenHeight = 768
#####
#HFreq;
#Used for analog overlay to program the pixel clock frequency for Nova chip.
#At IDecoderBoard_HwReset time the analog overlay chip will detect the horizontal
#frequency of the VGA mode and will program its PLL trying to match the VGA pixel
#frequency. Because of not enough accuracy the result can be slightly different from
#one detection to another and this causes a one pixel change in position of the video.
#In order to avoid this, the user should save the horizontal frequency got from HwLib
#in the registry and when the new detection will happen the HwLib will use the registry
#value if the value is in a +/-200Hz range. If the value is too different the
#auto-detected frequency will be used - the VGA mode was probably changed.
#If the user doesn't use the value should be 0.
HFreq = 0
#####
#TotalPixelsPerLine;
```

```

#Used for analog overlay to program the pixel clock frequency for Nova chip.
#At IDecoderBoard_HwReset time the analog overlay chip will program the PLL frequency
#based on horizontal frequency and an estimation of the pixels per line number.
#For a better accuracy this number can be programmed by user in registry.
#If the user don't use the value should be 0.
#The default value is 0.
TotalPixelsPerLine = 0
#####
#AcpiEnable;
# - 0 will not power off/on the EM8400 (no ACPI ON/OFF)
# - 1 IDecoder_Init will switch to ACPI_ON, IDecoder_Delete will switch to ACPI_OFF
#The default value is 0.
AcpiEnable = 0
#####
#DecoderIsSlave;
# Used for Ventura2k:
# - 0 EM8400 is master - EM8400 generates the HSync and VSync
# - 1 EM8400 is slave - EM8400 doesn't generate the HSync and VSync
#The default value is 0.
DecoderIsSlave = 0
#####
#ZoomEnable;
# - 0 no Zoom
# - 1 enable AcqWnd window to be the zoomed video window
# Default :
ZoomEnable = 1
#####
#WindowTvEnable;
# - 0 fullscreen on TV
# - 1 enable destination window on TV
# Default :
WindowTvEnable = 0
#####
#WindowHdtvEnable;
# - 0 fullscreen on HDTV
# - 1 enable destination window on HDTV
# Default :
WindowHdtvEnable = 0
#####
#OsdVideoIndependent; 1 to enable the zoom
# - 0 OSD will be displayed relative to the video window
# - 1 OSD will be displayed relative to the output device screen
# Default :
OsdVideoIndependent = 0
#####
#MaximumDvclk;
#Used to program the digital video pixel clock Dvclk for EM9010.
# - 0 Dvclk will match the VGA pixel frequency
# - 1 Dvclk will be set to maximum limit 80000kHz = 80MHz
# - any required Dvclk in kHz, limited between hardware limits (30000 and 80000 for EM9010)
#The default value is 0.
MaximumDvclk = 0

```

[ VGA ]

```
#####
#InvertField;      - 0 for top-bottom display, 1 for bottom-top display
# Default :
InvertField = 0
#####
#Vmi_16bits;       - 0 for 8bits, 1 for 16 bits
# Default :
Vmi_16bits = 0
#####
#Ccir_656;         - one of 0 (CCIR_601) or 1 (CCIR_656)
# Default :
Ccir_656 = 1
#####
#SyncEnable;
#                 - 2: VSync enabled, HSync disabled, VVLD/HS=HS enabled,
#                 - 1: VSync enabled, HSync enabled, VVLD/HS=VVLD enabled,
#                 - 0: VSync, HSync, VVLD/HS=VVLD disabled.
# Default :
SyncEnable = 0 (by default disable VS,HS, VVLD)
#####
#Vip20;
#                 - enable (1) or disable (0) vip20 mode
# Default :
Vip20 = 0;
```

## 6.2 Os - WinCE

Setting up the registry - Please copy the DVD registries and Hwlibray values into your platform.reg file.

### 6.2.1 DVD registries

```
#####
;# FOR SIGMA DESIGNS DVD NAVIGATION
#####
[HKEY_LOCAL_MACHINE\REALmagic]
; Language code : default language code
; RM_LANGUAGECODE_ENGLISH 0x656e // TEXT ("en")
; RM_LANGUAGECODE_FRENCH 0x6672 // TEXT ("fr")
; RM_LANGUAGECODE_SPANISH 0x7370 // TEXT ("sp")
; RM_LANGUAGECODE_ITALIAN 0x6974 // TEXT ("it")
; RM_LANGUAGECODE_GERMAN 0x6765 // TEXT ("ge")
; RM_LANGUAGECODE_DUTCH 0x6475 // TEXT ("du")
; RM_LANGUAGECODE_PORTUGUESE 0x706f // TEXT ("po")
; RM_LANGUAGECODE_CHINESE 0x6368 // TEXT ("ch")
; RM_LANGUAGECODE_JAPANESE 0x6a61 // TEXT ("ja")
; RM_LANGUAGECODE_ORIGINAL 0x6f72 // TEXT ("or")
; RM_LANGUAGECODE_OTHERS 0x6f74 // TEXT ("ot")
"LanguageCode"=dword: 656e
"MenuLanguageCode"=dword: 656e
"AudioLanguageCode"=dword: 6f72
; RM_LANGUAGECODE_AUDIOFOLLOW 0x6166 // TEXT ("af")
"SubpictureLanguageCode"=dword: 6166
"SubpictureDisplayMode"=dword: 0000
```

```

; Aspect ratio of the screen
; ASPECT_RATIO_4_3 0x0000
; ASPECT_RATIO_16_9 0x0003
"UserAspectRatio"=dword: 0

; Output mode for sources with a different aspect ratio
; DISPLAY_MODE_NORMAL_OR_WIDE 0x0000
; DISPLAY_MODE_PAN_SCAN 0x0001
; DISPLAY_MODE_LETTERBOX 0x0002
; DISPLAY_MODE_ZOOM_ON 0x0003
"UserVideoOutputMode"=dword: 2

; Country code for parental control
; RM_COUNTRYCODE_US 0x5553 // TEXT ("US")
; RM_COUNTRYCODE_OTHERS 0x4f54 // TEXT ("OT")
"CountryCode"=dword: 5553
; RM_PARENTALCONTROL_OFF 0x000f
; RM_PARENTALCONTROL_LEVEL1 0x0001
; RM_PARENTALCONTROL_LEVEL2 0x0002
; RM_PARENTALCONTROL_LEVEL3 0x0003
; RM_PARENTALCONTROL_LEVEL4 0x0004
; RM_PARENTALCONTROL_LEVEL5 0x0005
; RM_PARENTALCONTROL_LEVEL6 0x0006
; RM_PARENTALCONTROL_LEVEL7 0x0007
; RM_PARENTALCONTROL_LEVEL8 0x0008
"ParentalControl"=dword: 000f

