

Programming interface for MVC 20x media block

Copyright (c) 2006-2012 MikroM GmbH. All rights reserved. MikroM GmbH

Version: 9.2.0.0

# Table of Contents

Design Guide	1
Overview	1
Getting started	1
Basic Types	1
Find existing MVC cards	2
Authentication	3
Symbol Reference	4
mvc2 Namespace	4
ActiveDecoderInfo Class	5
ActiveDecoderInfo::~ActiveDecoderInfo Destructor	6
ActiveDecoderInfo::ActiveDecoderInfo Constructor (ActiveDecoderInfo &)	6
ActiveDecoderInfo::ActiveDecoderInfo Constructor (ActiveDecoderInfoPrivate *)	6
ActiveDecoderInfo Methods	6
ActiveDecoderInfo::getChannel Method	6
ActiveDecoderInfo Operators	7
ActiveDecoderInfo::= Operator	7
ActiveDecoderIterator Class	7
ActiveDecoderIterator::~ActiveDecoderIterator Destructor	8
ActiveDecoderIterator::ActiveDecoderIterator Constructor (ActiveDecoderIterator &)	8
ActiveDecoderIterator::ActiveDecoderIterator Constructor (TMmRc *, MvcDevice &)	8
ActiveDecoderIterator Methods	8
ActiveDecoderIterator::getNext Method	8
ActiveDecoderIterator Operators	8
ActiveDecoderIterator::= Operator	8
AudioDecoderErrors Class	9
AudioDecoderErrors::~AudioDecoderErrors Destructor	10
AudioDecoderErrors::AudioDecoderErrors Constructor ()	11
AudioDecoderErrors::AudioDecoderErrors Constructor (AudioDecoderErrors &)	11
AudioDecoderErrors::AudioDecoderErrors Constructor (MvcDecoderPrivate *)	11
AudioDecoderErrors::AudioDecoderErrors Constructor (MvcDevice &, uint32_t)	11
AudioDecoderErrors Methods	11
AudioDecoderErrors::getClippedSamples Method	11
AudioDecoderErrors::getInputUnderruns Method	12
AudioDecoderErrors::getMaxClippingPeriode Method	12
AudioDecoderErrors::getNumberOfClippings Method	12
AudioDecoderErrors::getPlayedSamples Method	12

	AudioDecoderErrors::getSilentSamples Method	12
	AudioDecoderErrors::getSkippedSamples Method	12
	AudioDecoderErrors Operators	12
	AudioDecoderErrors::= Operator	13
Αu	dioDecoderStatus Class	13
	AudioDecoderStatus::AudioDecoderStatus Constructor ()	14
	AudioDecoderStatus::AudioDecoderStatus Constructor (AudioDecoderStatus &)	15
	AudioDecoderStatus::AudioDecoderStatus Constructor (MvcDecoderPrivate *)	15
	AudioDecoderStatus::AudioDecoderStatus Constructor (MvcDecoderPrivate *, AudioDecoderStatus &)	15
	AudioDecoderStatus::AudioDecoderStatus Constructor (MvcDevice &, uint32_t)	15
	AudioDecoderStatus Enumerations	15
	mvc2::AudioDecoderStatus::INTERVAL Enumeration	15
	AudioDecoderStatus Methods	16
	AudioDecoderStatus::getAudioLevel Method	16
	AudioDecoderStatus::isAudioChannelActive Method	16
	AudioDecoderStatus Operators	16
	AudioDecoderStatus::= Operator	16
Αu	dioOutput Class	16
	AudioOutput::~AudioOutput Destructor	18
	AudioOutput::AudioOutput Constructor ()	18
	AudioOutput::AudioOutput Constructor (AudioOutput &)	18
	AudioOutput::AudioOutput Constructor (TMmRc *, MvcDevice &, uint32_t, uint32_t)	18
	AudioOutput Enumerations	19
	mvc2::AudioOutput::AUDIOPROPERTY Enumeration	19
	AudioOutput Methods	19
	AudioOutput::getAudioOutputStatus Method	19
	AudioOutput::getChannelMute Method	19
	AudioOutput::getNumberOfChannels Method	20
	AudioOutput::getOuputFrequency Method	20
	AudioOutput::getOutputDelay Method	20
	AudioOutput::setChannelMute Method	20
	AudioOutput::setOutputDelay Method	21
	AudioOutput::setOutputFrequency Method	21
	AudioOutput::setUserData Method	22
	AudioOutput::setVolume Method	22
	AudioOutput::setVolumeDB Method	22
	AudioOutput Operators	22
	AudioOutput::= Operator	23
	AudioOutput::bool Operator	23
Αu	dioOutputStatus Class	23
	AudioOutputStatus::~AudioOutputStatus Destructor	24
	AudioOutputStatus::AudioOutputStatus Constructor ()	25

AudioOutputStatus::AudioOutputStatus Constructor (AudioOutputStatus &)	25
AudioOutputStatus::AudioOutputStatus Constructor (AudioOutputStatusPrivate *)	25
AudioOutputStatus::AudioOutputStatus Constructor (MvcDevice &, uint32_t, TIME_RESOLUTION)	25
AudioOutputStatus Enumerations	25
mvc2::AudioOutputStatus::TIME_RESOLUTION Enumeration	25
AudioOutputStatus Data Members	26
AudioOutputStatus::m_StatusPriv Data Member	26
AudioOutputStatus Methods	26
AudioOutputStatus::getOvershootLength Method	26
AudioOutputStatus::getOvershootSamples Method	26
AudioOutputStatus::getSampleActivity Method	26
AudioOutputStatus::getSampleLevel Method	26
AudioOutputStatus::getSampleLevelDB Method	27
AudioOutputStatus::isChannelActive Method	27
AudioOutputStatus Operators	27
AudioOutputStatus::= Operator	27
AudioOutputStatus::bool Operator	27
BufferStatus Class	27
BufferStatus::~BufferStatus Destructor	29
BufferStatus::BufferStatus Constructor (BufferStatus &)	29
BufferStatus::BufferStatus Constructor (MvcDecoderPrivate *)	29
BufferStatus::BufferStatus Constructor (MvcDevice &, uint32_t)	29
BufferStatus Enumerations	29
mvc2::BufferStatus::INTERVAL Enumeration	29
BufferStatus Methods	30
BufferStatus::getBitrateDecoder Method	30
BufferStatus::getBitrateDriver Method	30
BufferStatus::getDecoderBufferMax Method	30
BufferStatus::getDecoderBufferSize Method	30
BufferStatus::getDecoderFullness Method	30
BufferStatus::getFreeDataBuffers Method	30
BufferStatus::getPendingFrames Method	31
BufferStatus::getReturnCode Method	31
BufferStatus Operators	31
BufferStatus::= Operator	31
BufferStatus::bool Operator	31
ColorConversionMatrix Class	31
ColorConversionMatrix::ColorConversionMatrix Constructor	32
ColorConversionMatrix Operators	32
ColorConversionMatrix::= Operator	32
ColorSpace Class	32
ColorSpace::ColorSpace Constructor (ColourPrimaries, TransferCharacteristics, MatrixCoefficients)	33

	ColorSpace::ColorSpace Constructor (Space)	34
	ColorSpace Enumerations	34
	mvc2::ColorSpace::ColourPrimaries Enumeration	34
	mvc2::ColorSpace::MatrixCoefficients Enumeration	34
	mvc2::ColorSpace::Space Enumeration	35
	mvc2::ColorSpace::TransferCharacteristics Enumeration	35
	ColorSpace Methods	35
	ColorSpace::getColorSpacePacked Method	35
	ColorSpace::getSpace Method	35
	ColorSpace Operators	36
	ColorSpace::= Operator	36
Cc	onfigAccess Class	36
	ConfigAccess::~ConfigAccess Destructor	37
	ConfigAccess::ConfigAccess Constructor ()	38
	ConfigAccess::ConfigAccess Constructor (ConfigAccess&)	38
	ConfigAccess::ConfigAccess Constructor (TMmRc *, MvcDevice &)	38
	ConfigAccess Methods	38
	ConfigAccess::getLogFilter Method	38
	ConfigAccess::getLogFilterConsole Method	38
	ConfigAccess::getMessageFifoClients Method	38
	ConfigAccess::getMessageFifoSize Method	39
	ConfigAccess::getNetworkDHCP Method	39
	ConfigAccess::getNetworkIPAddress Method	39
	ConfigAccess::getNetworkMask Method	39
	ConfigAccess::setLogFilter Method	39
	ConfigAccess::setLogFilterConsole Method	39
	ConfigAccess::setMessageFifoClients Method	40
	ConfigAccess::setMessageFifoSize Method	40
	ConfigAccess::setNetworkDHCP Method	40
	ConfigAccess::setNetworkIPAddress Method	40
	ConfigAccess::setNetworkMask Method	40
	ConfigAccess Operators	40
	ConfigAccess::= Operator	40
	ConfigAccess::bool Operator	41
CF	PUInfo Class	41
	CPUInfo::~CPUInfo Destructor	42
	CPUInfo::CPUInfo Constructor ()	42
	CPUInfo::CPUInfo Constructor (CPUInfo &)	42
	CPUInfo::CPUInfo Constructor (MvcDevicePrivate *, uint32_t)	42
	CPUInfo Enumerations	42
	mvc2::CPUInfo::CPUTYPE Enumeration	43
	CPUInfo Methods	43

CPUInfo::getCoreFrequency Method	43
CPUInfo::getCoreFrequencykHz Method	43
CPUInfo::getCPUType Method	43
CPUInfo::getCPUTypeString Method	43
CPUInfo::getHardwareVersion Method	43
CPUInfo::getNumberOfCores Method	44
CPUInfo Operators	44
CPUInfo::= Operator	44
CPUInfo::bool Operator	44
DataBuffer Class	44
DataBuffer::~DataBuffer Destructor	46
DataBuffer::DataBuffer Constructor ()	46
DataBuffer::DataBuffer Constructor (DataBuffer&)	46
DataBuffer::DataBuffer Constructor (MvcDecoderPrivate *, MemoryBuffer *)	46
DataBuffer Methods	46
DataBuffer::copy Method	46
DataBuffer::getBufferAddress Method	47
DataBuffer::getFreeSize Method	47
DataBuffer::getInSize Method	47
DataBuffer::getTimeStamp Method	48
DataBuffer::send Method ()	48
DataBuffer::send Method (uint32_t)	48
DataBuffer::setDecryptionSize Method	48
DataBuffer::setInSize Method	49
DataBuffer::setKeyId Method	49
DataBuffer::setKeyIndex Method	49
DataBuffer::setMicValue Method	50
DataBuffer::setTimeStamp Method	50
DataBuffer::setUserData Method	50
DataBuffer::wait Method	51
DataBuffer Operators	51
DataBuffer::= Operator	51
DataBuffer::bool Operator	51
DecoderErrors Class	52
DecoderErrors::~DecoderErrors Destructor	52
DecoderErrors::DecoderErrors Constructor ()	53
DecoderErrors::DecoderErrors Constructor (DecoderErrors &)	53
DecoderErrors::DecoderErrors Constructor (MvcDecoderPrivate *)	53
DecoderErrors::DecoderErrors Constructor (MvcDevice &, uint32_t)	53
DecoderErrors Methods	53
DecoderErrors::getReturnCode Method	53
DecoderErrors Operators	53

DecoderErrors::= Operator	54
DecoderErrors::bool Operator	54
DecoderStatus Class	54
DecoderStatus::~DecoderStatus Destructor	55
DecoderStatus::DecoderStatus Constructor ()	55
DecoderStatus::DecoderStatus Constructor (DecoderStatus &)	55
DecoderStatus::DecoderStatus Constructor (MvcDecoderPrivate *)	55
DecoderStatus::DecoderStatus Constructor (MvcDecoderPrivate *, DecoderStatus &)	56
DecoderStatus::DecoderStatus Constructor (MvcDevice &, uint32_t)	56
DecoderStatus::DecoderStatus Constructor (int32_t *, uint32_t)	56
DecoderStatus Methods	56
DecoderStatus::getReturnCode Method	56
DecoderStatus::getTimeStamp Method	56
DecoderStatus::getUserData Method	57
DecoderStatus Operators	57
DecoderStatus::= Operator	57
DecoderStatus::bool Operator	57
Dolby3Dchromaticity Class	57
Dolby3Dchromaticity::~Dolby3Dchromaticity Destructor	58
Dolby3Dchromaticity::Dolby3Dchromaticity Constructor	58
Dolby3Dchromaticity Enumerations	58
mvc2::Dolby3Dchromaticity::Measurement Enumeration	59
Dolby3Dchromaticity Methods	59
Dolby3Dchromaticity::getChroma Method	59
Dolby3Dchromaticity::getLuminance Method	59
Dolby3Dchromaticity::isCorrectionUsed Method	59
Dolby3Dchromaticity::isWPointIterUsed Method	59
Dolby3Dchromaticity::setChroma Method	60
Dolby3Dchromaticity::setLuminance Method	60
ExternalConnector Class	60
ExternalConnector::~ExternalConnector Destructor	61
ExternalConnector::ExternalConnector Constructor ()	61
ExternalConnector::ExternalConnector Constructor (ExternalConnector &)	61
ExternalConnector::ExternalConnector Constructor (ExternalConnectorPrivate *)	62
ExternalConnector::ExternalConnector Constructor (TMmRc *, MvcDevice &)	62
ExternalConnector Data Members	62
ExternalConnector::m_ConnectorPrivate Data Member	62
ExternalConnector Methods	62
ExternalConnector::readI2C Method (uint16_t, uint8_t, uint8_t *, uint32_t)	62
ExternalConnector::readI2C Method (uint16_t, uint8_t *, uint32_t)	62
ExternalConnector::writeI2C Method	63
ExternalConnector Operators	63

ExternalConnector::= Operator	63
ExternalConnector::bool Operator	63
FeatureAccess Class	63
FeatureAccess::~FeatureAccess Destructor	64
FeatureAccess::FeatureAccess Constructor ()	64
FeatureAccess::FeatureAccess Constructor (FeatureAccess&)	64
FeatureAccess::FeatureAccess Constructor (TMmRc *, MvcDevice &)	64
FeatureAccess Methods	65
FeatureAccess::getNextFeature Method	65
FeatureAccess::installFeature Method	65
FeatureAccess::queryFeature Method	65
FeatureAccess Operators	65
FeatureAccess::= Operator	65
FeatureAccess::bool Operator	65
FramePosition Class	66
FramePosition::FramePosition Constructor ()	67
FramePosition::FramePosition Constructor (FramePosition &)	67
FramePosition::FramePosition Constructor (int16_t, int16_t, uint16_t, uint16_t)	68
FramePosition Enumerations	68
mvc2::FramePosition::MODE Enumeration	68
FramePosition Methods	69
FramePosition::fgetX Method	69
FramePosition::fgetY Method	69
FramePosition::fsetX Method	69
FramePosition::fsetY Method	69
FramePosition::getModeX Method	70
FramePosition::getModeY Method	70
FramePosition::getX Method	70
FramePosition::getY Method	70
FramePosition::setModeX Method	70
FramePosition::setModeY Method	71
FramePosition::setX Method	71
FramePosition::setY Method	71
FramePosition Operators	71
FramePosition::!= Operator	71
FramePosition::= Operator	72
FramePosition::== Operator	72
Jpeg2kDecoder Class	72
Jpeg2kDecoder::~Jpeg2kDecoder Destructor	76
Jpeg2kDecoder::Jpeg2kDecoder Constructor ()	76
Jpeg2kDecoder::Jpeg2kDecoder Constructor (TMmRc *, Jpeg2kDecoder &)	76
Jpeg2kDecoder::Jpeg2kDecoder Constructor (TMmRc *, MvcDevice &, uint32_t)	77

Jpeg2kDecoder Enumerations	. <b>*</b>	77
mvc2::Jpeg2kDecoder::CREATION_FLAGS Enu	ımeration	77
mvc2::Jpeg2kDecoder::MODE_4K Enumeration		77
Jpeg2kDecoder Methods		78
Jpeg2kDecoder::getFrameRate Method		78
Jpeg2kDecoder::getFrameRateTicks Method		78
Jpeg2kDecoder::set4kMode Method		78
Jpeg2kDecoder::setColorSpace Method		78
Jpeg2kDecoder::setCutOffLevel Method		79
Jpeg2kDecoder::setFrameRate Method		79
Jpeg2kDecoder::setFrameRateTicks Method		79
Jpeg2kDecoder::setResolutionDivider Method		79
Jpeg2kDecoder Operators		80
Jpeg2kDecoder::= Operator		80
_ogAccess Class		80
LogAccess::~LogAccess Destructor		81
LogAccess::LogAccess Constructor ()		81
LogAccess::LogAccess Constructor (LogAccess&)		81
LogAccess::LogAccess Constructor (TMmRc *, Mvc	Device &, bool)	81
LogAccess Enumerations		82
mvc2::LogAccess::TIMEOUT Enumeration		82
LogAccess Methods		82
LogAccess::abortGetMessage Method		82
LogAccess::getMessage Method		82
LogAccess::setMessageFilter Method		83
LogAccess Operators		83
LogAccess::= Operator		83
LogAccess::bool Operator		83
_ogMessage Class		84
LogMessage::~LogMessage Destructor		85
LogMessage::LogMessage Constructor (LogMessage	je&)	85
LogMessage::LogMessage Constructor (Severity, LogMessage)	ogld, uint32_t, uint32_t, uint64_t, char *, uint32_t)	86
LogMessage::LogMessage Constructor (uint32_t)		86
LogMessage Enumerations		86
mvc2::LogMessage::LogId Enumeration		86
mvc2::LogMessage::LogSubIdFirmware Enumer	ation	87
mvc2::LogMessage::Severity Enumeration		87
LogMessage Methods		88
LogMessage::getCTime Method		88
LogMessage::getId Method		88
LogMessage::getMessageCounter Method		88
LogMessage::getMessageString Method		89

LogMessage::getMessageStringLength Method	89
LogMessage::getMicroSeconds Method	89
LogMessage::getSeverity Method	89
LogMessage::getSubId Method	90
LogMessage::getTimeStamp Method	90
LogMessage::setMessage Method	90
LogMessage Operators	90
LogMessage::= Operator	90
LogMessage::bool Operator	91
LogMessage Friends	91
friend class LogAccess Friend	91
Mpeg2Decoder Class	91
Mpeg2Decoder::~Mpeg2Decoder Destructor	94
Mpeg2Decoder::Mpeg2Decoder Constructor ()	94
Mpeg2Decoder::Mpeg2Decoder Constructor (Mpeg2Decoder&)	95
Mpeg2Decoder::Mpeg2Decoder Constructor (TMmRc *, Mpeg2Decoder &)	95
Mpeg2Decoder::Mpeg2Decoder Constructor (TMmRc *, MvcDevice &)	95
Mpeg2Decoder Enumerations	95
mvc2::Mpeg2Decoder::PROCESSING_3D Enumeration	96
Mpeg2Decoder Operators	96
Mpeg2Decoder::= Operator	96
MvcDecoder Class	96
MvcDecoder::~MvcDecoder Destructor	98
MvcDecoder::MvcDecoder Constructor ()	98
MvcDecoder::MvcDecoder Constructor (MvcDecoder&)	98
MvcDecoder::MvcDecoder Constructor (MvcDecoderPrivate *)	98
MvcDecoder Enumerations	98
mvc2::MvcDecoder::DecoderType Enumeration	98
mvc2::MvcDecoder::TIMEOUT Enumeration	99
MvcDecoder Methods	99
MvcDecoder::flush Method	99
MvcDecoder::getBufferStatus Method	99
MvcDecoder::getChannel Method	99
MvcDecoder::getDataBuffer Method	100
MvcDecoder::getDecoderErrors Method	100
MvcDecoder::getDecoderStatus Method	100
MvcDecoder::getDecoderType Method	101
MvcDecoder::getPrivate Method	101
MvcDecoder::setBackwardPlayback Method	101
MvcDecoder::setCplUid Method	101
MvcDecoder::setEndOfStream Method	101
MvcDecoder::setSecurityManager Method	101

MvcDecoder::setStartDelay Method	102
MvcDecoder::waitDecoderStatus Method	102
MvcDecoder::waitForTimeStamp Method	102
MvcDecoder::waitForTransferFinish Method	102
MvcDecoder::waitForUserData Method	102
MvcDecoder Operators	102
MvcDecoder::= Operator	103
MvcDecoder::bool Operator	103
MvcDevice Class	103
MvcDevice::~MvcDevice Destructor	105
MvcDevice::MvcDevice Constructor ()	105
MvcDevice Enumerations	105
mvc2::MvcDevice::CLOCKID Enumeration	106
mvc2::MvcDevice::OUTPUTMODE Enumeration	106
mvc2::MvcDevice::PRODUCTCODE Enumeration	106
MvcDevice Methods	106
MvcDevice::getAPIVersion Method	106
MvcDevice::getBootloaderVersion Method	107
MvcDevice::getCPUInfo Method	107
MvcDevice::getDeviceState Method	107
MvcDevice::getDriverVersion Method	108
MvcDevice::getErrorDescription Method	108
MvcDevice::getErrorString Method	108
MvcDevice::getFirmwareVersion Method	108
MvcDevice::getLastError Method	108
MvcDevice::getNetworkConfiguration Method	108
MvcDevice::getPCIBus Method	109
MvcDevice::getPCISlot Method	109
MvcDevice::getPowerDownStatus Method	109
MvcDevice::getProductCode Method	109
MvcDevice::getProductRevision Method	110
MvcDevice::getSecurityBootloaderVersion Method	110
MvcDevice::getSecurityManagerHwStatus Method	110
MvcDevice::getSecurityManagerVersion Method	110
MvcDevice::getSystemPosixTime Method	110
MvcDevice::getSystemStatus Method	111
MvcDevice::getUID Method	111
MvcDevice::getUptimeMs Method	111
MvcDevice::putMessage Method	111
MvcDevice::resetCard Method	112
MvcDevice::setAuthenticationPassword Method	112
MvcDevice::setOutputMode Method	112

MVcDevice::setPowerDown I ime Method	112
MvcDevice::setStatusLed Method	112
MvcDevice Operators	113
MvcDevice::= Operator	113
MvcDevice::bool Operator	113
MvcDeviceIterator Class	113
MvcDeviceIterator::~MvcDeviceIterator Destructor	114
MvcDeviceIterator::MvcDeviceIterator Constructor ()	114
MvcDeviceIterator Methods	114
MvcDeviceIterator::getFirst Method	114
MvcDeviceIterator::getIndex Method	115
MvcDeviceIterator::getNext Method	115
MvcNetDeviceIterator Class	115
MvcNetDeviceIterator::~MvcNetDeviceIterator Destructor	116
MvcNetDeviceIterator::MvcNetDeviceIterator Constructor (MvcNetDeviceIterator &)	116
MvcNetDeviceIterator::MvcNetDeviceIterator Constructor (char *)	116
MvcNetDeviceIterator Methods	116
MvcNetDeviceIterator::getFirst Method	117
MvcNetDeviceIterator::getIndex Method	117
MvcNetDeviceIterator::getNext Method	117
NetworkInterfaceInfo Class	117
NetworkInterfaceInfo::NetworkInterfaceInfo Constructor ()	118
NetworkInterfaceInfo::NetworkInterfaceInfo Constructor (NetworkInterfaceInfo &)	118
NetworkInterfaceInfo::NetworkInterfaceInfo Constructor (uint32_t, uint8_t *)	118
NetworkInterfaceInfo Methods	118
NetworkInterfaceInfo::getIPAddress Method ()	118
NetworkInterfaceInfo::getIPAddress Method (uint8_t *)	119
NetworkInterfaceInfo::getMACAddress Method	119
NetworkInterfaceInfo Operators	119
NetworkInterfaceInfo::= Operator	119
NetworkInterfaceInfo::bool Operator	119
OverlayDataBuffer Class	120
OverlayDataBuffer::OverlayDataBuffer Constructor ()	122
OverlayDataBuffer::OverlayDataBuffer Constructor (MvcDecoderPrivate *, MemoryBuffer *)	122
OverlayDataBuffer Enumerations	122
mvc2::OverlayDataBuffer::EFFECTSPEED Enumeration	123
mvc2::OverlayDataBuffer::MOVEEFFECT Enumeration	123
OverlayDataBuffer Methods	123
OverlayDataBuffer::addRenderCommand Method	123
OverlayDataBuffer::clearScreen Method	124
OverlayDataBuffer::setDisplayDuration Method	124
OverlayDataBuffer::setGlobalAlpha Method	124

OverlayDataBuffer::setGlobalPosition Method	124
OverlayDataBuffer::setIncompleteRender Method	124
OverlayDataBuffer::setSwapBuffer Method	124
OverlayDataBuffer Friends	124
friend class OverlayDecoder Friend	125
OverlayDecoder Class	125
OverlayDecoder::~OverlayDecoder Destructor	127
OverlayDecoder::OverlayDecoder Constructor ()	127
OverlayDecoder::OverlayDecoder Constructor (OverlayDecoder&)	128
OverlayDecoder::OverlayDecoder Constructor (TMmRc *, MvcDevice &)	128
OverlayDecoder Enumerations	128
mvc2::OverlayDecoder::RESOLUTION_FLAGS Enumeration	128
OverlayDecoder Methods	128
OverlayDecoder::connectOutput Method	128
OverlayDecoder::deleteOverlayElement Method	129
OverlayDecoder::disconnectOutput Method	129
OverlayDecoder::getDataBuffer Method (OverlayDataBuffer &, uint32_t)	129
OverlayDecoder::getDataBuffer Method (OverlayElementDataBuffer &, uint32_t, uint32_t)	130
OverlayDecoder::setOutputResolution Method	130
OverlayDecoder Operators	130
OverlayDecoder::= Operator	130
OverlayElementDataBuffer Class	130
OverlayElementDataBuffer::OverlayElementDataBuffer Constructor ()	132
OverlayElementDataBuffer::OverlayElementDataBuffer Constructor (MvcDecoderPrivate *, MemoryBuffe *)	er 132
OverlayElementDataBuffer Methods	133
OverlayElementDataBuffer::getElementName Method	133
OverlayElementDataBuffer::setElementName Method	133
OverlayElementDataBuffer::setPartialTransferInfo Method	133
OverlayElementDataBuffer Friends	133
friend class OverlayDecoder Friend	133
friend class SubtitleDecoder Friend	133
OverlayRenderCommand Class	133
OverlayRenderCommand::~OverlayRenderCommand Destructor	135
OverlayRenderCommand::OverlayRenderCommand Constructor	135
OverlayRenderCommand Data Members	135
OverlayRenderCommand::m_clearColor Data Member	135
OverlayRenderCommand::m_flags Data Member	135
OverlayRenderCommand::m_reserved1 Data Member	135
OverlayRenderCommand::m_reserved2 Data Member	136
OverlayRenderCommand Methods	136
OverlayRenderCommand::clearRenderArea Method	136

OverlayRenderCommand::renderIntoBuffer Method	136
OverlayRenderCommand Friends	136
friend class OverlayDataBuffer Friend	136
OverlaySubtitleDataBuffer Class	136
OverlaySubtitleDataBuffer::OverlaySubtitleDataBuffer Constructor ()	138
OverlaySubtitleDataBuffer::OverlaySubtitleDataBuffer Constructor (MvcDecoderPrivate *, MemoryB	uffer *) 139
OverlaySubtitleDataBuffer Methods	139
OverlaySubtitleDataBuffer::getResourceId Method	139
OverlaySubtitleDataBuffer::setTimeValues Method	139
OverlaySubtitleDataBuffer Friends	139
friend class SubtitleDecoder Friend	139
PCMDataBuffer Class	139
PCMDataBuffer::~PCMDataBuffer Destructor	142
PCMDataBuffer::PCMDataBuffer Constructor ()	142
PCMDataBuffer::PCMDataBuffer Constructor (MvcDecoderPrivate *, MemoryBuffer *)	142
PCMDataBuffer Methods	142
PCMDataBuffer::setBitsPerSample Method	142
PCMDataBuffer::setChannelMapping Method	142
PCMDataBuffer::setMixingChannel Method	142
PCMDataBuffer::setNumberOfChannels Method	143
PCMDataBuffer::setSampleFrequency Method	143
PCMDataBuffer Friends	143
friend class PCMDecoder Friend	143
PCMDecoder Class	143
PCMDecoder::~PCMDecoder Destructor	146
PCMDecoder::PCMDecoder Constructor ()	146
PCMDecoder::PCMDecoder Constructor (PCMDecoder&)	147
PCMDecoder::PCMDecoder Constructor (TMmRc *, MvcDevice &, uint32_t, uint32_t, uint32_t)	147
PCMDecoder Data Members	147
PCMDecoder::CHANNEL_MAPPING_FLAGS Data Member	147
PCMDecoder Methods	148
PCMDecoder::connectOutput Method	148
PCMDecoder::disconnectOutput Method	148
PCMDecoder::getBitsPerSample Method	148
PCMDecoder::getDecoderErrors Method	149
PCMDecoder::getDecoderStatus Method	149
PCMDecoder::getNumberOfChannels Method	149
PCMDecoder::getPCMDataBuffer Method	149
PCMDecoder::getSampleFrequency Method	149
PCMDecoder::setBitsPerSample Method	150
PCMDecoder::setChannelMapping Method	150
PCMDecoder::setDolbyPrologicIIChannels Method	150

PCMDecoder::setMixingChannel Method	150
PCMDecoder::setNumberOfChannels Method	150
PCMDecoder::setSampleFrequency Method	150
PCMDecoder::waitDecoderStatus Method	151
PCMDecoder Operators	151
PCMDecoder::= Operator	151
PlaybackControl Class	151
PlaybackControl::~PlaybackControl Destructor	152
PlaybackControl::PlaybackControl Constructor ()	152
PlaybackControl::PlaybackControl Constructor (PlaybackControl&)	153
PlaybackControl::PlaybackControl Constructor (TMmRc *, MvcDevice &)	153
PlaybackControl Enumerations	153
mvc2::PlaybackControl::PLAYBACK_STATE Enumeration	153
PlaybackControl Methods	154
PlaybackControl::connect Method	154
PlaybackControl::disconnect Method	154
PlaybackControl::flush Method	155
PlaybackControl::getState Method	155
PlaybackControl::pause Method	155
PlaybackControl::run Method	155
PlaybackControl::runSpeed Method	156
PlaybackControl::setSyncSlave Method	156
PlaybackControl::singleStep Method	156
PlaybackControl::stop Method	157
PlaybackControl::waitForEndOfStream Method	157
PlaybackControl Operators	157
PlaybackControl::= Operator	158
PlaybackControl::bool Operator	158
RenderFill Class	158
RenderFill::~RenderFill Destructor	159
RenderFill::RenderFill Constructor	160
RenderFill Methods	160
RenderFill::renderIntoBuffer Method	160
RenderFill::setBox Method	160
RenderFill::setColor Method	160
ProjectorAccess Class	160
ProjectorAccess::~ProjectorAccess Destructor	161
ProjectorAccess::ProjectorAccess Constructor ()	161
ProjectorAccess::ProjectorAccess Constructor (ProjectorAccess&)	162
ProjectorAccess::ProjectorAccess Constructor (TMmRc *, MvcDevice &, char *, uint16_t)	162
ProjectorAccess Methods	162
ProjectorAccess::commandlo Method	162

ProjectorAccess::login Method	163
ProjectorAccess::logout Method	163
ProjectorAccess Operators	163
ProjectorAccess::= Operator	163
ProjectorAccess::bool Operator	163
RenderPicture Class	164
RenderPicture::~RenderPicture Destructor	165
RenderPicture::RenderPicture Constructor	165
RenderPicture Methods	165
RenderPicture::renderIntoBuffer Method	165
RenderPicture::setPosition Method	166
RenderText Class	166
RenderText::~RenderText Destructor	168
RenderText::RenderText Constructor	168
RenderText Enumerations	168
mvc2::RenderText::TextFlags Enumeration	168
RenderText Methods	168
RenderText::renderIntoBuffer Method	168
RenderText::setColor Method	169
RenderText::setFont Method	169
RenderText::setPosition Method	169
SecurityAccess Class	169
SecurityAccess::~SecurityAccess Destructor	173
SecurityAccess::SecurityAccess Constructor ()	173
SecurityAccess::SecurityAccess Constructor (SecurityAccess&)	173
SecurityAccess::SecurityAccess Constructor (TMmRc *, MvcDevice &)	173
SecurityAccess Structures	174
mvc2::SecurityAccess::_CineLink2Para Structure	174
SecurityAccess Enumerations	174
mvc2::SecurityAccess::_HashAlgorithm Enumeration	174
mvc2::SecurityAccess::_MediaType Enumeration	174
mvc2::SecurityAccess::_SoftwareId Enumeration	175
mvc2::SecurityAccess::_SymmetricCryptoOperation Enumeration	175
SecurityAccess Methods	175
SecurityAccess::deleteCertificateChain Method	176
SecurityAccess::disableForensicMarking Method (AudioOutput &)	176
SecurityAccess::disableForensicMarking Method (VideoOutput &)	176
SecurityAccess::disableLLE Method	177
SecurityAccess::enableCinelink Method	177
SecurityAccess::enableCinelink2 Method	178
SecurityAccess::enableForensicMarking Method (AudioOutput &, uint16_t, uint32_t)	178
SecurityAccess::enableForensicMarking Method (VideoOutput &, uint16_t, uint32_t)	179

SecurityAccess::getAllHardwareInfo Method	179
SecurityAccess::getAppletStatusCode Method	180
SecurityAccess::getAppletVersion Method	180
SecurityAccess::getBuildTimeStrings Method	180
SecurityAccess::getCertificateSigningRequest Method	180
SecurityAccess::getErrorCode Method	180
SecurityAccess::getFipsFirmwareVersion Method	181
SecurityAccess::getFirmwareVersion Method	181
SecurityAccess::getForensicMarkingId Method	181
SecurityAccess::getHashValue Method	181
SecurityAccess::getMikromPublicKey Method	182
SecurityAccess::getPublicKey Method	182
SecurityAccess::getRtcHardwareStatus Method	182
SecurityAccess::getSelftestStatus Method	182
SecurityAccess::getTamperStatus Method	183
SecurityAccess::hashRtcSram Method	183
SecurityAccess::initHardwareInfo Method	183
SecurityAccess::initLogFlash Method	183
SecurityAccess::installCertificate Method	183
SecurityAccess::processSmartcardApdu Method	183
SecurityAccess::readLogFlashAttributes Method	184
SecurityAccess::readRtcReg Method	184
SecurityAccess::readRtcSram Method	184
SecurityAccess::readSecurityInterface Method	184
SecurityAccess::resetPowerfailBit Method	185
SecurityAccess::rsaPrivateKeyDecCmp Method	185
SecurityAccess::selfTest Method	185
SecurityAccess::setCineLink2Keys Method	185
SecurityAccess::setForensicMarkingId Method	185
SecurityAccess::setForensicMarkingIndicator Method	185
SecurityAccess::setHardwareInfo Method	186
SecurityAccess::setHashData Method	186
SecurityAccess::setMediaDecryptorKey Method	186
SecurityAccess::symmetricCryptoFinish Method	187
SecurityAccess::symmetricCryptoInit Method	187
SecurityAccess::symmetricCryptoProcess Method	187
SecurityAccess::validateHardwareInfo Method	188
SecurityAccess::verify Method	188
SecurityAccess::writeRtcReg Method	188
SecurityAccess::writeSecurityInterface Method	188
SecurityAccess Operators	188
SecurityAccess::= Operator	188

SecurityAccess::bool Operator	189
SecurityManager Class	189
SecurityManager Data Members	192
SecurityManager::c_ProjectorCertificate Data Member	192
SecurityManager Enumerations	192
mvc2::SecurityManager::CPL_ASSET_EXCEPTION Enumeration	192
mvc2::SecurityManager::SM_OPERATION Enumeration	192
mvc2::SecurityManager::SUITE_STATUS Enumeration	193
SecurityManager Friends	193
friend class MvcDecoder Friend	193
SecurityManager Methods	193
SecurityManager::addCertificate Method	193
SecurityManager::adjustTime Method	194
SecurityManager::checkShow Method	194
SecurityManager::connect Method	194
SecurityManager::getCertificate Method	194
SecurityManager::getCplList Method	194
SecurityManager::getKdmList Method	195
SecurityManager::getKeyMap Method	195
SecurityManager::getLogReport Method	195
SecurityManager::initiateMarriage Method	195
SecurityManager::loadCertificateChainFile Method	195
SecurityManager::loadPrivateKeyFile Method	195
SecurityManager::playShow Method	195
SecurityManager::purgeCpl Method	196
SecurityManager::queryStatus Method	196
SecurityManager::serviceDoorTamperTermination Method	196
SecurityManager::startSuite Method (SpblpAddress *, uint32_t)	196
SecurityManager::startSuite Method (char *)	197
SecurityManager::stopShow Method	197
SecurityManager::stopSuite Method	197
SecurityManager::uploadCplBuffer Method	197
SecurityManager::uploadCplFile Method	198
SecurityManager::uploadKdmBuffer Method	198
SecurityManager::uploadKdmFile Method	198
SecurityManager::usePrivateKey Method	198
SecurityManager::zeroize Method	198
SecurityManager Nested Types	199
SecurityManager::AesKeyMap Nested Type	199
SecurityManager::KeyMap Nested Type	199
SecurityManager::SpbIpAddress Nested Type	199
SecurityManager Operators	199

SecurityManager::= Operator	198
SecurityManager::bool Operator	200
SecurityManager::~SecurityManager Destructor	200
SecurityManager::SecurityManager Constructor ()	200
SecurityManager::SecurityManager Constructor (SecurityManager&)	200
SecurityManager::SecurityManager Constructor (TMmRc *, MvcDevice &)	200
SubtitleDecoder Class	200
SubtitleDecoder::~SubtitleDecoder Destructor	203
SubtitleDecoder::SubtitleDecoder Constructor ()	203
SubtitleDecoder::SubtitleDecoder Constructor (SubtitleDecoder&)	204
SubtitleDecoder::SubtitleDecoder Constructor (TMmRc *, MvcDevice &)	204
SubtitleDecoder Enumerations	204
mvc2::SubtitleDecoder::RENDERFLAGS Enumeration	204
SubtitleDecoder Methods	204
SubtitleDecoder::connectOutput Method	204
SubtitleDecoder::disableSubtitles Method	205
SubtitleDecoder::disconnectOutput Method	205
SubtitleDecoder::enableSubtitles Method	205
SubtitleDecoder::getDataBuffer Method (OverlayElementDataBuffer &, uint32_t, uint	t32_t) 206
SubtitleDecoder::getDataBuffer Method (OverlaySubtitleDataBuffer &, uint32_t, uint3	32_t) 206
SubtitleDecoder::getOverlayElementRequest Method	206
SubtitleDecoder::sendOverlayElement Method	206
SubtitleDecoder::sendSubtitleFile Method	206
SubtitleDecoder::setOutputResolution Method	206
SubtitleDecoder::setZOffset Method	207
SubtitleDecoder Operators	207
SubtitleDecoder::= Operator	207
SystemStatus Class	207
SystemStatus::~SystemStatus Destructor	208
SystemStatus::SystemStatus Constructor ()	209
SystemStatus Enumerations	209
mvc2::SystemStatus::CPULOAD Enumeration	209
SystemStatus Methods	209
SystemStatus::getBoardTemperature Method	209
SystemStatus::getCPULoad Method	209
SystemStatus::getDecoderTemperature Method	210
SystemStatus::getFanSpeed Method	210
SystemStatus::getFreeMemory Method	210
SystemStatus::getMaxMemory Method	210
SystemStatus::getOutputTemperature Method	211
SystemStatus::getUsedMemory Method	211
SystemStatus Operators	211

SystemStatus::= Operator	211
SystemStatus::bool Operator	211
TimeCode Class	211
TimeCode::TimeCode Constructor ()	212
TimeCode::TimeCode Constructor (TimeCode &)	213
TimeCode::TimeCode Constructor (uint32_t)	213
TimeCode::TimeCode Constructor (uint8_t, uint8_t, uint8_t, uint8_t, bool)	213
TimeCode Methods	214
TimeCode::getFrames Method	214
TimeCode::getHours Method	214
TimeCode::getMinutes Method	214
TimeCode::getSeconds Method	214
TimeCode::getTimeCodePacked Method	215
TimeCode::isDropFrame Method	215
TimeCode Operators	215
TimeCode::= Operator	215
UuidValue Class	216
UuidValue::UuidValue Constructor (UuidValue &)	217
UuidValue::UuidValue Constructor (char *)	217
UuidValue::UuidValue Constructor (uint8_t)	217
UuidValue::UuidValue Constructor (uint8_t, uint8_t, uint8	217
UuidValue::UuidValue Constructor (uint8_t *)	218
UuidValue Data Members	218
UuidValue::m_key Data Member	218
UuidValue Methods	218
UuidValue::toArray Method	218
UuidValue::toString Method	218
UuidValue Operators	218
UuidValue::!= Operator	218
UuidValue::[] Operator (uint8_t)	219
UuidValue::[] Operator (uint8_t)	219
UuidValue::= Operator	219
UuidValue::== Operator	219
UuidValue::bool Operator	219
TamperStatus Class	219
TamperStatus::TamperStatus Constructor	220
TamperStatus Enumerations	220
mvc2::TamperStatus::TAMPER_EVENTS Enumeration	220
mvc2::TamperStatus::TAMPER_SWITCHES Enumeration	221
TamperStatus Methods	221
TamperStatus::getSwitchStatus Method	221

TamperStatus::getTamperEvent Method	221
TamperStatus Friends	221
friend class SecurityAccess Friend	222
VersionValue Class	222
VersionValue::VersionValue Constructor (VersionValue &)	223
VersionValue::VersionValue Constructor (uint32_t)	223
VersionValue::VersionValue Constructor (uint32_t, uint32_t, uint32_t, uint32_t)	223
VersionValue Methods	223
VersionValue::getBuildRevision Method	224
VersionValue::getBuildVersion Method	224
VersionValue::getRevision Method	224
VersionValue::getVersion Method	224
VersionValue Operators	224
VersionValue::= Operator	225
VersionValue::bool Operator	225
VideoDataBuffer Class	225
VideoDataBuffer::~VideoDataBuffer Destructor	227
VideoDataBuffer::VideoDataBuffer Constructor ()	227
VideoDataBuffer::VideoDataBuffer Constructor (MvcDecoderPrivate *, MemoryBuffer *)	227
VideoDataBuffer Methods	227
VideoDataBuffer::setTimeCode Method	227
VideoDataBuffer Friends	228
friend class VideoDecoder Friend	228
VideoDecoder Class	228
VideoDecoder::VideoDecoder Constructor ()	230
VideoDecoder::VideoDecoder Constructor (MvcDecoderPrivate *)	230
VideoDecoder Methods	230
VideoDecoder::calcDolby3DMatrix Method	231
VideoDecoder::connectOutput Method	231
VideoDecoder::disconnectOutput Method	231
VideoDecoder::getDecoderErrors Method	232
VideoDecoder::getDecoderStatus Method	232
VideoDecoder::getVideoDataBuffer Method	232
VideoDecoder::setCloneVideo Method	232
VideoDecoder::setDolby3DChroma Method	232
VideoDecoder::setDolby3DMatrix Method	232
VideoDecoder::waitDecoderStatus Method	233
VideoDecoderErrors Class	233
VideoDecoderErrors::~VideoDecoderErrors Destructor	235
VideoDecoderErrors::VideoDecoderErrors Constructor ()	235
VideoDecoderErrors::VideoDecoderErrors Constructor (MvcDecoderPrivate *)	235
VideoDecoderErrors::VideoDecoderErrors Constructor (MvcDevice &, uint32_t)	235

VideoDecoderErrors::VideoDecoderErrors Constructor (VideoDecoderErrors &)	235
VideoDecoderErrors Methods	235
VideoDecoderErrors::getInputUnderruns Method	236
VideoDecoderErrors::getPartialFrames Method	236
VideoDecoderErrors::getPlayedFramesError Method	236
VideoDecoderErrors::getPlayedFramesOk Method	236
VideoDecoderErrors::getRepeatedFramesAVSync Method	236
VideoDecoderErrors::getRepeatedFramesUnderrun Method	236
VideoDecoderErrors::getSkippedFramesAVSync Method	236
VideoDecoderErrors::getSkippedFramesError Method	237
VideoDecoderErrors Operators	237
VideoDecoderErrors::= Operator	237
VideoDecoderStatus Class	237
VideoDecoderStatus::VideoDecoderStatus Constructor ()	239
VideoDecoderStatus::VideoDecoderStatus Constructor (MvcDecoderPrivate *)	239
VideoDecoderStatus::VideoDecoderStatus Constructor (MvcDecoderPrivate *, VideoDecoderStatus &	) 239
VideoDecoderStatus::VideoDecoderStatus Constructor (MvcDevice &, uint32_t)	239
VideoDecoderStatus::VideoDecoderStatus Constructor (VideoDecoderStatus &)	239
VideoDecoderStatus Methods	240
VideoDecoderStatus::getPictureHeight Method	240
VideoDecoderStatus::getPictureWidth Method	240
VideoDecoderStatus::getTimeCode Method	240
VideoDecoderStatus Operators	240
VideoDecoderStatus::= Operator	240
VideoMode Class	240
VideoMode::VideoMode Constructor	241
VideoMode Enumerations	241
mvc2::VideoMode::Mode Enumeration	241
VideoMode Methods	242
VideoMode::getMode Method	242
VideoMode Operators	243
VideoMode::= Operator	243
VideoOutput Class	243
VideoOutput::~VideoOutput Destructor	244
VideoOutput::VideoOutput Constructor ()	244
VideoOutput::VideoOutput Constructor (TMmRc *, MvcDevice &, uint32_t)	245
VideoOutput::VideoOutput Constructor (TMmRc *, VideoOutput &)	245
VideoOutput::VideoOutput Constructor (VideoOutput &)	245
VideoOutput Enumerations	246
mvc2::VideoOutput::EFFECTSPEED Enumeration	246
mvc2::VideoOutput::MOVEEFFECT Enumeration	246
mvc2::VideoOutput::OUTPUTACTIVITY Enumeration	247

Index	а
VSCConnector::waitForGPIOInterrupt Method	254
VSCConnector::setGPIOValue Method	254
VSCConnector::setGPIOInterrupt Method	254
VSCConnector::setGPIODirection Method	254
VSCConnector::getGPIOValue Method	254
VSCConnector::abortGPIOInterrupt Method	254
VSCConnector Methods	254
VSCConnector::VSCConnector Constructor (VSCConnector &)	253
VSCConnector::VSCConnector Constructor (TMmRc *, MvcDevice &)	253
VSCConnector::VSCConnector Constructor (ExternalConnectorPrivate *)	253
VSCConnector::VSCConnector Constructor ()	253
VSCConnector::~VSCConnector Destructor	253
VSCConnector Class	251
VideoOutput::bool Operator	251
VideoOutput::= Operator	250
VideoOutput Operators	250
VideoOutput::setVideoMode Method	250
VideoOutput::setPosition Method	249
VideoOutput::setOutputActivity Method	249
VideoOutput::setAudioEmbedding Method	248
VideoOutput::getActiveVideoMode Method	248
VideoOutput::enableDeGhosting Method	248
VideoOutput::disableDeGhosting Method	248
VideoOutput Methods	248
mvc2::VideoOutput::VIDEOPROPERTY Enumeration	247

## 1 Design Guide

## 1.1 Overview

The MVC API is the interface to the MikroM MVC200-DC or MVC201 mediablock. It is designed as C++ class structure to be highly variable and easy extendable.