; RM_PASSWORD_OFF 0x0000
; RM_PASSWORD_ON 0x0001
; RM_NO_PASSWORD 0xffff
"PasswordState"=dword: 0000
"PasswordLowWord"=dword: ffff
"PasswordHighWord"=dword: ffff
; set your region code if the DVDROM device is not RPC2
; valid numbers are 1-6
"RegionCode"=dword: 1

```

## 6.2.2 Hwlibrary

Here are some registry entries for EM8400:

```

#####
;# FOR EM84xx DECODER
#####
[HKEY_LOCAL_MACHINE\Drivers\SigmaDesigns\EM84xx]
#####
;# TvOut byte:
;#Bit7,Bit6 = YcYuvRgb      - one of COMPOSITE, COMPONENT_YUV, COMPONENT_RGB
;#Bit4      = TvAsSource    - one of SET_TV_AS_SOURCE, SET_TV_AS_USER
;#Bit2      = TvNoScaling   - one of SET_ONETOONE, SET_SCALE
;#Bit3,Bit1 = Standard_TvOut - one of SET_NTSC, SET_PAL, SET_PAL60, SET_PALM
;#Bit5,Bit0 = TvOut        - one of SET_VGA, SET_TV, SET_HDTV
;#

```

```

#define COMPONENT_MASK 0x00C0
#define COMPOSITE      0x0000
#define COMPONENT_YUV  0x0080
#define COMPONENT_RGB  0x00C0
;
#define VIDEOOUT_MASK  0x0021
#define SET_VGA         0x0000
#define SET_TV          0x0001
#define SET_HDTV        0x0020
;
#define STANDARDTV_MASK 0x000A
#define SET_NTSC        0x0000
#define SET_PAL         0x0002
#define SET_PAL60       0x0008
#define SET_PALM        0x000A

#define SET_ONETOONE    0x0000
#define SET_SCALE       0x0004

#define SET_TV_AS_SOURCE 0x0010
#define SET_TV_AS_USER  0x0000

;# Example :
;# TvOut = 0 == SET_VGA | SET_NTSC | SET_ONETOONE | SET_TV_AS_USER will set:
;# -the analog overlay boards to play on Vga monitor
;# -the digital overlay boards to play in same time on vga and Tv , Ntsc
;# TvOut = 1 == SET_TV | SET_NTSC | SET_ONETOONE | SET_TV_AS_USER will set:
;# -the analog overlay boards to play on Tv Ntsc
;# -the digital overlay boards to play on 480P connector
;# (only Ventura2k has this connector)
;# TvOut = 32 (0x20) == SET_HDTV | SET_NTSC | SET_ONETOONE | SET_TV_AS_USER
;# will set:
;# -the analog overlay boards to play on Hdtv
;# -not used for the digital overlay boards
;# Default :
;# "TvOut" = dword:0
"TvOut" = dword:00000001
;#####
;# TvMRO - 0x10 for AnalogDevice TvEncoders (extended NTSC filters)
;# Default :
"TvMRO" = dword:00000010
;#####
;# ActiveVideoWidthNtsc
;# Number of active video pixels for NTSC resolution
;# Default :
;"ActiveVideoWidthNtsc" = dword:000002d0
;#####
;# ActiveVideoWidthPal
;# Number of active video pixels for PAL resolution
;# Default :
;"ActiveVideoWidthPal" = dword:000002d0
;#####
;# ForcedProgressiveSourceOff
;# - 100 for FORCED_PROGRESSIVE_OFF - displays progressive

```

```

;#          or interlaced like in the video stream
;#          - 0 for FORCED_PROGRESSIVE_ON - displays only progressive
;#          if the stream switches very fast from interlaced to progressive
;# Default :
"ForcedProgressiveSourceOff" = dword:00000100
#####
;# ForcedProgressiveAlways
;#          - 1 for PROGRESSIVE_ALWAYS - displays progressive for any video stream
;#          - 0 for PROGRESSIVE_MOVIES - displays progressive only for movies
;# Default :
"ForcedProgressiveAlways" = dword:00000000
#####
;# NtscPalFrameDrop
;#          - 1 for FRAME_DROP - it drops frames when it converts Ntsc clip to PAL TV
;#          - 0 for FIELD_DROP - it drops fields when it converts Ntsc clip to PAL TV
;# Default :
"NtscPalFrameDrop" = dword:00000000
#####
;# VGAForcedInterlaced
;#          - 1 for VGA_INTERLACED - displays only BOB on VGA
;#          - 0 VGA_PROGRESSIVE - displays BOB or WEAVE on VGA, depending on stream
;# Default :
"VGAForcedInterlaced" = dword:00000000
#####
;# RightVolume
;#          audio volume [0x0..0x64]
"RightVolume" = dword:00000019
#####
;# LeftVolume
;#          audio volume [0x0..0x64]
"LeftVolume" = dword:00000019
#####
;# Brightness;          - brightness from 0 to 1000
;# Default :
"Brightness" = dword:000001f4
#####
;# Contrast;           - brightness from 0 to 1000
;# Default :
"Contrast" = dword:000001f4
#####
;# Saturation;         - brightness from 0 to 1000
;# Default :
"Saturation" = dword:000001f4
#####
;# BitsPerPixel
;# Default:
"BitsPerPixel" = dword:00000010
#####
;# DecoderIsSlave;
;#          - 0 for Ventura2000 when EM8400 is master - normal situation
;#          - 1 for Ventura2000 when EM8400 is slave
;# Default :
"DecoderIsSlave" = dword:00000000
#####

```

```

;# ScreenWidth
;# Default:
"ScreenWidth" = dword:00000280
#####
;# ScreenHeight
;# Default:
"ScreenHeight" = dword:000001e0
#####
;# 16bitsVMI;      - 0 for 8bits, 1 for 16 bits
;# Default :
"16bitsVMI" = dword:00000000
#####
;# ZoomEnable
;# Default :
;"ZoomEnable" = dword:00000001
#####
;# WindowTvEnable
;# Default :
;"WindowTvEnable" = dword:00000000
"WindowTvEnable" = dword:00000001
#####
;# WindowHdtvEnable
;# Default :
;"WindowHdtvEnable" = dword:00000000
#####
;# OsdVideoIndependent
;# Default :
;"OsdVideoIndependent" = dword:00000000
"OsdVideoIndependent" = dword:00000001
#####
;# InvertField      - 0 for top-bottom display, 1 for bottom-top display
;# Default :
;"InvertField" = dword:00000000
#####
;# Ccir_656          - one of 0 (CCIR_601) or 1 (CCIR_656)
;# Default :
;"Ccir_656" = dword:00000001
"Ccir_656" = dword:00000000
#####
;# SyncEnable
;#          - 2: VSync enabled, HSync disabled, VVLD/HS=HS enabled,
;#          - 1: VSync enabled, HSync enabled, VVLD/HS=VVLD enabled,
;#          - 0: VSync, HSync, VVLD/HS=VVLD disabled.
;# Default :
;"SyncEnable" = dword:00000000
"SyncEnable" = dword:00000001
#####
;# Vip20
;#          - enable (1) or disable (0) vip20 mode
;# Default :
;"Vip20" = dword:00000000
#####
;# PI07; - if you need to pull the PI07 up or down, register it here
;#

```