#### **Features**

- platform independent C++ interface (currently Linux and Windows, more on request)
- objects have reference counting, used for automatic garbage collection
- · freely configurable decoder, output and control connections
- · Jpeg2000 and MPEG-2 decoding
- PCM output up to 16 channels, 24 bit, 96 kHz
- any data format can be decrypted with an AES-128 decryption block on the media block
- · optional MIC check for any data format
- complete control over playback: play, stop, pause, multi-speed playback (from slow motion up to fast forward)
- · backwards playback (needs special player support)
- · extensive status information about the playback
- · full screen 32 bit overlay and subtitle renderer
- 3D playback including seamless 2D/3D switching
- Dolby 3D processing and RealD deghosting
- · wide range of output color spaces independent from input format

## 1.2 Getting started

This section will give an overview how to use the MVC API and some basic principals.

- fundamentals how to handle MVC API objects
- · find the existing cards in the system

## 1.2.1 Basic Types

All MVC20x API objects consists of public objects with a reference to a private objects which holds the actual information. This design has main benefit in a garbage collector. Private objects will be deleted automatically if no more references exists, no need for new and delete operators.

A number of operators makes the work with these objects very easy:

The assignment operator = copies the pointer to the private object and increases the reference count. The bool operator can be used to check if a object has a working private object reference. Copy constructors are usefull to create new objects from existing ones.

Example of creating a Jpeg2kDecoder object:

```
Jpeq2kDecoder j2kdec;
                                 // an empty Jpeg2kDecoder object will be created
if (j2kdec)
                                 // bool operator checks if the object is empty or not
{
    printf("object is empty\n");
}
// create a Jpeg2kDecoder
TMmRc status;
j2kdec = Jpeg2kDecoder(&status, g_mvcdevice); // assign a Jpeg2kDecoder to the empty
object
if (status != MMRC_Ok) // check return code of the object creation
   printf("failed to create decoder: %d\n",status);
if (j2kdec) // check if the object really exits, this is redundant due to the status check
before
   BufferStatus bufstat = j2kdec.getBufferStatus(); // get a BufferStatus object from the
decoder
    if (bufstat)
        printf("Decoder Buffer Size: %d bytes\n",bufstat.getDecoderBufferMax());
    else
       printf("failed to get buffer status\n");
} // BufferStatus object bufstat will automatically be deleted by leaving this region
m_j2kdec = j2kdec; // save the decoder object in a member variable for later use
```

## 1.2.2 Find existing MVC cards

The base of all the API functionality is the MvcDevice class. So first a MvcDevice object has to be created, which belongs to one card in the system. There could be any number of cards available and to open them the MvcDeviceIterator class should be used.

The first example shows how to directly open a card with a specific index. This method is the most common way to create a MvcDevice object, especially with index 0.

```
MvcDevice g_mvcdevice;
if (!(g_mvcdevice = MvcDeviceIterator().getIndex(card_index)))
{
    printf("MVC card %d not found\n",card_index);
    exit(0);
}
```

The second example shows how to parse through all available cards in the system.

```
MvcDeviceIterator mvcitor;
MvcDevice mvcdev;
uint32_t index = 0;
```

```
while(mvcdev = mvcitor.getNext())
{
    printf("MVC card found at %d:\n",index);
    if (ret = mvcdev.getDeviceState())
    {
        printf(" card condition bad: %d\n",ret);
    }
    else
    {
        printf(" UID: %d\n",mvcdev.getUID());
    }
    index++;
}
```

## 1.2.3 Authentication

The MVC media block needs to check the authentication of the application which uses the API. There for any program needs to set the authentication password to make sure a specific type of communication is allowed with the media block.

## 2 Symbol Reference

## 2.1 mvc2 Namespace

Namespace of all MVC20x API classes.

#### Classes

	Name	Description
<b>4</b> \$	ActiveDecoderInfo	This is class mvc2::ActiveDecoderInfo.
<b>4</b> \$	ActiveDecoderIterator	This is class mvc2::ActiveDecoderIterator.
<b>4</b> \$	AudioDecoderErrors	This is class mvc2::AudioDecoderErrors.
<b>4</b> \$	AudioDecoderStatus	This is class mvc2::AudioDecoderStatus.
<b>4</b> \$	AudioOutput	The AudioOutput class represents physical or virtual audio outputs.
<b>4</b> \$	AudioOutputStatus	This is class mvc2::AudioOutputStatus.
<b>4</b> 3	BufferStatus	This object will hold information about the current status of the input buffer and calculated data rates.
<b>4</b> \$	ColorConversionMatrix	This is class mvc2::ColorConversionMatrix.
<b>4</b> \$	ColorSpace	Colorspace definition class.
<b>4</b> \$	ConfigAccess	This class is used to access configuration for the MVC20x firmware and driver.
<b>4</b> \$	CPUInfo	This is class mvc2::CPUInfo.
<b>4</b> \$	DataBuffer	Transfer data buffer class.
<b>4</b> \$	DecoderErrors	Base class for decoder error counters.
<b>4</b> \$	DecoderStatus	These object hold information about the current status of a decoder.
<b>4</b> \$	Dolby3Dchromaticity	This is class mvc2::Dolby3Dchromaticity.
<b>4</b> \$	ExternalConnector	This is class mvc2::ExternalConnector.
43	FeatureAccess	This is class mvc2::FeatureAccess.
<b>4</b> \$	FramePosition	Positioning class for an area.
<b>4</b> \$	Jpeg2kDecoder	JPEG 2000 decoder class.
<b>4</b> \$	LogAccess	This class is used to access the debug log of the MVC card.
<b>4</b> \$	LogMessage	Log Message class
<del>%</del> \$	Mpeg2Decoder	MPEG-2 decoder class.
<b>4</b> \$	MvcDecoder	This is the base class of all decoders of a MVC card. It handles the basic functions like data transfers and end-of-stream.
<b>₽</b> \$	MvcDevice	
<b>4</b> \$	MvcDeviceIterator	MVC 200 device iterator class.
<b>4</b> \$	MvcNetDeviceIterator	This is class mvc2::MvcNetDeviceIterator.
<b>4</b> \$	NetworkInterfaceInfo	Network interface information class
<b>4</b> \$	OverlayDataBuffer	This is class mvc2::OverlayDataBuffer.
<b>4</b> 3	OverlayDecoder	This is class mvc2::OverlayDecoder.
<b>4</b> 3	OverlayElementDataBuffer	This is class mvc2::OverlayElementDataBuffer.
<del>^</del> \$	OverlayRenderCommand	This is class mvc2::OverlayRenderCommand.

<del>^</del> \$	OverlaySubtitleDataBuffer	This is class mvc2::OverlaySubtitleDataBuffer.
<b>4</b> \$	PCMDataBuffer	This is class mvc2::PCMDataBuffer.
<b>?</b> \$	PCMDecoder	The PCM decoder class does the pre-processing of PCM samples before transferring it to an Audio-Output.
4\$	PlaybackControl	This class controls the playback and the decoders.
<b>4</b> \$	RenderFill	This is class mvc2::RenderFill.
<b>4</b> \$	ProjectorAccess	Interface to the projector connected to the MVC201.
<b>4</b> \$	RenderPicture	This is class mvc2::RenderPicture.
<b>4</b> \$	RenderText	This is class mvc2::RenderText.
<b>₹</b> \$	SecurityAccess	Class to access security features of the card, if the security manager is not active.
<b>4</b> \$	SecurityManager	Security manager API implementation.
<b>4</b> \$	SubtitleDecoder	Subtitle overlay renderer.
<b>*</b> \$	SystemStatus	This class holds the collected information about the current system status. It is the result of the MvcDevice::getSystemStatus() method.
<del>^</del> \$	TimeCode	TimeCode value type.
<b>4</b> \$	UuidValue	Helper class for UUIDs.
<b>%</b> \$	TamperStatus	This is class mvc2::TamperStatus.
<b>?</b> \$	VersionValue	Class to hold version numbers
<del>^</del> \$	VideoDataBuffer	This is class mvc2::VideoDataBuffer.
4\$	VideoDecoder	This is class mvc2::VideoDecoder.
<b>4</b> \$	VideoDecoderErrors	These objects collect a couple of error counters of a specific decoder.
Pt\$	VideoDecoderStatus	This is class mvc2::VideoDecoderStatus.
Pt\$	VideoMode	Video mode definition class.
<b>4</b> \$	VideoOutput	The VideoOutput class represents physical or virtual video outputs.
<u>4</u> 3	VSCConnector	This is class mvc2::VSCConnector.

## 2.1.1 ActiveDecoderInfo Class

#### **Inheritance Hierarchy**

C++

class ActiveDecoderInfo;

File

mvc2api\_decoder.h

Remarks

This is class mvc2::ActiveDecoderInfo.

#### Members

#### Methods

	Name	Description
<b>≡∳</b>	~ActiveDecoderInfo	This is ~ActiveDecoderInfo, a member of class ActiveDecoderInfo.
<b>≡</b>	ActiveDecoderInfo	This is ActiveDecoderInfo, a member of class ActiveDecoderInfo.
<b>≡</b>	ActiveDecoderInfo	This is ActiveDecoderInfo, a member of class ActiveDecoderInfo.

#### **ActiveDecoderInfo Methods**

	Name	Description
<b>≡⋄</b>	getChannel	This is getChannel, a member of class ActiveDecoderInfo.

#### **ActiveDecoderInfo Operators**

	Name	Description
<u>(/−</u> =+)	= 67	This is =, a member of class ActiveDecoderInfo.

#### **ActiveDecoderInfo Methods**

Name		Description
<b>=♦</b>	getChannel	This is getChannel, a member of class ActiveDecoderInfo.

#### **ActiveDecoderInfo Operators**

	Name	Description
=+)	=	This is =, a member of class ActiveDecoderInfo.

### 2.1.1.1 ActiveDecoderInfo::~ActiveDecoderInfo Destructor

#### C++

~ActiveDecoderInfo();

#### Remarks

This is ~ActiveDecoderInfo, a member of class ActiveDecoderInfo.

# 2.1.1.2 ActiveDecoderInfo::ActiveDecoderInfo Constructor (ActiveDecoderInfo &)

C++

ActiveDecoderInfo(const ActiveDecoderInfo & other);

#### Remarks

This is ActiveDecoderInfo, a member of class ActiveDecoderInfo.

# 2.1.1.3 ActiveDecoderInfo::ActiveDecoderInfo Constructor (ActiveDecoderInfoPrivate \*)

C++

ActiveDecoderInfo(ActiveDecoderInfoPrivate \* info = 0);

#### Remarks

This is ActiveDecoderInfo, a member of class ActiveDecoderInfo.

### 2.1.1.4 ActiveDecoderInfo Methods

## 2.1.1.4.1 ActiveDecoderInfo::getChannel Method

#### $C^{++}$

uint32\_t getChannel() const;

#### Remarks

This is getChannel, a member of class ActiveDecoderInfo.

## 2.1.1.5 ActiveDecoderInfo Operators

## 2.1.1.5.1 ActiveDecoderInfo::= Operator

#### C++

ActiveDecoderInfo& operator =(const ActiveDecoderInfo& other);

#### Remarks

This is =, a member of class ActiveDecoderInfo.

## 2.1.2 ActiveDecoderIterator Class

#### **Inheritance Hierarchy**

#### C++

class ActiveDecoderIterator;

#### File

mvc2api\_decoder.h

#### Remarks

This is class mvc2::ActiveDecoderIterator.

#### **Members**

#### Methods

	Name	Description
<b>≡∳</b>	~ActiveDecoderIterator	This is ~ActiveDecoderIterator, a member of class ActiveDecoderIterator.
<b>≡∳</b>	ActiveDecoderIterator	This is ActiveDecoderIterator, a member of class ActiveDecoderIterator.
<b>≡♦</b>	ActiveDecoderIterator	This is ActiveDecoderIterator, a member of class ActiveDecoderIterator.

#### **ActiveDecoderIterator Methods**

	Name	Description
<b>=♦</b>	getNext	This is getNext, a member of class ActiveDecoderIterator.

#### **ActiveDecoderIterator Operators**

	Name	Description
( <u>/-</u> =+)	=	This is =, a member of class ActiveDecoderIterator.

#### **ActiveDecoderIterator Methods**

	Name	Description
<b>≡♦</b>	getNext	This is getNext, a member of class ActiveDecoderIterator.

#### **ActiveDecoderIterator Operators**

	Name	Description
=+)	=	This is =, a member of class ActiveDecoderIterator.

### 2.1.2.1 ActiveDecoderIterator::~ActiveDecoderIterator Destructor

C++

```
~ActiveDecoderIterator();
```

#### Remarks

This is ~ActiveDecoderIterator, a member of class ActiveDecoderIterator.

# 2.1.2.2 ActiveDecoderIterator::ActiveDecoderIterator Constructor (ActiveDecoderIterator &)

C++

```
ActiveDecoderIterator(const ActiveDecoderIterator & other);
```

#### Remarks

This is ActiveDecoderIterator, a member of class ActiveDecoderIterator.

# 2.1.2.3 ActiveDecoderIterator::ActiveDecoderIterator Constructor (TMmRc \*, MvcDevice &)

C++

```
ActiveDecoderIterator(TMmRc * resultPointer, const MvcDevice & mvcdev);
```

#### Remarks

This is ActiveDecoderIterator, a member of class ActiveDecoderIterator.

#### 2.1.2.4 ActiveDecoderIterator Methods

## 2.1.2.4.1 ActiveDecoderIterator::getNext Method

C++

```
ActiveDecoderInfo getNext();
```

#### Remarks

This is getNext, a member of class ActiveDecoderIterator.

## 2.1.2.5 ActiveDecoderIterator Operators

## 2.1.2.5.1 ActiveDecoderIterator::= Operator

C++

```
ActiveDecoderIterator& operator =(const ActiveDecoderIterator& other);
```

#### Remarks

This is =, a member of class ActiveDecoderIterator.

## 2.1.3 AudioDecoderErrors Class

#### **Inheritance Hierarchy**

#### C++

class AudioDecoderErrors : public DecoderErrors;

#### File

mvc2api\_decoder.h

#### Remarks

This is class mvc2::AudioDecoderErrors.

#### Members

#### Methods

	Name	Description
<b>■</b>	~DecoderErrors	This is ~DecoderErrors, a member of class DecoderErrors.
<b>≡⋄</b>	DecoderErrors	This is DecoderErrors, a member of class DecoderErrors.
<b>≡⋄</b>	DecoderErrors	This is DecoderErrors, a member of class DecoderErrors.
<b>≡⋄</b>	DecoderErrors	This is DecoderErrors, a member of class DecoderErrors.
<b>=♦</b>	DecoderErrors	This is DecoderErrors, a member of class DecoderErrors.

#### **AudioDecoderErrors Class**

	Name	Description
<b>=♦</b> ₩	~AudioDecoderErrors	This is ~AudioDecoderErrors, a member of class AudioDecoderErrors.
<b>=♦</b>	AudioDecoderErrors	This is AudioDecoderErrors, a member of class AudioDecoderErrors.
<b>=♦</b>	AudioDecoderErrors	This is AudioDecoderErrors, a member of class AudioDecoderErrors.
=•	AudioDecoderErrors	This is AudioDecoderErrors, a member of class AudioDecoderErrors.
<b>=</b> ♦	AudioDecoderErrors	This is AudioDecoderErrors, a member of class AudioDecoderErrors.

#### **DecoderErrors Methods**

	Name	Description
<b>≡</b>	getReturnCode	This is getReturnCode, a member of class DecoderErrors.

#### **AudioDecoderErrors Class**

	Name	Description
<b>=</b> ♦	getClippedSamples	This is getClippedSamples, a member of class AudioDecoderErrors.
<b>=♦</b>	getInputUnderruns	This is getInputUnderruns, a member of class AudioDecoderErrors.
<b>=</b> ♦	getMaxClippingPeriode	This is getMaxClippingPeriode, a member of class AudioDecoderErrors.
<b>≟</b>	getNumberOfClippings	This is getNumberOfClippings, a member of class AudioDecoderErrors.

<b>=</b> ♦	getPlayedSamples	This is getPlayedSamples, a member of class AudioDecoderErrors.
<b>=♦</b>	getSilentSamples	This is getSilentSamples, a member of class AudioDecoderErrors.
<b>≅♦</b>	getSkippedSamples	This is getSkippedSamples, a member of class AudioDecoderErrors.

#### **DecoderErrors Operators**

	Name	Description
<del>(/-</del> =+)	-	This is =, a member of class DecoderErrors.
<u>(/-</u> =+)	bool	Returns if the object is valid.

#### **AudioDecoderErrors Class**

	Name	Description
=+)	=	This is =, a member of class AudioDecoderErrors.

#### **DecoderErrors Methods**

	Name	Description
<b>≡♦</b>	getReturnCode	This is getReturnCode, a member of class DecoderErrors.

#### **AudioDecoderErrors Class**

	Name	Description
<b>≟</b> ∳	getClippedSamples	This is getClippedSamples, a member of class AudioDecoderErrors.
<b>=</b>	getInputUnderruns	This is getInputUnderruns, a member of class AudioDecoderErrors.
<b>=∳</b>	getMaxClippingPeriode	This is getMaxClippingPeriode, a member of class AudioDecoderErrors.
<b>=</b>	getNumberOfClippings	This is getNumberOfClippings, a member of class AudioDecoderErrors.
<b>≡</b>	getPlayedSamples	This is getPlayedSamples, a member of class AudioDecoderErrors.
<b>≡</b>	getSilentSamples	This is getSilentSamples, a member of class AudioDecoderErrors.
<b>≟∳</b>	getSkippedSamples	This is getSkippedSamples, a member of class AudioDecoderErrors.

#### **DecoderErrors Operators**

	Name	Description
( <u>/-</u> =+)	=	This is =, a member of class DecoderErrors.
<u>(/-</u> =+)	bool	Returns if the object is valid.

#### AudioDecoderErrors Class

	Name	Description
<u>(/-</u> =+)	=	This is =, a member of class AudioDecoderErrors.

## 2.1.3.1 AudioDecoderErrors::~AudioDecoderErrors Destructor

#### C++

virtual ~AudioDecoderErrors();

#### Remarks

This is ~AudioDecoderErrors, a member of class AudioDecoderErrors.

## 2.1.3.2 AudioDecoderErrors::AudioDecoderErrors Constructor ()

C++

AudioDecoderErrors();

#### Remarks

This is AudioDecoderErrors, a member of class AudioDecoderErrors.

# 2.1.3.3 AudioDecoderErrors::AudioDecoderErrors Constructor (AudioDecoderErrors &)

C++

AudioDecoderErrors(const AudioDecoderErrors & other);

#### Remarks

This is AudioDecoderErrors, a member of class AudioDecoderErrors.

# 2.1.3.4 AudioDecoderErrors::AudioDecoderErrors Constructor (MvcDecoderPrivate \*)

C++

AudioDecoderErrors(MvcDecoderPrivate \* mvcdec);

#### Remarks

This is AudioDecoderErrors, a member of class AudioDecoderErrors.

# 2.1.3.5 AudioDecoderErrors::AudioDecoderErrors Constructor (MvcDevice &, uint32\_t)

C++

AudioDecoderErrors(const MvcDevice & mvcdev, uint32\_t channel);

#### Remarks

This is AudioDecoderErrors, a member of class AudioDecoderErrors.

#### 2.1.3.6 AudioDecoderErrors Methods

## 2.1.3.6.1 AudioDecoderErrors::getClippedSamples Method

C++

```
uint32_t getClippedSamples(uint32_t audioChannel) const;
```

#### Remarks

This is getClippedSamples, a member of class AudioDecoderErrors.

### 2.1.3.6.2 AudioDecoderErrors::getInputUnderruns Method

#### C++

```
uint32_t getInputUnderruns(uint32_t audioChannel) const;
```

#### Remarks

This is getInputUnderruns, a member of class AudioDecoderErrors.

### 2.1.3.6.3 AudioDecoderErrors::getMaxClippingPeriode Method

#### C++

```
uint32_t getMaxClippingPeriode(uint32_t audioChannel) const;
```

#### Remarks

This is getMaxClippingPeriode, a member of class AudioDecoderErrors.

#### 2.1.3.6.4 AudioDecoderErrors::getNumberOfClippings Method

#### C++

```
uint32_t getNumberOfClippings(uint32_t audioChannel) const;
```

#### Remarks

This is getNumberOfClippings, a member of class AudioDecoderErrors.

#### 2.1.3.6.5 AudioDecoderErrors::getPlayedSamples Method

#### C++

```
uint32_t getPlayedSamples(uint32_t audioChannel) const;
```

#### Remarks

This is getPlayedSamples, a member of class AudioDecoderErrors.

### 2.1.3.6.6 AudioDecoderErrors::getSilentSamples Method

#### C++

```
uint32_t getSilentSamples(uint32_t audioChannel) const;
```

#### Remarks

This is getSilentSamples, a member of class AudioDecoderErrors.

### 2.1.3.6.7 AudioDecoderErrors::getSkippedSamples Method

#### C++

```
uint32_t getSkippedSamples(uint32_t audioChannel) const;
```

#### Remarks

This is getSkippedSamples, a member of class AudioDecoderErrors.

## 2.1.3.7 AudioDecoderErrors Operators

# 2.1.3.7.1 AudioDecoderErrors::= Operator

#### C++

AudioDecoderErrors & operator =(const AudioDecoderErrors & other);

## Remarks

This is =, a member of class AudioDecoderErrors.

# 2.1.4 AudioDecoderStatus Class

# **Inheritance Hierarchy**

## C++

class AudioDecoderStatus : public DecoderStatus;

# File

mvc2api\_decoder.h

# Remarks

This is class mvc2::AudioDecoderStatus.

## **Members**

## Methods

	Name	Description
<b>=</b> ♦ ₩	~DecoderStatus	This is ~DecoderStatus, a member of class DecoderStatus.
<b>≡♦</b>	DecoderStatus	This is DecoderStatus, a member of class DecoderStatus.
<b>=</b> ♦	DecoderStatus	This is DecoderStatus, a member of class DecoderStatus.
<b>=</b> ♦	DecoderStatus	This is DecoderStatus, a member of class DecoderStatus.
<b>=</b> ♦	DecoderStatus	This is DecoderStatus, a member of class DecoderStatus.
<b>=</b> ♦	DecoderStatus	This is DecoderStatus, a member of class DecoderStatus.
<b>≡</b>	DecoderStatus	This is DecoderStatus, a member of class DecoderStatus.

# AudioDecoderStatus Class

	Name	Description
<b>≡♦</b>	AudioDecoderStatus	This is AudioDecoderStatus, a member of class AudioDecoderStatus.
<b>≡♦</b>	AudioDecoderStatus	This is AudioDecoderStatus, a member of class AudioDecoderStatus.
<b>≡♦</b>	AudioDecoderStatus	This is AudioDecoderStatus, a member of class AudioDecoderStatus.
<b>=</b> ♦	AudioDecoderStatus	This is AudioDecoderStatus, a member of class AudioDecoderStatus.
<b>≡♦</b>	AudioDecoderStatus	This is AudioDecoderStatus, a member of class AudioDecoderStatus.

# **DecoderStatus Methods**

	Name	Description
<b>=♦</b>	getReturnCode	This is getReturnCode, a member of class DecoderStatus.
<b>=</b> ♦	getTimeStamp	This is getTimeStamp, a member of class DecoderStatus.
<b>≡</b>	getUserData	This is getUserData, a member of class DecoderStatus.

# **AudioDecoderStatus Class**

	Name	Description
<b>∉</b> ∳	getAudioLevel	This is getAudioLevel, a member of class AudioDecoderStatus.
<b>≡♦</b>	isAudioChannelActive	This is isAudioChannelActive, a member of class AudioDecoderStatus.

# **DecoderStatus Operators**

	Name		Description
=+)	=	Y(O),	This is =, a member of class DecoderStatus.
<del>(/_</del> =+)	bool		Returns if the object is valid.

## **AudioDecoderStatus Class**

	Name	Description
<u>&gt;-</u> =+)	=	This is =, a member of class AudioDecoderStatus.

# **AudioDecoderStatus Enumerations**

## AudioDecoderStatus Class

	Name	Description
<b>a</b>	INTERVAL	This is record mvc2::AudioDecoderStatus::INTERVAL.

## **AudioDecoderStatus Enumerations**

## **AudioDecoderStatus Class**

	Name	Description
<b>a</b>	INTERVAL	This is record mvc2::AudioDecoderStatus::INTERVAL.

## **DecoderStatus Methods**

	Name	Description
<b>≡♦</b>	getReturnCode	This is getReturnCode, a member of class DecoderStatus.
<b>≡</b>	getTimeStamp	This is getTimeStamp, a member of class DecoderStatus.
<b>≡♦</b>	getUserData	This is getUserData, a member of class DecoderStatus.

# AudioDecoderStatus Class

	Name	Description
<b>≡∳</b>	getAudioLevel	This is getAudioLevel, a member of class AudioDecoderStatus.
<b>≡♦</b>	isAudioChannelActive	This is isAudioChannelActive, a member of class AudioDecoderStatus.

# **DecoderStatus Operators**

	Name	Description
( <u>/-</u> =+)	=	This is =, a member of class DecoderStatus.
(/ <del>_</del> =+)	bool	Returns if the object is valid.

## **AudioDecoderStatus Class**

	Name	Description
<u>(&gt;</u> =+)	=	This is =, a member of class AudioDecoderStatus.

# 2.1.4.1 AudioDecoderStatus::AudioDecoderStatus Constructor ()

# C++

AudioDecoderStatus();

## Remarks

This is AudioDecoderStatus, a member of class AudioDecoderStatus.

# 2.1.4.2 AudioDecoderStatus::AudioDecoderStatus Constructor (AudioDecoderStatus &)

C++

```
AudioDecoderStatus(const AudioDecoderStatus & other);
```

#### Remarks

This is AudioDecoderStatus, a member of class AudioDecoderStatus.

# 2.1.4.3 AudioDecoderStatus::AudioDecoderStatus Constructor (MvcDecoderPrivate \*)

C++

```
AudioDecoderStatus(MvcDecoderPrivate * mvcdec);
```

## Remarks

This is AudioDecoderStatus, a member of class AudioDecoderStatus.

# 2.1.4.4 AudioDecoderStatus::AudioDecoderStatus Constructor (MvcDecoderPrivate \*, AudioDecoderStatus &)

C++

```
AudioDecoderStatus(MvcDecoderPrivate * mvcdec, const AudioDecoderStatus & other);
```

### Remarks

This is AudioDecoderStatus, a member of class AudioDecoderStatus.

# 2.1.4.5 AudioDecoderStatus::AudioDecoderStatus Constructor (MvcDevice &, uint32\_t)

C++

```
AudioDecoderStatus(const MvcDevice & mvcdev, uint32_t channel);
```

### Remarks

This is AudioDecoderStatus, a member of class AudioDecoderStatus.

# 2.1.4.6 AudioDecoderStatus Enumerations

# 2.1.4.6.1 mvc2::AudioDecoderStatus::INTERVAL Enumeration

C++

```
enum INTERVAL {
   Interval_50ms = 0,
```

```
Interval_1s = 1
File
```

mvc2api\_decoder.h

#### Remarks

This is record mvc2::AudioDecoderStatus::INTERVAL

# 2.1.4.7 AudioDecoderStatus Methods

# 2.1.4.7.1 AudioDecoderStatus::getAudioLevel Method

C++

```
uint32_t getAudioLevel(uint32_t audioChannel, INTERVAL interval = Interval_50ms) const;
```

#### Remarks

This is getAudioLevel, a member of class AudioDecoderStatus.

# 2.1.4.7.2 AudioDecoderStatus::isAudioChannelActive Method

```
bool isAudioChannelActive(uint32_t audioChannel) const;
```

#### Remarks

This is isAudioChannelActive, a member of class AudioDecoderStatus.

# 2.1.4.8 AudioDecoderStatus Operators

# 2.1.4.8.1 AudioDecoderStatus::= Operator

C++

```
AudioDecoderStatus & operator = (const AudioDecoderStatus & instat);
```

# Remarks

This is =, a member of class AudioDecoderStatus.

# 2.1.5 AudioOutput Class

The AudioOutput class represents physical or virtual audio outputs.

# Inheritance Hierarchy

C++

```
class AudioOutput;
```

File

mvc2api\_output.h

# Description

Creating a AudioOutput will allocation physical resources of a specific kind on the MVC20x media block. These resources

can be used by a decoder of the output process. The decoder must be connected using connectOutput() method from a decoder.

## **Members**

# Methods

	Name	Description
<b>=</b> ♦ <b>W</b>	~AudioOutput	This is ~AudioOutput, a member of class AudioOutput.
<b>=</b> ♦	AudioOutput	This is AudioOutput, a member of class AudioOutput.
<b>=</b> ♦	AudioOutput	This is AudioOutput, a member of class AudioOutput.
<b>=♦</b>	AudioOutput	Create AudioOutput object.

# **AudioOutput Enumerations**

	Name	Description
<b>a</b>	AUDIOPROPERTY	Possible properties of a audio output.

# **AudioOutput Methods**

	Name	Description
<b>=♦</b>	getAudioOutputStatus	This is getAudioOutputStatus, a member of class AudioOutput.
<b>≡∳</b>	getChannelMute	Gets channel muting
<b>∉</b> ∳	getNumberOfChannels	This is getNumberOfChannels, a member of class AudioOutput.
<b>=♦</b>	getOuputFrequency	Gets the current output frequency
<b>≡∳</b>	getOutputDelay	Gets the current audio output delay
<b>≡</b>	setChannelMute	Selects audio channels to mute.
<b>≡</b>	setOutputDelay	Sets audio delay.
<b>≡</b>	setOutputFrequency	Sets the audio output frequency.
<b>=♦</b>	setUserData	Sets audio user data.
<b>≡</b>	setVolume	This is setVolume, a member of class AudioOutput.
<b>≡∳</b>	setVolumeDB	This is setVolumeDB, a member of class AudioOutput.

# **AudioOutput Operators**

	Name	Description
<u>(/-</u> =+)	=	This is =, a member of class AudioOutput.
——————————————————————————————————————	bool	Returns if the object is valid.

# **AudioOutput Enumerations**

	Name	Description
<b>a</b>	AUDIOPROPERTY	Possible properties of a audio output.

# **AudioOutput Methods**

	Name	Description
<b>≡</b>	getAudioOutputStatus	This is getAudioOutputStatus, a member of class AudioOutput.
<b>=∳</b>	getChannelMute	Gets channel muting
<b>≡</b>	getNumberOfChannels	This is getNumberOfChannels, a member of class AudioOutput.
<b>≡</b>	getOuputFrequency	Gets the current output frequency
<b>≡∳</b>	getOutputDelay	Gets the current audio output delay
<b>≡∳</b>	setChannelMute	Selects audio channels to mute.
<b>≡∳</b>	setOutputDelay	Sets audio delay.
<b>≡</b>	setOutputFrequency	Sets the audio output frequency.

<b>=♦</b>	setUserData	Sets audio user data.
<b>=♦</b>	setVolume	This is setVolume, a member of class AudioOutput.
<b>=♦</b>	setVolumeDB	This is setVolumeDB, a member of class AudioOutput.

## **AudioOutput Operators**

	Name	Description
( <u>/-</u> =+)	=	This is =, a member of class AudioOutput.
=+)	bool	Returns if the object is valid.

# 2.1.5.1 AudioOutput::~AudioOutput Destructor

#### C++

```
virtual ~AudioOutput();
```

#### Remarks

This is ~AudioOutput, a member of class AudioOutput.

# 2.1.5.2 AudioOutput::AudioOutput Constructor ()

#### C++

AudioOutput();

### Remarks

This is AudioOutput, a member of class AudioOutput.

# 2.1.5.3 AudioOutput::AudioOutput Constructor (AudioOutput &)

## C++

AudioOutput(const AudioOutput & other);

### Remarks

This is AudioOutput, a member of class AudioOutput.

# 2.1.5.4 AudioOutput::AudioOutput Constructor (TMmRc \*, MvcDevice &, uint32\_t, uint32\_t)

Create AudioOutput object.

### C++

AudioOutput(TMmRc \* resultPointer, const MvcDevice & mvcDevice, uint32\_t numberOfChannels,
uint32\_t audioProperty = 0);

## **Parameters**

Parameters	Description
TMmRc * resultPointer	Pointer where to store the result in. if 0 no result will be written.
const MvcDevice & mvcDevice	MvcDevice to create the output on
uint32_t audioProperty = 0	optional properties to influence the output selection

# Returns

MMRC\_Ok if the output could be created, else an error.

## **Return Values**

Return Values	Description
MMRC_Ok	Output successfully created

# Description

This method creates a object for a physical or virtual audio output. Before the object will be created the list of available outputs will be parsed to find an output which fits best to the AudioProperty.

# 2.1.5.5 AudioOutput Enumerations

# 2.1.5.5.1 mvc2::AudioOutput::AUDIOPROPERTY Enumeration

#### C++

```
enum AUDIOPROPERTY {
  AudioProperty_16bit_Samples = (1<<0),
  AudioProperty_20bit_Samples = (1<<1),
  AudioProperty_24bit_Samples = (1<<2),
  AudioProperty_Bitstream_Support = (1<<3),
  AudioProperty_Video_Embedding = (1<<4)
};</pre>
```

#### File

mvc2api\_output.h

#### **Members**

Members	Description
AudioProperty_16bit_Samples = (1<<0)	Output supports 16 bit samples.
AudioProperty_20bit_Samples = (1<<1)	Output supports 20 bit samples.
AudioProperty_24bit_Samples = (1<<2)	Output supports 24 bit samples.
AudioProperty_Bitstream_Support = (1<<3)	Output can be used to transfer a compressed audio bit stream.
AudioProperty_Video_Embedding = (1<<4)	Output will transfer the data over video using audio embedding.

# Remarks

Possible properties of a audio output.

# 2.1.5.6 AudioOutput Methods

# 2.1.5.6.1 AudioOutput::getAudioOutputStatus Method

### C++

```
TMmRc getAudioOutputStatus(AudioOutputStatus & status, AudioOutputStatus::TIME_RESOLUTION
res = AudioOutputStatus::Resolution_1s) const;
```

### Remarks

This is getAudioOutputStatus, a member of class AudioOutput.

# 2.1.5.6.2 AudioOutput::getChannelMute Method



Gets channel muting

## C++

uint32\_t getChannelMute() const;

#### Returns

Returns output muting mask.

# Description

This functions gets the current channel muting mask from the output. The mask has a bit for every channel of the output, starting from bit 0 for channel 0. A 1 in the mask means the channel is muted.

# 2.1.5.6.3 AudioOutput::getNumberOfChannels Method



#### C++

```
uint32_t getNumberOfChannels() const;
```

#### Remarks

This is getNumberOfChannels, a member of class AudioOutput.

# 2.1.5.6.4 AudioOutput::getOuputFrequency Method

Gets the current output frequency

#### C++

```
uint32_t getOuputFrequency() const;
```

#### Returns

Returns the sample frequency in Hz.

### Description

This function reads the currently active output frequency from the audio output.

# 2.1.5.6.5 AudioOutput::getOutputDelay Method



Gets the current audio output delay

## C++

```
int32_t getOutputDelay() const;
```

# Returns

Returns the audio delay in milli-seconds.

# Description

The current active audio output delay can be read by this function.

# 2.1.5.6.6 AudioOutput::setChannelMute Method



Selects audio channels to mute.

### C++

TMmRc setChannelMute(uint32\_t mask);

# **Parameters**

Parameters	Description
uint32_t mask	channel mask with bits set for muting

# Returns

MMRC\_Ok if muting setup was successful.

Return Value List

#### **Return Values**

Return Values	Description
MMRC_Ok	on success

# Description

This method mutes selected audio channels. The mask is a bit array with one bit per channel starting with the lowest bit (bit 0) for the first channel. A 1 in the mask means the channel is muted.

# 2.1.5.6.7 AudioOutput::setOutputDelay Method



Sets audio delay.

#### C++

TMmRc setOutputDelay(int32\_t delay);

#### **Parameters**

Parameters	Description
int32_t delay	delay in milli-seconds, any positive or negative value is allowed, too big values may influence buffer fullness or overall playback

## Returns

MMRC\_Ok if new value was successfully setup.

#### **Return Values**

Return Values	Description
MMRC_Ok	Return value 1

# Description

This method sets the delay between the audio output and the clock playback control, respectively the audio video delay. The delay is processed on the fly, so any change will be audioable immediately.

# 2.1.5.6.8 AudioOutput::setOutputFrequency Method

Sets the audio output frequency.



# C++

TMmRc setOutputFrequency(uint32\_t frequency);

### **Parameters**

Parameters	Description
uint32_t frequency	sample frequency in Hz, values of 48000 or 96000 are allowed

## Returns

MMRC\_Ok on success, otherwise an error code.

# **Return Values**

Return Values	Description
MMRC_Ok	Sucessfull frequency setup
MMRC_MVC2_SampleFrequencyNotSupported	unsupported sample frequency

## Description

The audio output can run at 48kHz or at 96kHz. This method is used to set the frequency of the output.

# 2.1.5.6.9 AudioOutput::setUserData Method

Sets audio user data.



#### C++

TMmRc setUserData(uint32\_t channel, const uint8\_t \* userData, uint32\_t size);

## **Parameters**

Parameters	Description
uint32_t channel	channel number of that audio output
const uint8_t * userData	an array of up to 24 bytes with the userdata, lsb first
uint32_t size	size of the userData array, if greater than 24, the array will be truncated to 24, if lower than 24, the array will be filled up with zeros

## **Returns**

MMRC\_Ok in successful setup.

#### **Return Values**

Return Values	Description
MMRC_Ok	on success
MMRC_MM_API_Index2OutOfRange	if the channel number is not available inside this output

# Description

A AES/3 audio channel can transfer optional channel with user bits. It is an array of 192 bits which can totally user defined or comply to AES/3 or IEC 60958 standard. This method can be used to set these 192 bits per audio channel. Changes take place immediately.

# 2.1.5.6.10 AudioOutput::setVolume Method



# C++

TMmRc setVolume(uint32\_t firstChannel, uint32\_t numChannels, const float \* volumeArray);

### Remarks

This is setVolume, a member of class AudioOutput.

# 2.1.5.6.11 AudioOutput::setVolumeDB Method



# C++

TMmRc setVolumeDB(uint32\_t firstChannel, uint32\_t numChannels, const float \* volumeArray);

## Remarks

This is setVolumeDB, a member of class AudioOutput.

# 2.1.5.7 AudioOutput Operators

# 2.1.5.7.1 AudioOutput::= Operator

#### C++

AudioOutput & operator =(const AudioOutput & other);

## Remarks

This is =, a member of class AudioOutput.

# 2.1.5.7.2 AudioOutput::bool Operator

Returns if the object is valid.

#### C++

operator bool() const;

# 2.1.6 AudioOutputStatus Class

# **Inheritance Hierarchy**

C++

class AudioOutputStatus;

#### File

mvc2api\_output.h

## Remarks

This is class mvc2::AudioOutputStatus.

# Members

# Methods

	Name	Description
<b>=</b> ♦ ₩	~AudioOutputStatus	This is ~AudioOutputStatus, a member of class AudioOutputStatus.
<b>≡</b>	AudioOutputStatus	This is AudioOutputStatus, a member of class AudioOutputStatus.
<b>=♦</b>	AudioOutputStatus	This is AudioOutputStatus, a member of class AudioOutputStatus.
<b>≡</b>	AudioOutputStatus	This is AudioOutputStatus, a member of class AudioOutputStatus.
<b>≡</b>	AudioOutputStatus	This is AudioOutputStatus, a member of class AudioOutputStatus.

# AudioOutputStatus Enumerations

	Name	Description
e <sup>e</sup>	TIME_RESOLUTION	This is record
		mvc2::AudioOutputStatus::TIME_RESOLUTION.

## AudioOutputStatus Data Members

	Name	Description
<b>∳</b> 9	m_StatusPriv	This is m_StatusPriv, a member of class AudioOutputStatus.

# AudioOutputStatus Methods

	Name	Description
<b>=</b> ♦	getOvershootLength	This is getOvershootLength, a member of class AudioOutputStatus.
<b>=♦</b>	getOvershootSamples	This is getOvershootSamples, a member of class AudioOutputStatus.
<b>=♦</b>	getSampleActivity	This is getSampleActivity, a member of class AudioOutputStatus.
<b>=</b> ♦	getSampleLevel	This is getSampleLevel, a member of class AudioOutputStatus.
<b>=</b> ♦	getSampleLevelDB	This is getSampleLevelDB, a member of class AudioOutputStatus.
=•	isChannelActive	This is isChannelActive, a member of class AudioOutputStatus.

# AudioOutputStatus Operators

	Name	Description
<u>(/-</u> =+)	=	This is =, a member of class AudioOutputStatus.
( <u>/-</u> =+)	bool	Returns if the object is valid.

# AudioOutputStatus Data Members

	Name	Description
<b>∳</b> 9	m_StatusPriv	This is m_StatusPriv, a member of class AudioOutputStatus.

# **AudioOutputStatus Enumerations**

	Name	Description
<b>a</b>	TIME_RESOLUTION	This is record
		mvc2::AudioOutputStatus::TIME_RESOLUTION.

# AudioOutputStatus Methods

	Name	Description
<b>=</b> ♦	getOvershootLength	This is getOvershootLength, a member of class AudioOutputStatus.
<b>≟♦</b>	getOvershootSamples	This is getOvershootSamples, a member of class AudioOutputStatus.
<b>=</b> ♦	getSampleActivity	This is getSampleActivity, a member of class AudioOutputStatus.
<b>≡♦</b>	getSampleLevel	This is getSampleLevel, a member of class AudioOutputStatus.
<b>≡♦</b>	getSampleLevelDB	This is getSampleLevelDB, a member of class AudioOutputStatus.
<b>≡♦</b>	isChannelActive	This is isChannelActive, a member of class AudioOutputStatus.

# AudioOutputStatus Operators

	Name	Description
=+)	=	This is =, a member of class AudioOutputStatus.
<u>(~</u> =+)	bool	Returns if the object is valid.

# 2.1.6.1 AudioOutputStatus::~AudioOutputStatus Destructor

## C++

virtual ~AudioOutputStatus();

#### Remarks

This is ~AudioOutputStatus, a member of class AudioOutputStatus.

# 2.1.6.2 AudioOutputStatus::AudioOutputStatus Constructor ()

#### C++

```
AudioOutputStatus();
```

#### Remarks

This is AudioOutputStatus, a member of class AudioOutputStatus.

# 2.1.6.3 AudioOutputStatus::AudioOutputStatus Constructor (AudioOutputStatus &)

 $C^{++}$ 

```
AudioOutputStatus(const AudioOutputStatus & other);
```

#### Remarks

This is AudioOutputStatus, a member of class AudioOutputStatus.

# 2.1.6.4 AudioOutputStatus::AudioOutputStatus Constructor (AudioOutputStatusPrivate \*)

C++

```
AudioOutputStatus(AudioOutputStatusPrivate * priv);
```

## Remarks

This is AudioOutputStatus, a member of class AudioOutputStatus.

# 2.1.6.5 AudioOutputStatus::AudioOutputStatus Constructor (MvcDevice &, uint32 t, TIME RESOLUTION)

C++

```
AudioOutputStatus(const MvcDevice & mvcdev, uint32_t index, TIME_RESOLUTION res =
Resolution_1s);
```

### Remarks

This is AudioOutputStatus, a member of class AudioOutputStatus.

# 2.1.6.6 AudioOutputStatus Enumerations

# 2.1.6.6.1 mvc2::AudioOutputStatus::TIME\_RESOLUTION Enumeration

C++

```
enum TIME_RESOLUTION {
  Resolution_100ms = 100,
  Resolution_1s = 1000
```

```
};
```

File

mvc2api\_output.h

#### Remarks

This is record mvc2::AudioOutputStatus::TIME\_RESOLUTION.

# 2.1.6.7 AudioOutputStatus Data Members

# 2.1.6.7.1 AudioOutputStatus::m\_StatusPriv Data Member

#### C++

```
AudioOutputStatusPrivate * m_StatusPriv;
```

#### Remarks

This is m\_StatusPriv, a member of class AudioOutputStatus.

# 2.1.6.8 AudioOutputStatus Methods

# 2.1.6.8.1 AudioOutputStatus::getOvershootLength Method

## C++

```
uint32_t getOvershootLength(uint32_t channel) const;
```

# Remarks

This is getOvershootLength, a member of class AudioOutputStatus.

# 2.1.6.8.2 AudioOutputStatus::getOvershootSamples Method

### C++

```
uint32_t getOvershootSamples(uint32_t channel) const;
```

# Remarks

This is getOvershootSamples, a member of class AudioOutputStatus.

# 2.1.6.8.3 AudioOutputStatus::getSampleActivity Method

# C++

```
uint32_t getSampleActivity(uint32_t channel) const;
```

### Remarks

This is getSampleActivity, a member of class AudioOutputStatus.

# 2.1.6.8.4 AudioOutputStatus::getSampleLevel Method

# C++

```
uint32_t getSampleLevel(uint32_t channel) const;
```

## Remarks

This is getSampleLevel, a member of class AudioOutputStatus.

# 2.1.6.8.5 AudioOutputStatus::getSampleLeveIDB Method

C++

```
float getSampleLevelDB(uint32_t channel) const;
```

#### Remarks

This is getSampleLevelDB, a member of class AudioOutputStatus.

# 2.1.6.8.6 AudioOutputStatus::isChannelActive Method

C++

```
bool isChannelActive(uint32_t channel) const;
```

### Remarks

This is isChannelActive, a member of class AudioOutputStatus.

# 2.1.6.9 AudioOutputStatus Operators

# 2.1.6.9.1 AudioOutputStatus::= Operator

C++

```
AudioOutputStatus & operator =(const AudioOutputStatus & other);
```

#### Remarks

This is =, a member of class AudioOutputStatus.

# 2.1.6.9.2 AudioOutputStatus::bool Operator

Returns if the object is valid.

C++

```
operator bool() const;
```

# Description

Returns true if the object is valid or false if the object is invalid.

# 2.1.7 BufferStatus Class

This object will hold information about the current status of the input buffer and calculated data rates.

# **Inheritance Hierarchy**

C++

```
class BufferStatus;
```

File

mvc2api\_decoder.h

# Description

# Members

# Methods

	Name	Description
<b>≡</b>	~BufferStatus	This is ~BufferStatus, a member of class BufferStatus.
<b>≡</b>	BufferStatus	This is BufferStatus, a member of class BufferStatus.
<b>≡</b>	BufferStatus	This is BufferStatus, a member of class BufferStatus.
<b>≡</b>	BufferStatus	This is BufferStatus, a member of class BufferStatus.

# **BufferStatus Enumerations**

	Name	Description
<b>*</b>	INTERVAL	This is record mvc2::BufferStatus::INTERVAL.

## **BufferStatus Methods**

	Name	Description
<b>=♦</b>	getBitrateDecoder	This is getBitrateDecoder, a member of class BufferStatus.
<b>=♦</b>	getBitrateDriver	This is getBitrateDriver, a member of class BufferStatus.
<b>=♦</b>	getDecoderBufferMax	This is getDecoderBufferMax, a member of class BufferStatus.
=0	getDecoderBufferSize	This is getDecoderBufferSize, a member of class BufferStatus.
<b>=♦</b>	getDecoderFullness	This is getDecoderFullness, a member of class BufferStatus.
<b>=♦</b>	getFreeDataBuffers	This is getFreeDataBuffers, a member of class BufferStatus.
<b>=♦</b>	getPendingFrames	This is getPendingFrames, a member of class BufferStatus.
<b>=♦</b>	getReturnCode	This is getReturnCode, a member of class BufferStatus.

# **BufferStatus Operators**

	Name	Description
( <u>/-</u>	=	This is =, a member of class BufferStatus.
( <u>/−</u> =+)	bool	This is bool, a member of class BufferStatus.

# **BufferStatus Enumerations**

	Name	Description
<b>a</b>	INTERVAL	This is record mvc2::BufferStatus::INTERVAL.

# **BufferStatus Methods**

	Name	Description
<b>=</b>	getBitrateDecoder	This is getBitrateDecoder, a member of class BufferStatus.
<b>≡</b>	getBitrateDriver	This is getBitrateDriver, a member of class BufferStatus.
<b>∉</b> ∳	getDecoderBufferMax	This is getDecoderBufferMax, a member of class BufferStatus.
<b>≡</b>	getDecoderBufferSize	This is getDecoderBufferSize, a member of class BufferStatus.
<b>≡</b>	getDecoderFullness	This is getDecoderFullness, a member of class BufferStatus.
<b>≡</b>	getFreeDataBuffers	This is getFreeDataBuffers, a member of class BufferStatus.
<b>=♦</b>	getPendingFrames	This is getPendingFrames, a member of class BufferStatus.
<b>∉</b>	getReturnCode	This is getReturnCode, a member of class BufferStatus.

# **BufferStatus Operators**

		Name	Description
<u>C</u>	<u>/-</u> :+)	=	This is =, a member of class BufferStatus.
=	/ <del>-</del> +)	bool	This is bool, a member of class BufferStatus.

# 2.1.7.1 BufferStatus::~BufferStatus Destructor

#### C++

```
~BufferStatus();
```

#### Remarks

This is ~BufferStatus, a member of class BufferStatus.

# 2.1.7.2 BufferStatus::BufferStatus Constructor (BufferStatus &)

C++

```
BufferStatus(const BufferStatus & other);
```

#### Remarks

This is BufferStatus, a member of class BufferStatus.

# 2.1.7.3 BufferStatus::BufferStatus Constructor (MvcDecoderPrivate \*)

C++

```
BufferStatus(MvcDecoderPrivate * mvcdec);
```

#### Remarks

This is BufferStatus, a member of class BufferStatus.

# 2.1.7.4 BufferStatus::BufferStatus Constructor (MvcDevice &, uint32 t)

C++

```
BufferStatus(const MvcDevice & mvcdev, uint32_t channel);
```

# Remarks

This is BufferStatus, a member of class BufferStatus.

# 2.1.7.5 BufferStatus Enumerations

# 2.1.7.5.1 mvc2::BufferStatus::INTERVAL Enumeration

```
C++
```

```
enum INTERVAL {
   Interval_100ms = 0,
   Interval_1s = 1,
   Interval_10s = 2
};
```

File

mvc2api\_decoder.h

#### Remarks

This is record mvc2::BufferStatus::INTERVAL.

# 2.1.7.6 BufferStatus Methods

# 2.1.7.6.1 BufferStatus::getBitrateDecoder Method

#### C++

```
uint32_t getBitrateDecoder(INTERVAL interval = Interval_1s) const;
```

#### Remarks

This is getBitrateDecoder, a member of class BufferStatus.

# 2.1.7.6.2 BufferStatus::getBitrateDriver Method

#### C++

```
uint32_t getBitrateDriver(INTERVAL interval = Interval_1s) const;
```

# Remarks

This is getBitrateDriver, a member of class BufferStatus.

# 2.1.7.6.3 BufferStatus::getDecoderBufferMax Method

#### C++

```
uint32_t getDecoderBufferMax() const;
```

## Remarks

This is getDecoderBufferMax, a member of class BufferStatus.

# 2.1.7.6.4 BufferStatus::getDecoderBufferSize Method

# C++

```
uint32_t getDecoderBufferSize() const;
```

### Remarks

This is getDecoderBufferSize, a member of class BufferStatus.

# 2.1.7.6.5 BufferStatus::getDecoderFullness Method

# C++

```
float getDecoderFullness() const;
```

## Remarks

This is getDecoderFullness, a member of class BufferStatus.

# 2.1.7.6.6 BufferStatus::getFreeDataBuffers Method

### C++

```
uint32_t getFreeDataBuffers() const;
```

## Remarks

This is getFreeDataBuffers, a member of class BufferStatus.

# 2.1.7.6.7 BufferStatus::getPendingFrames Method

#### C++

```
uint32_t getPendingFrames() const;
```

#### Remarks

This is getPendingFrames, a member of class BufferStatus.

# 2.1.7.6.8 BufferStatus::getReturnCode Method

## C++

```
TMmRc getReturnCode() const;
```

#### Remarks

This is getReturnCode, a member of class BufferStatus.

# 2.1.7.7 BufferStatus Operators

# 2.1.7.7.1 BufferStatus::= Operator

#### C++

```
BufferStatus & operator =(const BufferStatus & inbuf);
```

## Remarks

This is =, a member of class BufferStatus.

# 2.1.7.7.2 BufferStatus::bool Operator

# C++

```
operator bool() const;
```

## Remarks

This is bool, a member of class BufferStatus.

# 2.1.8 ColorConversionMatrix Class



# Inheritance Hierarchy

### C++

class ColorConversionMatrix;

# File

mvc2api\_output.h

## Remarks

This is class mvc2::ColorConversionMatrix.

# Members

## Methods

	Name	Description
<b>≡♦</b>	ColorConversionMatrix	This is ColorConversionMatrix, a member of class
		ColorConversionMatrix.

# **ColorConversionMatrix Operators**

	Name	Description
<u>(/-</u> =+)	=	This is =, a member of class ColorConversionMatrix.

## **ColorConversionMatrix Operators**

	Name	Description
=+)	=	This is =, a member of class ColorConversionMatrix.

# 2.1.8.1 ColorConversionMatrix::ColorConversionMatrix Constructor

C++

ColorConversionMatrix(const float \* matrix, const int32\_t \* post\_add);

## Remarks

This is ColorConversionMatrix, a member of class ColorConversionMatrix.

# 2.1.8.2 ColorConversionMatrix Operators

# 2.1.8.2.1 ColorConversionMatrix::= Operator

C++

ColorConversionMatrix & operator =(const ColorConversionMatrix & conv);

## Remarks

This is =, a member of class ColorConversionMatrix.

# 2.1.9 ColorSpace Class



Colorspace definition class.

# **Inheritance Hierarchy**

C++

class ColorSpace;

File

mvc2api\_output.h

# Description

## **Members**

## Methods

	Name	Description
<b>=♦</b>	ColorSpace	This is ColorSpace, a member of class ColorSpace.
<b>≡♦</b>	ColorSpace	This is ColorSpace, a member of class ColorSpace.

# **ColorSpace Enumerations**

	Name	Description
<b>a</b>	ColourPrimaries	This is record mvc2::ColorSpace::ColourPrimaries.
<b>a</b>	MatrixCoefficients	This is record mvc2::ColorSpace::MatrixCoefficients.
<b>a</b>	Space	This is record mvc2::ColorSpace::Space.
	TransferCharacteristics	This is record mvc2::ColorSpace::TransferCharacteristics.

# **ColorSpace Methods**

	Name	Description
<b>∉</b> ∳	getColorSpacePacked	This is getColorSpacePacked, a member of class ColorSpace.
<b>≡♦</b>	getSpace	This is getSpace, a member of class ColorSpace.

# **ColorSpace Operators**

	Name	Description
<u>(/-</u> =+)	=	This is =, a member of class ColorSpace.

# **ColorSpace Enumerations**

	Name	Description
<b>a</b>	ColourPrimaries	This is record mvc2::ColorSpace::ColourPrimaries.
<b></b>	MatrixCoefficients	This is record mvc2::ColorSpace::MatrixCoefficients.
<b></b>	Space	This is record mvc2::ColorSpace::Space.
<b>a</b>	TransferCharacteristics	This is record mvc2::ColorSpace::TransferCharacteristics.

# **ColorSpace Methods**

	Name	Description
<b>≡</b> ♦	0 1	This is getColorSpacePacked, a member of class ColorSpace.
<b>=♦</b>	getSpace	This is getSpace, a member of class ColorSpace.

# **ColorSpace Operators**

	Name	Description
<u>∨−</u> =+)	=	This is =, a member of class ColorSpace.

# 2.1.9.1 ColorSpace::ColorSpace Constructor (ColourPrimaries, TransferCharacteristics, MatrixCoefficients)

# C++

ColorSpace(ColourPrimaries prim, TransferCharacteristics trans, MatrixCoefficients coeff);

### Remarks

This is ColorSpace, a member of class ColorSpace.

# 2.1.9.2 ColorSpace::ColorSpace Constructor (Space)

#### C++

```
ColorSpace(Space spc = ColorSpace::Space_Default);
```

This is ColorSpace, a member of class ColorSpace.

# 2.1.9.3 ColorSpace Enumerations

# 2.1.9.3.1 mvc2::ColorSpace::ColourPrimaries Enumeration

#### C++

```
enum ColourPrimaries {
    Prim_Default = 0,
    Prim_ITUR_BT_709 = 1,
    Prim_ITUR_BT_470_M = 4,
    Prim_ITUR_BT_470_BG = 5,
    Prim_SMPTE_170M = 6,
    Prim_SMPTE_240M = 7,
    Prim_P3 = 8,
    Prim_XYZ = 100,
    Prim_sRGB = 101
};
File

mvc2api_output.h
```

## Remarks

This is record mvc2::ColorSpace::ColourPrimaries.

# 2.1.9.3.2 mvc2::ColorSpace::MatrixCoefficients Enumeration

# C++

```
enum MatrixCoefficients {
   Coeff_Default = 0,
   Coeff_ITUR_BT_709 = 1,
   Coeff_FCC = 4,
   Coeff_ITUR_BT_470_BG = 5,
   Coeff_SMPTE_170M = 6,
   Coeff_SMPTE_240M = 7,
   Coeff_DCI_P3 = 8,
   Coeff_YCxCz = 20,
   Coeff_RGB = 100,
   Coeff_XYZ = 101
   };

File
   mvc2api_output.h
```

# Remarks

This is record mvc2::ColorSpace::MatrixCoefficients.

# 2.1.9.3.3 mvc2::ColorSpace::Space Enumeration

```
C++
```

```
enum Space {
   Space_Default = 0,
   Space_RGB = 1,
   Space_YCbCr = 2,
   Space_XYZ = 3,
   Space_YCxCz = 4,
   Space_YCbCr_P3 = 5,
   Space_Max
};
File
mvc2api_output.h
```

#### Remarks

This is record mvc2::ColorSpace::Space.

# 2.1.9.3.4 mvc2::ColorSpace::TransferCharacteristics Enumeration

## C++

```
enum TransferCharacteristics {
   Trans_Default = 0,
   Trans_ITUR_BT_709 = 1,
   Trans_ITUR_BT_470_M = 4,
   Trans_ITUR_BT_470_BG = 5,
   Trans_SMPTE_170M = 6,
   Trans_SMPTE_240M = 7,
   Trans_Linear = 8,
   Trans_DCI = 9,
   Trans_ITUR_BT_1361 = 10,
   Trans_SRGB = 11
   };

File
   mvc2api_output.h
```

# Remarks

This is record mvc2::ColorSpace::TransferCharacteristics.

# 2.1.9.4 ColorSpace Methods

# 2.1.9.4.1 ColorSpace::getColorSpacePacked Method

```
C++
```

```
uint32_t getColorSpacePacked() const;
```

## Remarks

This is getColorSpacePacked, a member of class ColorSpace.

# 2.1.9.4.2 ColorSpace::getSpace Method

### C++

```
ColorSpace::Space getSpace() const;
```

# Remarks

This is getSpace, a member of class ColorSpace.

# 2.1.9.5 ColorSpace Operators

# 2.1.9.5.1 ColorSpace::= Operator

## C++

```
ColorSpace & operator =(const ColorSpace & spc);
```

## Remarks

This is =, a member of class ColorSpace.

# 2.1.10 ConfigAccess Class

# **Inheritance Hierarchy**

# C++

class ConfigAccess;

# File

mvc2api\_config.h

# Remarks

This class is used to access configuration for the MVC20x firmware and driver.

## **Members**

## Methods

	Name	Description
<b>=</b> ♦ ₩	~ConfigAccess	This is ~ConfigAccess, a member of class ConfigAccess.
<b>≡♦</b>	ConfigAccess	This is ConfigAccess, a member of class ConfigAccess.
<b>≡♦</b>	ConfigAccess	This is ConfigAccess, a member of class ConfigAccess.
<b>≡♦</b>	ConfigAccess	This is ConfigAccess, a member of class ConfigAccess.

# **ConfigAccess Methods**

	Name	Description
<b>=♦</b>	getLogFilter	This is getLogFilter, a member of class ConfigAccess.
<b>=♦</b>	getLogFilterConsole	This is getLogFilterConsole, a member of class ConfigAccess.
<b>=</b> ♦	getMessageFifoClients	This is getMessageFifoClients, a member of class ConfigAccess.
<b>=</b> ♦	getMessageFifoSize	This is getMessageFifoSize, a member of class ConfigAccess.
<b>=♦</b>	getNetworkDHCP	This is getNetworkDHCP, a member of class ConfigAccess.
<b>=</b> ♦	getNetworkIPAddress	This is getNetworkIPAddress, a member of class ConfigAccess.
<b>=♦</b>	getNetworkMask	This is getNetworkMask, a member of class ConfigAccess.
<b>=♦</b>	setLogFilter	This is setLogFilter, a member of class ConfigAccess.
<b>=</b> ♦	setLogFilterConsole	This is setLogFilterConsole, a member of class ConfigAccess.

<b>=♦</b>	setMessageFifoClients	This is setMessageFifoClients, a member of class ConfigAccess.
<b>=♦</b>	setMessageFifoSize	This is setMessageFifoSize, a member of class ConfigAccess.
<b>≡♦</b>	setNetworkDHCP	This is setNetworkDHCP, a member of class ConfigAccess.
=•	setNetworkIPAddress	This is setNetworkIPAddress, a member of class ConfigAccess.
<b>≡♦</b>	setNetworkMask	This is setNetworkMask, a member of class ConfigAccess.

# **ConfigAccess Operators**

	Name	Description
( <u>/-</u> =+)	-	This is =, a member of class ConfigAccess.
( <u>/-</u> =+)	bool	This is bool, a member of class ConfigAccess.

# **ConfigAccess Methods**

	Name	Description
<b>≡</b>	getLogFilter	This is getLogFilter, a member of class ConfigAccess.
<b>=</b> ♦	getLogFilterConsole	This is getLogFilterConsole, a member of class ConfigAccess.
<b>≡♦</b>	getMessageFifoClients	This is getMessageFifoClients, a member of class ConfigAccess.
<b>≟♦</b>	getMessageFifoSize	This is getMessageFifoSize, a member of class ConfigAccess.
<b>≡</b>	getNetworkDHCP	This is getNetworkDHCP, a member of class ConfigAccess.
<b>≡♦</b>	getNetworkIPAddress	This is getNetworkIPAddress, a member of class ConfigAccess.
<b>≡</b>	getNetworkMask	This is getNetworkMask, a member of class ConfigAccess.
<b>≡♦</b>	setLogFilter	This is setLogFilter, a member of class ConfigAccess.
<b>=</b> ♦	setLogFilterConsole	This is setLogFilterConsole, a member of class ConfigAccess.
<b>≡♦</b>	setMessageFifoClients	This is setMessageFifoClients, a member of class ConfigAccess.
<b>≡♦</b>	setMessageFifoSize	This is setMessageFifoSize, a member of class ConfigAccess.
<b>≡♦</b>	setNetworkDHCP	This is setNetworkDHCP, a member of class ConfigAccess.
<b>≡♦</b>	setNetworkIPAddress	This is setNetworkIPAddress, a member of class ConfigAccess.
<b>≡</b>	setNetworkMask	This is setNetworkMask, a member of class ConfigAccess.

# **ConfigAccess Operators**

	Name	Description
=+)	=	This is =, a member of class ConfigAccess.
<del>V=</del> =+)	bool	This is bool, a member of class ConfigAccess.

# 2.1.10.1 ConfigAccess::~ConfigAccess Destructor

# C++

virtual ~ConfigAccess();

# Remarks

This is ~ConfigAccess, a member of class ConfigAccess.

# 2.1.10.2 ConfigAccess::ConfigAccess Constructor ()

#### C++

ConfigAccess();

#### Remarks

This is ConfigAccess, a member of class ConfigAccess.

# 2.1.10.3 ConfigAccess::ConfigAccess Constructor (ConfigAccess&)

C++

ConfigAccess(const ConfigAccess& other);

#### Remarks

This is ConfigAccess, a member of class ConfigAccess.

# 2.1.10.4 ConfigAccess::ConfigAccess Constructor (TMmRc \*, MvcDevice &)

C++

ConfigAccess(TMmRc \* resultPointer, const MvcDevice & mvcDevice);

## Remarks

This is ConfigAccess, a member of class ConfigAccess.

# 2.1.10.5 ConfigAccess Methods

# 2.1.10.5.1 ConfigAccess::getLogFilter Method

### C++

```
TMmRc getLogFilter(uint32_t * filter);
```

# Remarks

This is getLogFilter, a member of class ConfigAccess.

# 2.1.10.5.2 ConfigAccess::getLogFilterConsole Method

## C++

```
TMmRc getLogFilterConsole(uint32_t * filter);
```

## Remarks

This is getLogFilterConsole, a member of class ConfigAccess.

# 2.1.10.5.3 ConfigAccess::getMessageFifoClients Method

### C++

```
TMmRc getMessageFifoClients(uint32_t * clients);
```

## Remarks

This is getMessageFifoClients, a member of class ConfigAccess.

# 2.1.10.5.4 ConfigAccess::getMessageFifoSize Method

#### C++

```
TMmRc getMessageFifoSize(uint32_t * size);
```

#### Remarks

This is getMessageFifoSize, a member of class ConfigAccess.

# 2.1.10.5.5 ConfigAccess::getNetworkDHCP Method

### C++

```
TMmRc getNetworkDHCP(bool * enable);
```

#### Remarks

This is getNetworkDHCP, a member of class ConfigAccess.

# 2.1.10.5.6 ConfigAccess::getNetworklPAddress Method

#### C++

```
TMmRc getNetworkIPAddress(uint32_t * ip);
```

### Remarks

This is getNetworkIPAddress, a member of class ConfigAccess.

# 2.1.10.5.7 ConfigAccess::getNetworkMask Method

## C++

```
TMmRc getNetworkMask(uint32_t * mask);
```

## Remarks

This is getNetworkMask, a member of class ConfigAccess.

# 2.1.10.5.8 ConfigAccess::setLogFilter Method

### C++

```
TMmRc setLogFilter(uint32_t filter);
```

# Remarks

This is setLogFilter, a member of class ConfigAccess.

# 2.1.10.5.9 ConfigAccess::setLogFilterConsole Method

### C++

```
TMmRc setLogFilterConsole(uint32_t filter);
```

# Remarks

This is setLogFilterConsole, a member of class ConfigAccess.

# 2.1.10.5.10 ConfigAccess::setMessageFifoClients Method

#### C++

TMmRc setMessageFifoClients(uint32\_t clients);

#### Remarks

This is setMessageFifoClients, a member of class ConfigAccess.

# 2.1.10.5.11 ConfigAccess::setMessageFifoSize Method

#### C++

TMmRc setMessageFifoSize(uint32\_t size);

### Remarks

This is setMessageFifoSize, a member of class ConfigAccess.

# 2.1.10.5.12 ConfigAccess::setNetworkDHCP Method

## C++

TMmRc setNetworkDHCP(bool enable);

#### Remarks

This is setNetworkDHCP, a member of class ConfigAccess.

# 2.1.10.5.13 ConfigAccess::setNetworkIPAddress Method

#### C++

TMmRc setNetworkIPAddress(uint32\_t ip);

## Remarks

This is setNetworkIPAddress, a member of class ConfigAccess.

# 2.1.10.5.14 ConfigAccess::setNetworkMask Method

# C++

TMmRc setNetworkMask(uint32\_t mask);

## Remarks

This is setNetworkMask, a member of class ConfigAccess.

# 2.1.10.6 ConfigAccess Operators

# 2.1.10.6.1 ConfigAccess::= Operator

### C++

ConfigAccess& operator =(const ConfigAccess& other);

## Remarks

This is =, a member of class ConfigAccess.

# 2.1.10.6.2 ConfigAccess::bool Operator

#### C++

operator bool() const;

## Remarks

This is bool, a member of class ConfigAccess.

# 2.1.11 CPUInfo Class

# **Inheritance Hierarchy**

## C++

class CPUInfo;

# File

mvc2api\_device.h

# Remarks

This is class mvc2::CPUInfo.

## **Members**

## Methods

	Name	Description
<b>≡♦</b>	~CPUInfo	This is ~CPUInfo, a member of class CPUInfo.
<b>=</b> ♦	CPUInfo	This is CPUInfo, a member of class CPUInfo.
<b>≡♦</b>	CPUInfo	This is CPUInfo, a member of class CPUInfo.
<b>=♦</b>	CPUInfo	This is CPUInfo, a member of class CPUInfo.

# **CPUInfo Enumerations**

	Name	Description
<b>a</b>	CPUTYPE	This is record mvc2::CPUInfo::CPUTYPE.

## **CPUInfo Methods**

	Name	Description
<b>=</b> ♦	getCoreFrequency	This is getCoreFrequency, a member of class CPUInfo.
<b>=♦</b>	getCoreFrequencykHz	This is getCoreFrequencykHz, a member of class CPUInfo.
<b>=♦</b>	getCPUType	This is getCPUType, a member of class CPUInfo.
<b>=♦</b>	getCPUTypeString	This is getCPUTypeString, a member of class CPUInfo.
<b>=♦</b>	getHardwareVersion	This is getHardwareVersion, a member of class CPUInfo.
<b>=</b> ♦	getNumberOfCores	This is getNumberOfCores, a member of class CPUInfo.

# **CPUInfo Operators**

	Name	Description
( <u>/-</u> =+)	=	This is =, a member of class CPUInfo.
( <u>/-</u> =+)	bool	This is bool, a member of class CPUInfo.

# **CPUInfo Enumerations**

	Name	Description
<b>P</b>	CPUTYPE	This is record mvc2::CPUInfo::CPUTYPE.

# **CPUInfo Methods**

	Name	Description
<b>≡♦</b>	getCoreFrequency	This is getCoreFrequency, a member of class CPUInfo.
<b>≡♦</b>	getCoreFrequencykHz	This is getCoreFrequencykHz, a member of class CPUInfo.
<b>≡♦</b>	getCPUType	This is getCPUType, a member of class CPUInfo.
<b>≡♦</b>	getCPUTypeString	This is getCPUTypeString, a member of class CPUInfo.
<b>≡♦</b>	getHardwareVersion	This is getHardwareVersion, a member of class CPUInfo.
<b>≡♦</b>	getNumberOfCores	This is getNumberOfCores, a member of class CPUInfo.

# **CPUInfo Operators**

	Name	Description
=+)	=	This is =, a member of class CPUInfo.
( <u>/-</u> =+)	bool	This is bool, a member of class CPUInfo.

# 2.1.11.1 CPUInfo::~CPUInfo Destructor

## C++

~CPUInfo();

#### Remarks

This is ~CPUInfo, a member of class CPUInfo.

# 2.1.11.2 CPUInfo::CPUInfo Constructor ()

## C++

CPUInfo();

## Remarks

This is CPUInfo, a member of class CPUInfo.

# 2.1.11.3 CPUInfo::CPUInfo Constructor (CPUInfo &)

## C++

CPUInfo(const CPUInfo & other);

# Remarks

This is CPUInfo, a member of class CPUInfo.

# 2.1.11.4 CPUInfo::CPUInfo Constructor (MvcDevicePrivate \*, uint32\_t)

# C++

CPUInfo(MvcDevicePrivate \* mvcdev, uint32\_t index);

### Remarks

This is CPUInfo, a member of class CPUInfo.

# 2.1.11.5 CPUInfo Enumerations

# 2.1.11.5.1 mvc2::CPUInfo::CPUTYPE Enumeration

#### C++

```
enum CPUTYPE {
    CPUTYPE_Unknown = 0,
    CPUTYPE_PMCSierra_MSP8510 = 1
    };

File
    mvc2api_device.h
```

## Remarks

This is record mvc2::CPUInfo::CPUTYPE

# 2.1.11.6 CPUInfo Methods

# 2.1.11.6.1 CPUInfo::getCoreFrequency Method

## C++

```
uint64_t getCoreFrequency() const;
```

## Remarks

This is getCoreFrequency, a member of class CPUInfo.

# 2.1.11.6.2 CPUInfo::getCoreFrequencykHz Method

### C++

```
uint32_t getCoreFrequencykHz() const;
```

# Remarks

This is getCoreFrequencykHz, a member of class CPUInfo.

# 2.1.11.6.3 CPUInfo::getCPUType Method

## C++

```
CPUTYPE getCPUType() const;
```

## Remarks

This is getCPUType, a member of class CPUInfo.

# 2.1.11.6.4 CPUInfo::getCPUTypeString Method

## C++

```
const char * getCPUTypeString() const;
```

### Remarks

This is getCPUTypeString, a member of class CPUInfo.

# 2.1.11.6.5 CPUInfo::getHardwareVersion Method

### C++

```
VersionValue getHardwareVersion() const;
```

# Remarks

This is getHardwareVersion, a member of class CPUInfo.

# 2.1.11.6.6 CPUInfo::getNumberOfCores Method

#### C++

```
uint32_t getNumberOfCores() const;
```

#### Remarks

This is getNumberOfCores, a member of class CPUInfo.

# 2.1.11.7 CPUInfo Operators

# 2.1.11.7.1 CPUInfo::= Operator

#### C++

```
CPUInfo & operator = (const CPUInfo & inbuf);
```

#### Remarks

This is =, a member of class CPUInfo.

# 2.1.11.7.2 CPUInfo::bool Operator

#### C++

```
operator bool() const;
```

## Remarks

This is bool, a member of class CPUInfo.

# 2.1.12 DataBuffer Class

Transfer data buffer class.

# Inheritance Hierarchy

# C++

```
class DataBuffer;
```

## File

mvc2api\_decoder.h

# Description

This class is used to transfer data to the MVC card. A decoder will create a number of DataBuffers which will be provided by the decoder. A application should get a DataBuffer from the decoder, fill it with one frame and set additional information (like timestamp or userdata). After the data was copied to the buffer, the application will set the size to transfer using setInSize() method. The transfer will be started by the send() method. After calling send(), the buffer is invalid an cannot be used anymore.

## Remarks

# Members

# Methods

	Name	Description
<b>=</b> ♦ <b>W</b>	~DataBuffer	Destructor
<b>≡♦</b>	DataBuffer	This is DataBuffer, a member of class DataBuffer.
<b>≟</b> ♦	DataBuffer	This is DataBuffer, a member of class DataBuffer.
<b>=</b> ♦•	DataBuffer	This is DataBuffer, a member of class DataBuffer.

# **DataBuffer Methods**

	Name	Description
<b>=</b>	сору	Copy a data block to DataBuffer.
<b>≡</b>	getBufferAddress	Gets the buffer memory address.
<b>=</b>	getFreeSize	Get the available buffer size.
<b>=</b>	getInSize	Get the transfer size.
<b>=</b>	getTimeStamp	Get current timestamp of the DataBuffer.
<b>=</b>	send	Send buffer to MVC card.
<b>≡</b>	send	Send DataBuffer to the MVC card.
<b>≡</b>	setDecryptionSize	Set decryption size.
<b>=</b>	setInSize	Sets the transfer size.
<b>=♦</b>	setKeyId	Set decryption information.
<b>=</b>	setKeyIndex	Set decryption information.
<b>≡</b>	setMicValue	Set the MIC value extracted from the MXF file.
<b>≡</b>	setTimeStamp	Sets the presentation time stamp.
<b>≡</b>	setUserData	Sets user data.
<b>≡</b>	wait	Wait for transfer completion

# **DataBuffer Operators**

	Name	Description
=+)	=	Assignment operator.
<u>(&gt;−</u> =+)	bool	This is bool, a member of class DataBuffer.

# **DataBuffer Methods**

copy getBufferAddress getFreeSize	Copy a data block to DataBuffer.  Gets the buffer memory address.
	Gets the buffer memory address.
getFreeSize	
9011 1000120	Get the available buffer size.
getInSize	Get the transfer size.
getTimeStamp	Get current timestamp of the DataBuffer.
send	Send buffer to MVC card.
send	Send DataBuffer to the MVC card.
setDecryptionSize	Set decryption size.
setInSize	Sets the transfer size.
setKeyld	Set decryption information.
setKeyIndex	Set decryption information.
setMicValue	Set the MIC value extracted from the MXF file.
setTimeStamp	Sets the presentation time stamp.
setUserData	Sets user data.
wait	Wait for transfer completion
	getTimeStamp send send setDecryptionSize setInSize setKeyId setKeyIndex setMicValue setTimeStamp setUserData

# **DataBuffer Operators**

	Name	Description
=+)	=	Assignment operator.
<u>(/-</u> =+)	bool	This is bool, a member of class DataBuffer.

# 2.1.12.1 DataBuffer::~DataBuffer Destructor

Destructor

C++

```
virtual ~DataBuffer();
```

# Description

Default destructor of a DataBuffer. All references to memory will be removed.

# 2.1.12.2 DataBuffer::DataBuffer Constructor ()

C++

DataBuffer();

## Remarks

This is DataBuffer, a member of class DataBuffer.

# 2.1.12.3 DataBuffer::DataBuffer Constructor (DataBuffer&)

C++

DataBuffer(const DataBuffer& other);

## Remarks

This is DataBuffer, a member of class DataBuffer.

# 2.1.12.4 DataBuffer::DataBuffer Constructor (MvcDecoderPrivate \*, MemoryBuffer \*)

C++

```
DataBuffer(MvcDecoderPrivate * mvcdec, MemoryBuffer * mem);
```

# Remarks

This is DataBuffer, a member of class DataBuffer.

# 2.1.12.5 DataBuffer Methods

# 2.1.12.5.1 DataBuffer::copy Method

Copy a data block to DataBuffer.

C++

```
uint32_t copy(const uint8_t * srcblock, uint32_t size);
```

#### **Parameters**

Parameters	Description
const uint8_t * srcblock	pointer to a array of bytes to copy from
uint32_t size	number of bytes to copy

#### **Returns**

number of bytes which were copied

## Description

This method will copy data to the DataBuffer object and adjusts the size. It's more easy to use the getBufferAddress() + memcpy() +setInSize().

# 2.1.12.5.2 DataBuffer::getBufferAddress Method

Gets the buffer memory address.

#### C++

```
uint8_t * getBufferAddress();
```

## Returns

Memory address or NULL if the buffer doesn't hold any memory.

## Description

This method will return the memory address of the DataBuffer. Do not copy more data into this buffer then the maximal size. The buffer size will be return by getFreeSize() method.

# 2.1.12.5.3 DataBuffer::getFreeSize Metho

Get the available buffer size.

### C++

```
uint32_t getFreeSize();
```

## Returns

Number of bytes available in this buffer.

# Description

This method returns the number of bytes available in this DataBuffer. setInSize() should be used to set the data size which needs to be transferred to the MVC card.

# 2.1.12.5.4 DataBuffer::getInSize Method



Get the transfer size.

# C++

```
uint32_t getInSize();
```

### Returns

Number of bytes to transfer to the card.

# **Description**

This method will return the size which the application has selected for the transfer. It's the size which was set using the setInSize() method.

# 2.1.12.5.5 DataBuffer::getTimeStamp Method



Get current timestamp of the DataBuffer.

#### C++

```
uint32_t getTimeStamp() const;
```

#### Returns

Returns the time stamp of the DataBuffer which is based on 90kHz ticks.

### Description

Some decoder may calculate the time stamp for a new DataBuffer automatically (e.g. by setting a frame rate). This function will return the current time stamp. It's the calculated time stamp for a new DataBuffer or the stamp set using setTimeStamp().

# 2.1.12.5.6 DataBuffer::send Method ()



Send buffer to MVC card.

## C++

```
TMmRc send();
```

#### Description

After the buffer was filled, the data can be written to the MVC card. The send method will transfer the buffer to the driver. The number of bytes which will be transferred must be specified by the application before calling send() using setInSize() method. After calling send(), the DataBuffer is empty, it doesn't hold a memory space anymore.

# 2.1.12.5.7 DataBuffer::send Method (uint32\_t)



Send DataBuffer to the MVC card.

# C++

```
TMmRc send(uint32_t size);
```

## **Parameters**

Parameters	Description
uint32_t size	number of bytes to transfer

# Description

Same method as send(), but the transfer size will be set before.

# 2.1.12.5.8 DataBuffer::setDecryptionSize Method

Set decryption size.



# C++

```
void setDecryptionSize(uint32_t plainOffset, uint32_t sourceLength);
```

## **Parameters**

Parameters	Description
uint32_t plainOffset	number of plain bytes at the start of the data block
uint32_t sourceLength	total number of bytes after decryption (plainOffset + encryptionSize - encryptionStuffing)

### Description

A data transfer is split into three parts. First a number of unencrypted bytes following the encrypted data. The encrypted data itself is stuff to be dividable by 16.

# 2.1.12.5.9 DataBuffer::setInSize Meth

Sets the transfer size.

### C++

```
void setInSize(uint32_t size);
```

### **Parameters**

Parameters	Description
uint32_t size	number of bytes to transfer to the MVC card

### Description

After the data buffer was filled, the application needs to tell how many bytes are in the buffer and need to be transferred to the MVC card. The default size is zero, so no data will be transferred by default.

## 2.1.12.5.10 DataBuffer::setKeyld Method



Set decryption information.

### C++

```
TMmRc setKeyId(const mvc2::UuidValue & keyid, const uint8_t * iv, const uint8_t * cv);
```

### **Parameters**

Parameters	Description
const mvc2::UuidValue & keyid	Key UUID is for decryption.
const uint8_t * iv	128 bit (16 bytes array) initialization vector for the AES decryption
const uint8_t * cv	128 bit (16 bytes array) check value for the AES decryption

### Description

This method will set all necessary information needed to decrypt the data. The decryption will only be enabled if all fields are filled with valid data and the plain size is smaller then the size of the DataBuffer. This function will use the security manager if available and set to the decoder to translate the keyid to the keyindex. The security manager must have the CPL and KDM and playShow() must already been called.

# 2.1.12.5.11 DataBuffer::setKeyIndex Method



Set decryption information.

### C++

```
TMmRc setKeyIndex(uint16_t keyindex, const uint8_t * iv, const uint8_t * cv);
```

### Parameters

Parameters	Description
uint16_t keyindex	index of the AES key for the decryption or NO_DECRYPTION for plain data.
const uint8_t * iv	128 bit (16 bytes array) initialization vector for the AES decryption
const uint8_t * cv	128 bit (16 bytes array) check value for the AES decryption

### Description

This method will set all necessary information needed to decrypt the data. The decryption will only be enabled if all fields are filled with valid data and the plain size is smaller then the size of the DataBuffer.

## 2.1.12.5.12 DataBuffer::setMicValue Method



Set the MIC value extracted from the MXF file.

### C++

TMmRc setMicValue(const uint8\_t \* micBlock, uint32\_t size);

### **Parameters**

Parameters	Description
const uint8_t * micBlock	pointer to the MIC block of the MXF
uint32_t size	size of the micBlock in bytes

### Returns

Returns MMRC\_Ok if internal save went well.

### Description

The function sets the MIC which will be used for integrity checks. It starts with the first byte of the BER of the track UUID and ends with the last byte of the MIC value.

# 2.1.12.5.13 DataBuffer::setTimeStamp Method



Sets the presentation time stamp.

### C++

void setTimeStamp(uint32\_t stamp);

### **Parameters**

Parameters	Description
uint32_t stamp	presentation time stamp in 90kHz ticks

### Description

This method will set the presentation time stamp for the frame inside of the DataBuffer. The time stamp will start at 0 for the first frame and will count in 90kHz ticks.

# 2.1.12.5.14 DataBuffer::setUserData Method





### C++

void setUserData(uint32\_t data);

### **Parameters**

Parameters	Description
uint32_t data	32 bit user data

### **Description**

Every frame holds a user data field which can be read out if the frame is displayed. The usage of this field is up to the application.

# 2.1.12.5.15 DataBuffer::wait Method



Wait for transfer completion

### C++

```
TMmRc wait(uint32_t timeout);
```

### **Parameters**

Parameters	Description
uint32_t timeout	number of milliseconds to wait for the transfer completion

### Returns

TMmRc of the operation

### **Return Values**

Return Values	Description
MMRC_Timeout	operation timed out
MMRC_Ok	transfer completed

### Description

After send() method was called, this method could be used to wait for the transfer completion of this buffer. Normally this method is not needed for transfers, because the number of buffers is limited, so the API will automatically wait for a free buffer from time to time.

# 2.1.12.6 DataBuffer Operators

# 2.1.12.6.1 DataBuffer::= Operator

Assignment operator.

### C++

DataBuffer & operator = (const DataBuffer & other);

### **Parameters**

Parameters	Description
inbuf	Source DataBuffer

### Returns

If the source DataBuffer is not empty, a true will be returned, otherwise false.

### Description

This operator copies the information from one DataBuffer to another.

# 2.1.12.6.2 DataBuffer::bool Operator

### C++

```
operator bool() const;
```

### Remarks

This is bool, a member of class DataBuffer.

# 2.1.13 DecoderErrors Class

### **Inheritance Hierarchy**

### C++

class DecoderErrors;

### File

mvc2api\_decoder.h

### Remarks

Base class for decoder error counters.

### **Members**

### Methods

	Name	Description
<b>=</b> ♦ <b>W</b>	~DecoderErrors	This is ~DecoderErrors, a member of class DecoderErrors.
<b>≡⋄</b>	DecoderErrors	This is DecoderErrors, a member of class DecoderErrors.
<b>≡♦</b>	DecoderErrors	This is DecoderErrors, a member of class DecoderErrors.
<b>≡♦</b>	DecoderErrors	This is DecoderErrors, a member of class DecoderErrors.
<b>=♦</b>	DecoderErrors	This is DecoderErrors, a member of class DecoderErrors.

### **DecoderErrors Methods**

	Name	Description
<b>≡♦</b>	getReturnCode	This is getReturnCode, a member of class DecoderErrors.

### **DecoderErrors Operators**

	Name	Description
( <u>/-</u> =+)	=	This is =, a member of class DecoderErrors.
( <u>/-</u> =+)	bool	Returns if the object is valid.

### **DecoderErrors Methods**

	Name	Description
<b>=♦</b>	getReturnCode	This is getReturnCode, a member of class DecoderErrors.

### **DecoderErrors Operators**

	Name	Description
( <u>/-</u> =+)	=	This is =, a member of class DecoderErrors.
( <u>/-</u> =+)	bool	Returns if the object is valid.

# 2.1.13.1 DecoderErrors::~DecoderErrors Destructor

### $C^{++}$

virtual ~DecoderErrors();

### Remarks

This is ~DecoderErrors, a member of class DecoderErrors.

# 2.1.13.2 DecoderErrors::DecoderErrors Constructor ()

C++

DecoderErrors();

### Remarks

This is DecoderErrors, a member of class DecoderErrors.

# 2.1.13.3 DecoderErrors::DecoderErrors Constructor (DecoderErrors &)

C++

DecoderErrors(const DecoderErrors & other);

### Remarks

This is DecoderErrors, a member of class DecoderErrors.

# 2.1.13.4 DecoderErrors::DecoderErrors Constructor (MvcDecoderPrivate \*)

C++

DecoderErrors(MvcDecoderPrivate \* mvcdec);

### Remarks

This is DecoderErrors, a member of class DecoderErrors.

# 2.1.13.5 DecoderErrors::DecoderErrors Constructor (MvcDevice &, uint32\_t)

C++

DecoderErrors(const MvcDevice & mvcdev, uint32\_t channel);

### Remarks

This is DecoderErrors, a member of class DecoderErrors.

### 2.1.13.6 DecoderErrors Methods

# 2.1.13.6.1 DecoderErrors::getReturnCode Method

C++

TMmRc getReturnCode() const;

### Remarks

This is getReturnCode, a member of class DecoderErrors.

# 2.1.13.7 DecoderErrors Operators

# 2.1.13.7.1 DecoderErrors::= Operator

### C++

DecoderErrors & operator =(const DecoderErrors & other);

### Remarks

This is =, a member of class DecoderErrors.

# 2.1.13.7.2 DecoderErrors::bool Operator

Returns if the object is valid.

### C++

operator bool() const;

### Description

Returns true if the object is valid or false if the object is invalid.

# 2.1.14 DecoderStatus Class

### **Inheritance Hierarchy**

### C++

class DecoderStatus;

### File

mvc2api\_decoder.h

### Remarks

These object hold information about the current status of a decoder.

### **Members**

### Methods

	Name	Description
<b>=♦</b> ₩	~DecoderStatus	This is ~DecoderStatus, a member of class DecoderStatus.
<b>≡</b>	DecoderStatus	This is DecoderStatus, a member of class DecoderStatus.
<b>≡</b>	DecoderStatus	This is DecoderStatus, a member of class DecoderStatus.
<b>=</b> ♦	DecoderStatus	This is DecoderStatus, a member of class DecoderStatus.
<b>=</b> ♦	DecoderStatus	This is DecoderStatus, a member of class DecoderStatus.
<b>=♦</b>	DecoderStatus	This is DecoderStatus, a member of class DecoderStatus.
<b>≡</b>	DecoderStatus	This is DecoderStatus, a member of class DecoderStatus.