```
;"PI07" = dword:00000000

;#####
;# TV Window
;# Default :
;#         for NTSC:
;#             x = 0
;#             y = 0
;#             width = 720
;#             height = 480
;#         for PAL:
;#             x = 0
;#             y = 0
;#             width = 720
;#             height = 576
"TVWindowX" = dword:00000000
"TVWindowY" = dword:00000000
"TVWindowW" = dword:000002d0
"TVWindowH" = dword:000001e0
```



# Chapter 7

## Settings

Following are the different settings and status for the MPEG driver.

They can be read with the FMPIGet command (1.5.3) and written with the FMPISet command (1.5.9) using the FMPI\_XXX index.

Some settings are read only and are marked as r, others can be written and are marked r/w.

When it is r/s it means that you can write the value only if it is not yet determined. To be used carefully.

The FMPIGet and FMPISet commands always use 32 bits values. When a value is less than 32 bits long, the unused bits are zero.

### 7.1 CDDA settings

CDDAI_TOC	Returns zero for succeed	implemented
-----------	--------------------------	-------------

Figure 7.1: CDDA settings

#### 7.1.1 CDDAI\_TOC

Query the table of content of the CD-Audio

```
DWORD DVDQueryAttribute (CDDAI_TOC, dwArg);
```

*Arguments*

dwArg DWORD dwArg:

*Return value*

Zero if successful.

*Remarks*

If a track does not exist, its length will be 0xFFFFFFFF

Here is a small code :

```
#define LBA_TO_MSF(LBN) (((0x00FF0000 & (((LBN)/(60*75)) << 16)) | \
                        (0x0000FF00 & (((LBN)%(60*75))/75 << 8)) | \
                        (0x000000FF & (((LBN)%(60*75))%75) ))
#define MSF_M(MSF) ((MSF) >> 16) & (0x000000FF)
#define MSF_S(MSF) ((MSF) >> 8) & (0x000000FF)
```

```

#define MSF_F(MSF) ((MSF) & 0x000000FF)

void getTracksDuration()
{
    FMP_OPENSTRUCT FMPOpenStruct;
    DWORD tmp [101];
    DWORD MSF, i;
    BYTE bSec, bMin, bFrame;
    MPEGDriverEntry(DRIVE_NUMBER);
    FMPOpenStruct.dwStructSize = sizeof (FMP_OPENSTRUCT);
    FMPOpenStruct.bDriverNumber = DRIVE_NUMBER;
    FMPOpenStruct.dwSize = 32 * 1024;
    FMPOpenStruct.dwCount = 10;
    FMPOpenStruct.pCallback = NULL;
    FMPOpenStruct.dwContext = 0;
    FMPOpenStruct.sFileName = NULL;
    FMPOpenStruct.dwFlags = FMPCF_DVD;
    FMPOpenStruct.bDiscContent = FMPC_UNKNOWN;
    FMPOpenStruct.bFileSystem = 0x00;

    tmp [0] = FMPOpenDiscPlayback (&FMPOpenStruct);

    if (tmp [0] != 0) {
        printf("Failed to open the playback\n");
        return;
    }
    memset ( tmp, 0, 101 * sizeof(DWORD));
    DVDQueryAttribute ( CDDAI_TOC, (DWORD)tmp );

    //TOC starts from the element 1 of the 'tmp'
    //print first ten elements of the TOC

    for ( i = 1; i <= 10; i++ ) {
        if ( tmp [i] == 0xffffffff ) {
            printf("Cannot get duration of the track %d\n", i);
        }
        else {
            MSF      = LBA_TO_MSF( tmp [i] );
            bMin     = (BYTE)MSF_M(MSF);
            bSec     = (BYTE)MSF_S(MSF);
            bFrame   = (BYTE)MSF_F(MSF);
            printf ("TRACK  %d:  %d: %d: %d \n", i, bMin, bSec, bFrame);
        }
    }
    FMPClose();
    MPEGDriverUnload();
}

```

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks					•
Windows CE					
Linux					•

Figure 7.2: CDDAI.TOC Operating Systems and Streaming Models

## 7.2 DVD settings

Following are the different settings and status for streams.

They can be read with the `DVDQueryAttribute` command and using the `DVDI_xxx` indices. Please see the following sections for how to query the DVD attribute.

The driver settings are considered as default settings for future opened streams.

DVDLTT_SRPTI	Title Search Pointer Table Information: pointer to a TT_SRPTI struct	implemented
DVDLAST_ATR	Audio Attributes of the current VTS: pointer to a AST_ATR struct	implemented
DVDI_SPST_ATR	Subpicture Attributes of the current VTS: pointer to a SPST_ATR struct	implemented
DVDI_SPRM	System Parameters of the decoder	implemented
DVDI_VIDEO_MODE	Video Mode Attributes	implemented
DVDI_CURRENT_UOPS	Current UOPs bitfield	implemented

Figure 7.3: DVD settings

### 7.2.1 DVDI\_LAST\_ATR

Query the attributes of the Audio Streams of the current VTS.