### **DecoderStatus Methods**

	Name	Description
<b>=♦</b>	getReturnCode	This is getReturnCode, a member of class DecoderStatus.
<b>=</b> ♦	getTimeStamp	This is getTimeStamp, a member of class DecoderStatus.
<b>≡</b>	getUserData	This is getUserData, a member of class DecoderStatus.

### **DecoderStatus Operators**

	Name	Description
<del></del>	=	This is =, a member of class DecoderStatus.

bool Returns if the object is valid.
--------------------------------------

### **DecoderStatus Methods**

	Name	Description
<b>=</b> ♦	getReturnCode	This is getReturnCode, a member of class DecoderStatus.
<b>=</b> ♦	getTimeStamp	This is getTimeStamp, a member of class DecoderStatus.
<b>≡♦</b>	getUserData	This is getUserData, a member of class DecoderStatus.

### **DecoderStatus Operators**

	Name	Description
=+)	=	This is =, a member of class DecoderStatus.
=+)	bool	Returns if the object is valid.

# 2.1.14.1 DecoderStatus::~DecoderStatus Destructor

### C++

```
virtual ~DecoderStatus();
```

### Remarks

This is ~DecoderStatus, a member of class DecoderStatus.

# 2.1.14.2 DecoderStatus::DecoderStatus Constructor ()

### C++

DecoderStatus();

### Remarks

This is DecoderStatus, a member of class DecoderStatus.

# 2.1.14.3 DecoderStatus::DecoderStatus Constructor (DecoderStatus &)

### C++

DecoderStatus(const DecoderStatus & other);

### Remarks

This is DecoderStatus, a member of class DecoderStatus.

# 2.1.14.4 DecoderStatus::DecoderStatus Constructor (MvcDecoderPrivate \*)

### C++

```
DecoderStatus(MvcDecoderPrivate * mvcdec);
```

### Remarks

This is DecoderStatus, a member of class DecoderStatus.

# 2.1.14.5 DecoderStatus::DecoderStatus Constructor (MvcDecoderPrivate \*, DecoderStatus &)

C++

DecoderStatus(MvcDecoderPrivate \* mvcdec, const DecoderStatus & other);

### Remarks

This is DecoderStatus, a member of class DecoderStatus.

# 2.1.14.6 DecoderStatus::DecoderStatus Constructor (MvcDevice &, uint32\_t)

C++

DecoderStatus(const MvcDevice & mvcdev, uint32\_t channel);

### Remarks

This is DecoderStatus, a member of class DecoderStatus.

# 2.1.14.7 DecoderStatus::DecoderStatus Constructor (int32\_t \*, uint32\_t)

C++

DecoderStatus(const int32\_t \* values, uint32\_t size);

### Remarks

This is DecoderStatus, a member of class DecoderStatus.

# 2.1.14.8 DecoderStatus Methods

# 2.1.14.8.1 DecoderStatus: getReturnCode Method



C++

TMmRc getReturnCode() const;

### Remarks

This is getReturnCode, a member of class DecoderStatus.

# 2.1.14.8.2 DecoderStatus: getTimeStamp Method

C++

uint32\_t getTimeStamp() const;

### Remarks

This is getTimeStamp, a member of class DecoderStatus.

# 2.1.14.8.3 DecoderStatus::getUserData Method



C++

```
uint32_t getUserData() const;
```

### Remarks

This is getUserData, a member of class DecoderStatus.

# 2.1.14.9 DecoderStatus Operators

## 2.1.14.9.1 DecoderStatus::= Operator

### C++

```
DecoderStatus & operator =(const DecoderStatus & other);
```

### Remarks

This is =, a member of class DecoderStatus.

## 2.1.14.9.2 DecoderStatus::bool Operator

Returns if the object is valid.

C++

```
operator bool() const;
```

### Description

Returns true if the object is valid or false if the object is invalid.

# 2.1.15 Dolby3Dchromaticity Class

### **Inheritance Hierarchy**

C++

class Dolby3Dchromaticity;

File

mvc2api\_decoder.h

### Remarks

This is class mvc2::Dolby3Dchromaticity.

### Members

### Methods

	Name	Description
<b>=</b> ♦ ₩	~Dolby3Dchromaticity	This is ~Dolby3Dchromaticity, a member of class Dolby3Dchromaticity.
<b>≟</b> ∳	Dolby3Dchromaticity	This is Dolby3Dchromaticity, a member of class Dolby3Dchromaticity.

### **Dolby3Dchromaticity Enumerations**

	Name	Description
<b>3</b>	Measurement	This is record mvc2::Dolby3Dchromaticity::Measurement.

### **Dolby3Dchromaticity Methods**

	Name	Description
<b>=</b> ♦	getChroma	This is getChroma, a member of class Dolby3Dchromaticity.
<b>=♦</b>	getLuminance	This is getLuminance, a member of class Dolby3Dchromaticity.
<b>=</b> •	isCorrectionUsed	This is isCorrectionUsed, a member of class Dolby3Dchromaticity.
<b>=♦</b>	isWPointIterUsed	This is isWPointIterUsed, a member of class Dolby3Dchromaticity.
<b>=♦</b>	setChroma	This is setChroma, a member of class Dolby3Dchromaticity.
=•	setLuminance	This is setLuminance, a member of class Dolby3Dchromaticity.

### **Dolby3Dchromaticity Enumerations**

	Name	Description
<b>a</b> 10	Measurement	This is record mvc2::Dolby3Dchromaticity::Measurement.

### **Dolby3Dchromaticity Methods**

	Name	Description
<b>≡</b>	getChroma	This is getChroma, a member of class Dolby3Dchromaticity.
<b>≡♦</b>	getLuminance	This is getLuminance, a member of class Dolby3Dchromaticity.
<b>≡♦</b>	isCorrectionUsed	This is isCorrectionUsed, a member of class Dolby3Dchromaticity.
<b>≡♦</b>	isWPointIterUsed	This is isWPointIterUsed, a member of class Dolby3Dchromaticity.
<b>≡</b>	setChroma	This is setChroma, a member of class Dolby3Dchromaticity.
<b>≡♦</b>	setLuminance	This is setLuminance, a member of class Dolby3Dchromaticity.

# 2.1.15.1 Dolby3Dchromaticity::~Dolby3Dchromaticity Destructor

### C++

virtual ~Dolby3Dchromaticity();

### Remarks

This is ~Dolby3Dchromaticity, a member of class Dolby3Dchromaticity.

# 2.1.15.2 Dolby3Dchromaticity::Dolby3Dchromaticity Constructor

### C++

Dolby3Dchromaticity();

### Remarks

This is Dolby3Dchromaticity, a member of class Dolby3Dchromaticity.

# 2.1.15.3 Dolby3Dchromaticity Enumerations

58

## 2.1.15.3.1 mvc2::Dolby3Dchromaticity::Measurement Enumeration

### C++

```
enum Measurement {
    Chroma_red = 0,
    Chroma_green = 1,
    Chroma_blue = 2,
    Chroma_white = 4,
    Chroma_white_iter = 5,
    Chroma_red_proj = 6,
    Chroma_green_proj = 7,
    Chroma_blue_proj = 8,
    Chroma_white_proj = 9
};
File

mvc2api_decoder.h
```

### Remarks

This is record mvc2::Dolby3Dchromaticity::Measurement.

# 2.1.15.4 Dolby3Dchromaticity Methods

## 2.1.15.4.1 Dolby3Dchromaticity::getChroma Method

### C++

```
void getChroma(Measurement measurement, double & x, double & y) const;
```

### Remarks

This is getChroma, a member of class Dolby3Dchromaticity.

## 2.1.15.4.2 Dolby3Dchromaticity::getLuminance Method

### C++

```
double getLuminance() const;
```

### Remarks

This is getLuminance, a member of class Dolby3Dchromaticity.

# 2.1.15.4.3 Dolby3Dchromaticity::isCorrectionUsed Method

### C++

```
bool isCorrectionUsed() const;
```

### Remarks

This is isCorrectionUsed, a member of class Dolby3Dchromaticity.

# 2.1.15.4.4 Dolby3Dchromaticity::isWPointIterUsed Method

### C++

```
bool isWPointIterUsed() const;
```

### Remarks

This is isWPointIterUsed, a member of class Dolby3Dchromaticity.

59

# 2.1.15.4.5 Dolby3Dchromaticity::setChroma Method

### C++

void setChroma(Measurement measurement, double x, double y);

### Remarks

This is setChroma, a member of class Dolby3Dchromaticity.

# 2.1.15.4.6 Dolby3Dchromaticity::setLuminance Method

### C++

void setLuminance(double luma);

### Remarks

This is setLuminance, a member of class Dolby3Dchromaticity.

# 2.1.16 ExternalConnector Class

### Inheritance Hierarchy

### C++

class ExternalConnector;

### File

mvc2api\_output.h

### Remarks

This is class mvc2::ExternalConnector.

### Members

### **Methods**

	Name	Description
<b>=</b> ♦ ₩	~ExternalConnector	This is ~ExternalConnector, a member of class ExternalConnector.
<b>≡</b> ♦	ExternalConnector	This is ExternalConnector, a member of class ExternalConnector.
<b>≡</b> ♦	ExternalConnector	This is ExternalConnector, a member of class ExternalConnector.
<b>≡♦</b> •	ExternalConnector	This is ExternalConnector, a member of class ExternalConnector.
<b>≡</b> ♦	ExternalConnector	This is ExternalConnector, a member of class ExternalConnector.

### **ExternalConnector Data Members**

	Name	Description
<b>₽</b> ₽	m_ConnectorPrivate	This is m_ConnectorPrivate, a member of class
		ExternalConnector.

### **ExternalConnector Methods**

	Name	Description
<b>≡</b>	readl2C	This is readl2C, a member of class ExternalConnector.
<b>≡♦</b>	readI2C	This is readl2C, a member of class ExternalConnector.

	■ writeI2C	This is writeI2C, a member of class ExternalConnector.
--	------------	--

### **ExternalConnector Operators**

	Name	Description
<del>(/=</del>	=	This is =, a member of class ExternalConnector.
<u>(/−</u> =+)	bool	Returns if the object is valid.

### **ExternalConnector Data Members**

	Name		Description
49	m_ConnectorPrivate	0,	This is m_ConnectorPrivate, a member of class
			ExternalConnector.

### **ExternalConnector Methods**

	Name	Description
<b>≡♦</b>	readl2C	This is readI2C, a member of class ExternalConnector.
<b>=♦</b>	readI2C	This is readI2C, a member of class ExternalConnector.
<b>=♦</b>	writeI2C	This is writeI2C, a member of class ExternalConnector.

### **ExternalConnector Operators**

	Name	Description
=+)	=	This is =, a member of class ExternalConnector.
=+)	bool	Returns if the object is valid.

## 2.1.16.1 ExternalConnector::~ExternalConnector Destructor

### C++

virtual ~ExternalConnector();

### Remarks

This is ~ExternalConnector, a member of class ExternalConnector.

# 2.1.16.2 ExternalConnector::ExternalConnector Constructor ()

### C++

ExternalConnector();

### Remarks

This is ExternalConnector, a member of class ExternalConnector.

# 2.1.16.3 ExternalConnector::ExternalConnector Constructor (ExternalConnector &)

### C++

ExternalConnector(const ExternalConnector & other);

### Remarks

This is ExternalConnector, a member of class ExternalConnector.

# 2.1.16.4 ExternalConnector::ExternalConnector Constructor (ExternalConnectorPrivate \*)

C++

ExternalConnector(ExternalConnectorPrivate \* dev);

### Remarks

This is ExternalConnector, a member of class ExternalConnector.

# 2.1.16.5 ExternalConnector::ExternalConnector Constructor (TMmRc \*, MvcDevice &)

C++

ExternalConnector(TMmRc \* resultPointer, const MvcDevice & mvcDevice);

### Remarks

This is ExternalConnector, a member of class ExternalConnector.

## 2.1.16.6 ExternalConnector Data Members

### 2.1.16.6.1 ExternalConnector::m ConnectorPrivate Data Member

C++

ExternalConnectorPrivate \* m\_ConnectorPrivate;

### Remarks

This is m\_ConnectorPrivate, a member of class ExternalConnector.

## 2.1.16.7 ExternalConnector Methods

# 2.1.16.7.1 ExternalConnector::readI2C Method (uint16\_t, uint8\_t, uint8\_t \*, uint32\_t)

C++

```
TMmRc readI2C(uint16_t deviceAddr, uint8_t registerAddr, uint8_t * buffer, uint32_t
bufferSize);
```

### Remarks

This is readI2C, a member of class ExternalConnector.

# 2.1.16.7.2 ExternalConnector::readI2C Method (uint16\_t, uint8\_t \*, uint32\_t)

C++

```
TMmRc readI2C(uint16_t deviceAddr, uint8_t * buffer, uint32_t bufferSize);
```

### Remarks

This is readI2C, a member of class ExternalConnector.

### 2.1.16.7.3 ExternalConnector::writel2C Method

### C++

```
TMmRc writeI2C(uint16_t deviceAddr, const uint8_t * buffer, uint32_t bufferSize);
```

### Remarks

This is writel2C, a member of class ExternalConnector.

# 2.1.16.8 ExternalConnector Operators

## 2.1.16.8.1 ExternalConnector::= Operator

### C++

```
ExternalConnector & operator =(const ExternalConnector & other);
```

### Remarks

This is =, a member of class ExternalConnector.

## 2.1.16.8.2 ExternalConnector::bool Operator

Returns if the object is valid.

### C++

operator bool() const;

# 2.1.17 FeatureAccess Class

### **Inheritance Hierarchy**

### C++

class FeatureAccess;

### File

mvc2api\_config.h

### Remarks

This is class mvc2::FeatureAccess.

### **Members**

### Methods

	Name	Description
<b>=</b> ♦ ₩	~FeatureAccess	This is ~FeatureAccess, a member of class FeatureAccess.
<b>=</b> ♦	FeatureAccess	This is FeatureAccess, a member of class FeatureAccess.
<b>=♦</b>	FeatureAccess	This is FeatureAccess, a member of class FeatureAccess.
<b>=♦</b>	FeatureAccess	This is FeatureAccess, a member of class FeatureAccess.

### **FeatureAccess Methods**

	Name	Description
<b>≡♦</b>	getNextFeature	This is getNextFeature, a member of class FeatureAccess.
<b>≡♦</b>	installFeature	This is installFeature, a member of class FeatureAccess.

	<b>≡♦</b>	queryFeature	This is queryFeature, a member of class FeatureAccess.
--	-----------	--------------	--

### **FeatureAccess Operators**

	Name	Description
( <u>/-</u> =+)	=	This is =, a member of class FeatureAccess.
( <u>/-</u> =+)	bool	This is bool, a member of class FeatureAccess.

### **FeatureAccess Methods**

	Name	Description
<b>=♦</b>	getNextFeature	This is getNextFeature, a member of class FeatureAccess.
<b>=♦</b>	installFeature	This is installFeature, a member of class FeatureAccess.
<b>≡∳</b>	queryFeature	This is queryFeature, a member of class FeatureAccess.

### **FeatureAccess Operators**

	Name	Description
=+)	=	This is =, a member of class FeatureAccess.
(/ <u>−</u> =+)	bool	This is bool, a member of class FeatureAccess.

### 2.1.17.1 FeatureAccess::~FeatureAccess Destructor

### C++

virtual ~FeatureAccess();

### Remarks

This is ~FeatureAccess, a member of class FeatureAccess.

# 2.1.17.2 FeatureAccess::FeatureAccess Constructor ()

### C++

FeatureAccess();

### Remarks

This is FeatureAccess, a member of class FeatureAccess.

# 2.1.17.3 FeatureAccess::FeatureAccess Constructor (FeatureAccess&)

C++

FeatureAccess(const FeatureAccess& other);

### Remarks

This is FeatureAccess, a member of class FeatureAccess.

# 2.1.17.4 FeatureAccess::FeatureAccess Constructor (TMmRc \*, MvcDevice &)

### C++

FeatureAccess(TMmRc \* resultPointer, const MvcDevice & mvcDevice);

64

### Remarks

This is FeatureAccess, a member of class FeatureAccess.

# 2.1.17.5 FeatureAccess Methods

# 2.1.17.5.1 FeatureAccess::getNextFeature Method

### C++

```
TMmRc getNextFeature(char * featureString, char * featureValue, uint32_t * validFrom,
uint32_t * validThrough);
```

### Remarks

This is getNextFeature, a member of class FeatureAccess.

### 2.1.17.5.2 FeatureAccess::installFeature Method

### C++

```
TMmRc installFeature(const void * buffer, uint32_t buffersize);
```

### Remarks

This is installFeature, a member of class FeatureAccess.

# 2.1.17.5.3 FeatureAccess::queryFeature Method

### C++

```
TMmRc queryFeature(const char * featureString);
```

### Remarks

This is queryFeature, a member of class FeatureAccess.

# 2.1.17.6 FeatureAccess Operators

# 2.1.17.6.1 FeatureAccess::= Operator

### C++

```
FeatureAccess& operator =(const FeatureAccess& other);
```

### Remarks

This is =, a member of class FeatureAccess.

# 2.1.17.6.2 FeatureAccess::bool Operator

### C++

```
operator bool() const;
```

### Remarks

This is bool, a member of class FeatureAccess.

# 2.1.18 FramePosition Class

Positioning class for an area.

### **Inheritance Hierarchy**

### C++

class FramePosition;

### File

mvc2api\_output.h

### Description

This class is used to specify a position of a rectangular area (e.g. video frame) on screen. It is used by VideoOutput::setPosition().

### **Members**

### Methods

	Name	Description
<b>≡♦</b>	FramePosition	Default constructor.
<b>=</b> ♦	FramePosition	Copy constructor.
<b>≡</b>	FramePosition	Constructor to set a position.

### **FramePosition Enumerations**

	Name	Description
<b>a</b>	MODE	Possible values for offset representation.

### **FramePosition Methods**

	Name	Description
<b>≡⋄</b>	fgetX	Gets x offset as float value.
<b>≡⋄</b>	fgetY	Gets y offset as float value.
<b>≡⋄</b>	fsetX	Sets the x offset as float value.
<b>≡</b>	fsetY	Sets y offset as a float value.
<b>≡</b>	getModeX	Gets mode of x offset.
<b>≡</b>	getModeY	Gets mode of y offset.
<b>≡⋄</b>	getX	Gets x offset.
<b>=</b> ♦	getY	Gets y offset.
<b>=♦</b>	setModeX	Sets mode of x value.
<b>≡♦</b>	setModeY	Sets mode of y offset.
<b>=♦</b>	setX	Sets x offset.
<b>=♦</b>	setY	Sets y offset.

### **FramePosition Operators**

	Name	Description
=+)	!=	Compares two objects.
=+)	=	Assignment operator.
=+)	==	Compares two objects.

### **FramePosition Enumerations**

	Name	Description
<b>.</b>	MODE	Possible values for offset representation.

### **FramePosition Methods**

	Name	Description
<b>=♦</b>	fgetX	Gets x offset as float value.
<b>≡</b>	fgetY	Gets y offset as float value.
<b>≡♦</b>	fsetX	Sets the x offset as float value.
<b>=</b> ♦	fsetY	Sets y offset as a float value.
<b>=</b> ♦	getModeX	Gets mode of x offset.
<b>=♦</b>	getModeY	Gets mode of y offset.
<b>=♦</b>	getX	Gets x offset.
<b>≡</b>	getY	Gets y offset.
<b>=♦</b>	setModeX	Sets mode of x value.
<b>≡</b>	setModeY	Sets mode of y offset.
<b>≡</b>	setX	Sets x offset.
<b>=</b> ♦	setY	Sets y offset.

### **FramePosition Operators**

	Name	Description
=+)	!=	Compares two objects.
=+)	=	Assignment operator.
=+)	==	Compares two objects.

# 2.1.18.1 FramePosition::FramePosition Constructor ()

Default constructor.

C++

FramePosition();

### Description

The default constructor produces a FramePosition object for centering in horizontal and vertical direction.

# 2.1.18.2 FramePosition::FramePosition Constructor (FramePosition &)

Copy constructor.

C++

FramePosition(const FramePosition & other);

### **Parameters**

Parameters	Description
const FramePosition & other	source object to copy information from

### Description

This constructor creates a copy of a source FramePosition object.

# 2.1.18.3 FramePosition::FramePosition Constructor (int16\_t, int16\_t, uint16\_t, uint16\_t)

Constructor to set a position.

### C++

```
FramePosition(int16_t x, int16_t y, uint16_t mode_x = (Mode_Pixels | Mode_Align_Center),
uint16_t mode_y = (Mode_Pixels | Mode_Align_Center));
```

### **Parameters**

Parameters	Description
int16_t x	horizontal position offset as pixels or percent
int16_t y	vertical position offset as pixels or percent
uint16_t mode_x = (Mode_Pixels   Mode_Align_Center)	a combination of FramePosition::MODE bits to specify the horizontal offset
uint16_t mode_y = (Mode_Pixels   Mode_Align_Center)	a combination of FramePosition::MODE bits to specify the vertical offset

### Description

This constructor sets a integer based position.

## 2.1.18.4 FramePosition Enumerations

### 2.1.18.4.1 mvc2::FramePosition::MODE Enumeration

### C++

```
enum MODE {
   Mode_Pixels = (0<<0),
   Mode_Percent = (1<<0),
   Mode_Align_Center = (0<<1),
   Mode_Align_Left = (1<<1),
   Mode_Align_Top = (1<<1),
   Mode_Align_Right = (2<<1),
   Mode_Align_Bottom = (2<<1)
};</pre>
```

### File

mvc2api\_output.h

### Members

Members	Description
Mode_Pixels = (0<<0)	Offset values are pixel values.
Mode_Percent = (1<<0)	Offset values are percent values.
Mode_Align_Center = (0<<1)	Center the frame inside the destination frame plus offset.
Mode_Align_Left = (1<<1)	Align the left edge of the frame at the left edge of the destination frame plus x offset.
Mode_Align_Top = (1<<1)	Align the top edge of the frame at the top edge of the destination frame plus y offset.
Mode_Align_Right = (2<<1)	Align the right edge of the frame at the right edge of the destination frame plus x offset.
Mode_Align_Bottom = (2<<1)	Align the bottom edge of the frame at the bottom edge of the destination frame plus y offset.

### Remarks

Possible values for offset representation.

# 2.1.18.5 FramePosition Methods

# 2.1.18.5.1 FramePosition::fgetX Method

Gets x offset as float value.

### C++

```
float fgetX() const;
```

### **Returns**

Returns the horizontal offset as float point value. See FramePosition::fsetX() for details.

### 2.1.18.5.2 FramePosition::fgetY Method

Gets y offset as float value.

### C++

```
float fgetY() const;
```

### Returns

Returns the vertical offset as float point value. See FramePosition::fsetY() for details.

### 2.1.18.5.3 FramePosition::fsetX Method

Sets the x offset as float value.

### C++

```
void fsetX(float x);
```

### **Parameters**

Parameters	Description
float x	float representation of a offset in percent

### Description

Offset, especially percent values can also have float values. All values are internally represented as fixed value, so the accuracy is not very high. Setting a float value automatically set the mode to FrameMode::Mode\_Percent.

### 2.1.18.5.4 FramePosition::fsetY Method

Sets y offset as a float value.

### C++

```
void fsetY(float y);
```

### **Parameters**

Parameters	Description
float y	float representation of a offset in percent

### Description

Offset, especially percent values can also have float values. All values are internally represented as fixed value, so the

69

accuracy is not very high. Setting a float value automatically set the mode to FrameMode::Mode\_Percent.

# 2.1.18.5.5 FramePosition::getModeX Method

Gets mode of x offset.

### C++

```
uint16_t getModeX() const;
```

### **Returns**

Returns the current mode of horizontal offset. See setModeX() and FramePosition::MODE for details.

# 2.1.18.5.6 FramePosition::getModeY Method

Gets mode of y offset.

### C++

```
uint16_t getModeY() const;
```

### Returns

Returns the current mode of vertical offset. See setModeY() and FramePosition::MODE for details.

## 2.1.18.5.7 FramePosition::getX Method

Gets x offset.

### C++

```
int16_t getX() const;
```

### **Returns**

Returns the current horizontal offset as integer. See setX() for details.

# 2.1.18.5.8 FramePosition::getY Method

Gets y offset.

### C++

```
int16_t getY() const;
```

### Returns

Returns the current vertical offset as integer. See setY() for details.

### 2.1.18.5.9 FramePosition::setModeX Method

Sets mode of x value.

### C++

```
void setModeX(uint16_t mode);
```

### **Parameters**

Parameters	Description
uint16_t mode	a combination of FramePosition::MODE bits

### Description

This method selects the way how the horizontal offset value x should be interpret.

# 2.1.18.5.10 FramePosition::setModeY Method

Sets mode of y offset.

### C++

```
void setModeY(uint16_t mode);
```

### **Parameters**

Parameters	Description
uint16_t mode	a combination of FramePosition::MODE bits

### Description

This method selects the way how the vertical offset value x should be interpret.

### 2.1.18.5.11 FramePosition::setX Method

Sets x offset.

### C++

```
void setX(int16_t x);
```

### Parameters

Parameters	Description
int16_t x	integer offset in pixels or percent

### Description

Set the horizontal offset value as integer. The value represents a pixel offset or a percent value depending on ModeX setup.

### 2.1.18.5.12 FramePosition::setY Method

Sets y offset.

### C++

```
void setY(int16_t y);
```

### **Parameters**

Parameters	Description
int16_t y	offset in pixels or percent

### Description

Set the vertical offset value as integer. The value represents a pixel offset or a percent value depending on ModeY setup.

# 2.1.18.6 FramePosition Operators

# 2.1.18.6.1 FramePosition::!= Operator

Compares two objects.

### C++

```
bool operator !=(const FramePosition & other) const;
```

### **Parameters**

Parameters	Description
const FramePosition & other	source object

### Returns

Returns false of both objects are the same, otherwise true.

### Description

This operator compares a source object with the current one.

# 2.1.18.6.2 FramePosition::= Operator

Assignment operator.

### C++

FramePosition & operator =(const FramePosition & other);

### **Parameters**

Parameters	Description
const FramePosition & other	source object

### Returns

Returns a reference to the current object.

### Description

This operator copies all values from a source object to the current one.

## 2.1.18.6.3 FramePosition::== Operator

Compares two objects.

### C++

bool operator ==(const FramePosition & other) const;

### **Parameters**

Parameters	Description
const FramePosition & other	source object

### Returns

Returns true if both objects are the same, otherwise false.

### Description

This operator compares a source object with the current one.

# 2.1.19 Jpeg2kDecoder Class

JPEG 2000 decoder class.

### **Inheritance Hierarchy**

### C++

class Jpeg2kDecoder : public VideoDecoder;

### File

mvc2api\_decoder.h

### Description

This class will create a JPEG 2000 video decoder on the MVC card and establish a virtual transfer channel.

### Remarks

### **Members**

### Methods

	Name	Description
<b>=</b> ♦ ₩	~MvcDecoder	This is ~MvcDecoder, a member of class MvcDecoder.
<b>=</b> ♦	MvcDecoder	This is MvcDecoder, a member of class MvcDecoder.
<b>=</b> ♦	MvcDecoder	This is MvcDecoder, a member of class MvcDecoder.
<b>=♦</b> <sub><b>?</b></sub>	MvcDecoder	This is MvcDecoder, a member of class MvcDecoder.

### **VideoDecoder Class**

	Name	Description
<b>=Q</b> <sub><b>Q</b></sub>	VideoDecoder	This is VideoDecoder, a member of class VideoDecoder.
<b>=Q</b> <sub><b>Q</b></sub>	VideoDecoder	This is VideoDecoder, a member of class VideoDecoder.

### Jpeg2kDecoder Class

	Name	Description
<b>=</b> ♦ <b>W</b>	~Jpeg2kDecoder	Default destructor.
<b>≡♦</b>	Jpeg2kDecoder	Creates an empty (invalid) Jpeg 2000 Decoder object.
<b>≡♦</b>	Jpeg2kDecoder	Creates an empty (invalid) Jpeg 2000 Decoder object.
<b>≡♦</b>	Jpeg2kDecoder	Creates a linked Jpeg2k decoder for 3D playback.
<b>≡♦</b>	Jpeg2kDecoder	This method will create a JPEG 2000 decoder object.

### **MvcDecoder Enumerations**

	Name	Description
a <sup>a</sup>	DecoderType	This is record mvc2::MvcDecoder::DecoderType.
a <sup>a</sup>	TIMEOUT	This is record mvc2::MvcDecoder::TIMEOUT.

### Jpeg2kDecoder Class

	Name	Description
<b>a</b>	CREATION_FLAGS	This is record mvc2::Jpeg2kDecoder::CREATION_FLAGS.
<b>a</b>	MODE_4K	This is record mvc2::Jpeg2kDecoder::MODE_4K.

### **MvcDecoder Methods**

	Name	Description
<b>≡</b>	flush	This is flush, a member of class MvcDecoder.
<b>≡</b>	getBufferStatus	This is getBufferStatus, a member of class MvcDecoder.
<b>≡</b>	getChannel	Gets the used channel number.
<b>■♦</b>	getDataBuffer	Return a buffer for data transfer.
<b>≡</b>	getDecoderErrors	This is getDecoderErrors, a member of class MvcDecoder.
<b>≡</b> •	getDecoderStatus	This is getDecoderStatus, a member of class MvcDecoder.
<b>≡</b>	getDecoderType	This is getDecoderType, a member of class MvcDecoder.
<b>≡</b> ∳ş	getPrivate	This is getPrivate, a member of class MvcDecoder.
<b>≓</b> ♦	setBackwardPlayback	This is setBackwardPlayback, a member of class MvcDecoder.

<b>=</b> ♦	setCplUid	This is setCplUid, a member of class MvcDecoder.
<b>=</b> ♦	setEndOfStream	Go to end of stream state.
<b>=</b> ♦	setSecurityManager	This is setSecurityManager, a member of class MvcDecoder.
<b>=</b> ♦	setStartDelay	Setup start delay for the decoder.
<b>=♦</b>	waitDecoderStatus	This is waitDecoderStatus, a member of class MvcDecoder.
<b>=</b> ♦	waitForTimeStamp	This is waitForTimeStamp, a member of class MvcDecoder.
<b>≟♦</b>	waitForTransferFinish	This is waitForTransferFinish, a member of class MvcDecoder.
<b>=♦</b>	waitForUserData	This is waitForUserData, a member of class MvcDecoder.

### VideoDecoder Class

	Name	Description
<b>≡♦</b>	calcDolby3DMatrix	This is calcDolby3DMatrix, a member of class VideoDecoder.
<b>=</b> ♦	connectOutput	Connects an output object to the decoder.
<b>=</b> ♦	disconnectOutput	Disconnects the output.
<b>=</b> ♦	getDecoderErrors	This is getDecoderErrors, a member of class VideoDecoder.
<b>≡♦</b>	getDecoderStatus	This is getDecoderStatus, a member of class VideoDecoder.
<b>≡♦</b>	getVideoDataBuffer	This is getVideoDataBuffer, a member of class VideoDecoder.
<b>=</b> ♦	setCloneVideo	This is setCloneVideo, a member of class VideoDecoder.
<b>=</b> ♦	setDolby3DChroma	This is setDolby3DChroma, a member of class VideoDecoder.
<b>≡</b>	setDolby3DMatrix	This is setDolby3DMatrix, a member of class VideoDecoder.
<b>≡</b>	waitDecoderStatus	This is waitDecoderStatus, a member of class VideoDecoder.

### Jpeg2kDecoder Class

	Name	Description
<b>=♦</b>	getFrameRate	Gets the frame rate.
<b>≡</b>	getFrameRateTicks	Gets frame rate ticks.
<b>≡</b>	set4kMode	This is set4kMode, a member of class Jpeg2kDecoder.
<b>≡</b>	setColorSpace	This is setColorSpace, a member of class Jpeg2kDecoder.
<b>≡</b>	setCutOffLevel	This is setCutOffLevel, a member of class Jpeg2kDecoder.
<b>≡</b>	setFrameRate	Sets the frame rate.
<b>≡</b>	setFrameRateTicks	Sets the frame rate in ticks.
<b>≡</b>	setResolutionDivider	This is setResolutionDivider, a member of class Jpeg2kDecoder.

### **MvcDecoder Operators**

	Name	Description
=+)	=	This is =, a member of class MvcDecoder.
<u>(/-</u> =+)	bool	Returns if the object is valid.

### Jpeg2kDecoder Class

	Name	Description
/ <del>-</del> =+)	=	This is =, a member of class Jpeg2kDecoder.

### **MvcDecoder Enumerations**

	Name	Description
<b>a</b>	DecoderType	This is record mvc2::MvcDecoder::DecoderType.
<b>a</b>	TIMEOUT	This is record mvc2::MvcDecoder::TIMEOUT.

### Jpeg2kDecoder Class

	Name	Description
<u>_</u>	CREATION_FLAGS	This is record mvc2::Jpeg2kDecoder::CREATION_FLAGS.
<b>.</b>	MODE_4K	This is record mvc2::Jpeg2kDecoder::MODE_4K.

### **MvcDecoder Methods**

	Name	Description
<b>=♦</b>	flush	This is flush, a member of class MvcDecoder.
<b>≡♦</b>	getBufferStatus	This is getBufferStatus, a member of class MvcDecoder.
<b>≡</b>	getChannel	Gets the used channel number.
<b>=</b> ♦ ₩	getDataBuffer	Return a buffer for data transfer.
<b>≡♦</b>	getDecoderErrors	This is getDecoderErrors, a member of class MvcDecoder.
<b>≡♦</b>	getDecoderStatus	This is getDecoderStatus, a member of class MvcDecoder.
<b>≡</b>	getDecoderType	This is getDecoderType, a member of class MvcDecoder.
<b>=\$</b> <sub><b>9</b></sub>	getPrivate	This is getPrivate, a member of class MvcDecoder.
<b>=</b>	setBackwardPlayback	This is setBackwardPlayback, a member of class MvcDecoder.
<b>≡♦</b>	setCplUid	This is setCplUid, a member of class MvcDecoder.
<b>≡</b>	setEndOfStream	Go to end of stream state.
<b>≡</b>	setSecurityManager	This is setSecurityManager, a member of class MvcDecoder.
<b>≡♦</b>	setStartDelay	Setup start delay for the decoder.
<b>≡</b>	waitDecoderStatus	This is waitDecoderStatus, a member of class MvcDecoder.
<b>≡</b>	waitForTimeStamp	This is waitForTimeStamp, a member of class MvcDecoder.
<b>≓∳</b>	waitForTransferFinish	This is waitForTransferFinish, a member of class MvcDecoder.
<b>≡</b>	waitForUserData	This is waitForUserData, a member of class MvcDecoder.

### VideoDecoder Class

	Name	Description
<b>=♦</b>	calcDolby3DMatrix	This is calcDolby3DMatrix, a member of class VideoDecoder.
<b>=♦</b>	connectOutput	Connects an output object to the decoder.
<b>=♦</b>	disconnectOutput	Disconnects the output.
<b>=♦</b>	getDecoderErrors	This is getDecoderErrors, a member of class VideoDecoder.
<b>=♦</b>	getDecoderStatus	This is getDecoderStatus, a member of class VideoDecoder.
<b>≡∳</b>	getVideoDataBuffer	This is getVideoDataBuffer, a member of class VideoDecoder.
<b>=♦</b>	setCloneVideo	This is setCloneVideo, a member of class VideoDecoder.
<b>∉</b> ∳	setDolby3DChroma	This is setDolby3DChroma, a member of class VideoDecoder.
<b>=♦</b>	setDolby3DMatrix	This is setDolby3DMatrix, a member of class VideoDecoder.
<b>=♦</b>	waitDecoderStatus	This is waitDecoderStatus, a member of class VideoDecoder.

### Jpeg2kDecoder Class

	Name	Description
<b>=</b> ♦	getFrameRate	Gets the frame rate.
<b>=</b> ♦	getFrameRateTicks	Gets frame rate ticks.
<b>=♦</b>	set4kMode	This is set4kMode, a member of class Jpeg2kDecoder.
<b>=♦</b>	setColorSpace	This is setColorSpace, a member of class Jpeg2kDecoder.
<b>≡♦</b>	setCutOffLevel	This is setCutOffLevel, a member of class Jpeg2kDecoder.
<b>=♦</b>	setFrameRate	Sets the frame rate.
<b>≡</b>	setFrameRateTicks	Sets the frame rate in ticks.

<b>=♦</b>	setResolutionDivider	This is setResolutionDivider, a member of class
		Jpeg2kDecoder.

### **MvcDecoder Operators**

	Name	Description
=+)	=	This is =, a member of class MvcDecoder.
=+0	bool	Returns if the object is valid.

### Jpeg2kDecoder Class

	Name	Description
=+)	-	This is =, a member of class Jpeg2kDecoder.

# 2.1.19.1 Jpeg2kDecoder::~Jpeg2kDecoder Destructor

Default destructor.

### C++

```
virtual ~Jpeg2kDecoder();
```

### Description

The default destructor of the Jpeg2kDecoder will release the virtual transfer channel and all other resources allocated by the decoder. The decoder will also detached from the PlaybackControl.

# 2.1.19.2 Jpeg2kDecoder::Jpeg2kDecoder Constructor ()

Creates an empty (invalid) Jpeg 2000 Decoder object.

### C++

```
Jpeg2kDecoder();
Jpeg2kDecoder(const Jpeg2kDecoder& other);
```

### Description

Creates an empty (invalid) Jpeg 2000 Decoder object.

# 2.1.19.3 Jpeg2kDecoder::Jpeg2kDecoder Constructor (TMmRc \*, Jpeg2kDecoder &)

Creates a linked Jpeg2k decoder for 3D playback.

### C++

```
Jpeg2kDecoder(TMmRc * resultPointer, const Jpeg2kDecoder & linkedDecoder);
```

### **Parameters**

Parameters	Description
TMmRc * resultPointer	error code from the object creation
const Jpeg2kDecoder & linkedDecoder	Jpeg2k decoder to link the new decoder to

### Description

This constructor creates a secondary decoder to use with 3D playback.

# 2.1.19.4 Jpeg2kDecoder::Jpeg2kDecoder Constructor (TMmRc \*, MvcDevice &, uint32\_t)

This method will create a JPEG 2000 decoder object.

### C++

```
Jpeg2kDecoder(TMmRc * resultPointer, const MvcDevice & mvcDevice, uint32_t creationFlags =
0);
```

### **Parameters**

Parameters	Description
TMmRc * resultPointer	Pointer where to store the result in. if 0 no result will be written.
const MvcDevice & mvcDevice	MvcDevice object to which the decoder will be attached.
uint32_t creationFlags = 0	specify which capabilities this decoder should have

### Description

This method will create a JPEG 2000 decoder object and allocate the virtual transfer channel for it.

The decoder could be created the object will be valid the resultPointers value is set to MMRC\_Ok. If not the object is set to an invalid state and the resultPointers value contains the error code. The default decoder is a 2K decoder which can decode up to 60p if running in 2D mode (not linked). Using the createFlags a 4K and high frame rate decoder can created.

# 2.1.19.5 Jpeg2kDecoder Enumerations

## 2.1.19.5.1 mvc2::Jpeg2kDecoder::CREATION\_FLAGS Enumeration

### C++

```
enum CREATION_FLAGS {
   Create_4K_Decoder = (1<<0),
   Create_HighFrameRate = (1<<1)
};</pre>
```

### File

mvc2api\_decoder.h

### Members

Members	Description
Create_4K_Decoder = (1<<0)	4K decoding capability
Create_HighFrameRate = (1<<1)	need support for high frame rates, beyond 60p 2D or 30p 3D

### Remarks

This is record mvc2::Jpeg2kDecoder::CREATION\_FLAGS.

# 2.1.19.5.2 mvc2::Jpeg2kDecoder::MODE\_4K Enumeration

### C++

```
enum MODE_4K {
   MODE_4K_Off = 0,
   MODE_4K_Automatic = 1,
   MODE_4K_Always = 2,
   MODE_4K_DropScale = 3
};
```

### File

mvc2api\_decoder.h

### **Members**

Members	Description
MODE_4K_Off = 0	only decode 2K resolution
MODE_4K_Automatic = 1	automatically switch to 4K depending on the video material
MODE_4K_Always = 2	upscale any resolution to 4K

### Remarks

This is record mvc2::Jpeg2kDecoder::MODE\_4K.

# 2.1.19.6 Jpeg2kDecoder Methods

# 2.1.19.6.1 Jpeg2kDecoder::getFrameRate Method



Gets the frame rate.

### C++

```
float getFrameRate() const;
```

### Returns

Frames per second or zero of the frame rate was setup.

### Description

This method returns the current frame rate based on frames per second.

# 2.1.19.6.2 Jpeg2kDecoder::getFrameRateTicks Method

Gets frame rate ticks.

### C++

```
uint32_t getFrameRateTicks() const;
```

### Returns

Frame rate in 27 MHz ticks or zero if no frame rate was setup.

### Description

This method returns the current frame rate based on 27 MHz ticks. For example 27000000 means 10 frames per second.

# 2.1.19.6.3 Jpeg2kDecoder::set4kMode Method



### C++

```
TMmRc set4kMode(MODE_4K mode);
```

### Remarks

This is set4kMode, a member of class Jpeg2kDecoder.

# 2.1.19.6.4 Jpeg2kDecoder::setColorSpace Method



### C++

TMmRc setColorSpace(const ColorSpace & space);

### Remarks

This is setColorSpace, a member of class Jpeg2kDecoder.

# 2.1.19.6.5 Jpeg2kDecoder::setCutOffLevel Method



C++

TMmRc setCutOffLevel(uint8\_t cutoff);

### Remarks

This is setCutOffLevel, a member of class Jpeg2kDecoder.

## 2.1.19.6.6 Jpeg2kDecoder::setFrameRate Method

Sets the frame rate.



C++

void setFrameRate(float rate);

### **Parameters**

Parameters	Description
float rate	frame rate in frames per second

### Description

The decoder has the ability to automatically calculate the time stamps of the DataBuffers if it has the information about the frame rate. It will then calculate the new time stamp if getDataBuffer() is called based on the time stamp of the last DataBuffer::send() call. Internally the frame rate and time stamp calculation is done using 27 MHz ticks, which means there will be done some rounding of the frame rate value. The methods getFrameRate() and getFrameRateTicks() should be used to verify the used frame rate.

# 2.1.19.6.7 Jpeg2kDecoder: setFrameRateTicks Method

Sets the frame rate in ticks.

### C++

void setFrameRateTicks(uint32\_t ticks);

### **Parameters**

Parameters	Description
uint32_t ticks	Frame rate in 27 MHz ticks (e.g. 1125000 means 24 frames per second)

### Description

The decoder has the ability to automatically calculate the time stamps of the DataBuffers if it has the information about the frame rate. It will then calculate the new time stamp if getDataBuffer() is called based on the time stamp of the last DataBuffer::send() call.

# 2.1.19.6.8 Jpeg2kDecoder::setResolutionDivider Method

### C++

TMmRc setResolutionDivider(uint8\_t div);

### Remarks

This is setResolutionDivider, a member of class Jpeg2kDecoder.

# 2.1.19.7 Jpeg2kDecoder Operators

# 2.1.19.7.1 Jpeg2kDecoder::= Operator

### C++

Jpeg2kDecoder& operator =(const Jpeg2kDecoder& other);

### Remarks

This is =, a member of class Jpeg2kDecoder.

# 2.1.20 LogAccess Class

### **Inheritance Hierarchy**

C++

class LogAccess;

### File

mvc2api.h

### Remarks

This class is used to access the debug log of the MVC card.

### **Members**

### Methods

	Name	Description
<b>=</b> ♦ ₩	~LogAccess	This is ~LogAccess, a member of class LogAccess.
<b>=</b> ♦	LogAccess	This is LogAccess, a member of class LogAccess.
<b>≟</b> ♦	LogAccess	This is LogAccess, a member of class LogAccess.
<b>≟</b> ♦	LogAccess	Create LogAccess object.

### **LogAccess Enumerations**

	Name	Description
<b>a</b>	TIMEOUT	This is record mvc2::LogAccess::TIMEOUT.

### **LogAccess Methods**

	Name	Description
<b>=</b> ♦	abortGetMessage	Aborts a pending getMessage() operation.
<b>=</b> ♦	getMessage	Get a log message.
<b>=</b> ♦	setMessageFilter	Set the message filter.

### **LogAccess Operators**

	Name	Description
=+)	=	This is =, a member of class LogAccess.
=+)	bool	Returns if the object is valid.

### **LogAccess Enumerations**

	Name	Description
a P	TIMEOUT	This is record mvc2::LogAccess::TIMEOUT.

80

### **LogAccess Methods**

	Name	Description
<b>≡</b>	abortGetMessage	Aborts a pending getMessage() operation.
<b>≡⋄</b>	getMessage	Get a log message.
<b>≡</b>	setMessageFilter	Set the message filter.

### **LogAccess Operators**

	Name		Description
<del>(/=</del>	=	0,2	This is =, a member of class LogAccess.
<u>(/−</u> =+)	bool		Returns if the object is valid.

# 2.1.20.1 LogAccess::~LogAccess Destructor

### C++

```
virtual ~LogAccess();
```

### Remarks

This is ~LogAccess, a member of class LogAccess.

# 2.1.20.2 LogAccess::LogAccess Constructor ()

### C++

LogAccess();

### Remarks

This is LogAccess, a member of class LogAccess.

# 2.1.20.3 LogAccess::LogAccess Constructor (LogAccess&)

### C++

```
LogAccess(const LogAccess& other);
```

### Remarks

This is LogAccess, a member of class LogAccess.

# 2.1.20.4 LogAccess::LogAccess Constructor (TMmRc \*, MvcDevice &, bool)

Create LogAccess object.

### C++

LogAccess(TMmRc \* resultPointer, const MvcDevice & mvcdev, bool ignoreExistingMessages);

### **Parameters**

Parameters	Description
TMmRc * resultPointer	Pointer where to store the result in. if 0 no result will be written.
const MvcDevice & mvcdev	MvcDevice object to create a LogAccess object to
bool ignoreExistingMessages	ignore all existing messages the log buffer and output only new ones

### **Returns**

MMRC\_Ok for success, else an error

### **Return Values**

Return Values	Description
MMRC_Ok	object creation successful
MMRC_MVC2_DeviceBusy	MvcDevice is busy and cannot be accessed

### Description

This method creates a LogAccess object.

# 2.1.20.5 LogAccess Enumerations

## 2.1.20.5.1 mvc2::LogAccess::TIMEOUT Enumeration

### C++

```
enum TIMEOUT {
   WaitInfinite = 0xffffffff,
   DonotWait = 0
};
```

### File

mvc2api.h

### **Members**

Members	Description
WaitInfinite = 0xffffffff	Wait forever.
DonotWait = 0	Do not wait. Return function immediately.

### Remarks

This is record mvc2::LogAccess::TIMEOUT.

# 2.1.20.6 LogAccess Methods

# 2.1.20.6.1 LogAccess::abortGetMessage Method

Aborts a pending getMessage() operation.

### C++

```
void abortGetMessage();
```

### **Description**

The getMessage() method will wait for an incoming message. There may situations where this waiting should be aborted (e.g. to close a application). The getMessage() method will quickly return with a new message or a timeout.

# 2.1.20.6.2 LogAccess::getMessage Method

Get a log message.

### C++

```
TMmRc getMessage(LogMessage & msg, bool * moreMessagesAvailable, uint32_t timeout =
WaitInfinite);
```

82

### **Parameters**

Parameters	Description
LogMessage & msg	LogMessage object which receives the message
bool * moreMessagesAvailable	true, if there are messages waiting, else false
uint32_t timeout = WaitInfinite	timeout in milliseconds to wait for a new message

### **Returns**

MMRC\_Ok or an error.

### **Return Values**

Return Values	Description
MMRC_Ok	log message received
MMRC_Timeout	timeout occurred

### **Description**

This method will return the currently available log message or wait for a new message to come in.

# 2.1.20.6.3 LogAccess::setMessageFilter Method

Set the message filter.

### C++

void setMessageFilter(uint32\_t severityMask, uint32\_t idMask, uint32\_t subIdValue, uint32\_t
subIdMask);

### **Parameters**

Parameters	Description
uint32_t severityMask	wanted severity levels
uint32_t idMask	wanted id's
uint32_t subIdValue	wanted sub id's
uint32_t subIdMask	mask for the wanted sub id's

### **Description**

The log messages can be filtered before they will be returned, which makes the log access much more efficient. By default all message will be return without any filter.

# 2.1.20.7 LogAccess Operators

# 2.1.20.7.1 LogAccess::= Operator

### C++

LogAccess& operator =(const LogAccess& other);

### Remarks

This is =, a member of class LogAccess.

# 2.1.20.7.2 LogAccess::bool Operator

Returns if the object is valid.

### C++

operator bool() const;

### Description

Returns true if the object is valid or false if the object is invalid.

# 2.1.21 LogMessage Class Inheritance Hierarchy

class LogMessage;

### File

mvc2api.h

### Remarks

Log Message class

### **Members**

### Methods

	Name	Description
<b>=</b> ♦	~LogMessage	Default destructor.
<b>≡♦</b>	LogMessage	LogMessage copy constructor.
<b>=</b> ♦	LogMessage	specific constructor
<b>≡</b> ♦	LogMessage	default constructor

### **LogMessage Enumerations**

	Name	Description
<b>a</b>	LogId	Identifier for the source of the LogMessage.
<b></b>	LogSubIdFirmware	The sub id identifies a specific source of the LogMessage for messages with LogId set to IdFirmware. The bits are used as a mask, e.g. messages related to a video output will have bit0 and bit3 set.
	Severity	Severity level of the LogMessage.

### **LogMessage Methods**

	Name	Description
<b>≡</b>	getCTime	Return the time stamp as ANSI-C time.
<b>≡</b>	getId	Return the message id.
<b>≡</b>	getMessageCounter	Return the message counter.
<b>≡</b>	getMessageString	Return message string.
<b>≡</b>	getMessageStringLength	Return valid message bytes.
<b>≡</b>	getMicroSeconds	Returns microseconds of the message.
<b>≡</b>	getSeverity	Return the severity level.
<b>≡</b>	getSubId	Return sub id.
<b>≡</b>	getTimeStamp	Return timestamp.
<b>≡</b>	setMessage	Modify the message.

### **LogMessage Operators**

	Name	Description
<u>(/-</u> =+)	=	Assignment operator
(/ <del>_</del> =+)	bool	This is bool, a member of class LogMessage.

### LogMessage Friends

Name	Description
class LogAccess	This is friend friend class LogAccess.

### **LogMessage Enumerations**

	Name	Description	
<b>a</b>	LogId	Identifier for the source of the LogMessage.	
	LogSubIdFirmware	The sub id identifies a specific source of the LogMessage messages with LogId set to IdFirmware. The bits are used a mask, e.g. messages related to a video output will have bit0 and bit3 set.	d as
<b>a</b>	Severity	Severity level of the LogMessage.	

### LogMessage Friends

Name	Description
class LogAccess	This is friend friend class LogAccess.

### **LogMessage Methods**

	Name	Description
<b>=♦</b>	getCTime	Return the time stamp as ANSI-C time.
<b>=♦</b>	getld	Return the message id.
<b>≡</b>	getMessageCounter	Return the message counter.
<b>≡</b>	getMessageString	Return message string.
<b>=</b>	getMessageStringLength	Return valid message bytes.
<b>≡</b>	getMicroSeconds	Returns microseconds of the message.
<b>≡</b>	getSeverity	Return the severity level.
<b>≡</b>	getSubId	Return sub id.
<b>≡</b>	getTimeStamp	Return timestamp.
<b>=♦</b>	setMessage	Modify the message.

### **LogMessage Operators**

	Name	Description
=+)	=	Assignment operator
=+)	bool	This is bool, a member of class LogMessage.

# 2.1.21.1 LogMessage::~LogMessage Destructor

Default destructor.

C++

~LogMessage();

### Description

The destructor frees all resources of the LogMessage.

# 2.1.21.2 LogMessage::LogMessage Constructor (LogMessage&)

LogMessage copy constructor.

C++

LogMessage(const LogMessage& other);

### **Parameters**

Parameters	Description
msg	source LogMessage

### Description

This constructor creates a new LogMessage as exact copy of the source LogMessage.

# 2.1.21.3 LogMessage::LogMessage Constructor (Severity, LogId, uint32\_t, uint32\_t, uint64\_t, char \*, uint32\_t)

specific constructor

### C++

```
LogMessage(const Severity sev, const LogId id, const uint32_t subid, const uint32_t
counter, const uint64_t timestamp, const char * msgstr, const uint32_t msgsize);
```

### **Parameters**

Parameters	Description
const Severity sev	Severity level
const LogId id	ld
const uint32_t subid	SubId
const uint32_t counter	Message counter
const uint64_t timestamp	Time stamp in milliseconds
const char * msgstr	pointer to the message string (zero terminated)
const uint32_t msgsize	number of bytes of the message string

### Description

This constructor creates a complete LogMessage object.

### 2.1.21.4 LogMessage::LogMessage Constructor (uint32\_t)

default constructor

C++

```
LogMessage(uint32_t preallocatedSize = 0);
```

### Description

The constructor create a empty LogMessage object. If (preallocatedSize > 0) a message buffer gets allocated already. This can be used to avoid multiple free/allocates during message manipulation.

## 2.1.21.5 LogMessage Enumerations

### 2.1.21.5.1 mvc2::LogMessage::LogId Enumeration

### C++

```
enum LogId {
   IdNone = 0,
   IdApi = (1<<0),
   IdDriver = (1<<1),
   IdSecurityManager = (1<<2),
   IdFirmware = (1<<3)</pre>
```

```
};
```

File

mvc2api.h

### **Members**

Members	Description
IdNone = 0	LogMessage is empty and therefore it doesn't have an identifier.
IdApi = (1<<0)	Message create by MVC API.
IdDriver = (1<<1)	Message create by MVC driver.
IdSecurityManager = (1<<2)	Message create by the security manager on the MVC card.
IdFirmware = (1<<3)	Message create by the MVC card's firmware.

### Remarks

Identifier for the source of the LogMessage.

### 2.1.21.5.2 mvc2::LogMessage::LogSubIdFirmware Enumeration

### C++

```
enum LogSubIdFirmware {
   SubIdVideo = (1<<0),
   SubIdAudio = (1<<1),
   SubIdDecoding = (1<<2),
   SubIdOutput = (1<<3),
   SubIdControl = (1<<4),
   SubIdSecurity = (1<<5),
   SubIdNetwork = (1<<6)
};</pre>
```

### File

mvc2api.h

### **Members**

Members	Description
SubIdVideo = (1<<0)	LogMessage is related to video.
SubIdAudio = (1<<1)	LogMessage is related to audio.
SubIdDecoding = (1<<2)	LogMessage is related to decoding processes.
SubIdOutput = (1<<3)	LogMessage is related to outputs.
SubIdControl = (1<<4)	LogMessage is related to controlling.
SubIdSecurity = (1<<5)	LogMessage is related to security.
SubIdNetwork = (1<<6)	LogMessage is related to networking.

### Remarks

The sub id identifies a specific source of the LogMessage for messages with LogId set to IdFirmware. The bits are used as a mask, e.g. messages related to a video output will have bit0 and bit3 set.

### 2.1.21.5.3 mvc2::LogMessage::Severity Enumeration

### C++

```
enum Severity {
   SeverityNone = 0,
   SeverityTrace = (1<<0),
   SeverityInfo = (1<<1),
   SeverityWarning = (1<<2),
   SeverityError = (1<<3)
};</pre>
```

### File

mvc2api.h

### **Members**

Members	Description
SeverityNone = 0	LogMessage is empty and therefore it doesn't have a severity level.
SeverityTrace = (1<<0)	LogMessage is a trace message.
SeverityInfo = (1<<1)	LogMessage is an information.
SeverityWarning = (1<<2)	LogMessage is a warning.
SeverityError = (1<<3)	LogMessage is an error.

### Remarks

Severity level of the LogMessage.

# 2.1.21.6 LogMessage Methods

### 2.1.21.6.1 LogMessage::getCTime Method

Return the time stamp as ANSI-C time.

### C++

```
time_t getCTime() const;
```

### Returns

Number of seconds starting from 1.1.1970 of the Log message.

### Description

This method returns the time of the message as time\_t type to be used with ANSI-C time functions. The time is a counter of seconds starting from 1.1.1970.

### 2.1.21.6.2 LogMessage::getId Method

Return the message id.

### C++

```
LogId getId() const;
```

### Returns

A LogId or IdNone if the message is empty.

### Description

Returns the Id of the log message.

# 2.1.21.6.3 LogMessage::getMessageCounter Method

Return the message counter.

### C++

```
uint32_t getMessageCounter() const;
```

### Returns

The message number or zero for a empty log message.

### Description

Gets the message counter of the current log message.

### 2.1.21.6.4 LogMessage::getMessageString Method

Return message string.

### C++

```
char * getMessageString() const;
```

### **Returns**

Pointer to a string or NULL for a empty log message.

### Description

Gets the pointer to the internal string buffer with the message string. This string is valid as long as the object exists and it's not assigned from another LogMessage object.

### 2.1.21.6.5 LogMessage::getMessageStringLength Method

Return valid message bytes.

### C++

```
uint32_t getMessageStringLength() const;
```

#### Returns

Valid message bytes, or zero for a empty log message.

### Description

Gets the valid message bytes contained at address obtained by getMessageString(). Termination bytes are not included.

### 2.1.21.6.6 LogMessage::getMicroSeconds Method

Returns microseconds of the message.

### C++

```
uint32_t getMicroSeconds() const;
```

### Returns

Number of microseconds of the message.

### Description

Get the number of microseconds of the log message. This is useful in conjunction with getCTime().

### 2.1.21.6.7 LogMessage::getSeverity Method

Return the severity level.

### C++

```
Severity getSeverity() const;
```

### Returns

Severity level of SeverityNone for an empty log message.

### Description

Gets the severity level of the log message.

### 2.1.21.6.8 LogMessage::getSubId Method

Return sub id.

### C++

```
uint32_t getSubId() const;
```

### **Returns**

Sub id of the message.

### Description

Gets the sub id of the log message.

### 2.1.21.6.9 LogMessage::getTimeStamp Method

Return timestamp.

### C++

```
uint64_t getTimeStamp() const;
```

### Returns

Time of the message or zero for an empty log message.

### Description

Gets the time stamp of the message. The time is counted in microseconds starting from 1.1.1970. Time zone is UTC.

### 2.1.21.6.10 LogMessage::setMessage Method

Modify the message.

### C++

```
void setMessage(const Severity sev, const LogId id, const uint32_t subid, const uint32_t
counter, const uint64_t timestamp, const char * message, uint32_t messageSize);
```

### **Parameters**

Parameters	Description
const Severity sev	severity level
const LogId id	id
const uint32_t subid	sub id
const uint32_t counter	message counter
const uint64_t timestamp	time stamp
const char * message	pointer to the message string
uint32_t messageSize	size in bytes of the message string

### Description

Sets the message information. The message string will be copied to the LogMessage object.

# 2.1.21.7 LogMessage Operators

### 2.1.21.7.1 LogMessage::= Operator

Assignment operator

### C++

LogMessage & operator = (const LogMessage & other)

### **Parameters**

Parameters	Description
msg	input LogMessage object

### Returns

True if the LogMessage is not empty after the copy.

### Description

This operator copies all information from one LogMessage object to another.

### 2.1.21.7.2 LogMessage::bool Operator

### C++

```
operator bool() const;
```

### Remarks

This is bool, a member of class LogMessage.

### 2.1.21.8 LogMessage Friends

### 2.1.21.8.1 friend class LogAccess Friend

### C++

friend class LogAccess;

### Remarks

This is friend friend class LogAccess.

# 2.1.22 Mpeg2Decoder Class

MPEG-2 decoder class.

### **Inheritance Hierarchy**

### C++

```
class Mpeg2Decoder : public VideoDecoder;
```

### File

mvc2api\_decoder.h

### **Description**

This class will create a MPEG-2 video decoder on the MVC card and establish a virtual transfer channel.

### Remarks

### Members

### Methods

	Name	Description
<b>=</b> ♦ <b>W</b>	~MvcDecoder	This is ~MvcDecoder, a member of class MvcDecoder.
<b>≡♦</b>	MvcDecoder	This is MvcDecoder, a member of class MvcDecoder.
<b>=</b> ♦	MvcDecoder	This is MvcDecoder, a member of class MvcDecoder.
<b>=♦9</b>	MvcDecoder	This is MvcDecoder, a member of class MvcDecoder.

### VideoDecoder Class

	Name	Description
<b>=♦</b> ?	VideoDecoder	This is VideoDecoder, a member of class VideoDecoder.
<b>≡∳?</b>	VideoDecoder	This is VideoDecoder, a member of class VideoDecoder.

### Mpeg2Decoder Class

	Name	Description
<b>=</b> ♦ <b>W</b>	~Mpeg2Decoder	Default destructor.
<b>≡♦</b>	Mpeg2Decoder	This is Mpeg2Decoder, a member of class Mpeg2Decoder.
<b>≡♦</b>	Mpeg2Decoder	This is Mpeg2Decoder, a member of class Mpeg2Decoder.
<b>≡♦</b>	Mpeg2Decoder	This is Mpeg2Decoder, a member of class Mpeg2Decoder.
<b>≡♦</b>	Mpeg2Decoder	Creates a MPEG-2 decoder object.

### **MvcDecoder Enumerations**

	Name	Description
<b>a</b>	DecoderType	This is record mvc2::MvcDecoder::DecoderType.
e <sup>p</sup>	TIMEOUT	This is record mvc2::MvcDecoder::TIMEOUT.

### Mpeg2Decoder Class

	Name	Description
<b>a</b>	PROCESSING_3D	This is record mvc2::Mpeg2Decoder::PROCESSING_3D.

### **MvcDecoder Methods**

	Name	Description
<b>≡♦</b>	flush	This is flush, a member of class MvcDecoder.
<b>≡♦</b>	getBufferStatus	This is getBufferStatus, a member of class MvcDecoder.
<b>≡♦</b>	getChannel	Gets the used channel number.
<b>=</b> ♦ ₩	getDataBuffer	Return a buffer for data transfer.
<b>≡♦</b>	getDecoderErrors	This is getDecoderErrors, a member of class MvcDecoder.
<b>≡♦</b>	getDecoderStatus	This is getDecoderStatus, a member of class MvcDecoder.
<b>≡♦</b>	getDecoderType	This is getDecoderType, a member of class MvcDecoder.
=00	getPrivate	This is getPrivate, a member of class MvcDecoder.
<b>≡</b>	setBackwardPlayback	This is setBackwardPlayback, a member of class MvcDecoder.
<b>≡♦</b>	setCplUid	This is setCplUid, a member of class MvcDecoder.
<b>≡♦</b>	setEndOfStream	Go to end of stream state.
<b>≡♦</b>	setSecurityManager	This is setSecurityManager, a member of class MvcDecoder.
<b>≡♦</b>	setStartDelay	Setup start delay for the decoder.
<b>≡♦</b>	waitDecoderStatus	This is waitDecoderStatus, a member of class MvcDecoder.
<b>≡♦</b>	waitForTimeStamp	This is waitForTimeStamp, a member of class MvcDecoder.
<b>≡</b>	waitForTransferFinish	This is waitForTransferFinish, a member of class MvcDecoder.
<b>=</b> ♦	waitForUserData	This is waitForUserData, a member of class MvcDecoder.

### VideoDecoder Class

	Name	Description
<b>=♦</b>	calcDolby3DMatrix	This is calcDolby3DMatrix, a member of class VideoDecoder.
<b>=♦</b>	connectOutput	Connects an output object to the decoder.
<b>=♦</b>	disconnectOutput	Disconnects the output.
<b>=♦</b>	getDecoderErrors	This is getDecoderErrors, a member of class VideoDecoder.
<b>=♦</b>	getDecoderStatus	This is getDecoderStatus, a member of class VideoDecoder.
<b>≡∳</b>	getVideoDataBuffer	This is getVideoDataBuffer, a member of class VideoDecoder.
<b>=♦</b>	setCloneVideo	This is setCloneVideo, a member of class VideoDecoder.
<b>∉</b> ∳	setDolby3DChroma	This is setDolby3DChroma, a member of class VideoDecoder.
<b>=♦</b>	setDolby3DMatrix	This is setDolby3DMatrix, a member of class VideoDecoder.
<b>≡</b>	waitDecoderStatus	This is waitDecoderStatus, a member of class VideoDecoder.

### **MvcDecoder Operators**

	Name	Description
=+)	=	This is =, a member of class MvcDecoder.
=+)	bool	Returns if the object is valid.

### Mpeg2Decoder Class

	Name	Description
<u>(/-</u> =+)	=	This is =, a member of class Mpeg2Decoder.

### **MvcDecoder Enumerations**

	Name	Description
<b>a</b>	DecoderType	This is record mvc2::MvcDecoder::DecoderType.
<b>a</b>	TIMEOUT	This is record mvc2::MvcDecoder::TIMEOUT.

### Mpeg2Decoder Class

	Name	Description
<b>a</b>	PROCESSING_3D	This is record mvc2::Mpeg2Decoder::PROCESSING_3D.

### **MvcDecoder Methods**

	Name	Description
<b>=</b>	flush	This is flush, a member of class MvcDecoder.
<b>≡∳</b>	getBufferStatus	This is getBufferStatus, a member of class MvcDecoder.
<b>≡</b>	getChannel	Gets the used channel number.
<b>■♦</b>	getDataBuffer	Return a buffer for data transfer.
<b>≡</b>	getDecoderErrors	This is getDecoderErrors, a member of class MvcDecoder.
<b>≡</b>	getDecoderStatus	This is getDecoderStatus, a member of class MvcDecoder.
<b>≡∳</b>	getDecoderType	This is getDecoderType, a member of class MvcDecoder.
<b>≡∳</b> •	getPrivate	This is getPrivate, a member of class MvcDecoder.
<b>≡♦</b>	setBackwardPlayback	This is setBackwardPlayback, a member of class MvcDecoder.
<b>≡</b> •	setCplUid	This is setCplUid, a member of class MvcDecoder.
<b>≡</b>	setEndOfStream	Go to end of stream state.
<b>≡∳</b>	setSecurityManager	This is setSecurityManager, a member of class MvcDecoder.
<b>≡∳</b>	setStartDelay	Setup start delay for the decoder.
<b>≡∳</b>	waitDecoderStatus	This is waitDecoderStatus, a member of class MvcDecoder.
<b>≡</b>	waitForTimeStamp	This is waitForTimeStamp, a member of class MvcDecoder.

93

÷	<b>≡</b>	waitForTransferFinish	This is waitForTransferFinish, a member of class MvcDecoder.
÷	<b>≡</b>	waitForUserData	This is waitForUserData, a member of class MvcDecoder.

### **VideoDecoder Class**

	Name	Description
<b>=</b>	calcDolby3DMatrix	This is calcDolby3DMatrix, a member of class VideoDecoder.
<b>=♦</b>	connectOutput	Connects an output object to the decoder.
<b>=♦</b>	disconnectOutput	Disconnects the output.
<b>=♦</b>	getDecoderErrors	This is getDecoderErrors, a member of class VideoDecoder.
<b>≡</b>	getDecoderStatus	This is getDecoderStatus, a member of class VideoDecoder.
<b>≡♦</b>	getVideoDataBuffer	This is getVideoDataBuffer, a member of class VideoDecoder.
<b>≡</b>	setCloneVideo	This is setCloneVideo, a member of class VideoDecoder.
<b>≡♦</b>	setDolby3DChroma	This is setDolby3DChroma, a member of class VideoDecoder.
<b>≡</b>	setDolby3DMatrix	This is setDolby3DMatrix, a member of class VideoDecoder.
<b>≡∳</b>	waitDecoderStatus	This is waitDecoderStatus, a member of class VideoDecoder.

### **MvcDecoder Operators**

	Name	Description
( <u>/-</u>	=	This is =, a member of class MvcDecoder.
( <u>/-</u> =+)	bool	Returns if the object is valid.

### Mpeg2Decoder Class

	Name	Description
=+)	=	This is =, a member of class Mpeg2Decoder.

# 2.1.22.1 Mpeg2Decoder::~Mpeg2Decoder Destructor

Default destructor.

### C++

virtual ~Mpeg2Decoder();

### Description

The default destructor of the Mpeg2Decoder will release the virtual transfer channel and all other resources allocated by the decoder. The decoder will also detached from the PlaybackControl.

# 2.1.22.2 Mpeg2Decoder::Mpeg2Decoder Constructor ()

### C++

Mpeg2Decoder();

### Remarks

This is Mpeg2Decoder, a member of class Mpeg2Decoder.

# 2.1.22.3 Mpeg2Decoder::Mpeg2Decoder Constructor (Mpeg2Decoder&)

C++

Mpeg2Decoder(const Mpeg2Decoder& other)

#### Remarks

This is Mpeg2Decoder, a member of class Mpeg2Decoder.

# 2.1.22.4 Mpeg2Decoder::Mpeg2Decoder Constructor (TMmRc \*, Mpeg2Decoder &)

C++

Mpeg2Decoder(TMmRc \* resultPointer, const Mpeg2Decoder & linkedDecoder);

### Remarks

This is Mpeg2Decoder, a member of class Mpeg2Decoder.

# 2.1.22.5 Mpeg2Decoder::Mpeg2Decoder Constructor (TMmRc \*, MvcDevice &)

Creates a MPEG-2 decoder object.

C++

Mpeg2Decoder(TMmRc \* resultPointer, const MvcDevice & mvcdev);

### **Parameters**

Parameters	Description
const MvcDevice & mvcdev	MvcDevice object to which the decoder will be attached
dec	pointer to a Mpeg2Decoder object which receives the newly created object

### Returns

MMRC\_Ok for a successful creation, otherwise an error.

### **Return Values**

Return Values	Description
MMRC_Ok	operation successfully completed
MMRC_MVC2_DeviceBusy	MvcDevice is already in use

### Description

This method will create a MPEG-2 decoder object and allocate the virtual transfer channel for it.

### 2.1.22.6 Mpeg2Decoder Enumerations

### 2.1.22.6.1 mvc2::Mpeg2Decoder::PROCESSING\_3D Enumeration

### C++

```
enum PROCESSING_3D {
    Processing_None = 0,
    Processing_SideBySide = 1,
    Processing_TopBottom = 2
};

File
    mvc2api_decoder.h
```

### Remarks

This is record mvc2::Mpeg2Decoder::PROCESSING\_3D.

### 2.1.22.7 Mpeg2Decoder Operators

### 2.1.22.7.1 Mpeg2Decoder::= Operator

### C++

```
Mpeg2Decoder& operator =(const Mpeg2Decoder& other);
```

### Remarks

This is =, a member of class Mpeg2Decoder.

### 2.1.23 MvcDecoder Class

### Inheritance Hierarchy

### C++

class MvcDecoder;

### File

mvc2api\_decoder.h

### Remarks

This is the base class of all decoders of a MVC card. It handles the basic functions like data transfers and end-of-stream.

### **Members**

### Methods

	Name	Description
<b>=</b> ♦ <b>W</b>	~MvcDecoder	This is ~MvcDecoder, a member of class MvcDecoder.
<b>≡⋄</b>	MvcDecoder	This is MvcDecoder, a member of class MvcDecoder.
<b>=</b> ♦	MvcDecoder	This is MvcDecoder, a member of class MvcDecoder.
<b>=♦9</b>	MvcDecoder	This is MvcDecoder, a member of class MvcDecoder.

### **MvcDecoder Enumerations**

	Name	Description
<b>1</b>	DecoderType	This is record mvc2::MvcDecoder::DecoderType.
<b>a</b>	TIMEOUT	This is record mvc2::MvcDecoder::TIMEOUT.

### **MvcDecoder Methods**

	Name	Description
<b>≡∳</b>	flush	This is flush, a member of class MvcDecoder.
<b>≡</b>	getBufferStatus	This is getBufferStatus, a member of class MvcDecoder.
<b>≡∳</b>	getChannel	Gets the used channel number.
<b>■♦</b> ₩	getDataBuffer	Return a buffer for data transfer.
<b>≡</b>	getDecoderErrors	This is getDecoderErrors, a member of class MvcDecoder.
<b>≡</b>	getDecoderStatus	This is getDecoderStatus, a member of class MvcDecoder.
<b>≡∳</b>	getDecoderType	This is getDecoderType, a member of class MvcDecoder.
<b>≡</b> ∳ <b>?</b>	getPrivate	This is getPrivate, a member of class MvcDecoder.
<b>≡</b> •	setBackwardPlayback	This is setBackwardPlayback, a member of class MvcDecoder.
<b>≡∳</b>	setCplUid	This is setCplUid, a member of class MvcDecoder.
<b>≡∳</b>	setEndOfStream	Go to end of stream state.
<b>≡∳</b>	setSecurityManager	This is setSecurityManager, a member of class MvcDecoder.
<b>≡∳</b>	setStartDelay	Setup start delay for the decoder.
<b>≡∳</b>	waitDecoderStatus	This is waitDecoderStatus, a member of class MvcDecoder.
<b>≡∳</b>	waitForTimeStamp	This is waitForTimeStamp, a member of class MvcDecoder.
<b>≡</b>	waitForTransferFinish	This is waitForTransferFinish, a member of class MvcDecoder.
<b>≡</b>	waitForUserData	This is waitForUserData, a member of class MvcDecoder.

### **MvcDecoder Operators**

	Name	Description
=+)	=	This is =, a member of class MvcDecoder.
=+)	bool	Returns if the object is valid.

### **MvcDecoder Enumerations**

	Name	Description
<b>a</b>	DecoderType	This is record mvc2::MvcDecoder::DecoderType.
<b>a</b>	TIMEOUT	This is record mvc2::MvcDecoder::TIMEOUT.

### **MvcDecoder Methods**

	Name	Description
<b>=</b> ♦	flush	This is flush, a member of class MvcDecoder.
<b>=</b> ♦	getBufferStatus	This is getBufferStatus, a member of class MvcDecoder.
<b>≡⋄</b>	getChannel	Gets the used channel number.
<b>=</b> ♦ ₩	getDataBuffer	Return a buffer for data transfer.
<b>≡♦</b>	getDecoderErrors	This is getDecoderErrors, a member of class MvcDecoder.
<b>=</b> ♦	getDecoderStatus	This is getDecoderStatus, a member of class MvcDecoder.
<b>=</b> ♦	getDecoderType	This is getDecoderType, a member of class MvcDecoder.
<b>=♦</b> •	getPrivate	This is getPrivate, a member of class MvcDecoder.
<b>=</b> ♦	setBackwardPlayback	This is setBackwardPlayback, a member of class MvcDecoder.
<b>=♦</b>	setCplUid	This is setCplUid, a member of class MvcDecoder.
<b>=♦</b>	setEndOfStream	Go to end of stream state.
<b>=♦</b>	setSecurityManager	This is setSecurityManager, a member of class MvcDecoder.
<b>≡</b>	setStartDelay	Setup start delay for the decoder.
<b>≡</b>	waitDecoderStatus	This is waitDecoderStatus, a member of class MvcDecoder.
<b>=♦</b>	waitForTimeStamp	This is waitForTimeStamp, a member of class MvcDecoder.

<b>≡♦</b>	waitForTransferFinish	This is waitForTransferFinish, a member of class MvcDecoder.
<b>≡</b>	waitForUserData	This is waitForUserData, a member of class MvcDecoder.

### **MvcDecoder Operators**

	Name	Description
=+)	=	This is =, a member of class MvcDecoder.
( <u>/-</u> =+)	bool	Returns if the object is valid.

### 2.1.23.1 MvcDecoder::~MvcDecoder Destructor

### C++

```
virtual ~MvcDecoder();
```

### Remarks

This is ~MvcDecoder, a member of class MvcDecoder.

### 2.1.23.2 MvcDecoder::MvcDecoder Constructor ()

### C++

MvcDecoder();

#### Remarks

This is MvcDecoder, a member of class MvcDecoder.

## 2.1.23.3 MvcDecoder::MvcDecoder Constructor (MvcDecoder&)

### C++

```
MvcDecoder(const MvcDecoder& other);
```

### Remarks

This is MvcDecoder, a member of class MvcDecoder.

# 2.1.23.4 MvcDecoder::MvcDecoder Constructor (MvcDecoderPrivate \*)

C++

```
MvcDecoder(MvcDecoderPrivate * mvcdec);
```

### Remarks

This is MvcDecoder, a member of class MvcDecoder.

### 2.1.23.5 MvcDecoder Enumerations

### 2.1.23.5.1 mvc2::MvcDecoder::DecoderType Enumeration

### C++

```
enum DecoderType {
```

98

```
Type_UnknownDecoder = 0,
Type_Jpeg2kDecoder = 1,
Type_PCMDecoder = 2,
Type_Mpeg2Decoder = 3,
Type_TestDecoder = 4,
Type_OverlayDecoder = 5,
Type_SubtitleDecoder = 6
};
File
mvc2api_decoder.h
```

### Remarks

This is record mvc2::MvcDecoder::DecoderType

### 2.1.23.5.2 mvc2::MvcDecoder::TIMEOUT Enumeration

#### C++

```
enum TIMEOUT {
   WaitInfinite = 0xffffffff,
   DonotWait = 0
};
```

### File

mvc2api\_decoder.h

### **Members**

Members	Description
WaitInfinite = 0xffffffff	Wait forever.
DonotWait = 0	Do not wait. Return function immediately.

### Remarks

This is record mvc2::MvcDecoder::TIMEOUT.

### 2.1.23.6 MycDecoder Methods

# 2.1.23.6.1 MvcDecoder::flush Method



### C++

```
TMmRc flush(uint32_t keepTime, uint32_t * nextTimeStamp);
```

### Remarks

This is flush, a member of class MvcDecoder.

### 2.1.23.6.2 MvcDecoder::getBufferStatus Method

### C++

```
BufferStatus getBufferStatus();
```

### Remarks

This is getBufferStatus, a member of class MvcDecoder.

### 2.1.23.6.3 MvcDecoder::getChannel Method

Gets the used channel number.



99

### C++

```
uint32_t getChannel() const;
```

#### Returns

Transfer channel id.

### **Description**

Every decoder allocates a virtual transfer channel. To identify this channel an id will be assigned which this method returns.

# 2.1.23.6.4 MvcDecoder::getDataBuffer Method



Return a buffer for data transfer.

#### C++

virtual TMmRc getDataBuffer(DataBuffer & databuf, uint32\_t minsize = 0, uint32\_t timeout =
WaitInfinite);

### **Parameters**

Parameters	Description
DataBuffer & databuf	A DataBuffer object which will be overwritten with the new DataBuffer
uint32_t minsize = 0	minimum number of bytes which are needed for the transfer, the method will try to take care of this size, but it's not guaranteed that the buffer has this size
uint32_t timeout = WaitInfinite	number of milliseconds to wait for a available DataBuffer

### **Returns**

MMRC\_Ok if the method call was successful, otherwise an error.

### **Return Values**

Return Values	Description
MMRC_Ok	operation successfully completed

### Description

This method will return a DataBuffer which is used to transfer a frame to the MVC card. A MvcDevice has a number of DataBuffers for the transfer. If all buffers are in use, the function can wait for the next empty buffer. The number of buffer available and the buffer size depend on the decoder, e.g. audio decoders will have smaller buffer because of the smaller frame sizes.

### 2.1.23.6.5 MvcDecoder::getDecoderErrors Method

### C++

TMmRc getDecoderErrors(DecoderErrors & errors) const;

### Remarks

This is getDecoderErrors, a member of class MvcDecoder.

### 2.1.23.6.6 MvcDecoder::getDecoderStatus Method

### C++

TMmRc getDecoderStatus(DecoderStatus & status) const;

### Remarks

This is getDecoderStatus, a member of class MvcDecoder.

# 2.1.23.6.7 MvcDecoder::getDecoderType Method



#### C++

DecoderType getDecoderType() const;

#### Remarks

This is getDecoderType, a member of class MvcDecoder.

# 2.1.23.6.8 MvcDecoder: getPrivate Method

### C++

MvcDecoderPrivate \* getPrivate() const;

### Remarks

This is getPrivate, a member of class MvcDecoder.

### 2.1.23.6.9 MvcDecoder::setBackwardPlayback Method

C++



TMmRc setBackwardPlayback(bool backwards);

#### Remarks

This is setBackwardPlayback, a member of class MvcDecoder.

# 2.1.23.6.10 MvcDecoder:setCplUid Method



C+

TMmRc setCplUid(const uint8\_t \* uid);

### Remarks

This is setCplUid, a member of class MvcDecoder.

# 2.1.23.6.11 MvcDecoder: setEndOfStream Method

Go to end of stream state.

### C++

void setEndOfStream();

### Description

After all data has been transferred, the decoder should be set to end-of-stream state. This ensures that even the last frame will be displayed and it will automatically go to stop state if the last frame was displayed. No more data can be transferred if a decoder is in end-of-stream state.

# 2.1.23.6.12 MvcDecoder: setSecurityManager Method

### C++

TMmRc setSecurityManager(const SecurityManager & sm);



### Remarks

This is setSecurityManager, a member of class MvcDecoder.

### 2.1.23.6.13 MvcDecoder::setStartDelay Method

Setup start delay for the decoder.



#### C++

```
void setStartDelay(float delay);
```

### **Parameters**

Parameters		Description
float delay	0,	start delay in milliseconds

### Description

The playback will start at a time stamp of zero, setting the time stamp of the first frame to a different value will delay the start by the given time. This method does simply the same as setting the time stamp of the first frame.

### 2.1.23.6.14 MvcDecoder: waitDecoderStatus Method



### C++

TMmRc waitDecoderStatus(DecoderStatus & status) const;

#### Remarks

This is waitDecoderStatus, a member of class MvcDecoder.

### 2.1.23.6.15 MvcDecoder: waitForTimeStamp Method

```
TMmRc waitForTimeStamp(uint32_t timestampUpper, uint32_t timestampLower, uint32_t *
timestamp = 0);
```

### Remarks

This is waitForTimeStamp, a member of class MvcDecoder.

### 2.1.23.6.16 MvcDecoder: waitForTransferFinish Method



```
TMmRc waitForTransferFinish(uint32_t timeout = WaitInfinite);
```

### Remarks

This is waitForTransferFinish, a member of class MvcDecoder.

## 2.1.23.6.17 MvcDecoder: waitForUserData Method



### C++

```
TMmRc waitForUserData(uint32_t userdataUpper, uint32_t userdataLower, uint32_t * userdata =
```

### Remarks

This is waitForUserData, a member of class MvcDecoder.

### 2.1.23.7 MvcDecoder Operators

### 2.1.23.7.1 MvcDecoder::= Operator

MvcDecoder & operator =(const MvcDecoder& other);

### Remarks

This is =, a member of class MvcDecoder.

# 2.1.23.7.2 MvcDecoder::bool Operator

Returns if the object is valid.

### C++

operator bool() const;

### Description

Returns true if the object is valid or false if the object is invalid.

# 2.1.24 MvcDevice Class

### **Inheritance Hierarchy**

### C++

class MvcDevice;

### File

mvc2api\_device.h

### Description

This class is used to access a MVC20x media block. Objects of this class cannot be directly created, you need to use a MvcDeviceIterator or MvcNetDeviceIterator.

### **Members**

### **Methods**

	Name	Description
<b>≡</b>	~MvcDevice	MvcDevice destructor.
<b>=♦</b>	MvcDevice	Default constructor.
<b>≡</b>	MvcDevice	Default constructor.

### **MvcDevice Enumerations**

	Name	Description
<b>a</b>	CLOCKID	This is record mvc2::MvcDevice::CLOCKID.
<b>a</b>	OUTPUTMODE	This is record mvc2::MvcDevice::OUTPUTMODE.
<b>a</b>	PRODUCTCODE	This is record mvc2::MvcDevice::PRODUCTCODE.

### **MvcDevice Methods**

	Name	Description
<b>≡</b>	getAPIVersion	Return API version
<b>≡</b>	getBootloaderVersion	Return version of the card bootloader.
<b>=♦</b>	getCPUInfo	Get CPU information.
<b>≡</b>	getDeviceState	Get UID.

<b>≡</b>	getDriverVersion	Return driver version.
<b>≡∳</b>	getErrorDescription	This is getErrorDescription, a member of class MvcDevice.
<b>≡</b>	getErrorString	This is getErrorString, a member of class MvcDevice.
<b>≡</b>	getFirmwareVersion	Returns the firmware version.
<b>≡</b>	getLastError	This is getLastError, a member of class MvcDevice.
<b>≡∳</b>	getNetworkConfiguration	Return network information
<b>=♦</b>	getPCIBus	Return PCI bus number.
<b>≡</b>	getPCISlot	Return PCI slot number
<b>≡</b>	getPowerDownStatus	This is getPowerDownStatus, a member of class MvcDevice.
<b>≡</b>	getProductCode	Return product code
<b>≡</b>	getProductRevision	Return product revision
<b>≡∳</b>	getSecurityBootloaderVersion	Return version of the cards security bootloader.
<b>≡</b>	getSecurityManagerHwStatus	Reads the crc16 checksum of security application.
<b>≡</b>	getSecurityManagerVersion	Return version of the cards security manager.
<b>≡</b>	getSystemPosixTime	Reads the posix time.
<b>≡</b>	getSystemStatus	Gets the system status.
<b>=♦</b>	getUID	Get UID.
<b>=♦</b>	getUptimeMs	Reads the uptime in milliseconds.
<b>=♦</b>	putMessage	Ingest a log message in the card log.
<b>=♦</b>	resetCard	This is resetCard, a member of class MvcDevice.
<b>≡</b>	setAuthenticationPassword	This is setAuthenticationPassword, a member of class MvcDevice.
<b>≡</b>	setOutputMode	This is setOutputMode, a member of class MvcDevice.
<b>≡</b>	setPowerDownTime	This is setPowerDownTime, a member of class MvcDevice.
<b>≡</b>	setStatusLed	This is setStatusLed, a member of class MvcDevice.

### **MvcDevice Operators**

	Name	Description
<del>(/=</del> =+)	=	Assignment operator.
<u>(/-</u> =+)	bool	This is bool, a member of class MvcDevice.

### **MvcDevice Enumerations**

	Name	Description
<b>a</b>	CLOCKID	This is record mvc2::MvcDevice::CLOCKID.
<b>a</b>	OUTPUTMODE	This is record mvc2::MvcDevice::OUTPUTMODE.
<b>a</b>	PRODUCTCODE	This is record mvc2::MvcDevice::PRODUCTCODE.

### **MvcDevice Methods**

	Name	Description
<b>=</b> ♦	getAPIVersion	Return API version
<b>=</b> ♦	getBootloaderVersion	Return version of the card bootloader.
<b>=</b> ♦	getCPUInfo	Get CPU information.
<b>=♦</b>	getDeviceState	Get UID.
<b>=</b> ♦	getDriverVersion	Return driver version.
<b>=♦</b>	getErrorDescription	This is getErrorDescription, a member of class MvcDevice.
<b>=♦</b>	getErrorString	This is getErrorString, a member of class MvcDevice.
<b>=♦</b>	getFirmwareVersion	Returns the firmware version.
<b>=♦</b>	getLastError	This is getLastError, a member of class MvcDevice.
<b>=</b>	getNetworkConfiguration	Return network information
<b>=♦</b>	getPCIBus	Return PCI bus number.

<b>≡</b>	getPCISlot	Return PCI slot number
<b>≡</b>	getPowerDownStatus	This is getPowerDownStatus, a member of class MvcDevice.
<b>≡</b>	getProductCode	Return product code
<b>≡</b>	getProductRevision	Return product revision
<b>≡</b>	getSecurityBootloaderVersion	Return version of the cards security bootloader.
<b>≡</b>	getSecurityManagerHwStatus	Reads the crc16 checksum of security application.
<b>≡</b>	getSecurityManagerVersion	Return version of the cards security manager.
<b>≡</b>	getSystemPosixTime	Reads the posix time.
<b>≡</b>	getSystemStatus	Gets the system status.
<b>≡</b>	getUID	Get UID.
<b>≡</b>	getUptimeMs	Reads the uptime in milliseconds.
<b>≡</b>	putMessage	Ingest a log message in the card log.
<b>≡</b>	resetCard	This is resetCard, a member of class MvcDevice.
<b>≡∳</b>	setAuthenticationPassword	This is setAuthenticationPassword, a member of class MvcDevice.
<b>≡</b>	setOutputMode	This is setOutputMode, a member of class MvcDevice.
<b>≡∳</b>	setPowerDownTime	This is setPowerDownTime, a member of class MvcDevice.
<b>≡∳</b>	setStatusLed	This is setStatusLed, a member of class MvcDevice.

### **MvcDevice Operators**

	Name	Description
=+)	=	Assignment operator.
=+)	bool	This is bool, a member of class MvcDevice.

### 2.1.24.1 MvcDevice::~MvcDevice Destructor

MvcDevice destructor.

### C++

~MvcDevice();

### Description

This is the generic MvcDevice destructor, which will free any resources allocated by the MvcDevice.

# 2.1.24.2 MvcDevice::MvcDevice Constructor ()

Default constructor.

### C++

```
MvcDevice();
MvcDevice(const MvcDevice & indev);
```

### Description

The default constructor will generate a MvcDevice object without any information about a real MVC card.

### 2.1.24.3 MvcDevice Enumerations

### 2.1.24.3.1 mvc2::MvcDevice::CLOCKID Enumeration

```
enum CLOCKID {
   ClockId_SM = 0,
   ClockId_Firmware = 1,
   ClockId_System = 2
};
```

File

mvc2api\_device.h

### Remarks

This is record mvc2::MvcDevice::CLOCKID.

### 2.1.24.3.2 mvc2::MvcDevice::OUTPUTMODE Enumeration

```
C++
```

```
enum OUTPUTMODE {
   OutputMode_SDI = 0,
   OutputMode_DVI = 1
};
```

File

mvc2api\_device.h

### Remarks

This is record mvc2::MvcDevice::OUTPUTMODE.

### 2.1.24.3.3 mvc2::MvcDevice::PRODUCTCODE Enumeration

### C++

```
enum PRODUCTCODE {
   ProductCode_MVC200DC = 2224,
   ProductCode_MVC201 = 2250
};
```

File

mvc2api\_device.h

### Remarks

This is record mvc2::MvcDevice::PRODUCTCODE.

### 2.1.24.4 MvcDevice Methods

### 2.1.24.4.1 MvcDevice::getAPIVersion Method

Return API version

C++

```
VersionValue getAPIVersion() const;
```



### Returns

VersionValue object with the API version.

### Description

Gets the version of the API dll.

# 2.1.24.4.2 MvcDevice::getBootloaderVersion Method



Return version of the card bootloader.

### C++

VersionValue getBootloaderVersion() const;

### **Returns**

VersionValue object with the bootloader version.