```
DWORD DVDQueryAttribute (DVDI_LAST_ATR, PAST_ATR pAST_ATR);
```

*Arguments*

PAST\_ATR pAST\_ATR Pointer to a AST\_ATR structure:

```
typedef struct {
    BYTE bAST_Ns;                // Number of Audio Streams available (1 to 8)
    BOOL bAvailable[MAX_AST];    // If True, the audio stream is available in the current Program
    BYTE bAudioCodingMode [MAX_AST]; // Audio Coding Mode :
    BOOL bMultichannelExtension [MAX_AST]; // Multichannel extension
    BYTE bAudioType [MAX_AST];    // Audio Type
    BYTE bAudioApplicationMode [MAX_AST]; // Audio Application Mode
    BYTE bQuantization [MAX_AST]; // Quantization / DRC
    BYTE bfs [MAX_AST];          // Frequency
    BYTE bNumberOfAudioChannels [MAX_AST]; // Number of Audio channels
    WORD wLanguageCode [MAX_AST]; // Refer to Language codes
    WORD wLanguageCodeExtension [MAX_AST]; // Refer to Language codes
    BYTE bApplicationExtension [MAX_AST]; // Refer Annex C
} AST_ATR, *PAST_ASTR;
```

Audio coding modes are:

000b Dolby AC-3

010b MPEG-1

011b MPEG-2 with extension

100b Linear PCM audio

110b DTS

111b SDDS

Audio types are:

00b Not specified

01b Language

Audio application modes are:

00b Not specified

01b Karaoke mode

10b Surround mode

11b reserved

Number of audio channels are:

000b 1ch (mono)

001b 2ch (stereo)

010b 3ch

011b 4ch

100b 5ch

101b 6ch

110b 7ch

111b 8ch

All arrays go from 0 to b\_Ns.

### *Return value*

Zero if successful. Otherwise returns an error code.

DVDE\_ARG Arguments are not valid.

DVDE\_UNAUTHORIZED Unauthorized operation.

### *Remarks*

Here is a small code:

```
DWORD dwReturn;
AST_ATR ast_atr;
dwReturn = DVDQueryAttribute ( DVDI_AST_ATR, (DWORD)&ast_atr);
dwReturn = (DWORD) ast_atr.bAST_Ns;
dwReturn = (DWORD) ast_atr.bAudioCodingMode [1];
dwReturn = (DWORD) ast_atr.bMultichannelExtension [1];
dwReturn = (DWORD) ast_atr.bAudioType [1];
dwReturn = (DWORD) ast_atr.bAudioApplicationMode [1];
dwReturn = (DWORD) ast_atr.bQuantization [1];
dwReturn = (DWORD) ast_atr.bfs [1];
dwReturn = (DWORD) ast_atr.bNumberOfAudioChannels [1];
dwReturn = (DWORD) ast_atr.wLanguageCode [1];
dwReturn = (DWORD) ast_atr.wLanguageCodeExtension [1];
```

```

dwReturn = (DWORD) ast_atr.bApplicationExtension [1];
if ( ast_atr.bAvailable [1])
return TRUE;
else return FALSE;

```

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks		•			
Windows CE		•			
Linux		•			

Figure 7.4: DVDI\_LAST\_ATR Operating Systems and Streaming Models

### 7.2.2 DVDI\_SPRM

Query the current System Parameters of the decoder.

```
DWORD DVDQueryAttribute (DVDI_SPRM, value);
```

*Arguments*

DWORD value System Parameter value

SPRM	Value	Meaning
M_LCD	0	Menu Description Language Code
ASTN	1	Audio stream number
SPSTN	2	Sub-picture number and On/Off flag for TT_DOM
AGLN	3	Angle number
TTN	4	Title number for TT_DOM
VTN_TTN	5	VTN Title number for TT_DOM
TT_PGCN	6	Title PGC number
PTTN	7	Part_of_Title number for One_Sequential_PGC_Title
HL_BTNN	8	Highlighted Button number for Selection State
NV_TMR	9	Navigation Timer
NV_TMR_TT_PGCN	10	TT_PGCN for NV_TMR
P_AMXMD	11	Player Audio Mixing Mode for Karaoke
CTY_CD	12	Country Code for Parental Management
PTL_LVL	13	Parental Level
P_CFGV	14	Player Configuration for Video
P_CFGA	15	Player Configuration for Audio
IN_LCD_AST	16	Initial Language Code for AST
IN_LCD_EXT_AST	17	Initial Language Code extension for AST
IN_LCD_SPST	18	Initial Language Code for SPST
IN_LCD_EXT_SPST	19	Initial Language Code extension for SPST
PRC	20	Player Region Code
	21	Reserved
	22	Reserved
	23	Reserved for extended playback mode

Figure 7.5: System Parameters settings

*Return value*

Queried value if success, 0xFFFFFFFF if error (you can query error code using DVDGetLastError).

DVDE\_ARG Arguments are not valid.

DVDE\_UNAUTHORIZED Unauthorized operation.

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks		•			
Windows CE		•			
Linux		•			

Figure 7.6: DVDI\_SPRM Operating Systems and Streaming Models

### 7.2.3 DVDI\_SPST\_ATR

Query the attributes of the Subpicture Streams of the courant VTS.

`DWORD DVDQueryAttribute (DVDI_SPST_ATR, PSPST_ATR pSPST_ATR);`

#### Arguments

PSTST\_ATR pSPST\_ATR Pointer to a SPST\_ATR structure:

```
typedef struct {
    BYTE bSPST_Ns;           // Number of Subpicture Streams available (1..32)
    BOOL bAvailable[MAX_SP]; // If True, the subpicture stream is available in the current Program Chain.
    BYTE bSubpictureType [MAX_SP]; // Subpicture type : 00b Not specified, 01b Language
    WORD wLanguageCode [MAX_SP]; // See Language codes
    WORD wLanguageCodeExtension [MAX_SP]; // See Language codes
} SPST_ATR, *PSPST_ATR;
```

All arrays go from 0 to bSPST\_Ns-1.

#### Return value

Zero if successful. Otherwise returns an error code.

DVDE\_ARG Arguments are not valid.

DVDE\_UNAUTHORIZED Unauthorized operation.

#### See also

SubPictureStreamChange(1.4.31)

#### Remarks

Here is a small code:

```
DWORD dwReturn;
SPSR_ATR spst_atr;
dwReturn = DVDQueryAttribute(DVDI_SPST_ATR, (DWORD)& spst_atr);
dwReturn = (DWORD) spst_atr.bSPST_Ns;
dwReturn = (DWORD) spst_atr.bSubpictureType [0];
dwReturn = (DWORD) spst_atr.wLanguageCode [0];
dwReturn = (DWORD) spst_atr.wLanguageCodeExtension [0];
if (spst_atr.bAvailable [0])
    return TRUE;

else return FALSE;
```

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks		•			
Windows CE		•			
Linux		•			

Figure 7.7: DVDI\_SPST\_ATR Operating Systems and Streaming Models

#### 7.2.4 DVDI\_TT\_SRPTI

Query the Title Search Pointer Table Information of the current VMG.