### 2.1.24.4.3 MvcDevice::getCPUInfo Method



Get CPU information.

### C++

CPUInfo getCPUInfo(uint32\_t cpuIndex) const;

### **Parameters**

Parameters	Description
uint32_t cpuIndex	index of the CPU

### **Returns**

Returns a CPUInfo object.

### Description

Get basic information about the CPUs used on the MediaBlock. It could be useful to find out some differences on board revisions.

# 2.1.24.4.4 MvcDevice::getDeviceState Method



Get UID.

### C++

TMmRc getDeviceState() const;

### Returns

TMmRc device state

### **Return Values**

Return Values	Description
MMRC_MVC2_OpenFailed	CreateFile could not open device
MMRC_MVC2_WrongDriverInterfaceVersion	API and driver are not compatible
MMRC_MVC2_BadPowerSupply	power supply is not connected
MMRC_MVC2_BootImageNotLoaded	basic bootup procedure failed
MMRC_Ok	device is up and running

### Description

After a MVC card is opened with the MvcDeviceIterator, this functions return code should be checked for valid device state.

### 2.1.24.4.5 MvcDevice::getDriverVersion Method

Return driver version.



#### C++

VersionValue getDriverVersion() const;

### Returns

VersionValue object with the driver version.

### Description

Gets the version of the used driver.

# 2.1.24.4.6 MvcDevice::getErrorDescription Method



static const char \* getErrorDescription(TMmRc rc);

### Remarks

This is getErrorDescription, a member of class MvcDevice.

### 2.1.24.4.7 MvcDevice::getErrorString Method



### C++

static const char \* getErrorString(TMmRc rc);

### Remarks

This is getErrorString, a member of class MvcDevice.

# 2.1.24.4.8 MvcDevice: getFirmwareVersion Method

Returns the firmware version.



### C++

VersionValue getFirmwareVersion() const;

### Returns

VersionValue object with the firmware version.

### Description

Gets the firmware version number.

### 2.1.24.4.9 MvcDevice::getLastError Method



### C++

TMmRc getLastError() const;

### Remarks

This is getLastError, a member of class MvcDevice.

## 2.1.24.4.10 MvcDevice::getNetworkConfiguration Method

Return network information

### C++

NetworkInterfaceInfo getNetworkConfiguration() const

### Returns

NetworkInterfaceInfo object, which is empty if no information could be found.

### Description

Gets network information of the device. The MVC card driver provides a virtual network driver which can be used to communicate with the device.

### 2.1.24.4.11 MvcDevice::getPCIBus Method

Return PCI bus number.



#### C++

```
uint32_t getPCIBus() const;
```

### Returns

PCI bus number.

### Description

Gets the PCI bus number of the MVC card.

### 2.1.24.4.12 MvcDevice::getPCISIot Method



Return PCI slot number

### C++

```
uint32_t getPCISlot() const;
```

### Returns

PCI slot number.

### Description

Gets the PCI slot number of the MVC card.

# 2.1.24.4.13 MvcDevice::getPowerDownStatus Method

### C++



```
TMmRc getPowerDownStatus(int32_t * powerState, int32_t * powerDownTime, int32_t *
timeToPowerDown);
```

### Remarks

This is getPowerDownStatus, a member of class MvcDevice.

# 2.1.24.4.14 MvcDevice::getProductCode Method



Return product code

### C++

```
uint32_t getProductCode() const;
```

### Returns

Product code number.

### Description

Returns the product code of the MVC card to identify the card type.

# 2.1.24.4.15 MvcDevice::getProductRevision Method



Return product revision

### C++

uint32\_t getProductRevision() const;

#### Returns

Product revision number.

### **Description**

Gets product revision number, were one is the first revision.

### 2.1.24.4.16 MvcDevice::getSecurityBootloaderVersion Method



Return version of the cards security bootloader.

### C++

VersionValue getSecurityBootloaderVersion() const;

#### Returns

VersionValue object with the security bootloader version.

# 2.1.24.4.17 MvcDevice: getSecurityManagerHwStatus Method

Reads the crc16 checksum of security application.



### C++

TMmRc getSecurityManagerHwStatus(uint16\_t\* secAppCrc16, int32\_t\* mainFpgaProgRc, int32\_t\* outFpgaProgRc, int32\_t\* rtcInitRc);

### **Parameters**

Parameters	Description
checksum	crc16 calculated over internal program flash

### Returns

MMRC\_Ok on success with valid checksum

### 2.1.24.4.18 MvcDevice::getSecurityManagerVersion Method

Return version of the cards security manager.



### C++

VersionValue getSecurityManagerVersion() const;

### **Returns**

VersionValue object with the security manager version.

### 2.1.24.4.19 MvcDevice::getSystemPosixTime Method

Reads the posix time.



### C++

time\_t getSystemPosixTime(CLOCKID clockId = ClockId\_SM);

#### **Parameters**

Parameters	Description
CLOCKID clockid = Clockid_SM	select the clock to read out, 0 for SM and 1 for firmware

### Returns

posix time or -1 on error

# 2.1.24.4.20 MvcDevice::getSystemStatus Method



Gets the system status.

#### C++

SystemStatus getSystemStatus(bool readTemperatures = true) const;

#### Returns

SystemStatus object with the collected information. The SystemStatus object will be empty (filled with zeros) if an error happened.

### Description

The method will return useful information about the status of the firmware.

# 2.1.24.4.21 MvcDevice::getUID Method



Get UID.

### C++

uint32\_t getUID() const;

### **Returns**

32 bit unique identifier of the card. Zero will be returned if an error happened.

### Description

Every MVC card has a unique serial number. This method will return this number.

## 2.1.24.4.22 MvcDevice::getUptimeMs Method

Reads the uptime in milliseconds.



### C++

uint64\_t getUptimeMs(CLOCKID clockId = ClockId\_SM);

### **Parameters**

Parameters	Description
CLOCKID clockId = ClockId_SM	select the clock to read out, 0 for SM, 1 for firmware, 2 for local OS

### Returns

uptime in milliseconds

## 2.1.24.4.23 MvcDevice::putMessage Method



Ingest a log message in the card log.

### C++

TMmRc putMessage(uint32\_t severity, uint32\_t subid, char \* msg, uint32\_t msgsize);

#### **Parameters**

Parameters	Description
uint32_t severity	severity of the message (use LogMessage::Severity)
uint32_t subid	sub id of the message
char * msg	ansi string to ingest
uint32_t msgsize	size in bytes of the message including string termination

#### Returns

MMRC\_Ok or an error.

### Description

This method will ingest a log message in the cards log.

### 2.1.24.4.24 MvcDevice::resetCard Method



#### C++

TMmRc resetCard();

#### Remarks

This is resetCard, a member of class MvcDevice.

## 2.1.24.4.25 MvcDevice::setAuthenticationPassword Method



### C++

TMmRc setAuthenticationPassword(const char \* passwd);

### Remarks

This is setAuthenticationPassword, a member of class MvcDevice.

### 2.1.24.4.26 MvcDevice::setOutputMode Method



### C++

TMmRc setOutputMode(OUTPUTMODE mode);

### Remarks

This is setOutputMode, a member of class MvcDevice.

### 2.1.24.4.27 MvcDevice::setPowerDownTime Method



TMmRc setPowerDownTime(uint32\_t seconds);

### Remarks

This is setPowerDownTime, a member of class MvcDevice.

### 2.1.24.4.28 MycDevice::setStatusLed Method



C++

TMmRc setStatusLed(uint32\_t which, uint32\_t color, uint32\_t blinkmask);

### Remarks

This is setStatusLed, a member of class MvcDevice.

### 2.1.24.5 MvcDevice Operators

### 2.1.24.5.1 MvcDevice::= Operator

Assignment operator.

### C++

MvcDevice & operator =(const MvcDevice & mvcdev);

### **Parameters**

Parameters	Description
const MvcDevice & mvcdev	Input MvcDevice to copy data from.

#### Returns

true will be returned if the input MvcDevice is not NULL, which means it actual holds information about a MVC card.

### Description

This operator will copy all information from the incoming MvcDevice and frees the resources which were used be the MvcDevice before.

### 2.1.24.5.2 MvcDevice::bool Operator

### C++

operator bool() const;

### Remarks

This is bool, a member of class MvcDevice.

### 2.1.25 MvcDeviceIterator Class

MVC 200 device iterator class.

### **Inheritance Hierarchy**

C++

class MvcDeviceIterator;

### File

mvc2api\_device.h

### Description

This class is used to parse the MVC 200 cards available on the system. It holds a list of cards which can be parsed with getNext() method.

### Remarks

113

### **Members**

### Methods

	Name	Description
<b>≡♦</b>	~MvcDeviceIterator	Default destructor.
<b>=</b> ♦	MvcDeviceIterator	Default constructor.
<b>=</b> ♦	MvcDeviceIterator	Default constructor.

### **MvcDeviceIterator Methods**

	Name	Description
<b>≡♦</b>	getFirst	Gets first MvcDevice.
<b>≡♦</b>	getIndex	Get MvcDevice by index.
<b>≡♦</b>	getNext	Gets next MVC device.

### **MvcDeviceIterator Methods**

	Name	Description
<b>≡♦</b>	getFirst	Gets first MvcDevice.
<b>=</b> ♦	getIndex	Get MvcDevice by index.
<b>≡♦</b>	getNext	Gets next MVC device.

### 2.1.25.1 MvcDeviceIterator::~MvcDeviceIterator Destructor

Default destructor.

### C++

~MvcDeviceIterator();

### Description

All resources which were need to parse MVC card list will be freed.

### 2.1.25.2 MvcDeviceIterator::MvcDeviceIterator Constructor ()

Default constructor.

### C++

```
MvcDeviceIterator();
MvcDeviceIterator(const MvcDeviceIterator & initor);
```

### Description

The list of MVC cards will be initialized. To get the first MVC card found, use the getNext() method.

# 2.1.25.3 MvcDeviceIterator Methods



### 2.1.25.3.1 MvcDeviceIterator::getFirst Method

Gets first MvcDevice.

### C++

MvcDevice getFirst();

### Returns

A MvcDevice. If an error occurred, the MvcDevice will be empty.

### Description

This method returns the first MvcDevice found on the computer, if there is one.

# 2.1.25.3.2 MvcDeviceIterator::getIndex Method

Get MvcDevice by index.

### C++

MvcDevice getIndex(uint32\_t index);

### **Parameters**

Parameters	Description
uint32_t index	card index starting by zero

### Returns

A MvcDevice. If an error occurred, the MvcDevice will be empty.

### Description

The iterator will return the MvcDevice at a specific index. It will also set the iterator at that position, so getNext() can be used to continue parsing. It's recommended to use getNext() method instead of getIndex().

### 2.1.25.3.3 MvcDeviceIterator::getNext Method

Gets next MVC device.

### C++

MvcDevice getNext();

### Returns

An MvcDevice object.

### Description

This method is used to get the next MvcDevice for the iterator. If there's no more MVC card found an empty MvcDevice will be returned.

# 2.1.26 MycNetDeviceIterator Class



### **Inheritance Hierarchy**

### C++

class MvcNetDeviceIterator;

### File

mvc2api\_device.h

### Remarks

This is class mvc2::MvcNetDeviceIterator.

### **Members**

### Methods

	Name	Description
<b>≡♦</b>	~MvcNetDeviceIterator	This is ~MvcNetDeviceIterator, a member of class
		MvcNetDeviceIterator.

<b>≡♦</b>	MvcNetDeviceIterator	This is MvcNetDeviceIterator, a member of class MvcNetDeviceIterator.
<b>=♦</b>	MvcNetDeviceIterator	This is MvcNetDeviceIterator, a member of class MvcNetDeviceIterator.

### **MvcNetDeviceIterator Methods**

	Name	Description
<b>=♦</b>	getFirst	This is getFirst, a member of class MvcNetDeviceIterator.
<b>=♦</b>	getIndex	This is getIndex, a member of class MvcNetDeviceIterator.
<b>≡</b>	getNext	This is getNext, a member of class MvcNetDeviceIterator.

### **MvcNetDeviceIterator Methods**

	Name	Description
<b>≡♦</b>	getFirst	This is getFirst, a member of class MvcNetDeviceIterator.
<b>≡♦</b>	getIndex	This is getIndex, a member of class MvcNetDeviceIterator.
<b>≡♦</b>	getNext	This is getNext, a member of class MvcNetDeviceIterator.

### 2.1.26.1 MvcNetDeviceIterator::~MvcNetDeviceIterator Destructor

### C++

~MvcNetDeviceIterator();

#### Remarks

This is ~MvcNetDeviceIterator, a member of class MvcNetDeviceIterator.

# 2.1.26.2 MvcNetDeviceIterator::MvcNetDeviceIterator Constructor (MvcNetDeviceIterator &)

C++

MvcNetDeviceIterator(const MvcNetDeviceIterator & initor);

### Remarks

This is MvcNetDeviceIterator, a member of class MvcNetDeviceIterator.

# 2.1.26.3 MvcNetDeviceIterator::MvcNetDeviceIterator Constructor (char \*)

C++

MvcNetDeviceIterator(const char \* hostname);

### Remarks

This is MvcNetDeviceIterator, a member of class MvcNetDeviceIterator.

### 2.1.26.4 MvcNetDeviceIterator Methods

### 2.1.26.4.1 MvcNetDeviceIterator::getFirst Method

#### C++

```
MvcDevice getFirst();
```

### Remarks

This is getFirst, a member of class MvcNetDeviceIterator.

### 2.1.26.4.2 MvcNetDeviceIterator::getIndex Method

### C++

```
MvcDevice getIndex(uint32_t index);
```

### Remarks

This is getIndex, a member of class MvcNetDeviceIterator.

### 2.1.26.4.3 MvcNetDeviceIterator::getNext Method

### C++

```
MvcDevice getNext();
```

### Remarks

This is getNext, a member of class MvcNetDeviceIterator.

### 2.1.27 NetworkInterfaceInfo Class

### **Inheritance Hierarchy**

### C++

class NetworkInterfaceInfo;

### File

mvc2api\_device.h

### Remarks

Network interface information class

### **Members**

### Methods

	Name	Description
<b>≞</b> ♦	NetworkInterfaceInfo	This is NetworkInterfaceInfo, a member of class NetworkInterfaceInfo.
<b>≡♦</b>	NetworkInterfaceInfo	This is NetworkInterfaceInfo, a member of class NetworkInterfaceInfo.
<b>=</b> ♦	NetworkInterfaceInfo	This is NetworkInterfaceInfo, a member of class NetworkInterfaceInfo.

### **NetworkInterfaceInfo Methods**

	Name	Description
<b>≡⋄</b>	getIPAddress	Return the IP address
<b>≡⋄</b>	getIPAddress	Return IP address
<b>≡♦</b>	getMACAddress	Return MAC address

### **NetworkInterfaceInfo Operators**

	Name	Description
=+)	=	This is =, a member of class NetworkInterfaceInfo.
=+)	bool	This is bool, a member of class NetworkInterfaceInfo.

### **NetworkInterfaceInfo Methods**

	Name	Description
<b>≡</b>	getIPAddress	Return the IP address
<b>≡</b>	getIPAddress	Return IP address
<b>=♦</b>	getMACAddress	Return MAC address

### **NetworkInterfaceInfo Operators**

	Name	Description
=+)	=	This is =, a member of class NetworkInterfaceInfo.
( <u>/-</u> =+)	bool	This is bool, a member of class NetworkInterfaceInfo.

### 2.1.27.1 NetworkInterfaceInfo::NetworkInterfaceInfo Constructor ()

### C++

NetworkInterfaceInfo();

#### Remarks

This is NetworkInterfaceInfo, a member of class NetworkInterfaceInfo.

# 2.1.27.2 NetworkInterfaceInfo::NetworkInterfaceInfo Constructor (NetworkInterfaceInfo &)

C++

NetworkInterfaceInfo(const NetworkInterfaceInfo & ininfo);

### Remarks

This is NetworkInterfaceInfo, a member of class NetworkInterfaceInfo.

# 2.1.27.3 NetworkInterfaceInfo::NetworkInterfaceInfo Constructor (uint32\_t, uint8\_t \*)

C++

NetworkInterfaceInfo(const uint32\_t ipaddr, const uint8\_t \* mac = 0);

### Remarks

This is NetworkInterfaceInfo, a member of class NetworkInterfaceInfo.

### 2.1.27.4 NetworkInterfaceInfo Methods

### 2.1.27.4.1 NetworkInterfaceInfo::getIPAddress Method ()

Return the IP address

### C++

uint32\_t getIPAddress() const;

#### Returns

IP address

### **Description**

Gets the IP address as a 32 bit value.

### 2.1.27.4.2 NetworkInterfaceInfo::getIPAddress Method (uint8\_t \*)

Return IP address

### C++

```
void getIPAddress(uint8_t * ipaddr) const;
```

### **Parameters**

Parameters	Description
uint8_t * ipaddr	Array of bytes which receives 4 bytes of the IP address

### Description

Gets the IP address as four 8bit values.

### 2.1.27.4.3 NetworkInterfaceInfo::getMACAddress Method

Return MAC address

### C++

void getMACAddress(uint8\_t \* mac) const;

### **Parameters**

Parameters	Description
uint8_t * mac	pointer to an array which receives the 6 bytes of MAC
	address

### Description

Gets the MAC address.

### 2.1.27.5 NetworkInterfaceInfo Operators

### 2.1.27.5.1 NetworkInterfaceInfo::= Operator

### C++

NetworkInterfaceInfo & operator =(const NetworkInterfaceInfo & innet);

### Remarks

This is =, a member of class NetworkInterfaceInfo.

### 2.1.27.5.2 NetworkInterfaceInfo::bool Operator

### C++

```
operator bool() const;
```

### Remarks

This is bool, a member of class NetworkInterfaceInfo.

# 2.1.28 OverlayDataBuffer Class

### **Inheritance Hierarchy**

### C++

class OverlayDataBuffer : public DataBuffer;

### File

mvc2api\_overlay.h

### Remarks

This is class mvc2::OverlayDataBuffer.

### Members

### Methods

	Name	Description
<b>=</b> ♦ ₩	~DataBuffer	Destructor
<b>=</b> ♦	DataBuffer	This is DataBuffer, a member of class DataBuffer.
<b>=♦</b>	DataBuffer	This is DataBuffer, a member of class DataBuffer.
=00	DataBuffer	This is DataBuffer, a member of class DataBuffer.

### OverlayDataBuffer Class

	Name	Description
<b>≡♦</b>	OverlayDataBuffer	This is OverlayDataBuffer, a member of class OverlayDataBuffer.
<b>=♦</b> •	OverlayDataBuffer	This is OverlayDataBuffer, a member of class OverlayDataBuffer.

### **DataBuffer Methods**

	Name	Description
<b>≡</b>	сору	Copy a data block to DataBuffer.
<b>≡</b>	getBufferAddress	Gets the buffer memory address.
<b>≡</b>	getFreeSize	Get the available buffer size.
<b>≡</b> ∳	getInSize	Get the transfer size.
<b>≡</b>	getTimeStamp	Get current timestamp of the DataBuffer.
<b>≡</b>	send	Send buffer to MVC card.
<b>≡</b>	send	Send DataBuffer to the MVC card.
<b>≡</b>	setDecryptionSize	Set decryption size.
<b>≡∳</b>	setInSize	Sets the transfer size.
<b>≡</b>	setKeyId	Set decryption information.
<b>≡</b>	setKeyIndex	Set decryption information.
<b>≡∳</b>	setMicValue	Set the MIC value extracted from the MXF file.
<b>≡</b>	setTimeStamp	Sets the presentation time stamp.
<b>≡</b>	setUserData	Sets user data.
<b>≡</b>	wait	Wait for transfer completion

# OverlayDataBuffer Class

	Name	Description
<b>∉</b> ∳	addRenderCommand	This is addRenderCommand, a member of class OverlayDataBuffer.
<b>=♦</b>	clearScreen	This is clearScreen, a member of class OverlayDataBuffer.
<b>=♦</b>	setDisplayDuration	This is setDisplayDuration, a member of class OverlayDataBuffer.
<b>≡♦</b>	setGlobalAlpha	This is setGlobalAlpha, a member of class OverlayDataBuffer.
<b>∉∳</b>	setGlobalPosition	This is setGlobalPosition, a member of class OverlayDataBuffer.
<b>≡♦</b>	setIncompleteRender	This is setIncompleteRender, a member of class OverlayDataBuffer.
<b>≡</b>	setSwapBuffer	This is setSwapBuffer, a member of class OverlayDataBuffer.

# **DataBuffer Operators**

		Name	Description
=+:	5	=	Assignment operator.
<u>(/-</u> =+:	5	bool	This is bool, a member of class DataBuffer.

# OverlayDataBuffer Enumerations

# OverlayDataBuffer Class

	Name	Description
<b>a</b>	EFFECTSPEED	This is record mvc2::OverlayDataBuffer::EFFECTSPEED.
e <sup>20</sup>	MOVEEFFECT	This is record mvc2::OverlayDataBuffer::MOVEEFFECT.

# OverlayDataBuffer Friends

# OverlayDataBuffer Class

Name	Description
class OverlayDecoder	This is friend friend class OverlayDecoder.

# OverlayDataBuffer Enumerations

# OverlayDataBuffer Class

	Name	Description
a <sup>a</sup>	EFFECTSPEED	This is record mvc2::OverlayDataBuffer::EFFECTSPEED.
<b>a</b>	MOVEEFFECT	This is record mvc2::OverlayDataBuffer::MOVEEFFECT.

# OverlayDataBuffer Friends

# OverlayDataBuffer Class

Name	Description
class OverlayDecoder	This is friend friend class OverlayDecoder.

## **DataBuffer Methods**

	Name	Description
<b>≡♦</b>	сору	Copy a data block to DataBuffer.
<b>=♦</b>	getBufferAddress	Gets the buffer memory address.
<b>≡</b>	getFreeSize	Get the available buffer size.
<b>=</b>	getInSize	Get the transfer size.
<b>=</b>	getTimeStamp	Get current timestamp of the DataBuffer.
<b>=</b>	send	Send buffer to MVC card.
<b>=</b>	send	Send DataBuffer to the MVC card.

<b>≡♦</b>	setDecryptionSize	Set decryption size.
<b>=♦</b>	setInSize	Sets the transfer size.
<b>=</b> ♦	setKeyld	Set decryption information.
<b>=♦</b>	setKeyIndex	Set decryption information.
<b>=♦</b>	setMicValue	Set the MIC value extracted from the MXF file.
<b>=♦</b>	setTimeStamp	Sets the presentation time stamp.
<b>=♦</b>	setUserData	Sets user data.
<b>≡</b>	wait	Wait for transfer completion

## OverlayDataBuffer Class

	Name	Description
<b>≡♦</b>	addRenderCommand	This is addRenderCommand, a member of class OverlayDataBuffer.
<b>=</b> ♦	clearScreen	This is clearScreen, a member of class OverlayDataBuffer.
<b>≟</b> ♦	setDisplayDuration	This is setDisplayDuration, a member of class OverlayDataBuffer.
<b>≡♦</b>	setGlobalAlpha	This is setGlobalAlpha, a member of class OverlayDataBuffer.
<b>≡♦</b>	setGlobalPosition	This is setGlobalPosition, a member of class OverlayDataBuffer.
<b>≡♦</b>	setIncompleteRender	This is setIncompleteRender, a member of class OverlayDataBuffer.
<b>≡♦</b>	setSwapBuffer	This is setSwapBuffer, a member of class OverlayDataBuffer.

## **DataBuffer Operators**

	Name	Description
<del>(/=</del> =+)	=	Assignment operator.
<u>(/-</u> =+)	bool	This is bool, a member of class DataBuffer.

# 2.1.28.1 OverlayDataBuffer::OverlayDataBuffer Constructor ()

## C++

OverlayDataBuffer();

## Remarks

This is OverlayDataBuffer, a member of class OverlayDataBuffer.

# 2.1.28.2 OverlayDataBuffer::OverlayDataBuffer Constructor (MvcDecoderPrivate \*, MemoryBuffer \*)

## C++

OverlayDataBuffer(MvcDecoderPrivate \* mvcdec, MemoryBuffer \* mem);

## Remarks

This is OverlayDataBuffer, a member of class OverlayDataBuffer.

# 2.1.28.3 OverlayDataBuffer Enumerations

# 2.1.28.3.1 mvc2::OverlayDataBuffer::EFFECTSPEED Enumeration

## C++

```
enum EFFECTSPEED {
   Speed_Immediately = 0,
   Speed_ExtremelyFast = 1,
   Speed_VeryFast = 4,
   Speed_Fast = 8,
   Speed_Normal = 16,
   Speed_Slow = 32,
   Speed_VerySlow = 48
};
```

#### File

mvc2api\_overlay.h

#### **Members**

Members	Description
Speed_Immediately = 0	no transition effect
Speed_ExtremelyFast = 1	approx. 1/16 second
Speed_VeryFast = 4	approx. 1/4 second
Speed_Fast = 8	approx. 1/2 second
Speed_Normal = 16	approx. 1 second
Speed_Slow = 32	approx. 2 seconds
Speed_VerySlow = 48	approx. 3 seconds

### Remarks

This is record mvc2::OverlayDataBuffer::EFFECTSPEED.

# 2.1.28.3.2 mvc2::OverlayDataBuffer::MOVEEFFECT Enumeration

## C++

```
enum MOVEEFFECT {
    MoveEffect_Linear = 0,
    MoveEffect_SlowDown = 1,
    MoveEffect_SpeedUp = 2
};
File
mvc2api_overlay.h
```

## Remarks

This is record mvc2::OverlayDataBuffer::MOVEEFFECT.

# 2.1.28.4 OverlayDataBuffer Methods

# 2.1.28.4.1 OverlayDataBuffer::addRenderCommand Method

## C++

```
TMmRc addRenderCommand(const OverlayRenderCommand & cmd);
```

## Remarks

This is addRenderCommand, a member of class OverlayDataBuffer.

# 2.1.28.4.2 OverlayDataBuffer::clearScreen Method

#### C++

```
void clearScreen();
```

#### Remarks

This is clearScreen, a member of class OverlayDataBuffer.

# 2.1.28.4.3 OverlayDataBuffer::setDisplayDuration Method

### C++

```
void setDisplayDuration(uint32_t milliseconds);
```

## Remarks

This is setDisplayDuration, a member of class OverlayDataBuffer.

# 2.1.28.4.4 OverlayDataBuffer::setGlobalAlpha Method

#### C++

```
void setGlobalAlpha(float alpha, EFFECTSPEED speed = Speed_Immediately, MOVEEFFECT effect =
MoveEffect_Linear);
```

#### Remarks

This is setGlobalAlpha, a member of class OverlayDataBuffer.

# 2.1.28.4.5 OverlayDataBuffer::setGlobalPosition Method

## C++

```
void setGlobalPosition(const FramePosition & position, EFFECTSPEED speed =
Speed_Immediately, MOVEEFFECT effect = MoveEffect_Linear);
```

### Remarks

This is setGlobalPosition, a member of class OverlayDataBuffer.

# 2.1.28.4.6 OverlayDataBuffer::setIncompleteRender Method

## C++

```
void setIncompleteRender(bool completion);
```

## Remarks

This is setIncompleteRender, a member of class OverlayDataBuffer.

# 2.1.28.4.7 OverlayDataBuffer::setSwapBuffer Method

### Стт

```
void setSwapBuffer(bool swap);
```

## Remarks

This is setSwapBuffer, a member of class OverlayDataBuffer.

# 2.1.28.5 Overlay Data Buffer Friends

# 2.1.28.5.1 friend class OverlayDecoder Friend

#### C++

friend class OverlayDecoder;

### Remarks

This is friend friend class OverlayDecoder.

# 2.1.29 OverlayDecoder Class

# **Inheritance Hierarchy**

### C++

class OverlayDecoder : public MvcDecoder;

## File

mvc2api\_overlay.h

## Remarks

This is class mvc2::OverlayDecoder.

## **Members**

## Methods

	Name	Description
<b>=</b> ♦ ₩	~MvcDecoder	This is ~MvcDecoder, a member of class MvcDecoder.
<b>≡</b>	MvcDecoder	This is MvcDecoder, a member of class MvcDecoder.
<b>≡</b>	MvcDecoder	This is MvcDecoder, a member of class MvcDecoder.
<b>≡∳</b> ?	MvcDecoder	This is MvcDecoder, a member of class MvcDecoder.

# **OverlayDecoder Class**

	Name	Description
<b>=</b> ♦ ₩	~OverlayDecoder	This is ~OverlayDecoder, a member of class OverlayDecoder.
<b>≡♦</b>	OverlayDecoder	This is OverlayDecoder, a member of class OverlayDecoder.
<b>≡♦</b>	OverlayDecoder	This is OverlayDecoder, a member of class OverlayDecoder.
<b>≡♦</b>	OverlayDecoder	This is OverlayDecoder, a member of class OverlayDecoder.

## **MvcDecoder Enumerations**

	Name	Description
<b>a</b>	DecoderType	This is record mvc2::MvcDecoder::DecoderType.
<b>a</b>	TIMEOUT	This is record mvc2::MvcDecoder::TIMEOUT.

# **OverlayDecoder Class**

	Name	Description
<b>-</b>	RESOLUTION_FLAGS	This is record
		mvc2::OverlayDecoder::RESOLUTION_FLAGS.

## **MvcDecoder Methods**

	Name	Description
<b>≡♦</b>	flush	This is flush, a member of class MvcDecoder.
<b>≡♦</b>	getBufferStatus	This is getBufferStatus, a member of class MvcDecoder.

<b>≡</b>	getChannel	Gets the used channel number.
<b>⋄</b> ₩	getDataBuffer	Return a buffer for data transfer.
<b>≡∳</b>	getDecoderErrors	This is getDecoderErrors, a member of class MvcDecoder.
<b>=∳</b>	getDecoderStatus	This is getDecoderStatus, a member of class MvcDecoder.
<b>≡∳</b>	getDecoderType	This is getDecoderType, a member of class MvcDecoder.
≡ <b>∳</b> γ	getPrivate	This is getPrivate, a member of class MvcDecoder.
<b>≡∲</b>	setBackwardPlayback	This is setBackwardPlayback, a member of class MvcDecoder.
<b>≡∳</b>	setCplUid	This is setCplUid, a member of class MvcDecoder.
•	setEndOfStream	Go to end of stream state.
<b>≡∳</b>	setSecurityManager	This is setSecurityManager, a member of class MvcDecoder.
<b>≡∳</b>	setStartDelay	Setup start delay for the decoder.
•	waitDecoderStatus	This is waitDecoderStatus, a member of class MvcDecoder.
<b>≡♦</b>	waitForTimeStamp	This is waitForTimeStamp, a member of class MvcDecoder.
<b>=</b>	waitForTransferFinish	This is waitForTransferFinish, a member of class MvcDecoder.
<b>≡∳</b>	waitForUserData	This is waitForUserData, a member of class MvcDecoder.

# **OverlayDecoder Class**

	Name	Description
<b>≡∳</b>	connectOutput	Connects an output object to the decoder.
<b>=♦</b>	deleteOverlayElement	This is deleteOverlayElement, a member of class OverlayDecoder.
<b>≡∳</b>	disconnectOutput	Disconnects the output.
<b>=♦</b> ₩	getDataBuffer	This is getDataBuffer, a member of class OverlayDecoder.
<b>=</b> ♦ ₩	getDataBuffer	This is getDataBuffer, a member of class OverlayDecoder.
<b>=</b> ♦	setOutputResolution	This is setOutputResolution, a member of class OverlayDecoder.

# **MvcDecoder Operators**

	Name	Description
=+)	=	This is =, a member of class MvcDecoder.
( <u>/-</u> =+)	bool	Returns if the object is valid.

# OverlayDecoder Class

	Name	Description
( <del>/-</del>	=	This is =, a member of class OverlayDecoder.

# **MvcDecoder Enumerations**

	Name	Description
a <sup>a</sup>	DecoderType	This is record mvc2::MvcDecoder::DecoderType.
<b>a</b>	TIMEOUT	This is record mvc2::MvcDecoder::TIMEOUT.

# OverlayDecoder Class

	Name	Description
<b>a</b>	RESOLUTION_FLAGS	This is record mvc2::OverlayDecoder::RESOLUTION_FLAGS.

# **MvcDecoder Methods**

	Name	Description
<b>≡</b>	flush	This is flush, a member of class MvcDecoder.
<b>≡</b> ∳	getBufferStatus	This is getBufferStatus, a member of class MvcDecoder.

<b>≡∳</b>	getChannel	Gets the used channel number.
<b>=♦</b> ₩	getDataBuffer	Return a buffer for data transfer.
<b>≡♦</b>	getDecoderErrors	This is getDecoderErrors, a member of class MvcDecoder.
<b>≡♦</b>	getDecoderStatus	This is getDecoderStatus, a member of class MvcDecoder.
<b>≡∳</b>	getDecoderType	This is getDecoderType, a member of class MvcDecoder.
<b>≡∳?</b>	getPrivate	This is getPrivate, a member of class MvcDecoder.
<b>≡</b>	setBackwardPlayback	This is setBackwardPlayback, a member of class MvcDecoder.
<b>≡</b>	setCplUid	This is setCplUid, a member of class MvcDecoder.
<b>≡∳</b>	setEndOfStream	Go to end of stream state.
<b>≡∳</b>	setSecurityManager	This is setSecurityManager, a member of class MvcDecoder.
<b>≡∳</b>	setStartDelay	Setup start delay for the decoder.
<b>≡∳</b>	waitDecoderStatus	This is waitDecoderStatus, a member of class MvcDecoder.
<b>≡∳</b>	waitForTimeStamp	This is waitForTimeStamp, a member of class MvcDecoder.
<b>≟</b>	waitForTransferFinish	This is waitForTransferFinish, a member of class MvcDecoder.
<b>≡</b>	waitForUserData	This is waitForUserData, a member of class MvcDecoder.

# **OverlayDecoder Class**

	Name	Description
<b>≡∳</b>	connectOutput	Connects an output object to the decoder.
<b>=♦</b>	deleteOverlayElement	This is deleteOverlayElement, a member of class OverlayDecoder.
<b>≡∳</b>	disconnectOutput	Disconnects the output.
<b>=♦</b> ₩	getDataBuffer	This is getDataBuffer, a member of class OverlayDecoder.
<b>=</b> ♦ ₩	getDataBuffer	This is getDataBuffer, a member of class OverlayDecoder.
<b>=</b> ♦	setOutputResolution	This is setOutputResolution, a member of class OverlayDecoder.

# **MvcDecoder Operators**

	Name	Description
<del>(/=</del> =+)	=	This is =, a member of class MvcDecoder.
=+)	bool	Returns if the object is valid.

# **OverlayDecoder Class**

	Name	Description
( <u>/-</u> =+)	=	This is =, a member of class OverlayDecoder.

# 2.1.29.1 OverlayDecoder::~OverlayDecoder Destructor

# C++

```
virtual ~OverlayDecoder();
```

## Remarks

This is ~OverlayDecoder, a member of class OverlayDecoder.

# 2.1.29.2 OverlayDecoder::OverlayDecoder Constructor ()

## C++

```
OverlayDecoder();
```

127

### Remarks

This is OverlayDecoder, a member of class OverlayDecoder.

# 2.1.29.3 OverlayDecoder::OverlayDecoder Constructor (OverlayDecoder&)

```
C++
```

```
OverlayDecoder(const OverlayDecoder& other);
```

#### Remarks

This is OverlayDecoder, a member of class OverlayDecoder.

# 2.1.29.4 OverlayDecoder::OverlayDecoder Constructor (TMmRc \*, MvcDevice &)

```
C++
```

```
OverlayDecoder(TMmRc * resultPointer, const MvcDevice & mvcdev);
```

### Remarks

This is OverlayDecoder, a member of class OverlayDecoder.

# 2.1.29.5 OverlayDecoder Enumerations

# 2.1.29.5.1 mvc2::OverlayDecoder::RESOLUTION\_FLAGS Enumeration

## C++

```
enum RESOLUTION_FLAGS {
   Resolution_Absolute = (0<<0),
   Resolution_Relative = (1<<0),
   Resolution_Fix_Width = (0<<1),
   Resolution_Fix_Height = (1<<1),
   Resolution_Full_Screen = ((1<<2) | (1<<0))
};</pre>
```

## File

mvc2api\_overlay.h

### Remarks

This is record mvc2::OverlayDecoder::RESOLUTION\_FLAGS.

# 2.1.29.6 OverlayDecoder Methods

# 2.1.29.6.1 OverlayDecoder::connectOutput Method

Connects an output object to the decoder.

## C++

```
TMmRc connectOutput(const VideoOutput& VideoOut);
```

128

## **Parameters**

Parameters	Description
const VideoOutput& VideoOut	dedicated VideoOutput object.

#### Returns

MMRC\_Ok or an error if the connection fails.

### **Return Values**

Return Values	Description
MMRC_Ok	Output successfully connected
MMRC_MVC2_ConnectionError	Output could not be connected, it may not be compatible

## Description

This method is used to assign an output to the decoder which is used to display the decoder frames.

# 2.1.29.6.2 OverlayDecoder::deleteOverlayElement Method

## C++

TMmRc deleteOverlayElement(const char \* elementName);

#### Remarks

This is deleteOverlayElement, a member of class OverlayDecoder.

# 2.1.29.6.3 OverlayDecoder::disconnectOutput Method

Disconnects the output.

### C++

TMmRc disconnectOutput(const VideoOutput& VideoOut);

## **Parameters**

Parameters	Description
const VideoOutput& VideoOut	previously connected output

## Returns

MMRC\_Ok if the disconnection was successful, else an error.

### **Return Values**

Return Values	Description
MMRC_Ok	successful disconnections
MMRC_MVC2_ConnectionError	object is not connected to the decoder

## Description

Disconnects a previously connected output from the decoder, so it can be reused for another decoder.

# 2.1.29.6.4 OverlayDecoder::getDataBuffer Method (OverlayDataBuffer &, uint32\_t)

## C++

virtual TMmRc getDataBuffer(OverlayDataBuffer & databuf, uint32\_t timeout = WaitInfinite);

## Remarks

This is getDataBuffer, a member of class OverlayDecoder.

# 2.1.29.6.5 OverlayDecoder::getDataBuffer Method (OverlayElementDataBuffer &, uint32\_t, uint32\_t)

#### C++

```
virtual TMmRc getDataBuffer(OverlayElementDataBuffer & databuf, uint32_t minsize = 0,
uint32_t timeout = WaitInfinite);
```

### Remarks

This is getDataBuffer, a member of class OverlayDecoder.

# 2.1.29.6.6 OverlayDecoder::setOutputResolution Method

#### C++

```
TMmRc setOutputResolution(uint32_t width, uint32_t height, uint32_t flags =
Resolution_Absolute);
```

### Remarks

This is setOutputResolution, a member of class OverlayDecoder.

# 2.1.29.7 OverlayDecoder Operators

# 2.1.29.7.1 OverlayDecoder::= Operator

#### C++

```
OverlayDecoder& operator =(const OverlayDecoder& other);
```

## Remarks

This is =, a member of class OverlayDecoder.

# 2.1.30 OverlayElementDataBuffer Class

## Inheritance Hierarchy

### C++

```
class OverlayElementDataBuffer : public DataBuffer;
```

### File

mvc2api\_overlay.h

## Remarks

This is class mvc2::OverlayElementDataBuffer.

### **Members**

## Methods

	Name	Description
<b>=</b> ♦ ₩	~DataBuffer	Destructor
<b>=</b> ♦	DataBuffer	This is DataBuffer, a member of class DataBuffer.
<b>≡⋄</b>	DataBuffer	This is DataBuffer, a member of class DataBuffer.
<b>=♦?</b>	DataBuffer	This is DataBuffer, a member of class DataBuffer.

# OverlayElementDataBuffer Class

	Name	Description
<b>≡♦</b>	OverlayElementDataBuffer	This is OverlayElementDataBuffer, a member of class OverlayElementDataBuffer.
<b>=♦</b> •	OverlayElementDataBuffer	This is OverlayElementDataBuffer, a member of class OverlayElementDataBuffer.

## **DataBuffer Methods**

	Name	Description
<b>=♦</b>	сору	Copy a data block to DataBuffer.
<b>=♦</b>	getBufferAddress	Gets the buffer memory address.
<b>=♦</b>	getFreeSize	Get the available buffer size.
<b>≡</b>	getInSize	Get the transfer size.
<b>=♦</b>	getTimeStamp	Get current timestamp of the DataBuffer.
<b>=♦</b>	send	Send buffer to MVC card.
<b>=♦</b>	send	Send DataBuffer to the MVC card.
<b>=♦</b>	setDecryptionSize	Set decryption size.
<b>≡</b>	setInSize	Sets the transfer size.
<b>=</b>	setKeyId	Set decryption information.
<b>=♦</b>	setKeyIndex	Set decryption information.
<b>=♦</b>	setMicValue	Set the MIC value extracted from the MXF file.
<b>=</b>	setTimeStamp	Sets the presentation time stamp.
<b>=</b>	setUserData	Sets user data.
<b>=♦</b>	wait	Wait for transfer completion

# OverlayElementDataBuffer Class

	Name	Description
<b>=</b> ♦	getElementName	This is getElementName, a member of class OverlayElementDataBuffer.
<b>≡∳</b>	setElementName	This is setElementName, a member of class OverlayElementDataBuffer.
<b>≡♦</b>	setPartialTransferInfo	This is setPartialTransferInfo, a member of class OverlayElementDataBuffer.

# **DataBuffer Operators**

	Name	Description
<u>(/−</u> =+)	=	Assignment operator.
<del>V=</del> =+)	bool	This is bool, a member of class DataBuffer.

# $Overlay Element Data Buffer\ Friends$

# OverlayElementDataBuffer Class

Name	Description
class OverlayDecoder	This is friend friend class OverlayDecoder.
class SubtitleDecoder	This is friend friend class SubtitleDecoder.

# OverlayElementDataBuffer Friends

# OverlayElementDataBuffer Class

Name	Description
class OverlayDecoder	This is friend friend class OverlayDecoder.
class SubtitleDecoder	This is friend friend class SubtitleDecoder.

## **DataBuffer Methods**

	Name	Description
<b>=♦</b>	сору	Copy a data block to DataBuffer.
<b>≡</b>	getBufferAddress	Gets the buffer memory address.
<b>≡</b>	getFreeSize	Get the available buffer size.
<b>≡</b>	getInSize	Get the transfer size.
<b>≡</b>	getTimeStamp	Get current timestamp of the DataBuffer.
<b>=♦</b>	send	Send buffer to MVC card.
<b>≡</b>	send	Send DataBuffer to the MVC card.
<b>≡</b>	setDecryptionSize	Set decryption size.
<b>≡</b>	setInSize	Sets the transfer size.
<b>≡</b>	setKeyId	Set decryption information.
<b>≡</b>	setKeyIndex	Set decryption information.
<b>≡</b>	setMicValue	Set the MIC value extracted from the MXF file.
<b>≡∳</b>	setTimeStamp	Sets the presentation time stamp.
<b>≡</b>	setUserData	Sets user data.
<b>=♦</b>	wait	Wait for transfer completion

## OverlayElementDataBuffer Class

	Name	Description
<b>≡♦</b>	getElementName	This is getElementName, a member of class OverlayElementDataBuffer.
<b>≡</b>	setElementName	This is setElementName, a member of class OverlayElementDataBuffer.
<b>≡♦</b>	setPartialTransferInfo	This is setPartialTransferInfo, a member of class OverlayElementDataBuffer.

# **DataBuffer Operators**

	Name	Description
=+)	=	Assignment operator.
<u>(/−</u> =+)	bool	This is bool, a member of class DataBuffer.

# 2.1.30.1 OverlayElementDataBuffer::OverlayElementDataBuffer Constructor ()

C++

explicit OverlayElementDataBuffer();

## Remarks

This is OverlayElementDataBuffer, a member of class OverlayElementDataBuffer.

# 2.1.30.2 OverlayElementDataBuffer::OverlayElementDataBuffer Constructor (MvcDecoderPrivate \*, MemoryBuffer \*)

C++

```
explicit OverlayElementDataBuffer(MvcDecoderPrivate * mvcdec, MemoryBuffer * mem);
```

## Remarks

 $This is \ Overlay Element Data Buffer, a \ member \ of \ class \ Overlay Element Data Buffer.$ 

132

# 2.1.30.3 OverlayElementDataBuffer Methods

# 2.1.30.3.1 OverlayElementDataBuffer::getElementName Method

#### C++

```
const char * getElementName(uint32_t * resourceId) const;
```

### Remarks

This is getElementName, a member of class OverlayElementDataBuffer.

# 2.1.30.3.2 OverlayElementDataBuffer::setElementName Method

### C++

```
void setElementName(const char * name, uint32_t resourceId);
```

### Remarks

This is setElementName, a member of class OverlayElementDataBuffer.

# 2.1.30.3.3 OverlayElementDataBuffer::setPartialTransferInfo Method

## C++

```
void setPartialTransferInfo(uint32_t completeSize, uint32_t offset);
```

#### Remarks

This is setPartialTransferInfo, a member of class OverlayElementDataBuffer.

# 2.1.30.4 OverlayElementDataBuffer Friends

# 2.1.30.4.1 friend class OverlayDecoder Friend

### C++

```
friend class OverlayDecoder;
```

## Remarks

This is friend friend class OverlayDecoder.

# 2.1.30.4.2 friend class SubtitleDecoder Friend

### C++

```
friend class SubtitleDecoder;
```

## Remarks

This is friend friend class SubtitleDecoder.

# 2.1.31 OverlayRenderCommand Class

# **Inheritance Hierarchy**

OverlayRenderCommand Class

## C++

# class OverlayRenderCommand;

## File

mvc2api\_overlay.h

# Remarks

This is class mvc2::OverlayRenderCommand.

## **Members**

## Methods

	Name	Description
<b>=</b> ♦ <b>W</b>	~OverlayRenderCommand	This is ~OverlayRenderCommand, a member of class OverlayRenderCommand.
=•	OverlayRenderCommand	This is OverlayRenderCommand, a member of class OverlayRenderCommand.

## **OverlayRenderCommand Data Members**

	Name	Description
49	m_clearColor	This is m_clearColor, a member of class OverlayRenderCommand.
<b>4</b> 9	m_flags	This is m_flags, a member of class OverlayRenderCommand.
49	m_reserved1	This is m_reserved1, a member of class OverlayRenderCommand.
<b>₽</b>	m_reserved2	This is m_reserved2, a member of class OverlayRenderCommand.

# OverlayRenderCommand Methods

	Name	Description
<b>≡♦</b>	clearRenderArea	This is clearRenderArea, a member of class OverlayRenderCommand.
<b>=</b> ♦ <b>?</b> 🔼	renderIntoBuffer	This is renderIntoBuffer, a member of class OverlayRenderCommand.

# OverlayRenderCommand Friends

Name	Description
class OverlayDataBuffer	This is friend friend class OverlayDataBuffer.

# **OverlayRenderCommand Data Members**

	Name	Description
<b>∳</b> ?	m_clearColor	This is m_clearColor, a member of class OverlayRenderCommand.
49	m_flags	This is m_flags, a member of class OverlayRenderCommand.
<b>∳</b> §	m_reserved1	This is m_reserved1, a member of class OverlayRenderCommand.
<b>∳</b> ≩	m_reserved2	This is m_reserved2, a member of class OverlayRenderCommand.

# OverlayRenderCommand Friends

Name	Description
class OverlayDataBuffer	This is friend friend class OverlayDataBuffer.

## OverlayRenderCommand Methods

	Name	Description
<b>≡♦</b>	clearRenderArea	This is clearRenderArea, a member of class OverlayRenderCommand.
= <b>0</b> <sub>2</sub> A	renderIntoBuffer	This is renderIntoBuffer, a member of class OverlayRenderCommand.

# 2.1.31.1 OverlayRenderCommand::~OverlayRenderCommand Destructor

C++

```
virtual ~OverlayRenderCommand();
```

## Remarks

This is ~OverlayRenderCommand, a member of class OverlayRenderCommand.

# 2.1.31.2 OverlayRenderCommand::OverlayRenderCommand Constructor

C++

OverlayRenderCommand();

### Remarks

This is OverlayRenderCommand, a member of class OverlayRenderCommand.

# 2.1.31.3 OverlayRenderCommand Data Members

# 2.1.31.3.1 OverlayRenderCommand::m\_clearColor Data Member

C++

```
uint32_t m_clearColor;
```

## Remarks

This is m\_clearColor, a member of class OverlayRenderCommand.

# 2.1.31.3.2 OverlayRenderCommand::m\_flags Data Member

C++

```
uint32_t m_flags;
```

## Remarks

This is m\_flags, a member of class OverlayRenderCommand.

# 2.1.31.3.3 OverlayRenderCommand::m\_reserved1 Data Member

C++

```
uint32_t m_reserved1;
```

### Remarks

This is m\_reserved1, a member of class OverlayRenderCommand.

# 2.1.31.3.4 OverlayRenderCommand::m\_reserved2 Data Member

#### C++

```
uint32_t m_reserved2[4];
```

### Remarks

This is m\_reserved2, a member of class OverlayRenderCommand.

# 2.1.31.4 OverlayRenderCommand Methods

# 2.1.31.4.1 OverlayRenderCommand::clearRenderArea Method

### C++

```
void clearRenderArea(uint32_t color);
```

### Remarks

This is clearRenderArea, a member of class OverlayRenderCommand.

# 2.1.31.4.2 OverlayRenderCommand::renderIntoBuffer Method

#### C++

```
virtual TMmRc renderIntoBuffer(MemoryBuffer * membuf) const = 0;
```

## Remarks

This is renderIntoBuffer, a member of class OverlayRenderCommand.

# 2.1.31.5 OverlayRenderCommand Friends

# 2.1.31.5.1 friend class OverlayDataBuffer Friend

## C++

```
friend class OverlayDataBuffer;
```

## Remarks

This is friend friend class OverlayDataBuffer.

# 2.1.32 OverlaySubtitleDataBuffer Class

## **Inheritance Hierarchy**

```
C++
```

```
class OverlaySubtitleDataBuffer : public DataBuffer;
```

### File

mvc2api overlay.h

## Remarks

This is class mvc2::OverlaySubtitleDataBuffer.

## **Members**

## Methods

	Name	Description
<b>=</b> ♦ ₩	~DataBuffer	Destructor
<b>=</b> ♦	DataBuffer	This is DataBuffer, a member of class DataBuffer.
<b>=</b> ♦	DataBuffer	This is DataBuffer, a member of class DataBuffer.
<b>≡∳9</b>	DataBuffer	This is DataBuffer, a member of class DataBuffer.

# OverlaySubtitleDataBuffer Class

	Name	Description
<b>=</b> ♦	OverlaySubtitleDataBuffer	This is OverlaySubtitleDataBuffer, a member of class OverlaySubtitleDataBuffer.
<b>=\$</b> <sub><b>?</b></sub>	OverlaySubtitleDataBuffer	This is OverlaySubtitleDataBuffer, a member of class OverlaySubtitleDataBuffer.

## **DataBuffer Methods**

	Name	Description
<b>=♦</b>	сору	Copy a data block to DataBuffer.
<b>=</b> ♦	getBufferAddress	Gets the buffer memory address.
<b>=♦</b>	getFreeSize	Get the available buffer size.
<b>=♦</b>	getInSize	Get the transfer size.
<b>=♦</b>	getTimeStamp	Get current timestamp of the DataBuffer.
<b>=</b> ♦	send	Send buffer to MVC card.
<b>=♦</b>	send	Send DataBuffer to the MVC card.
<b>=♦</b>	setDecryptionSize	Set decryption size.
<b>≡♦</b>	setInSize	Sets the transfer size.
<b>=♦</b>	setKeyId	Set decryption information.
<b>=♦</b>	setKeyIndex	Set decryption information.
<b>≡♦</b>	setMicValue	Set the MIC value extracted from the MXF file.
<b>≡</b>	setTimeStamp	Sets the presentation time stamp.
<b>≡</b>	setUserData	Sets user data.
<b>≡</b>	wait	Wait for transfer completion

# OverlaySubtitleDataBuffer Class

	Name	Description
<b>≡</b>	getResourceId	This is getResourceld, a member of class OverlaySubtitleDataBuffer.
<b>=</b> ♦	setTimeValues	This is setTimeValues, a member of class OverlaySubtitleDataBuffer.

# **DataBuffer Operators**

		Name	Description
€	<del>=</del>	=	Assignment operator.
€	+)	bool	This is bool, a member of class DataBuffer.

# OverlaySubtitleDataBuffer Friends

## OverlaySubtitleDataBuffer Class

Name	Description
class SubtitleDecoder	This is friend friend class SubtitleDecoder.

## OverlaySubtitleDataBuffer Friends

# OverlaySubtitleDataBuffer Class

Name	Description
class SubtitleDecoder	This is friend friend class SubtitleDecoder.

## **DataBuffer Methods**

	Name	Description
<b>=♦</b>	сору	Copy a data block to DataBuffer.
<b>=♦</b>	getBufferAddress	Gets the buffer memory address.
<b>≡</b>	getFreeSize	Get the available buffer size.
<b>≡</b>	getInSize	Get the transfer size.
<b>=♦</b>	getTimeStamp	Get current timestamp of the DataBuffer.
<b>≡</b>	send	Send buffer to MVC card.
<b>≡</b>	send	Send DataBuffer to the MVC card.
<b>=♦</b>	setDecryptionSize	Set decryption size.
<b>=♦</b>	setInSize	Sets the transfer size.
<b>=♦</b>	setKeyId	Set decryption information.
<b>=♦</b>	setKeyIndex	Set decryption information.
<b>≡</b>	setMicValue	Set the MIC value extracted from the MXF file.
<b>≡</b>	setTimeStamp	Sets the presentation time stamp.
<b>=♦</b>	setUserData	Sets user data.
<b>=♦</b>	wait	Wait for transfer completion

## OverlaySubtitleDataBuffer Class

	Name	Description
<b>∉</b> ∳	getResourceId	This is getResourceId, a member of class OverlaySubtitleDataBuffer.
<b>=♦</b>	setTimeValues	This is setTimeValues, a member of class OverlaySubtitleDataBuffer.

# **DataBuffer Operators**

	Name	Description
<del>(/=</del>	=	Assignment operator.
<del>(/=</del> =+)	bool	This is bool, a member of class DataBuffer.

# 2.1.32.1 OverlaySubtitleDataBuffer::OverlaySubtitleDataBuffer Constructor ()

### C++

explicit OverlaySubtitleDataBuffer();

## Remarks

This is OverlaySubtitleDataBuffer, a member of class OverlaySubtitleDataBuffer.

# 2.1.32.2 OverlaySubtitleDataBuffer::OverlaySubtitleDataBuffer Constructor (MvcDecoderPrivate \*, MemoryBuffer \*)

C++

```
explicit OverlaySubtitleDataBuffer(MvcDecoderPrivate * mvcdec, MemoryBuffer * mem);
```

### Remarks

This is OverlaySubtitleDataBuffer, a member of class OverlaySubtitleDataBuffer.

# 2.1.32.3 OverlaySubtitleDataBuffer Methods

# 2.1.32.3.1 OverlaySubtitleDataBuffer::getResourceld Method

C++

```
uint32_t getResourceId() const;
```

## Remarks

This is getResourceld, a member of class OverlaySubtitleDataBuffer.

# 2.1.32.3.2 OverlaySubtitleDataBuffer::setTimeValues Method

C++

```
void setTimeValues(int32_t timeOffset = 0, uint32_t startTime = 0, uint32_t endTime =
UINT32_MAX);
```

## Remarks

This is setTimeValues, a member of class OverlaySubtitleDataBuffer.

# 2.1.32.4 OverlaySubtitleDataBuffer Friends

# 2.1.32.4.1 friend class SubtitleDecoder Friend

C++

```
friend class SubtitleDecoder;
```

## Remarks

This is friend friend class SubtitleDecoder.

# 2.1.33 PCMDataBuffer Class

## Inheritance Hierarchy

```
C++
```

```
class PCMDataBuffer : public DataBuffer;
```

File

mvc2api\_decoder.h

## Remarks

This is class mvc2::PCMDataBuffer.

## **Members**

## Methods

	Name	Description
<b>=</b> ♦ <b>W</b>	~DataBuffer	Destructor
<b>≡♦</b>	DataBuffer	This is DataBuffer, a member of class DataBuffer.
<b>≡♦</b>	DataBuffer	This is DataBuffer, a member of class DataBuffer.
<b>=♦</b> •	DataBuffer	This is DataBuffer, a member of class DataBuffer.

# **PCMDataBuffer Class**

	Name	Description
<b>■♦ ₩</b>	~PCMDataBuffer	This is ~PCMDataBuffer, a member of class PCMDataBuffer.
<b>≡♦</b>	PCMDataBuffer	This is PCMDataBuffer, a member of class PCMDataBuffer.
<b>=♦</b> ?	PCMDataBuffer	This is PCMDataBuffer, a member of class PCMDataBuffer.

# **DataBuffer Methods**

	Name	Description
<b>≡♦</b>	сору	Copy a data block to DataBuffer.
<b>=♦</b>	getBufferAddress	Gets the buffer memory address.
<b>=♦</b>	getFreeSize	Get the available buffer size.
<b>=♦</b>	getInSize	Get the transfer size.
<b>=♦</b>	getTimeStamp	Get current timestamp of the DataBuffer.
<b>=♦</b>	send	Send buffer to MVC card.
<b>=♦</b>	send	Send DataBuffer to the MVC card.
<b>≡</b>	setDecryptionSize	Set decryption size.
<b>≡</b>	setInSize	Sets the transfer size.
<b>≡∳</b>	setKeyld	Set decryption information.
<b>≡∳</b>	setKeyIndex	Set decryption information.
<b>≡∳</b>	setMicValue	Set the MIC value extracted from the MXF file.
<b>≡</b>	setTimeStamp	Sets the presentation time stamp.
<b>≡♦</b>	setUserData	Sets user data.
<b>≡</b>	wait	Wait for transfer completion

## **PCMDataBuffer Class**

	Name	Description
<b>=</b> ♦	setBitsPerSample	This is setBitsPerSample, a member of class PCMDataBuffer.
<b>≡♦</b>	setChannelMapping	This is setChannelMapping, a member of class PCMDataBuffer.
<b>=</b> ♦	setMixingChannel	This is setMixingChannel, a member of class PCMDataBuffer.
<b>≡♦</b>	setNumberOfChannels	This is setNumberOfChannels, a member of class PCMDataBuffer.
<b>≟</b>	setSampleFrequency	This is setSampleFrequency, a member of class PCMDataBuffer.

# **DataBuffer Operators**

	Name	Description
=+)	=	Assignment operator.

- 0	/		
- 13	+)	hool	This is bool, a member of class DataBuffer.
		5001	This is book, a mornbor of slass Balabaner.

# **PCMDataBuffer Friends**

# **PCMDataBuffer Class**

Name	Description
class PCMDecoder	This is friend friend class PCMDecoder.

# **PCMDataBuffer Friends**

# **PCMDataBuffer Class**

Name		Description
class PCMDecoder	(J)	This is friend friend class PCMDecoder.

## **DataBuffer Methods**

	Name	Description
<b>=♦</b>	сору	Copy a data block to DataBuffer.
<b>=♦</b>	getBufferAddress	Gets the buffer memory address.
<b>≡</b>	getFreeSize	Get the available buffer size.
<b>≡</b>	getInSize	Get the transfer size.
<b>=</b>	getTimeStamp	Get current timestamp of the DataBuffer.
<b>≡</b>	send	Send buffer to MVC card.
<b>≡</b>	send	Send DataBuffer to the MVC card.
<b>∉</b> ∳	setDecryptionSize	Set decryption size.
<b>≡</b>	setInSize	Sets the transfer size.
<b>≡</b>	setKeyld	Set decryption information.
<b>≡∳</b>	setKeyIndex	Set decryption information.
<b>≡∳</b>	setMicValue	Set the MIC value extracted from the MXF file.
<b>≡</b>	setTimeStamp	Sets the presentation time stamp.
<b>≡</b>	setUserData	Sets user data.
<b>≡</b>	wait	Wait for transfer completion

# PCMDataBuffer Class

	Name	Description
<b>≡</b> •	setBitsPerSample	This is setBitsPerSample, a member of class PCMDataBuffer.
<b>≡∳</b>	setChannelMapping	This is setChannelMapping, a member of class PCMDataBuffer.
<b>≡∳</b>	setMixingChannel	This is setMixingChannel, a member of class PCMDataBuffer.
<b>≡∳</b>	setNumberOfChannels	This is setNumberOfChannels, a member of class PCMDataBuffer.
<b>≡</b>	setSampleFrequency	This is setSampleFrequency, a member of class PCMDataBuffer.

# **DataBuffer Operators**

		Name	Description
9	(+)	=	Assignment operator.
9	<del>-</del>	bool	This is bool, a member of class DataBuffer.

# 2.1.33.1 PCMDataBuffer::~PCMDataBuffer Destructor

#### C++

```
virtual ~PCMDataBuffer();
```

#### Remarks

This is ~PCMDataBuffer, a member of class PCMDataBuffer.

# 2.1.33.2 PCMDataBuffer::PCMDataBuffer Constructor ()

## C++

```
PCMDataBuffer();
```

#### Remarks

This is PCMDataBuffer, a member of class PCMDataBuffer.

# 2.1.33.3 PCMDataBuffer::PCMDataBuffer Constructor (MvcDecoderPrivate \*, MemoryBuffer \*)

### C++

```
PCMDataBuffer(MvcDecoderPrivate * mvcdec, MemoryBuffer * mem);
```

## Remarks

This is PCMDataBuffer, a member of class PCMDataBuffer.

# 2.1.33.4 PCMDataBuffer Methods

# 2.1.33.4.1 PCMDataBuffer::setBitsPerSample Method

### C++

```
TMmRc setBitsPerSample(uint8_t bitsPerSample);
```

## Remarks

This is setBitsPerSample, a member of class PCMDataBuffer.

# 2.1.33.4.2 PCMDataBuffer::setChannelMapping Method

## C++

```
TMmRc setChannelMapping(const uint8_t * mapping, uint32_t mappingSize);
```

# Remarks

This is setChannelMapping, a member of class PCMDataBuffer.

# 2.1.33.4.3 PCMDataBuffer::setMixingChannel Method

### C++

```
TMmRc setMixingChannel(uint8_t channel, const float * mixingValues, uint32_t mixingSize);
```

## Remarks

This is setMixingChannel, a member of class PCMDataBuffer.



# 2.1.33.4.4 PCMDataBuffer::setNumberOfChannels Method

#### C++

TMmRc setNumberOfChannels(uint8\_t numberOfchannels);

#### Remarks

This is setNumberOfChannels, a member of class PCMDataBuffer.

# 2.1.33.4.5 PCMDataBuffer::setSampleFrequency Method



### C++

TMmRc setSampleFrequency(uint32\_t frequency);

#### Remarks

This is setSampleFrequency, a member of class PCMDataBuffer.

# 2.1.33.5 PCMDataBuffer Friends

# 2.1.33.5.1 friend class PCMDecoder Friend

#### C++

friend class PCMDecoder;

## Remarks

This is friend friend class PCMDecoder.

# 2.1.34 PCMDecoder Class

## Inheritance Hierarchy

## C++

class PCMDecoder : public MvcDecoder;

## File

mvc2api\_decoder.h

## Remarks

The PCM decoder class does the pre-processing of PCM samples before transferring it to an Audio-Output.

# Members

# Methods

	Name	Description
<b>=</b> ♦ ₩	~MvcDecoder	This is ~MvcDecoder, a member of class MvcDecoder.
<b>≡⋄</b>	MvcDecoder	This is MvcDecoder, a member of class MvcDecoder.
<b>≡♦</b>	MvcDecoder	This is MvcDecoder, a member of class MvcDecoder.
<b>=♦</b> •	MvcDecoder	This is MvcDecoder, a member of class MvcDecoder.

# **PCMDecoder Class**

	Name	Description
<b>=♦</b> ₩	~PCMDecoder	This is ~PCMDecoder, a member of class PCMDecoder.
<b>≡</b>	PCMDecoder	This is PCMDecoder, a member of class PCMDecoder.
<b>≡</b>	PCMDecoder	This is PCMDecoder, a member of class PCMDecoder.
<b>≡</b>	PCMDecoder	Creates a PCMDecoder object.

# **MvcDecoder Enumerations**

	Name	Description
a <sup>2</sup>	DecoderType	This is record mvc2::MvcDecoder::DecoderType.
e P	TIMEOUT	This is record mvc2::MvcDecoder::TIMEOUT.

## **MvcDecoder Methods**

	Name	Description
<b>≡</b>	flush	This is flush, a member of class MvcDecoder.
<b>≡♦</b>	getBufferStatus	This is getBufferStatus, a member of class MvcDecoder.
<b>=♦</b>	getChannel	Gets the used channel number.
<b>=♦</b> ₩	getDataBuffer	Return a buffer for data transfer.
<b>=♦</b>	getDecoderErrors	This is getDecoderErrors, a member of class MvcDecoder.
<b>≡</b>	getDecoderStatus	This is getDecoderStatus, a member of class MvcDecoder.
<b>=♦</b>	getDecoderType	This is getDecoderType, a member of class MvcDecoder.
<b>≡\$</b> }	getPrivate	This is getPrivate, a member of class MvcDecoder.
<b>≡</b>	setBackwardPlayback	This is setBackwardPlayback, a member of class MvcDecoder.
<b>≡∳</b>	setCplUid	This is setCplUid, a member of class MvcDecoder.
<b>≡∳</b>	setEndOfStream	Go to end of stream state.
<b>≡</b>	setSecurityManager	This is setSecurityManager, a member of class MvcDecoder.
<b>≡</b>	setStartDelay	Setup start delay for the decoder.
<b>≡∳</b>	waitDecoderStatus	This is waitDecoderStatus, a member of class MvcDecoder.
<b>=♦</b>	waitForTimeStamp	This is waitForTimeStamp, a member of class MvcDecoder.
<b>≡</b>	waitForTransferFinish	This is waitForTransferFinish, a member of class MvcDecoder.
<b>∉</b>	waitForUserData	This is waitForUserData, a member of class MvcDecoder.

# **PCMDecoder Class**

	Name	Description
<b>≡</b>	connectOutput	Connects the audio output.
<b>≡</b>	disconnectOutput	Disconnects the audio output.
<b>=♦</b>	getBitsPerSample	Gets bits per sample.
<b>=♦</b>	getDecoderErrors	This is getDecoderErrors, a member of class PCMDecoder.
<b>=♦</b>	getDecoderStatus	This is getDecoderStatus, a member of class PCMDecoder.
<b>=♦</b>	getNumberOfChannels	Gets number of channels.
<b>=♦</b>	getPCMDataBuffer	This is getPCMDataBuffer, a member of class PCMDecoder.
<b>≡</b> ∳	getSampleFrequency	This is getSampleFrequency, a member of class PCMDecoder.
<b>=</b>	setBitsPerSample	This is setBitsPerSample, a member of class PCMDecoder.
<b>≡</b>	setChannelMapping	This is setChannelMapping, a member of class PCMDecoder.
<b>≡</b>	setDolbyPrologicIIChannels	This is setDolbyPrologicIIChannels, a member of class PCMDecoder.
<b>=♦</b>	setMixingChannel	This is setMixingChannel, a member of class PCMDecoder.

<b>∉</b> ∳	setNumberOfChannels	This is setNumberOfChannels, a member of class PCMDecoder.
<b>≟</b> ∳	setSampleFrequency	This is setSampleFrequency, a member of class PCMDecoder.
<b>≡♦</b>	waitDecoderStatus	This is waitDecoderStatus, a member of class PCMDecoder.

# **MvcDecoder Operators**

	N	lame	Description
=+)	=		This is =, a member of class MvcDecoder.
=+)	bo	ool	Returns if the object is valid.

## **PCMDecoder Class**

	Name	Description
<u>(/−</u> =+)	=	This is =, a member of class PCMDecoder.

# **PCMDecoder Data Members**

# **PCMDecoder Class**

	Name	Description
•	CHANNEL_MAPPING_FLAGS	This is CHANNEL_MAPPING_FLAGS, a member of class
		PCMDecoder.

## **PCMDecoder Data Members**

# **PCMDecoder Class**

	Name	Description
•	CHANNEL_MAPPING_FLAGS	This is CHANNEL_MAPPING_FLAGS, a member of class PCMDecoder.

# **MvcDecoder Enumerations**

	Name	Description
a <sup>2</sup>	DecoderType	This is record mvc2::MvcDecoder::DecoderType.
e P	TIMEOUT	This is record mvc2::MvcDecoder::TIMEOUT.

## **MvcDecoder Methods**

	Name	Description
<b>≡</b>	flush	This is flush, a member of class MvcDecoder.
<b>≡</b>	getBufferStatus	This is getBufferStatus, a member of class MvcDecoder.
<b>≡</b>	getChannel	Gets the used channel number.
<b>=♦</b> ₩	getDataBuffer	Return a buffer for data transfer.
<b>≡</b>	getDecoderErrors	This is getDecoderErrors, a member of class MvcDecoder.
<b>=</b>	getDecoderStatus	This is getDecoderStatus, a member of class MvcDecoder.
<b>=</b>	getDecoderType	This is getDecoderType, a member of class MvcDecoder.
<b>=\$</b>	getPrivate	This is getPrivate, a member of class MvcDecoder.
<b>∉</b> ∳	setBackwardPlayback	This is setBackwardPlayback, a member of class MvcDecoder.
<b>≡</b>	setCplUid	This is setCplUid, a member of class MvcDecoder.
<b>=</b>	setEndOfStream	Go to end of stream state.
<b>=</b>	setSecurityManager	This is setSecurityManager, a member of class MvcDecoder.
<b>≡</b>	setStartDelay	Setup start delay for the decoder.
<b>≡</b>	waitDecoderStatus	This is waitDecoderStatus, a member of class MvcDecoder.
<b>≡</b>	waitForTimeStamp	This is waitForTimeStamp, a member of class MvcDecoder.
<b>∉</b> ∳	waitForTransferFinish	This is waitForTransferFinish, a member of class MvcDecoder.

		This is a selection to a Detail a second and at all a selection of the sel
	waitForUserData	This is waitForUserData, a member of class MvcDecoder.
	Walli 010301Data	Triis is waiti bioscibata, a member of diass invebedder.

## **PCMDecoder Class**

	Name	Description
<b>≡</b>	connectOutput	Connects the audio output.
<b>≡</b>	disconnectOutput	Disconnects the audio output.
<b>≡</b>	getBitsPerSample	Gets bits per sample.
<b>≡</b>	getDecoderErrors	This is getDecoderErrors, a member of class PCMDecoder.
<b>≡</b>	getDecoderStatus	This is getDecoderStatus, a member of class PCMDecoder.
<b>≡</b>	getNumberOfChannels	Gets number of channels.
<b>≡</b>	getPCMDataBuffer	This is getPCMDataBuffer, a member of class PCMDecoder.
<b>∉</b> ∳	getSampleFrequency	This is getSampleFrequency, a member of class PCMDecoder.
<b>=</b>	setBitsPerSample	This is setBitsPerSample, a member of class PCMDecoder.
<b>=♦</b>	setChannelMapping	This is setChannelMapping, a member of class PCMDecoder.
<b>≓</b>	setDolbyPrologicIIChannels	This is setDolbyPrologicIIChannels, a member of class PCMDecoder.
<b>≡</b>	setMixingChannel	This is setMixingChannel, a member of class PCMDecoder.
<b>≟</b>	setNumberOfChannels	This is setNumberOfChannels, a member of class PCMDecoder.
<b>≟</b>	setSampleFrequency	This is setSampleFrequency, a member of class PCMDecoder.
<b>=</b>	waitDecoderStatus	This is waitDecoderStatus, a member of class PCMDecoder.

## **MvcDecoder Operators**

	Name	Description
=+)	=	This is =, a member of class MvcDecoder.
=+)	bool	Returns if the object is valid.

# **PCMDecoder Class**

	Name	Description
=+)	=	This is =, a member of class PCMDecoder.

# 2.1.34.1 PCMDecoder::~PCMDecoder Destructor

## C++

```
virtual ~PCMDecoder();
```

## Remarks

This is ~PCMDecoder, a member of class PCMDecoder.

# 2.1.34.2 PCMDecoder::PCMDecoder Constructor ()

## C++

```
PCMDecoder();
```

## Remarks

This is PCMDecoder, a member of class PCMDecoder.

# 2.1.34.3 PCMDecoder::PCMDecoder Constructor (PCMDecoder&)

#### C++

```
PCMDecoder(const PCMDecoder& other);
```

#### Remarks

This is PCMDecoder, a member of class PCMDecoder.

# 2.1.34.4 PCMDecoder::PCMDecoder Constructor (TMmRc \*, MvcDevice &, uint32\_t, uint32\_t, uint32\_t)

Creates a PCMDecoder object.

#### C++

```
PCMDecoder(TMmRc * resultPointer, const MvcDevice & mvcDevice, uint32_t bitsPerSample,
uint32_t numberOfChannels, uint32_t channelMask = 0xffffffff);
```

#### **Parameters**

Parameters	Description
TMmRc * resultPointer	Pointer where to store the result in. if 0 no result will be written.
const MvcDevice & mvcDevice	MVC device where the decoder will be created
uint32_t bitsPerSample	bits per sample to process (possible values: 16 and 24)
uint32_t numberOfChannels	number of channels
uint32_t channelMask = 0xffffffff	reserved

## **Return Values**

Return Values	Description
MMRC_Ok	Decoder created successfully
MMRC_MVC2_OutOfDecoderResources	Decoder could not be created, because there are no more resources available

## Description

This method will create a PCMDecoder object if possible. The PCM decoder supports various sample formats and and a number of input channels. The sample data must be channel interleave, so it starts with the first sample of channel 0 followed by the first sample of channel 1 and so on. If BitsPerChannel is 16 one sample consists of two bytes in little-endian order. If BitsPerSample is 17 to 24, a samples has 3 bytes in little-endian byte order.

# 2.1.34.5 PCMDecoder Data Members

# 2.1.34.5.1 PCMDecoder::CHANNEL\_MAPPING\_FLAGS Data Member

## C++

```
enum {
   ChannelMapping_UnusedChannel = 0xff,
   ChannelMapping_MixedChannel = 0x80
} CHANNEL_MAPPING_FLAGS;
```

## Remarks

This is CHANNEL\_MAPPING\_FLAGS, a member of class PCMDecoder.

# 2.1.34.6 PCMDecoder Methods



# 2.1.34.6.1 PCMDecoder::connectOutput Method

Connects the audio output.

### C++

TMmRc connectOutput(const AudioOutput& audioOutput);

#### **Parameters**

Parameters	Description
AudioOut	AudioOutput object

### Returns

MMRC\_Ok if the output was connected successfully, else an error.

# **Return Values**

Return Values	Description
MMRC_Ok	Connection successful
MMRC_MVC2_ConnectionError	Connection could not be established

## Description

This method assigns a AudioOutput to the decoder which is used to play the sample data to. The PCM data will be process to fit the capabilities of the output.

# 2.1.34.6.2 PCMDecoder::disconnectOutput Method

Disconnects the audio output.

## C++

TMmRc disconnectOutput(const AudioOutput& audioOutput);

## **Parameters**

Parameters	Description
const AudioOutput& audioOutput	Previously connected AudioOutput object

## Returns

MMRC\_Ok if the disconnection was successful, else an error.

## **Return Values**

Return Values	Description
MMRC_Ok	Disconnection successful
MMRC_MVC2_ConnectionError	an error occurred while disconnecting

# Description

A previously connected output could be reused by another decoder. Before this, it has to be disconnected.

# 2.1.34.6.3 PCMDecoder::getBitsPerSample Method

Gets bits per sample.

## C++

uint32\_t getBitsPerSample() const;

148

## **Returns**

Number of bits per sample.

## Description

This method returns the number of bits per sample currently in use.

# 2.1.34.6.4 PCMDecoder::getDecoderErrors Method

#### C++

```
TMmRc getDecoderErrors(AudioDecoderErrors & errors) const;
```

#### Remarks

This is getDecoderErrors, a member of class PCMDecoder.

# 2.1.34.6.5 PCMDecoder::getDecoderStatus Method

### C++

```
TMmRc getDecoderStatus(AudioDecoderStatus & status) const;
```

### Remarks

This is getDecoderStatus, a member of class PCMDecoder.

# 2.1.34.6.6 PCMDecoder::getNumberOfChannels Method

Gets number of channels.

### C++

```
uint32_t getNumberOfChannels() const;
```

## Returns

Number of channels.

## Description

This method returns the number of channels used by the decoder. It's normally the value specified at create().

# 2.1.34.6.7 PCMDecoder::getPCMDataBuffer Method

# C++

```
TMmRc getPCMDataBuffer(PCMDataBuffer & databuf, uint32_t minsize = 0, uint32_t timeout =
WaitInfinite);
```

## Remarks

This is getPCMDataBuffer, a member of class PCMDecoder.

# 2.1.34.6.8 PCMDecoder::getSampleFrequency Method

## C++

```
uint32_t getSampleFrequency() const;
```

## Remarks

This is getSampleFrequency, a member of class PCMDecoder.

# 2.1.34.6.9 PCMDecoder::setBitsPerSample Method

#### C++

TMmRc setBitsPerSample(uint8\_t bitsPerSample);

#### Remarks

This is setBitsPerSample, a member of class PCMDecoder.

# 2.1.34.6.10 PCMDecoder::setChannelMapping Method



C++

TMmRc setChannelMapping(const uint8\_t \* mapping, uint32\_t mappingSize);

## Remarks

This is setChannelMapping, a member of class PCMDecoder.

# 2.1.34.6.11 PCMDecoder::setDolbyPrologicIIChannels Method



TMmRc setDolbyPrologicIIChannels(uint8\_t destChannelLeft, uint8\_t destChannelRight, uint8\_t srcLeft, uint8\_t srcRight, uint8\_t srcCenter, uint8\_t srcSLeft, uint8\_t srcSRight);

### Remarks

This is setDolbyPrologicIIChannels, a member of class PCMDecoder.

# 2.1.34.6.12 PCMDecoder::setMixingChannel Method



C++

TMmRc setMixingChannel(uint8\_t channel, const float \* mixingValues, uint32\_t mixingSize);

### Remarks

This is setMixingChannel, a member of class PCMDecoder.

# 2.1.34.6.13 PCMDecoder::setNumberOfChannels Method

### C++



TMmRc setNumberOfChannels(uint8\_t numberOfchannels);

## Remarks

This is setNumberOfChannels, a member of class PCMDecoder.

# 2.1.34.6.14 PCMDecoder::setSampleFrequency Method

## C++

TMmRc setSampleFrequency(uint32\_t frequency);



### Remarks

This is setSampleFrequency, a member of class PCMDecoder.

# 2.1.34.6.15 PCMDecoder::waitDecoderStatus Method

#### C++

TMmRc waitDecoderStatus(AudioDecoderStatus & status) const;

### Remarks

This is waitDecoderStatus, a member of class PCMDecoder.

# 2.1.34.7 PCMDecoder Operators

# 2.1.34.7.1 PCMDecoder::= Operator

### C++

PCMDecoder& operator =(const PCMDecoder& other);

### Remarks

This is =, a member of class PCMDecoder.

# 2.1.35 PlaybackControl Class

## **Inheritance Hierarchy**

## C++

class PlaybackControl;

## File

mvc2api.h

## Remarks

This class controls the playback and the decoders.

## **Members**

## Methods

	Name	Description
<b>=</b> ♦ <b>W</b>	~PlaybackControl	Default destructor
<b>=</b> ♦	PlaybackControl	This is PlaybackControl, a member of class PlaybackControl.
<b>=</b> ♦	PlaybackControl	This is PlaybackControl, a member of class PlaybackControl.
<b>=</b> ♦	PlaybackControl	Creates a PlaybackControl object.

## **PlaybackControl Enumerations**

	Name	Description
<b>=</b>	PLAYBACK_STATE	Playback state.

## **PlaybackControl Methods**

	Name	Description
<b>≡⋄</b>	connect	Connect the playback control with a decoder.
<b>≡⋄</b>	disconnect	Disconnect a decoder from the playback control.
<b>=</b> ♦	flush	This is flush, a member of class PlaybackControl.
<b>≡♦</b>	getState	Gets the current state.

<b>=</b> ♦	pause	Pauses the video/audio
<b>≡♦</b>	run	Starts the playback.
<b>=♦</b>	runSpeed	Start playback with specific speed.
<b>=♦</b>	setSyncSlave	This is setSyncSlave, a member of class PlaybackControl.
<b>=</b>	singleStep	Single step a frame.
<b>≡</b>	stop	Stops the playback.
<b>≡</b>	waitForEndOfStream	Waits for end of stream.

# **PlaybackControl Operators**

	Name	Description
=+)	=	This is =, a member of class PlaybackControl.
<u>(/-</u>	bool	This is bool, a member of class PlaybackControl.

## **PlaybackControl Enumerations**

	Name	Description
	PLAYBACK_STATE	Playback state.

## **PlaybackControl Methods**

	Name	Description
<b>∉</b>	connect	Connect the playback control with a decoder.
<b>∉</b>	disconnect	Disconnect a decoder from the playback control.
<b>=♦</b>	flush	This is flush, a member of class PlaybackControl.
<b>=♦</b>	getState	Gets the current state.
<b>=♦</b>	pause	Pauses the video/audio
<b>=♦</b>	run	Starts the playback.
<b>=♦</b>	runSpeed	Start playback with specific speed.
<b>=♦</b>	setSyncSlave	This is setSyncSlave, a member of class PlaybackControl.
<b>=♦</b>	singleStep	Single step a frame.
<b>=</b>	stop	Stops the playback.
<b>=♦</b>	waitForEndOfStream	Waits for end of stream.

# **PlaybackControl Operators**

	Name	Description
<del>(/=</del>	=	This is =, a member of class PlaybackControl.
<u>(/-</u> =+)	bool	This is bool, a member of class PlaybackControl.

# 2.1.35.1 PlaybackControl::~PlaybackControl Destructor

Default destructor

C++

virtual ~PlaybackControl();

# Description

The PlaybackControl object will be freed and all of it's resources.

# 2.1.35.2 PlaybackControl::PlaybackControl Constructor ()

## C++

PlaybackControl();

## Remarks

This is PlaybackControl, a member of class PlaybackControl.

# 2.1.35.3 PlaybackControl::PlaybackControl Constructor (PlaybackControl&)

#### C++

```
PlaybackControl(const PlaybackControl& other);
```

#### Remarks

This is PlaybackControl, a member of class PlaybackControl.

# 2.1.35.4 PlaybackControl::PlaybackControl Constructor (TMmRc \*, MvcDevice &)

Creates a PlaybackControl object.

## C++

```
PlaybackControl(TMmRc * resultPointer, const MvcDevice & mvcdev);
```

#### **Parameters**

Parameters	Description
TMmRc * resultPointer	Pointer where to store the result in. if 0 no result will be written.
const MvcDevice & mvcdev	MvcDevice which should be controlled

# Returns

MMRC\_Ok will be returned if the creation was successful, otherwise an error.

### **Return Values**

Description
operation successfully completed
object could not be created because the device is already in use

# Description

This method will create a PlaybackControl object and attach it to a given MvcDevice.

# 2.1.35.5 PlaybackControl Enumerations

# 2.1.35.5.1 mvc2::PlaybackControl::PLAYBACK\_STATE Enumeration

## C++

```
enum PLAYBACK_STATE {
   Stopped = 0,
   Running = 1,
   Paused = 2
};
```

File

mvc2api.h

## **Members**

Members	Description
Stopped = 0	The card is in stopped state. No video will be displayed until run() or pause() was called.
Running = 1	The card is currently in running state. Video and audio will be played.
Paused = 2	The MVC card is in paused state. A freeze frame will be displayed or the card waits for the first decoded frame.

## Remarks

Playback state.

# 2.1.35.6 PlaybackControl Methods

# 2.1.35.6.1 PlaybackControl: connect Method



Connect the playback control with a decoder.

## C++

TMmRc connect(const MvcDecoder& dec);

## **Parameters**

Parameters	Description
const MvcDecoder& dec	decoder which will be connected to the playback control

## **Returns**

MMRC\_Ok if the connect succeeded, else an error

## **Return Values**

Return Values	Description
MMRC_Ok	connection successful

## Description

To control decoders with a playback control, the playback control must be connected to the decoders.

# 2.1.35.6.2 PlaybackControl::disconnect Method

Disconnect a decoder from the playback control.

### C++

TMmRc disconnect(const MvcDecoder& dec);

## **Parameters**

Parameters	Description
const MvcDecoder& dec	decoder which will be disconnected from the playback control

### Returns

MMRC\_Ok if the disconnect succeeded, else an error

## **Return Values**

Return Values	Description
MMRC_Ok	disconnect successful

# 2.1.35.6.3 PlaybackControl: flush Method



#### C++

```
TMmRc flush(uint32_t keepTime, uint32_t * nextTimeStamp);
```

### Remarks

This is flush, a member of class PlaybackControl.

# 2.1.35.6.4 PlaybackControl::getState Method

Gets the current state.



### C++

```
PLAYBACK_STATE getState(uint32_t * timeStamp = 0);
```

#### **Parameters**

Parameters	Description
uint32_t * timeStamp = 0	current time stamp used for play out (in 90kHz ticks)

## Returns

A PlaybackControl::PLAYBACK\_STATE will be returned.

## Description

This method will return the current active PlaybackControl state.

# 2.1.35.6.5 PlaybackControl::pause Method

Pauses the video/audio



### C++

```
TMmRc pause();
```

## Returns

MMRC\_Ok will be returned if the operation was successful, otherwise an error.

## **Return Values**

Return Values	Description
MMRC_Ok	operation successfully completed

# **Description**

After calling pause() the video and audio will freeze immediately. Audio mutes and video displays a still frame. If the video is not running, pause can be activated before. This display the first decoded video frame right after it was decoded. The playback can then be started with run() method.

# 2.1.35.6.6 PlaybackControl::run Method

Starts the playback.



### C++

```
TMmRc run();
```

## Returns

MMRC\_Ok will be returned if the operation was successful, otherwise an error.

## **Return Values**

Return Values	Description
MMRC_Ok	operation successfully completed

## Description

After calling the run() method, the playback will start.

# 2.1.35.6.7 PlaybackControl::runSpeed Method

Start playback with specific speed.



TMmRc runSpeed(float speed);

## **Parameters**

Parameters	Description
float speed	playback speed, 1.0 is normal speed

## **Returns**

Returns MMRC\_Ok if playback started successfully, otherwise an error.

### **Return Values**

Return Values	Description
MMRC_Ok	playback started

## Description

This function works like run() function, but here a running speed can be set. The speed should be bigger than 0 and not to high (not more than 3), because maximum the speed is limited by the decoding speed. For higher playback speed the player need to skip frames before transferring them to the card.

# 2.1.35.6.8 PlaybackControl: setSyncSlave Method

## C++

TMmRc setSyncSlave(bool enable);

### Remarks

This is setSyncSlave, a member of class PlaybackControl.

# 2.1.35.6.9 PlaybackControl::singleStep Method

Single step a frame.



# C++

TMmRc singleStep(int32\_t NumberOfFrames);

## **Parameters**

Parameters	Description
int32_t NumberOfFrames	number of frames to advance the playback

## Returns

MMRC\_Ok on success, otherwise an error.

#### **Return Values**

Return Values	Description
MMRC_Ok	operation successfully completed

#### Description

This function causes the playback to advance by a number of frames. If the playback is not already paused, pause will be activated. Normal usage would be singeStep(1), to step to the next frame. A singleStep(10) would skip 9 frames and show the following frame.

# 2.1.35.6.10 PlaybackControl::stop Method



Stops the playback.

#### C++

```
TMmRc stop();
```

## Returns

MMRC\_Ok will be returned if the operation was successful, otherwise an error.

#### **Return Values**

Return Values	Description
MMRC_Ok	operation successfully completed

## Description

The stop() method will stop the whole playback and display a stop specific image (normally this is a black frame). All decoder buffers will be flushed.

# 2.1.35.6.11 PlaybackControl::waitForEndOfStream Method

Waits for end of stream.

## C++



TMmRc waitForEndOfStream();

#### Returns

MMRC Ok if end of stream was reached.

## **Return Values**

Return Values	Description
MMRC_Ok	end of stream reached
MMRC_TimeOut	waitForEndOfStream was aborted, probably by calling stop().

## Description

This function will return if the end of stream marker is received by all decoders connected to the playback control. The end of stream maker will be queue using setEndOfStream() at a decoder. The function can be aborted by calling stop(), which resets all decoders and there end of stream marker.

# 2.1.35.7 PlaybackControl Operators

# 2.1.35.7.1 PlaybackControl::= Operator

#### C++

PlaybackControl& operator =(const PlaybackControl& other);

#### Remarks

This is =, a member of class PlaybackControl.

# 2.1.35.7.2 PlaybackControl::bool Operator

## C++

```
operator bool() const;
```

## Remarks

This is bool, a member of class PlaybackControl.

# 2.1.36 RenderFill Class

## **Inheritance Hierarchy**

#### C++

class RenderFill : public OverlayRenderCommand;

#### File

mvc2api\_overlay.h

## Remarks

This is class mvc2::RenderFill.

## **Members**

# Methods

	Name	Description
<b>=</b> ♦ ₩	~OverlayRenderCommand	This is ~OverlayRenderCommand, a member of class OverlayRenderCommand.
<b>=</b> ♦	OverlayRenderCommand	This is OverlayRenderCommand, a member of class OverlayRenderCommand.

#### RenderFill Class

	Name	Description
<b>=</b> ♦ ₩	~RenderFill	This is ~RenderFill, a member of class RenderFill.
<b>=♦</b>	RenderFill	This is RenderFill, a member of class RenderFill.

# OverlayRenderCommand Data Members

	Name	Description
49	m_clearColor	This is m_clearColor, a member of class OverlayRenderCommand.
99	m_flags	This is m_flags, a member of class OverlayRenderCommand.
49	m_reserved1	This is m_reserved1, a member of class OverlayRenderCommand.
49	m_reserved2	This is m_reserved2, a member of class OverlayRenderCommand.

## OverlayRenderCommand Methods

	Name	Description
<b>≡♦</b>	clearRenderArea	This is clearRenderArea, a member of class OverlayRenderCommand.
<b>=♦</b> A	renderIntoBuffer	This is renderIntoBuffer, a member of class OverlayRenderCommand.

#### RenderFill Class

	Name	Description
<b>=</b> ♦••	renderIntoBuffer	This is renderIntoBuffer, a member of class RenderFill.
<b>=</b> ♦	setBox	This is setBox, a member of class RenderFill.
<b>≡</b>	setColor	This is setColor, a member of class RenderFill.