```
DWORD DVDQueryAttribute (DVDI_TT_SRPTI, PTT_SRPTI pTT_SRPTI);
```

*Arguments*

PTT\_SRPTI pTT\_SRPTI Pointer to a TT\_SRPTI structure:

```
typedef struct {
    BYTE bTT_SRP_Ns;          // Number of Titles (1..99)
    BYTE bALG_Ns [MAX_AGL];   // Number of Angles (1-9)
    WORD wPTT_Ns [MAX_TT];    // Number of Part_of_Titles
} TT_SRPTI, * TT_SRPTI;
```

All arrays go from 1 to bTT\_SRP\_Ns.

*Return value*

Zero if successful. Otherwise returns an error code.

DVDE\_ARG Arguments are not valid

DVDE\_UNAUTHORIZED Unauthorized operation

*Remarks*

There can be between 1 and 99 titles and 1 to 999 Part\_of\_Titles in each Title

*See also*

TitlePlay(1.4.34)

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks		•			
Windows CE		•			
Linux		•			

Figure 7.8: DVDI\_TT\_SRPTI Operating Systems and Streaming Models

#### 7.2.5 DVDI\_VIDEO\_MODE

Query the attributes of the current Video Mode.

```
DWORD DVDQueryAttribute ( DVDI_VIDEO_MODE, PVIDEOMODE pVIDEO_MODE);
```

*Arguments*

PVIDEO\_MODE pVIDEO\_MODE Pointer to a VIDEO\_MODE structure

```
typedef struct {
  BYTE bAspectRatio // Aspect Ratio
  BYTE bDisplayMode // Describes the permitted display modes on 4:3 monitors.
} VIDEO_MODE, *PVIDEO_MODE;
```

See

Return value

Zero if successful. Otherwise returns an error code.

DVDE\_ARG Arguments are not valid.

DVDE\_UNAUTHORIZED Unauthorized operation.

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks		•			
Windows CE		•			
Linux		•			

Figure 7.9: DVDI\_VIDEO\_MODE Operating Systems and Streaming Models

7.2.6 DVDI\_CURRENT\_UOPS

Query which user commands are currently valid.

```
DWORD DVDQueryAttribute (DVDI_CURRENT_UOPS, NULL);
```

Arguments

NULL



UOP	Value(Bit number)	Function
UOP_FLAG_Title_Or_Time_Play	0	TitlePlay, TimePlay
UOP_FLAG_Chapter_Search_Or_Play	1	ChapterSearch, ChapterPlay
UOP_FLAG_Title_Play	2	TitlePlay
UOP_FLAG_Stop	3	StopForResume
UOP_FLAG_GoUp	4	GoUp
UOP_FLAG_Time_Or_Chapter_Search	5	TimeSearch, ChapterSearch
UOP_FLAG_Prev_Or_Top_PG_Search	6	PrevPGSearch, TopPGSearch
UOP_FLAG_Next_PG_Search	7	NextPGSearch
UOP_FLAG_Forward_Scan	8	ForwardScan
UOP_FLAG_Backward_Scan	9	BackwardScan
UOP_FLAG_Title_Menu_Call	10	MenuCall with a parameter value of 2 (DVD_MENU_Title)
UOP_FLAG_Root_Menu_Call	11	MenuCall with a parameter value of 3 (DVD_MENU_Root)
UOP_FLAG_SubPic_Menu_Call	12	MenuCall with a parameter value of 4 (DVD_MENU_Subpicture)
UOP_FLAG_Audio_Menu_Call	13	MenuCall with a parameter value of 5 (DVD_MENU_Audio)
UOP_FLAG_Angle_Menu_Call	14	MenuCall with a parameter value of 6 (DVD_MENU_Angle)
UOP_FLAG_Chapter_Menu_Call	15	MenuCall with a parameter value of 7 (DVD_MENU_Chapter)
UOP_FLAG_Resume	16	Resume
UOP_FLAG_Button_Select_Or_Activate	17	UpperButtonSelect, LowerButtonSelect, LeftButtonSelect, RightButtonSelect, ButtonActivate, ButtonSelectAndActivate
UOP_FLAG_Still_Off	18	StillOff
UOP_FLAG_Pause_On	19	PauseOn, MenuLanguageSelect
UOP_FLAG_Audio_Stream_Change	20	AudioStreamChange
UOP_FLAG_SubPic_Stream_Change	21	SubpictureStreamChange
UOP_FLAG_Angle_Change	22	AngleChange, ParentalLevelSelect
UOP_FLAG_Karaoke_Audio_Pres_Mode_Change	23	KaraokeAudioPresentationModeChange (not supported)
UOP_FLAG_Video_Pres_Mode_Change	24	VideoModePreference (not supported)

Figure 7.10: System Parameters settings

*Return value*

If successful, returns a DWORD value containing bits for all user operations (UOP). Each bit in the DWORD represents the state (valid or not valid) of a user operation. If the bit corresponding to a user operation is set, then that user operation is prohibited.

0xFFFFFFFF if error (you can query error code using DVDGetLastError).

DVDE\_ARG Arguments are not valid.

DVDE\_UNAUTHORIZED Unauthorized operation.

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks		•			
Windows CE		•			
Linux		•			

Figure 7.11: DVDI\_CURRENT\_UOPS Operating Systems and Streaming Models

## 7.3 DVD, SVCD, VCD settings

Setting code		Description	Models	Status
FMPL_DISC_TYPE	r	Retrieves the type of the disc (DVD/SVCD or VCD)	DVD/SVCD/VCD	Not implemented
FMPL_ANGLES_AVAILABLE	r	Retrieves a value specifying the number of available angles.	DVD	Implemented
FMPL_AUDIO_STREAMS_AVAILABLE	r	Retrieves a value specifying the number of available audio streams	DVD/SVCD/VCD	Implemented
FMPL_BALANCE	r/w	Sets or retrieves a value indicating the stereo balance.	All	Implemented
FMPL_BUTTONS_AVAILABLE	r	Retrieves a value specifying the number of available buttons.	DVD	Implemented
FMPL_CAN_SCAN	r	Retrieves a value specifying whether the current file supports scanning (fast-forwarding and rewinding).	DVD	Implemented
FMPL_CAN_SEEK	r	Retrieves a value specifying whether the current file has the ability to seek to a specific time.	DVD	Implemented
FMPL_CC_ACTIVE	r/w	Sets or retrieves a value specifying the closed captioning service state (on or off).		
FMPL_COLOR_KEY	r/w	Sets or retrieves the color key being used by the DVD playback.		
FMPL_CURRENT_ANGLE	r/w	Retrieves a the current angle of a multi-angle title.	DVD	Implemented
FMPL_CURRENT_AUDIO_STREAM	r/w	Sets or retrieves a value specifying the current audio stream.	DVD/SVCD/VCD	Implemented
FMPL_AUDIO_OUTPUT	r/w	Sets or retrieves audio output device; 1: AC3,DTS send to external decoder, 0: Audio decoded in hardware and sent to stereo and Spdif decoder. Can change the audio output only in stop state (not on fly).	DVD/SVCD/VCD	Implemented
FMPL_CURRENT_BUTTON	r	Retrieves a value specifying the number of the current button	DVD	Implemented
FMPL_CURRENT_CHAPTER	r	Retrieves a value specifying the chapter number currently being played	DVD	Implemented
FMPL_CURRENT_DISC_SIDE	r	Retrieves a value specifying the current disc side		
FMPL_CURRENT_DOMAIN	r	Retrieves a value specifying the current DVD domain of the DVD player	DVD	Implemented
FMPL_CURRENT_SUBPICT_STREAM	r/w	Sets or retrieves a value specifying the source of the subpicture	DVD	Implemented
FMPL_CURRENT_TIME	r	Retrieves a value specifying the current playback time, in seconds	DVD/VCD/SVCD/CD-Audio	Implemented
FMPL_CURRENT_TITLE	r	Retrieves a value specifying the title number currently being played	DVD	Implemented
DisplaySize	r/w	Sets or retrieves a value specifying the size of the image display window		
FMPL_FRAMES_PER_SECOND	r	Retrieves a value specifying the number of frames per second used by the DVD title		
FMPL_MUTE	r/w	Sets or retrieves a value indicating the current mute state		Implemented
to be ct'd...				

Setting code (ct'd)		Description (ct'd)	Models	Status
FMPI_RATE	r/w	Sets or retrieves a value specifying the clip's playback rate		
FMPI_SHOW_CAPTIONING	r/w	Sets or retrieves a value specifying whether the closed caption area is visible and closed captioning is enabled		
FMPI_SUBPICTURE_ON	r/w	Sets or retrieves a value specifying whether the subpicture is displayed	DVD	Implemented
FMPI_SUBPICTURE_STREAM_AVAILABLE	r	Retrieves a value specifying the number of available subpicture streams	DVD	Implemented
FMPI_TITLES_AVAILABLE	r	Retrieves a value specifying the number of titles available in the current volume	DVD/CD-Audio	Implemented
FMPI_TOTAL_TITLE_TIME	r	Retrieves a value specifying the total playback time for the current title	DVD	Implemented
FMPI_TOTAL_TRACK_TIME	r	Retrieves a value specifying the total playback time for the current track	CD-Audio	Implemented
FMPI_UNIQUE_ID	r	Retrieves a value specifying the unique identifier associated with the DVD volume		
FMPI_LEFT_VOLUME	r/w	Sets or retrieves a value specifying the volume, in hundredths of decibels		
FMPI_RIGHT_VOLUME	r/w	Sets or retrieves a value specifying the volume, in hundredths of decibels		
FMPI_CURRENT_TRACK	r	Retrieves a value specifying the track number currently being played	VCD/SVCD/CD-Audio	Implemented
FMPI_CURRENT_SPEED	r	Retrieves the current scanning speed	DVD	Implemented

## 7.4 General settings

Use FMPIGet function to get a value of specified setting and FMPISet function to change the value of driver with specified setting.

Setting code		Description
FMPI_TRICKMODE	r/w	Signal a TrickMode section. The driver waits the next section to go into or out of TrickMode playback (I-Frames only, no audio). Values are :  FMPIV_TRICKMODE_NOCHANGE No trick mode change  FMPIV_TRICKMODE_START Start trick mode after next section  FMPIV_TRICKMODE_END End trick mode after next section
FMPI_STC	r	System Time Clock of the Decoder in PTS (32bit value so most significant bit is ignored).
FMPI_PTS	r	Last Presentation Time Stamp from the stream depacketizer (32bit value so most significant bit is ignored).
FMPI_AUDIO_COUNT	r	Number of available audio streams
FMPI_AUDIO_SELECT	r/w	Select a desired audio stream for transport stream. There are two modes for the setting : if $\leq 8$ : the value is interpreted as zero-based. if $\geq 8$ : the value is interpreted as a PID.
FMPI_BRIGHTNESS	r/w	Set brightness (0..1000).
FMPI_SATURATION	r/w	Set saturation (0..1000).
FMPI_CONTRAST	r/w	Set contrast (0..1000).
FMPI_AUDIO_SPEED	r/w	User can modify the current audio sample rate by a multiplying factor(DWORD) using FMPISet/Get function. The default value is 1000 in decimal.

Figure 7.12: General settings

### 7.4.1 FMPI\_STANDARDTV

Select the output format for the TV, NTSC or PALs.

```
DWORD FMPI_Set (FMPI_STANDARDTV, DWORD dwValue);
```

#### Arguments

DWORD dwValue the desired format. Supported formats are:

- 0 NTSC
- 1 PAL
- 2 PAL-60
- 3 PAL-M

#### Return value

Zero if successful. Otherwise returns an error code.

OS	Push Model	DVD	VCD	SVCD	CD Audio
VxWorks	•	•	•	•	
Windows CE	•	•	•	•	
Linux	•	•	•	•	

Figure 7.13: FMPI\_STANDARDTV Operating Systems and Streaming Models

## 7.5 Source and destination settings

Use

```
DWORD FMPI_Set(FMPI_SOURCE_WINDOW, WND* SrcWnd);
```

with

```
typedef struct {
    DWORD x;
    DWORD y;
    DWORD w;
    DWORD h;
} WND;
```

Setting code	Description
FMPI_SOURCE_WINDOW	Sets or retrieves source window coordinates.
FMPI_VISIBLE_SOURCE_WINDOW	Sets or retrieves visible source window coordinates.
FMPI_DESTINATION_WINDOW	Sets or retrieves destination window coordinates.

and

```
DWORD FMPI_Set(FMPI_OVERLAY_FLAGS, DWORD dwOverlayFlags);
```

with dwOverlayFlags a combination of:

VIDEO\_ZOOM\_ENABLE Zoom VisibleSource (from Source) area in Destination window

VIDEO\_TV\_DEST\_ENABLE use Destination window instead of full screen on TV

VIDEO\_HDTV\_DEST\_ENABLE use Destination window instead of full screen on HDTV

OSD\_VIDEO\_INDEPENDENT\_DEST On screen display can have different destination than the MPEG Video

## Chapter 8

# Streaming video

This driver has been specially designed and optimized for streaming video.

The data interface is based on a push model: the application must create a Task or Thread and then push the data it receives from the network (or from the disk) into the driver.

Basically the application will create a Task dedicated to listen to a Live Source of MPEG data and bring it to system memory to be transferred to the MPEG driver.

### *Synchronization*

1. The application must send play (FMPPPlay) or pause (FMPPPause) to the driver before sending any data.
2. The application must stop sending data to the driver before sending a stop command (FMPStop).
3. The driver takes care of the memory management : the application has to ask a buffer to the driver. Then the application can fill this buffer with data and then can push the buffer back to the driver.
4. Overflow is handled by the data source: overflow condition happens when the memory manager cannot deliver a buffer (FMPGetBuffer block until a buffer is available or return an error code if the bBlockingCall argument is set to FALSE).
5. Underflow is handled by the MPEG driver: underflow condition happens when the video decoder is in starvation.  
The application can prevent this condition to happen by re-buffering data when a low-water mark condition is reached.  
It is recommended to use double-buffering to prevent underflow conditions.
6. The application cannot call any FMPStop or FMPClose before pushing back all the buffers to driver. If FMPStop or FMPClose is called between a FMPGetBuffer/FMPPush call sequence, the driver will not be able to release its memory.

### *PES Support*

To enable PES playback, open the driver (FMPOpen) with the FMPF\_PES option.

When you get a buffer from the driver (FMPGetBuffer) you have to specify if you want a video or an audio buffer in the FMP\_BUFFER structure.

Here is a small code snippet :

```
FMP_BUFFER FMPBuf;  
if (FMPOpen (FMPF_PES, 1024*32, 8, 0, 0) != FMPE_OK)  
{  
    FMPClose ();  
    goto exit_now;  
}
```

```

FMPPlay ();
while (1)
{
    FMPBuf.dwFlagsEx = FMP_VIDEO_PES;
    if (FMPGetBuffer (&FMPBuf, TRUE) == FMPE_OK)
    {
        nbytes = fread (FMPBuf.pBuffer, 1, 1024*32, file_handle);
        FMPBuf.dwDataSize = nbytes;
        if (FMPPush (&FMPBuf) != FMPE_OK)
        {
            printf ("FMPPush() error");
            printf ("FMPBuf.pBuffer = 0x%08lx", FMPBuf.pBuffer);
        }
    }
}

```

The pes packets must not have 0 as the packet length.

If there are 0 length (in the case of video packets inside a transport stream) then there is a special PES format that the MPEG driver can accept:

The PES file format is meant to model how PES data coming from a transport stream will be pushed to the MPEG driver for decoding. For both audio and video files, the data can be formatted as follows.

A data block may or may not contain an entire packet, but it will never contain more than one packet.

In other words, each data block may contain at most one PES packet.

If the whole PES packet cannot fit in the data block, then the next data block(s) will contain the rest of the packet. PES packets always start on data block boundaries.

DataBlock = { 00 00 01 BA — Flags (4 bytes) — PacketDataLength (4 bytes) — packet data }

DataBlockLength = 4 + 4 + 4 + PacketDataLength

Thus a stream that an application would push to the MPEG driver would look like the following.

PushedVideoStream = DataBlock, DataBlock, DataBlock, ..., DataBlock PushedAudioStream = DataBlock, DataBlock, DataBlock, ..., DataBlock

When the application calls FMPPush, the buffer will be filled with a single DataBlock. The length of the FMP\_BUFFER will be set to DataBlockLength. The application will also set the dwFlagsEx to flag either audio or video as appropriate.

Flag values are as follow:

1. - Continuation packet (flag 0)
2. - Discontinuity (flag 1)
3. - Packet start (flag 2)
4. - Packet checksum error (flag 3)

Here is the beginning of a video sequence.

```

** 0 - Len 1944 (0x798) Flags 0x00000002 Offset 12 (0xc)
** 1 - Len 1640 (0x668) Flags 0x00000002 Offset 1968 (0x7b0)
** 2 - Len 3000 (0xbb8) Flags 0x00000002 Offset 3620 (0xe24)
** 3 - Len 1656 (0x678) Flags 0x00000002 Offset 6632 (0x19e8)
** 4 - Len 1616 (0x650) Flags 0x00000002 Offset 8300 (0x206c)
** 5 - Len 1992 (0x7c8) Flags 0x00000002 Offset 9928 (0x26c8)
** 6 - Len 1824 (0x720) Flags 0x00000002 Offset 11932 (0x2e9c)
** 7 - Len 1672 (0x688) Flags 0x00000002 Offset 13768 (0x35c8)
** 8 - Len 3400 (0xd48) Flags 0x00000002 Offset 15452 (0x3c5c)
** 9 - Len 2192 (0x890) Flags 0x00000002 Offset 18864 (0x49b0)

```

\*\* 10 - Len 2264 (0x8d8) Flags 0x00000002 Offset 21068 (0x524c)  
\*\* 11 - Len 3944 (0xf68) Flags 0x00000002 Offset 23344 (0x5b30)  
\*\* 12 - Len 3264 (0xcc0) Flags 0x00000002 Offset 27300 (0x6aa4)  
\*\* 13 - Len 3496 (0xda8) Flags 0x00000002 Offset 30576 (0x7770)  
\*\* 14 - Len 16356 (0x3fe4) Flags 0x00000002 Offset 34084 (0x8524)  
\*\* 15 - Len 16356 (0x3fe4) Flags 0x00000000 Offset 50452 (0xc514)  
\*\* 16 - Len 16356 (0x3fe4) Flags 0x00000000 Offset 66820 (0x10504)  
\*\* 17 - Len 16356 (0x3fe4) Flags 0x00000000 Offset 83188 (0x144f4)  
\*\* 18 - Len 12696 (0x3198) Flags 0x00000000 Offset 99556 (0x184e4)  
\*\* 19 - Len 2024 (0x7e8) Flags 0x00000002 Offset 112264 (0x1b688)

For each line, the application would do an FMPPush, pushing the entire data block as described above.

Note that blocks 14 - 18 are actually a single PES packet. The others are all individual PES packets.

### *Implementing trick modes*

This section explains how to enter and exit trick mode using the push model.

By entering trick mode, you indicate the MPEG decoder to decode only I frames. B and P frames will be ignored. This is very useful for simulating fast forward and fast rewind.

To have a seamless effect (that means not stopping the streaming), the command has to be embedded inside the stream. In the FMP\_BUFFER structure, there is a dwFlags paramaters that can be or'ed to FMP\_TRICKMODE\_START to start the trick mode and FMP\_TRICKMODE\_END to end it.

When you flag your buffer with FMP\_TRICKMODE\_START, the driver will :

1. - Stop decoding B and P frame from byte 0 of the flagged buffer.
2. - Stop decoding audio from byte 0 of the flagged buffer.
3. - Stop synchronizing Video to Audio.

These effects will stay in place until a buffer is flagged FMP\_TRICKMODE\_END :

When you flag your buffer with FMP\_TRICKMODE\_START, the driver will :

1. - Start decoding B and P frame from byte 0 of the flagged buffer.
2. - Start decoding audio from byte 0 of the flagged buffer.
3. - Start synchronizing Video to Audio.

This will resume normal playback.





## Chapter 9

# Use guidelines

### 9.1 Programming notes

Some samples in C are included with the driver.

Please refer to them for more information on how to use the driver.

### 9.2 Technical support

If you need additional information, if you encounter problems or if you have any suggestions, please contact REALmagic developer support at <mailto:support@sdesigns.com>.

# Index

AST\_ATR, 91  
CDDAFastForward, 10  
CDDAGetNumberOfTracks, 11  
CDDAGetStatus, 11  
CDDAGetTrackDuration, 12  
CDDAI\_TOC, 89  
CDDANextTrack, 12  
CDDAPlayTrack, 12  
CDDAPrevTrack, 13  
CDDARepeat, 13  
CDDARewind, 14  
CDDA functions, 10  
CDDA settings, 89  
DVDI\_AST\_ATR, 91  
DVDI\_CURRENT\_UOPS, 96  
DVDI\_SPRM, 93  
DVDI\_SPST\_ATR, 94  
DVDI\_TT\_SRPTI, 95  
DVDI\_VIDEO\_MODE, 95  
DVD - celesteapp, 76  
DVD registries, 83  
DVD settings, 91  
DVD, SVCD, VCD settings, 98  
FMPAbout, 34  
FMPAnalogOverlay, 50  
FMPClose, 35  
FMPFlush, 53  
FMPGetBuffer, 53  
FMPGetProfileWord, 37  
FMPGet, 35  
FMPI\_STANDARDTV, 100  
FMPM\_AUTOPAUSE, 60  
FMPM\_CDDA\_PSD\_END, 60  
FMPM\_DVDROM\_ERR, 65  
FMPM\_DVDROM\_NOT\_READY, 66  
FMPM\_DVD\_ANGLES\_BLOCK, 61  
FMPM\_DVD\_ANGLE\_CHANGE, 60  
FMPM\_DVD\_AUDIO\_STREAM\_CHANGE, 61  
FMPM\_DVD\_BUTTONS\_CHANGE, 61  
FMPM\_DVD\_BUTTON\_CHANGE, 61  
FMPM\_DVD\_CHAPTER\_CHANGE, 62  
FMPM\_DVD\_CURRENT\_TIME, 62  
FMPM\_DVD\_DOMAIN\_CHANGE, 62  
FMPM\_DVD\_END\_PLAYBACK, 62  
FMPM\_DVD\_FATAL\_ERROR, 62  
FMPM\_DVD\_MACROVISION\_LEVEL, 63  
FMPM\_DVD\_NO\_FP\_PGC, 63  
FMPM\_DVD\_PARENTAL\_CHECK, 63  
FMPM\_DVD\_PARENTAL\_ERROR, 63  
FMPM\_DVD\_PARENTAL\_LEVEL\_CHANGE, 64  
FMPM\_DVD\_PLAYBACK\_STOPPED, 64  
FMPM\_DVD\_PROGRAM\_START, 64  
FMPM\_DVD\_STILL\_OFF, 64  
FMPM\_DVD\_STILL\_ON, 64  
FMPM\_DVD\_SUBPICTURE\_STREAM\_CHANGE, 65  
FMPM\_DVD\_TITLE\_CHANGE, 65  
FMPM\_DVD\_VALID\_UOPS\_CHANGE, 65  
FMPEOS, 66  
FMPEERROR, 66  
FMPEREGION\_MISMATCH, 66  
FMPESTARVATION, 66  
FMPE\_SVCD\_PSD\_END, 66  
FMPE\_TRICK\_MODE\_CHANGE, 67  
FMPOpenDiscPlayback, 38  
FMPOpen, 54  
FMPPause, 40  
FMPPPlay, 41  
FMPPProperty, 41  
FMPPush, 55  
FMPPQueryInterface, 49  
FMPPSetDestination, 51  
FMPPSetSource, 52  
FMPPSetVideoPortDimensions, 14  
FMPPSetVisibleSource, 52  
FMPPSet, 45  
FMPPStop, 46  
FMPPTrayOperation, 46  
FMPPWriteProfileWord, 47  
FMP\_BUFFER, 55  
FMP\_OPENSTRUCT, 38  
MPEGDriverEntry, 48  
MPEGDriverUnload, 49  
SPST\_ATR, 94  
TT\_SRPTI, 95  
VIDEO\_MODE, 96  
  
Analogoverlay, 75  
AngleChange, 15  
AudioStreamChange, 15  
  
BackwardScan, 16

- ButtonActivate, 16
- ButtonSelectAndActivate, 17
- Callback functions, 9, 10
- ChapterPlay, 17
- ChapterSearch, 18
- ClearRepeatAB, 18
- Command description, 9
- DefaultPGSearch, 19
- Digital overlay, 14
- Disc functions, 15
- Error messages, 57
- Events, 59
- FastForward, 19
- ForwardScan, 20
- General functions, 34
- General settings, 99
- GoUp, 20
- Hwlibrary, 78, 84
- Installation, 69
- KaraokeAudioPresentationModeChange, 20
- Language codes, 71
- LeftButtonSelect, 21
- Linux, 69
- LowerButtonSelect, 21
- MenuCall, 22
- NextPGSearch, 22
- NumericSelections, 23
- Os - Linux, 75
- Os - WinCE, 83
- Picture placement, 50
- PrevPGSearch, 24
- Programming notes, 105
- Push model functions, 53
- Registry values, 75
- RepeatAB, 24
- RepeatChapter, 25
- RepeatTitle, 25
- Resume, 25
- ResumeBookmark, 26
- ReturnPGSearch, 27
- Rewind, 27
- RightButtonSelect, 28
- Settings, 89
- Source and destination settings, 100
- StillOff, 28
- StopForResume, 28
- StoreBookmark, 29
- Streaming video, 101
- SubPictureStreamChange, 29
- Technical support, 105
- TimePlay, 30
- TimeSearch, 31
- TitlePlay, 33
- TopPGSearch, 34
- UpperButtonSelect, 34
- Use guidelines, 105
- variable types, 40
- VxWorks, 69
- Windows CE, 69