## OverlayRenderCommand Friends

١	Name	Description
	class OverlayDataBuffer	This is friend friend class OverlayDataBuffer.

## OverlayRenderCommand Data Members

	Name	Description
<b>∳</b> §	m_clearColor	This is m_clearColor, a member of class OverlayRenderCommand.
49	m_flags	This is m_flags, a member of class OverlayRenderCommand.
<b>♦</b> 9	m_reserved1	This is m_reserved1, a member of class OverlayRenderCommand.
49	m_reserved2	This is m_reserved2, a member of class OverlayRenderCommand.

## OverlayRenderCommand Friends

Name	Description
class OverlayDataBuffer	This is friend friend class OverlayDataBuffer.

## OverlayRenderCommand Methods

	Name	Description
<b>≡</b>	clearRenderArea	This is clearRenderArea, a member of class OverlayRenderCommand.
= <b>4</b> , <b>A</b>	renderIntoBuffer	This is renderIntoBuffer, a member of class OverlayRenderCommand.

## RenderFill Class

	Name	Description
<b>=</b> ♦•	renderIntoBuffer	This is renderIntoBuffer, a member of class RenderFill.
<b>≡</b> ♦	setBox	This is setBox, a member of class RenderFill.
<b>≡♦</b>	setColor	This is setColor, a member of class RenderFill.

# 2.1.36.1 RenderFill::~RenderFill Destructor

#### C++

virtual ~RenderFill();

#### Remarks

This is ~RenderFill, a member of class RenderFill.

# 2.1.36.2 RenderFill::RenderFill Constructor

#### C++

```
RenderFill();
```

#### Remarks

This is RenderFill, a member of class RenderFill.

# 2.1.36.3 RenderFill Methods

# 2.1.36.3.1 RenderFill::renderIntoBuffer Method

#### C++

```
virtual TMmRc renderIntoBuffer(MemoryBuffer * membuf) const;
```

#### Remarks

This is renderIntoBuffer, a member of class RenderFill.

# 2.1.36.3.2 RenderFill::setBox Method

## C++

```
void setBox(const FramePosition & startPos, const FramePosition & endPos);
```

#### Remarks

This is setBox, a member of class RenderFill.

# 2.1.36.3.3 RenderFill::setColor Method

## C++

```
void setColor(uint32_t fillColor);
```

#### Remarks

This is setColor, a member of class RenderFill.

# 2.1.37 ProjectorAccess Class

Interface to the projector connected to the MVC201.

## **Inheritance Hierarchy**

C++

```
class ProjectorAccess;
```

#### File

mvc2api.h

#### Description

This class is used to communicate with the projector connected to the MVC201. Please check TI or OEM documentation for available commands.

#### **Members**

#### Methods

	Name	Description
<b>=</b> ♦ <b>W</b>	~ProjectorAccess	This is ~ProjectorAccess, a member of class ProjectorAccess.
<b>=</b> ♦	ProjectorAccess	This is ProjectorAccess, a member of class ProjectorAccess.
<b>≡♦</b>	ProjectorAccess	This is ProjectorAccess, a member of class ProjectorAccess.
<b>≡</b>	ProjectorAccess	Constructs a projector access object.

## **ProjectorAccess Methods**

	Name	Description
<b>=</b> ♦	commandlo	Send and receive a projector command.
<b>≡</b>	login	Login into the projector.
<b>≡♦</b>	logout	Logout from projector

## **ProjectorAccess Operators**

	Name	Description
=+)	=	This is =, a member of class ProjectorAccess.
=+)	bool	Returns if the object is valid.

## **ProjectorAccess Methods**

	Name	Description
<b>≡♦</b>	commandlo	Send and receive a projector command.
<b>≡♦</b>	login	Login into the projector.
<b>≡♦</b>	logout	Logout from projector

## **ProjectorAccess Operators**

	Name	Description
=+)	=	This is =, a member of class ProjectorAccess.
<u>(/-</u> =+)	bool	Returns if the object is valid.

# 2.1.37.1 ProjectorAccess::~ProjectorAccess Destructor

#### C++

virtual ~ProjectorAccess();

## Remarks

This is ~ProjectorAccess, a member of class ProjectorAccess.

# 2.1.37.2 ProjectorAccess::ProjectorAccess Constructor ()

## C++

ProjectorAccess();

#### Remarks

This is ProjectorAccess, a member of class ProjectorAccess.

# 2.1.37.3 ProjectorAccess::ProjectorAccess Constructor (ProjectorAccess&)

#### C++

ProjectorAccess(const ProjectorAccess& other)

#### Remarks

This is ProjectorAccess, a member of class ProjectorAccess.

# 2.1.37.4 ProjectorAccess::ProjectorAccess Constructor (TMmRc \*, MvcDevice &, char \*, uint16\_t)

Constructs a projector access object.

#### C++

```
ProjectorAccess(TMmRc * resultPointer, const MvcDevice & mvcdev, const char * ipaddr = 0,
uint16_t port = 0xaaa1);
```

#### **Parameters**

Parameters	Description
TMmRc * resultPointer	return code from the creation
const MvcDevice & mvcdev	base mvc device
const char * ipaddr = 0	optional ip address, default ICP ip address
uint16_t port = 0xaaa1	optional network port, default 0xaaa1

#### Description

A projector communication tunnel will be created and the connection to the projector will be established.

# 2.1.37.5 ProjectorAccess Methods

# 2.1.37.5.1 ProjectorAccess::commandlo Method

Send and receive a projector command.

#### C++

```
TMmRc commandIo(uint8_t * data, uint32_t sizeIn, uint32_t * sizeOut);
```

#### **Parameters**

Parameters	Description
uint8_t * data	input data block and output buffer
uint32_t sizeIn	number of bytes to send
uint32_t * sizeOut	size of data block and number of bytes received

## Returns

If MMRC\_Ok is returned the sizeOut variable is set with the number of bytes received.

#### Description

Send a projector command and receive the response. The data starts with 3 bytes command followed by 2 bytes size and data block. Check sum is calculated and check automatically. Please make sure to initialize the sizeOut variable with the space of the data array before calling this function.

# 2.1.37.5.2 ProjectorAccess::login Method

Login into the projector.

#### C++

```
TMmRc login(const char * username, const char * password);
```

#### **Parameters**

Parameters	Description
const char * username	pointer to the user name
const char * password	pointer to the password

#### **Returns**

Return MMRC\_Ok if login was successful, otherwise an error.

## Description

This function is used to authenticate the projector connection. See TI and OEM documentation for available credentials.

# 2.1.37.5.3 ProjectorAccess::logout Method

Logout from projector

#### C++

```
TMmRc logout();
```

#### Returns

MMRC\_Ok if logout was successful.

## Description

Remove the authentication from the projector connection.

# 2.1.37.6 ProjectorAccess Operators

# 2.1.37.6.1 ProjectorAccess::= Operator

#### C++

```
ProjectorAccess& operator =(const ProjectorAccess& other);
```

#### Remarks

This is =, a member of class ProjectorAccess.

# 2.1.37.6.2 ProjectorAccess::bool Operator

Returns if the object is valid.

#### C++

```
operator bool() const;
```

## Description

Returns true if the object is valid or false if the object is invalid.

# 2.1.38 RenderPicture Class

## **Inheritance Hierarchy**

## C++

class RenderPicture : public OverlayRenderCommand;

#### File

mvc2api\_overlay.h

## Remarks

This is class mvc2::RenderPicture.

## **Members**

#### Methods

		Name	Description
=4	W	~OverlayRenderCommand	This is ~OverlayRenderCommand, a member of class OverlayRenderCommand.
=4		OverlayRenderCommand	This is OverlayRenderCommand, a member of class OverlayRenderCommand.

#### **RenderPicture Class**

	Name	Description
<b>=</b> ♦ ₩	~RenderPicture	This is ~RenderPicture, a member of class RenderPicture.
<b>≡♦</b>	RenderPicture	This is RenderPicture, a member of class RenderPicture.

## **OverlayRenderCommand Data Members**

	Name	Description
<b>∳</b> ?	m_clearColor	This is m_clearColor, a member of class OverlayRenderCommand.
<b>∳</b> ∳	m_flags	This is m_flags, a member of class OverlayRenderCommand.
<b>∳</b> ≩	m_reserved1	This is m_reserved1, a member of class OverlayRenderCommand.
<b>∳</b> ?	m_reserved2	This is m_reserved2, a member of class OverlayRenderCommand.

# **OverlayRenderCommand Methods**

	Name	Description
<b>≡</b>	clearRenderArea	This is clearRenderArea, a member of class OverlayRenderCommand.
<b>=</b> ♦ <sub>?</sub> 🔏	renderIntoBuffer	This is renderIntoBuffer, a member of class OverlayRenderCommand.

#### **RenderPicture Class**

	Name	Description
<b>=</b> ♦•	renderIntoBuffer	This is renderIntoBuffer, a member of class RenderPicture.
<b>=♦</b>	setPosition	This is setPosition, a member of class RenderPicture.

## OverlayRenderCommand Friends

Name	Description
class OverlayDataBuffer	This is friend friend class OverlayDataBuffer.

## OverlayRenderCommand Data Members

	Name	Description
<b>∳</b> ?	m_clearColor	This is m_clearColor, a member of class OverlayRenderCommand.
49	m_flags	This is m_flags, a member of class OverlayRenderCommand.
<b>∳</b> ?	m_reserved1	This is m_reserved1, a member of class OverlayRenderCommand.
<b>♦</b> *	m_reserved2	This is m_reserved2, a member of class OverlayRenderCommand.

## OverlayRenderCommand Friends

Name	Description
class OverlayDataBuffer	This is friend friend class OverlayDataBuffer.

## OverlayRenderCommand Methods

	Name	Description
<b>≡♦</b>	clearRenderArea	This is clearRenderArea, a member of class OverlayRenderCommand.
<b>=</b> ♦•• <b>A</b>	renderIntoBuffer	This is renderIntoBuffer, a member of class OverlayRenderCommand.

#### **RenderPicture Class**

	Name	Description
= <b>4</b> 9 W	renderIntoBuffer	This is renderIntoBuffer, a member of class RenderPicture.
<b>≡</b> ∳	setPosition	This is setPosition, a member of class RenderPicture.

# 2.1.38.1 RenderPicture::~RenderPicture Destructor

## C++

virtual ~RenderPicture();

## Remarks

This is ~RenderPicture, a member of class RenderPicture.

# 2.1.38.2 RenderPicture::RenderPicture Constructor

#### C++

RenderPicture(const uint8\_t \* pictureBuffer, uint32\_t size);

#### Remarks

This is RenderPicture, a member of class RenderPicture.

# 2.1.38.3 RenderPicture Methods

# 2.1.38.3.1 RenderPicture::renderIntoBuffer Method

#### C++

virtual TMmRc renderIntoBuffer(MemoryBuffer \* membuf) const;

#### Remarks

This is renderIntoBuffer, a member of class RenderPicture.

# 2.1.38.3.2 RenderPicture::setPosition Method

#### C++

void setPosition(const FramePosition & position);

#### Remarks

This is setPosition, a member of class RenderPicture.

# 2.1.39 RenderText Class

## **Inheritance Hierarchy**

## C++

class RenderText : public OverlayRenderCommand;

#### File

mvc2api\_overlay.h

#### Remarks

This is class mvc2::RenderText.

#### **Members**

#### Methods

		Name	Description
=6	W	~OverlayRenderCommand	This is ~OverlayRenderCommand, a member of class OverlayRenderCommand.
==		OverlayRenderCommand	This is OverlayRenderCommand, a member of class OverlayRenderCommand.

## RenderText Class

	Name	Description
<b>■</b> ♦ ₩	~RenderText	This is ~RenderText, a member of class RenderText.
<b>≡⋄</b>	RenderText	This is RenderText, a member of class RenderText.

## OverlayRenderCommand Data Members

	Name	Description
<b>∳</b> ≩	m_clearColor	This is m_clearColor, a member of class OverlayRenderCommand.
49	m_flags	This is m_flags, a member of class OverlayRenderCommand.
<b>∳</b> 9	m_reserved1	This is m_reserved1, a member of class OverlayRenderCommand.
<b>∳</b> 9	m_reserved2	This is m_reserved2, a member of class OverlayRenderCommand.

## OverlayRenderCommand Methods

	Name	Description
<b>≡♦</b>	clearRenderArea	This is clearRenderArea, a member of class OverlayRenderCommand.

= <b>4</b> A	renderIntoBuffer	This is renderIntoBuffer, a member of class
		OverlayRenderCommand.

## RenderText Class

	Name	Description
<b>=</b> ♦• ₩	renderIntoBuffer	This is renderIntoBuffer, a member of class RenderText.
<b>=</b> ♦	setColor	This is setColor, a member of class RenderText.
<b>=</b> ♦	setFont	This is setFont, a member of class RenderText.
<b>=</b> ♦	setPosition	This is setPosition, a member of class RenderText.

# OverlayRenderCommand Friends

Name	Description
class OverlayDataBuffer	This is friend friend class OverlayDataBuffer.

#### **RenderText Enumerations**

## RenderText Class

	Name	Description
<b>a</b>	TextFlags	This is record mvc2::RenderText::TextFlags.

# OverlayRenderCommand Data Members

	Name	Description
<b>∳</b> ?	m_clearColor	This is m_clearColor, a member of class OverlayRenderCommand.
<b>♦</b> *	m_flags	This is m_flags, a member of class OverlayRenderCommand.
<b>∳</b> ?	m_reserved1	This is m_reserved1, a member of class OverlayRenderCommand.
<b>∳</b> ?	m_reserved2	This is m_reserved2, a member of class OverlayRenderCommand.

#### **RenderText Enumerations**

## RenderText Class

		Name	Description
Œ	P	TextFlags	This is record mvc2::RenderText::TextFlags.

# OverlayRenderCommand Friends

Name	Description
class OverlayDataBuffer	This is friend friend class OverlayDataBuffer.

# OverlayRenderCommand Methods

	Name	Description
<b>≡♦</b>	clearRenderArea	This is clearRenderArea, a member of class OverlayRenderCommand.
= <b>4</b> ? A	renderIntoBuffer	This is renderIntoBuffer, a member of class OverlayRenderCommand.

## RenderText Class

	Name	Description
= <b>♦</b> • ₩	renderIntoBuffer	This is renderIntoBuffer, a member of class RenderText.
<b>=♦</b>	setColor	This is setColor, a member of class RenderText.
<b>≡</b>	setFont	This is setFont, a member of class RenderText.
<b>=♦</b>	setPosition	This is setPosition, a member of class RenderText.

# 2.1.39.1 RenderText::~RenderText Destructor

#### C++

```
virtual ~RenderText();
```

#### Remarks

This is ~RenderText, a member of class RenderText.

# 2.1.39.2 RenderText::RenderText Constructor

#### C++

```
RenderText(const char * textBuffer, uint32_t size = 0);
```

#### Remarks

This is RenderText, a member of class RenderText.

# 2.1.39.3 RenderText Enumerations

# 2.1.39.3.1 mvc2::RenderText::TextFlags Enumeration

#### C++

```
enum TextFlags {
   TextFlag_Italic = (1<<0),
   TextFlag_Bold = (1<<1),
   TextFlag_Underline = (1<<2),
   TextFlag_Strikethrough = (1<<3),
   TextFlag_Border = (1<<4),
   TextFlag_Shadow = (1<<5),
   TextFlag_SoftShadow = (1<<6)
};</pre>
```

## File

mvc2api\_overlay.h

#### Remarks

This is record mvc2::RenderText::TextFlags.

# 2.1.39.4 RenderText Methods

# 2.1.39.4.1 RenderText::renderIntoBuffer Method

## C++

```
virtual TMmRc renderIntoBuffer(MemoryBuffer * membuf) const;
```

#### Remarks

This is renderIntoBuffer, a member of class RenderText.

# 2.1.39.4.2 RenderText::setColor Method

#### C++

void setColor(uint32\_t textColor, uint32\_t shadowColor, uint32\_t borderColor);

#### Remarks

This is setColor, a member of class RenderText.

# 2.1.39.4.3 RenderText::setFont Method

#### C++

```
void setFont(const char * fontName, uint16_t fontSize, uint16_t textFlags = 0);
```

#### Remarks

This is setFont, a member of class RenderText.

# 2.1.39.4.4 RenderText::setPosition Method

#### C++

void setPosition(const FramePosition & position);

#### Remarks

This is setPosition, a member of class RenderText.

# 2.1.40 SecurityAccess Class

## **Inheritance Hierarchy**

#### C++

class SecurityAccess;

## File

mvc2api\_security.h

#### Remarks

Class to access security features of the card, if the security manager is not active.

## **Members**

## Methods

	Name	Description
<b>=</b> ♦ ₩	~SecurityAccess	Default destructor.
<b>≡</b>	SecurityAccess	Default constructor.
<b>≡♦</b>	SecurityAccess	Default copy constructor.
<b>=♦</b>	SecurityAccess	Creates a SecurityAccess object.

## SecurityAccess Structures

		Name	Description
9	<b>&gt;</b>	_CineLink2Para	This is record mvc2::SecurityAccess::_CineLink2Para.

# **SecurityAccess Enumerations**

	Name	Description
<b>a</b>	_HashAlgorithm	This is record mvc2::SecurityAccess::_HashAlgorithm.
<b>a</b>	_MediaType	Media type for decryption
<b>a</b>	_SoftwareId	This is record mvc2::SecurityAccess::_SoftwareId.
<b>3</b>	_SymmetricCryptoOperation	This is record mvc2::SecurityAccess::_SymmetricCryptoOperation.

# **SecurityAccess Methods**

	Name	Description
<b>≡</b> ∳	deleteCertificateChain	This is deleteCertificateChain, a member of class SecurityAccess.
<b>=♦</b>	disableForensicMarking	Disable forensic marking
<b>=♦</b>	disableForensicMarking	Disable forensic marking
<b>=♦</b>	disableLLE	Disable local link encryption.
<b>=</b>	enableCinelink	Enable CineLink(TM) for a video output.
<b>=</b>	enableCinelink2	Enable CineLink(TM) 2 for a video output.
<b>=♦</b>	enableForensicMarking	Enable forensic marking.
<b>=♦</b>	enableForensicMarking	Enable forensic marking.
<b>≡</b> ∳	getAllHardwareInfo	This is getAllHardwareInfo, a member of class SecurityAccess.
<b>∉</b> ∳	getAppletStatusCode	This is getAppletStatusCode, a member of class SecurityAccess.
<b>=♦</b>	getAppletVersion	This is getAppletVersion, a member of class SecurityAccess.
<b>≡♦</b>	getBuildTimeStrings	reads strings basing on the compiler definesDATE andTIME
<b>∉</b> ∳	getCertificateSigningRequest	This is getCertificateSigningRequest, a member of class SecurityAccess.
<b>=♦</b>	getErrorCode	made to address and read an error code
<b>∉</b> ∳	getFipsFirmwareVersion	This is getFipsFirmwareVersion, a member of class SecurityAccess.
<b>≡</b>	getFirmwareVersion	Returns the security application version.
<b>≡</b>	getForensicMarkingId	Reads the forensic marking id.
<b>=</b>	getHashValue	Retrieves the hash value from target
<b>≡∳</b>	getMikromPublicKey	This function reads the exponent and modulus of device specific MikroM public key.
<b>=♦</b>	getPublicKey	This is getPublicKey, a member of class SecurityAccess.
<b>=♦</b>	getRtcHardwareStatus	checks hardware conditions of RTC
<b>=♦</b>	getSelftestStatus	Reads the state of the FIPS self test.
<b>=♦</b>	getTamperStatus	This is getTamperStatus, a member of class SecurityAccess.
<b>=♦</b>	hashRtcSram	This is hashRtcSram, a member of class SecurityAccess.
<b>=</b>	initHardwareInfo	This is initHardwareInfo, a member of class SecurityAccess.
<b>≡</b>	initLogFlash	This is initLogFlash, a member of class SecurityAccess.
<b>=♦</b>	installCertificate	This is installCertificate, a member of class SecurityAccess.
<b>∉∳</b>	processSmartcardApdu	This is processSmartcardApdu, a member of class SecurityAccess.
<b>≡</b>	readLogFlashAttributes	This function reads hardware attributes of log flash device.
<b>=</b>	readRtcReg	This is readRtcReg, a member of class SecurityAccess.
<b>=</b>	readRtcSram	This is readRtcSram, a member of class SecurityAccess.
<b>=</b>	readSecurityInterface	Reads data from security interface.
<b>≡</b>	resetPowerfailBit	This is resetPowerfailBit, a member of class SecurityAccess.

<b>=♦</b>	rsaPrivateKeyDecCmp	This is rsaPrivateKeyDecCmp, a member of class SecurityAccess.
<b>≡</b>	selfTest	This is selfTest, a member of class SecurityAccess.
<b>=♦</b>	setCineLink2Keys	This is setCineLink2Keys, a member of class SecurityAccess.
<b>=♦</b>	setForensicMarkingId	This is setForensicMarkingId, a member of class SecurityAccess.
<b>=</b>	setForensicMarkingIndicator	This is setForensicMarkingIndicator, a member of class SecurityAccess.
<b>=♦</b>	setHardwareInfo	This is setHardwareInfo, a member of class SecurityAccess.
<b>=♦</b>	setHashData	Transfers data to target for sha256 hash calculation.
<b>≡</b>	setMediaDecryptorKey	Set media decryption key.
<b>∉</b> ∳	symmetricCryptoFinish	This is symmetricCryptoFinish, a member of class SecurityAccess.
<b>=</b>	symmetricCryptoInit	initializes the calculation of symmetric encryption/decryption
<b>=♦</b>	symmetricCryptoProcess	This function processes input data and gives the encrypted or decrypted result.
<b>∉</b> ∳	validateHardwareInfo	This is validateHardwareInfo, a member of class SecurityAccess.
<b>≡</b>	verify	This is verify, a member of class SecurityAccess.
<b>≡</b>	writeRtcReg	This is writeRtcReg, a member of class SecurityAccess.
<b>=</b>	writeSecurityInterface	Writes data to security interface.

# **SecurityAccess Operators**

	Name	Description
<u>(/−</u> =+)	=	Default assignment operator.
(/ <u>-</u>	bool	Check for object validity.

# **SecurityAccess Enumerations**

	Name	Description
<b>a</b>	_HashAlgorithm	This is record mvc2::SecurityAccess::_HashAlgorithm.
<b>a</b>	_MediaType	Media type for decryption
<b>a</b>	_SoftwareId	This is record mvc2::SecurityAccess::_SoftwareId.
<b>a</b> P	_SymmetricCryptoOperation	This is record mvc2::SecurityAccess::_SymmetricCryptoOperation.

# **SecurityAccess Methods**

	Name	Description
<b>≓</b>	deleteCertificateChain	This is deleteCertificateChain, a member of class SecurityAccess.
<b>≡</b>	disableForensicMarking	Disable forensic marking
<b>≡</b>	disableForensicMarking	Disable forensic marking
<b>≡</b>	disableLLE	Disable local link encryption.
<b>≡</b>	enableCinelink	Enable CineLink(TM) for a video output.
<b>≡</b>	enableCinelink2	Enable CineLink(TM) 2 for a video output.
<b>≡</b>	enableForensicMarking	Enable forensic marking.
<b>≡</b>	enableForensicMarking	Enable forensic marking.
<b>=♦</b>	getAllHardwareInfo	This is getAllHardwareInfo, a member of class SecurityAccess.
<b>≓</b>	getAppletStatusCode	This is getAppletStatusCode, a member of class SecurityAccess.
<b>≡</b>	getAppletVersion	This is getAppletVersion, a member of class SecurityAccess.
<b>≟</b>	getBuildTimeStrings	reads strings basing on the compiler definesDATE andTIME

<b>≡∳</b>	getCertificateSigningRequest	This is getCertificateSigningRequest, a member of class SecurityAccess.
<b>≡∳</b>	getErrorCode	made to address and read an error code
<b>≡∳</b>	getFipsFirmwareVersion	This is getFipsFirmwareVersion, a member of class SecurityAccess.
<b>≡∳</b>	getFirmwareVersion	Returns the security application version.
<b>=∳</b>	getForensicMarkingId	Reads the forensic marking id.
<b>≡∳</b>	getHashValue	Retrieves the hash value from target
<b>≡∳</b>	getMikromPublicKey	This function reads the exponent and modulus of device specific MikroM public key.
<b>≡∳</b>	getPublicKey	This is getPublicKey, a member of class SecurityAccess.
<b>:</b> ∳	getRtcHardwareStatus	checks hardware conditions of RTC
<b>≡∳</b>	getSelftestStatus	Reads the state of the FIPS self test.
≡ <b>∳</b>	getTamperStatus	This is getTamperStatus, a member of class SecurityAccess.
<b>≡∳</b>	hashRtcSram	This is hashRtcSram, a member of class SecurityAccess.
•	initHardwareInfo	This is initHardwareInfo, a member of class SecurityAccess.
•	initLogFlash	This is initLogFlash, a member of class SecurityAccess.
•	installCertificate	This is installCertificate, a member of class SecurityAccess.
<b>≡∳</b>	processSmartcardApdu	This is processSmartcardApdu, a member of class SecurityAccess.
•	readLogFlashAttributes	This function reads hardware attributes of log flash device.
•	readRtcReg	This is readRtcReg, a member of class SecurityAccess.
•	readRtcSram	This is readRtcSram, a member of class SecurityAccess.
<b>•</b>	readSecurityInterface	Reads data from security interface.
≡ <b>∳</b>	resetPowerfailBit	This is resetPowerfailBit, a member of class SecurityAccess.
<b>≡</b>	rsaPrivateKeyDecCmp	This is rsaPrivateKeyDecCmp, a member of class SecurityAccess.
<b>≡∳</b>	selfTest	This is selfTest, a member of class SecurityAccess.
•	setCineLink2Keys	This is setCineLink2Keys, a member of class SecurityAccess
<b>≡∳</b>	setForensicMarkingId	This is setForensicMarkingId, a member of class SecurityAccess.
<b>=</b>	setForensicMarkingIndicator	This is setForensicMarkingIndicator, a member of class SecurityAccess.
<b>=∳</b>	setHardwareInfo	This is setHardwareInfo, a member of class SecurityAccess.
•	setHashData	Transfers data to target for sha256 hash calculation.
•	setMediaDecryptorKey	Set media decryption key.
<b>≡∳</b>	symmetricCryptoFinish	This is symmetricCryptoFinish, a member of class SecurityAccess.
•	symmetricCryptoInit	initializes the calculation of symmetric encryption/decryption
<b>≡</b>	symmetricCryptoProcess	This function processes input data and gives the encrypted or decrypted result.
<b>≡∳</b>	validateHardwareInfo	This is validateHardwareInfo, a member of class SecurityAccess.
<b>≡∳</b>	verify	This is verify, a member of class SecurityAccess.
<b>•</b>	writeRtcReg	This is writeRtcReg, a member of class SecurityAccess.
<b>≡</b>	writeSecurityInterface	Writes data to security interface.

# **SecurityAccess Operators**

	Name	Description
=+)	=	Default assignment operator.
=+)	bool	Check for object validity.

## **SecurityAccess Structures**

	Name	Description
<b>*</b>	_CineLink2Para	This is record mvc2::SecurityAccess::_CineLink2Para.

# 2.1.40.1 SecurityAccess::~SecurityAccess Destructor

Default destructor.

#### C++

```
virtual ~SecurityAccess();
```

#### Description

The destructor will free all resources allocated by this object. Any setups will be kept. Local Link Encryption and key storage remain untouched.

# 2.1.40.2 SecurityAccess::SecurityAccess Constructor ()

Default constructor.

#### C++

SecurityAccess();

#### Description

This constructor will create a empty, non-functional SecurityAccess object.

# 2.1.40.3 SecurityAccess::SecurityAccess Constructor (SecurityAccess&)

Default copy constructor.

#### C++

SecurityAccess(const SecurityAccess& other);

## **Parameters**

Parameters	Description
const SecurityAccess& other	Input SecurityAccess

## Description

A new SecurityAccess will be created from input SecurityAccess.

# 2.1.40.4 SecurityAccess::SecurityAccess Constructor (TMmRc \*, MvcDevice &)

Creates a SecurityAccess object.

C++

```
SecurityAccess(TMmRc * resultPointer, const MvcDevice & mvcDevice);
```

## **Parameters**

Parameters	Description
TMmRc * resultPointer	Return code from the creation

const MvcDevice & mvcDevice

MvcDevice object for which the object should be created

#### Description

The SecurityAccess is an interface for a security manager which is implemented on the host system (e.g. a Player software). All function of the SecurityAccess are normally controlled by the on board security manager. If it is not available or the customer don't want to use it, the main security features can be accessed by this class.

#### Remarks

Using this class for security features is not save. Function calls can be easily traced by third party programs. It should only be used for debugging or if there's no security manager on board. The only safe and DCI complaint way is to use the on board security manager.

If the on board security manager is active, none of these functions will have an effect!

# 2.1.40.5 SecurityAccess Structures

# 2.1.40.5.1 mvc2::SecurityAccess::\_CineLink2Para Structure

#### C++

```
struct _CineLink2Para {
    uint8_t m_key[SecurityAccess::CineLink2KeyLen];
    uint8_t m_attribute[SecurityAccess::CineLink2AttrLen];
    uint16_t m_keyID;
    uint16_t m_nextKeyID;
};
File

mvc2api_security.h
```

## Remarks

This is record mvc2::SecurityAccess::\_CineLink2Para.

# 2.1.40.6 SecurityAccess Enumerations

# 2.1.40.6.1 mvc2::SecurityAccess::\_HashAlgorithm Enumeration

## C++

#### Remarks

This is record mvc2::SecurityAccess::\_HashAlgorithm.

# 2.1.40.6.2 mvc2::SecurityAccess::\_MediaType Enumeration

#### C++

```
enum _MediaType {
```

174

```
MEDIA_TYPE_Video = 0,
MEDIA_TYPE_Audio = 1,
MEDIA_TYPE_Other = 2
};
File
mvc2api_security.h
```

#### Members

Members	Description
MEDIA_TYPE_Video = 0	Media type is video
MEDIA_TYPE_Audio = 1	Media type is audio
MEDIA_TYPE_Other = 2	Media type is not audio nor video

#### Remarks

Media type for decryption

# 2.1.40.6.3 mvc2::SecurityAccess::\_SoftwareId Enumeration

#### C++

```
enum _SoftwareId {
   SW_ID_SecApplication = 0,
   SW_ID_MainFirmware = 1,
   SW_ID_SecBootloader = 2,
   SW_ID_MainBootloader = 3,
   SW_ID_MainFPGA = 4
  };

File
  mvc2api_security.h
```

#### Remarks

This is record mvc2::SecurityAccess::\_SoftwareId.

# 2.1.40.6.4 mvc2::SecurityAccess::\_SymmetricCryptoOperation Enumeration

## C++

```
enum _SymmetricCryptoOperation {
    AES1280P_cbc_enc = 0,
    AES1280P_cbc_dec = 1,
    AES1280P_ecb_enc = 2,
    AES1280P_ecb_dec = 3,
    TrippleDes_cbc_enc = 4,
    TrippleDes_cbc_dec = 5,
    MMTrippleDes_cbc_dec = 6,
    MMTrippleDes_cbc_dec = 9,
    AES1280P_cbc_dec_fpga = 10,
    AES2560P_cbc_enc = 11,
    AES2560P_cbc_dec = 12
};
File
```

## Remarks

mvc2api\_security.h

This is record mvc2::SecurityAccess::\_SymmetricCryptoOperation.

# 2.1.40.7 SecurityAccess Methods

# 2.1.40.7.1 SecurityAccess::deleteCertificateChain Method

#### C++

TMmRc deleteCertificateChain(uint32\_t chainNumber);

#### Remarks

This is deleteCertificateChain, a member of class SecurityAccess.

# 2.1.40.7.2 SecurityAccess::disableForensicMarking Method (AudioOutput &)

Disable forensic marking

#### C++

TMmRc disableForensicMarking(const AudioOutput & output);

#### **Parameters**

Parameters	Description
const AudioOutput & output	AudioOutput

## Returns

Returns MMRC\_Ok on success.

#### **Return Values**

Return Values	Description
MMRC_Ok	successful operation
MMRC_MVC2_OutputNotFound	the output could not be located

## Description

This method disables the forensic marking for an output.

# 2.1.40.7.3 SecurityAccess::disableForensicMarking Method (VideoOutput &)

Disable forensic marking

## C++

TMmRc disableForensicMarking(const VideoOutput & output);

#### **Parameters**

Parameters	Description
const VideoOutput & output	VideoOutput

## Returns

Returns MMRC\_Ok on success.

## **Return Values**

Return Values	Description
MMRC_Ok	successful operation
MMRC_MVC2_OutputNotFound	the output could not be located

## Description

This method disables the forensic marking for an output.

# 2.1.40.7.4 SecurityAccess::disableLLE Method

Disable local link encryption.

#### C++

TMmRc disableLLE(const VideoOutput & output, uint32\_t link);

#### **Parameters**

Parameters	Description
const VideoOutput & output	VideoOutput to disable the output encryption
uint32_t link	link number inside the VideoOutput

#### **Returns**

Returns MMRC\_Ok on successful deactivation.

#### **Return Values**

Return Values	Description
MMRC_Ok	successful operation
MMRC_MVC2_OutputNotFound	the output could not be located on the card

## Description

The HD-SDI connection between the MVC card and the projector is called local link. The method will deactivate any encryption for the HD-SDI connection of a video output. This will deactivate CineLink(TM) 1 and CineLink(TM) 2.

# 2.1.40.7.5 SecurityAccess::enableCinelink Method

Enable CineLink(TM) for a video output.

#### C++

TMmRc enableCinelink(const VideoOutput & output, uint32\_t link, uint8\_t key = 0, uint16\_t
iv = 0);

## **Parameters**

Parameters	Description
const VideoOutput & output	VideoOutput object
uint32_t link	link number of the video output
uint8_t key = 0	encryption key, if zero is specified, the default encryption key is used
uint16_t iv = 0	initialization vector, if zero is specified, the default initialization vector is used

## Returns

Returns MMRC\_Ok if the activation was successful.

## **Return Values**

Return Values	Description
MMRC_Ok	Successful activation
MMRC_MVC2_OutputNotFound	The output does not exist on the card.

## Description

This method enable the CineLine(TM) 1 for a given video output. CineLink(TM) is a encryption algorithm for the HD-SDI transmission between the MVC card and the projector. Due to the support for multi link HD-SDI of the MVC card (e.g. dual HD-SDI). You have to specify the link where to enable the encryption.

# 2.1.40.7.6 SecurityAccess::enableCinelink2 Method

Enable CineLink(TM) 2 for a video output.

#### C++

TMmRc enableCinelink2(const VideoOutput & output, uint32\_t link, const uint8\_t \* key, const
uint8\_t \* attr, uint16\_t keyid, uint16\_t nextkeyid);

#### Parameters 4 8 1

Parameters	Description
const VideoOutput & output	VideoOutput for which the encryption should be enabled
uint32_t link	link number in the video output
const uint8_t * key	128 bit (16 bytes) encryption key (big endian)
const uint8_t * attr	64 bit (8 bytes) attribute of the key id (big endian)
uint16_t keyid	key id which will be transmitted to the projector (12 bits used, zero is reserved)
uint16_t nextkeyid	key id which will be used next, after the current key id (is also transmitted to the projector)

#### Returns

Returns MMRC\_Ok on successful activation.

#### **Return Values**

Return Values	Description
MMRC_Ok	Successful activation
MMRC_MVC2_OutputNotFound	The output does not exist on the card.

#### Description

This method enable the CineLine(TM) 2 for a given video output. CineLink(TM) is a encryption algorithm for the HD-SDI transmission between the MVC card and the projector. CineLink(TM) 2 is the successor of CineLink(TM) 1 and is much more secure. It uses a better encryption algorithm and needs a TLS connection between the security manager and the projector for dynamic key change. In case of the SecurityAccess, which only works if the on board security manager is disabled, this dynamic key change must by done by the Player application.

This projector has a key storage which uses the key id as index. These means that a number of keys can be transferred to the projector and selected by the key id. This key id will be transmitted to the projector via the HD-SDI cable. A key id of zero is reserved and cannot be used.

Due to the support for multi link HD-SDI of the MVC card (e.g. dual HD-SDI). You have to specify the link where to enable the encryption.

# 2.1.40.7.7 SecurityAccess::enableForensicMarking Method (AudioOutput &, uint16\_t, uint32\_t)

Enable forensic marking.

#### C++

TMmRc enableForensicMarking(const AudioOutput & output, uint16\_t timecode, uint32\_t locationid);

## **Parameters**

Parameters	Description
const AudioOutput & output	AudioOutput to activate the forensic marking for
uint16_t timecode	16 bit timecode to insert (15 minutes counter)
uint32_t locationid	20 bit location info id

#### **Returns**

Returns MMRC\_Ok on successful activation.

#### **Return Values**

Return Values	Description
MMRC_Ok	successful operation
MMRC_MVC2_OutputNotFound	output could not be found on the card

#### Description

The method enables the forensic marking for an output. Timecode and location information id will be inserted into the output. The timecode is a counter of 15 minutes starting from January the first of the current year. The firmware will not increase this timecode, a application must call this function every 15 minutes to increase the timecode. Location info id is a identifier for the cinema.

# 2.1.40.7.8 SecurityAccess::enableForensicMarking Method (VideoOutput &, uint16\_t, uint32\_t)

Enable forensic marking.

#### C++

TMmRc enableForensicMarking(const VideoOutput & output, uint16\_t timecode, uint32\_t
locationid);

#### **Parameters**

Parameters	Description
const VideoOutput & output	VideoOutput to activate the forensic marking for
uint16_t timecode	16 bit timecode to insert (15 minutes counter)
uint32_t locationid	20 bit location info id

#### **Returns**

Returns MMRC\_Ok on successful activation.

## **Return Values**

Return Values	Description
MMRC_Ok	successful operation
MMRC_MVC2_OutputNotFound	output could not be found on the card

#### Description

The method enables the forensic marking for an output. Timecode and location information id will be inserted into the output. The timecode is a counter of 15 minutes starting from January the first of the current year. The firmware will not increase this timecode, a application must call this function every 15 minutes to increase the timecode. Location info id is a identifier for the cinema.

# 2.1.40.7.9 SecurityAccess::getAllHardwareInfo Method

#### C++

TMmRc getAllHardwareInfo(uint32\_t hwStatus[8], uint32\_t hwValues[8]);

## Remarks

This is getAllHardwareInfo, a member of class SecurityAccess.

# 2.1.40.7.10 SecurityAccess::getAppletStatusCode Method

#### C++

TMmRc getAppletStatusCode(uint16\_t& statusCode);

#### Remarks

This is getAppletStatusCode, a member of class SecurityAccess.

# 2.1.40.7.11 SecurityAccess::getAppletVersion Method

#### C++

TMmRc getAppletVersion(const int32\_t appletId, uint32\_t& major, uint32\_t& minor, uint32\_t& module, uint32\_t& svnVersion);

#### Remarks

This is getAppletVersion, a member of class SecurityAccess.

# 2.1.40.7.12 SecurityAccess::getBuildTimeStrings Method

reads strings basing on the compiler defines \_\_DATE\_\_ and \_\_TIME\_\_

#### C++

TMmRc getBuildTimeStrings(uint32\_t moduleId, char strDate[SecurityAccess::DateLen], char
strTime[SecurityAccess::TimeLen]);

#### **Parameters**

Parameters	Description
uint32_t moduleId	the id of the module that should be checked
date	gets the date C string
time	gets the time C string

# 2.1.40.7.13 SecurityAccess::getCertificateSigningRequest Method

#### C++

TMmRc getCertificateSigningRequest(uint32\_t certNumber, uint8\_t \* csrBuffer, uint32\_t \*
csrBufferSize);

## Remarks

This is getCertificateSigningRequest, a member of class SecurityAccess.

# 2.1.40.7.14 SecurityAccess::getErrorCode Method

made to address and read an error code

#### C++

TMmRc getErrorCode(int32\_t errorCodeVariableId, int32\_t \* errorCode, bool askSm);

## **Parameters**

Parameters	Description
int32_t errorCodeVariableId	address the variable that contains the error code with this id
bool askSm	trueforwarded to Security Application, falseforwarded to MIPS firmware
contains	the read error code when function returns MMRC_Ok

#### **Returns**

MMRC\_Ok on success

# 2.1.40.7.15 SecurityAccess::getFipsFirmwareVersion Method

#### C++

TMmRc getFipsFirmwareVersion(VersionValue & version) const;

#### Remarks

This is getFipsFirmwareVersion, a member of class SecurityAccess.

# 2.1.40.7.16 SecurityAccess::getFirmwareVersion Method

Returns the security application version.

#### C++

TMmRc getFirmwareVersion(VersionValue & version) const;

#### **Returns**

VersionValue object with the firmware version.

#### Description

Gets the security application version number.

# 2.1.40.7.17 SecurityAccess::getForensicMarkingId Method

Reads the forensic marking id.

#### C++

TMmRc getForensicMarkingId(uint32\_t \* locationId);

### **Parameters**

Parameters	Description
uint32_t * locationId	20 bit location info id

# 2.1.40.7.18 SecurityAccess::getHashValue Method

Retrieves the hash value from target

## C++

```
TMmRc getHashValue(mvc2::SecurityAccess::SoftwareId,
mvc2::SecurityAccess::HashAlgorithm eHashAlg, uint8_t* hashValue, uint32_t* readSize) const;
```

## **Parameters**

Parameters	Description
hashBuffer	when function returns MMRC_Ok this buffer contains the
	result of sha256 hash calculation

## Returns

MMRC\_Ok or MMRC\_NotImplemented

# 2.1.40.7.19 SecurityAccess::getMikromPublicKey Method

#### C++

TMmRc getMikromPublicKey(uint8\_t\* exponentResBuffer, size\_t exponentResBufferSize, uint8\_t\* modulusResBuffer, size\_t modulusResBufferSize, size\_t\* exponentResponseSize, size\_t\* modulusResponseSize);

#### **Parameters**

Parameters	Description
uint8_t* exponentResBuffer	buffer to write exponent to
size_t exponentResBufferSize	size of exponent buffer
uint8_t* modulusResBuffer	buffer to write modulus to
size_t modulusResBufferSize	size of modulus buffer
size_t* exponentResponseSize	size of written exponent in exponent buffer
size_t* modulusResponseSize	size of written modulus in modulus buffer

#### Returns

MMRC Ok MMRC\_ArgumentOutOfRange buffer on success when one the is to small MMRC\_MM\_API\_ReadSizeOutOfRange when response does not conform description in Tag MMTAG\_SECURITY\_DeviceKey

#### Remarks

This function reads the exponent and modulus of device specific MikroM public key.

# 2.1.40.7.20 SecurityAccess::getPublicKey Method

#### C++

TMmRc getPublicKey(const int32\_t keyPairId, uint8\_t\* modulusBuf, const size\_t modulusBufSize, size\_t\* modulusSize, uint8\_t\* exponentBuf, const size\_t expBufSize, size\_t\* expSize);

## Remarks

This is getPublicKey, a member of class SecurityAccess.

# 2.1.40.7.21 SecurityAccess::getRtcHardwareStatus Method

checks hardware conditions of RTC

## C++

TMmRc getRtcHardwareStatus(uint32\_t\* status);

## **Parameters**

Parameters	Description
uint32_t* status	gets status bit field, bit0RTC hardware initialization (0OK,1Error) if bit0==1 following bits are "don't care" bit1Power Loss (0OK,1Error Power was lost) bit2Cover Button state(0OK,1Cover Button activated) bit3CRC16
	(0OK,1Mismatch)

#### Returns

should be MMRC\_Ok

# 2.1.40.7.22 SecurityAccess::getSelftestStatus Method

Reads the state of the FIPS self test.

#### C++

```
TMmRc getSelftestStatus();
```

#### Returns

MMRC\_Ok self-test finished successfully MMRC\_SelftestError self-test ends with error MMRC\_SelftestPending self-test in progress

# 2.1.40.7.23 SecurityAccess::getTamperStatus Method

#### C++

```
TMmRc getTamperStatus(TamperStatus & stat);
```

#### Remarks

This is getTamperStatus, a member of class SecurityAccess.

# 2.1.40.7.24 SecurityAccess::hashRtcSram Method

#### C++

```
TMmRc hashRtcSram();
```

#### Remarks

This is hashRtcSram, a member of class SecurityAccess.

# 2.1.40.7.25 SecurityAccess::initHardwareInfo Method

#### C++

```
TMmRc initHardwareInfo();
```

## Remarks

This is initHardwareInfo, a member of class SecurityAccess.

# 2.1.40.7.26 SecurityAccess::initLogFlash Method

## C++

```
TMmRc initLogFlash(int32_t* initLogFlashRc, bool bMsgIf = false);
```

#### Remarks

This is initLogFlash, a member of class SecurityAccess.

# 2.1.40.7.27 SecurityAccess::installCertificate Method

#### C++

```
TMmRc installCertificate(uint32_t certNumber, uint8_t * certBuffer, uint32_t
certBufferSize);
```

#### Remarks

This is installCertificate, a member of class SecurityAccess.

# 2.1.40.7.28 SecurityAccess::processSmartcardApdu Method

## C++

```
TMmRc processSmartcardApdu(const uint8_t* cmd, size_t cmdSize, uint8_t* response, size_t responseBufSize, size_t* responseSize, int32_t* hardwareStatus, uint16_t* isoStatus);
```

#### Remarks

This is processSmartcardApdu, a member of class SecurityAccess.

# 2.1.40.7.29 SecurityAccess::readLogFlashAttributes Method

#### C++

```
TMmRc readLogFlashAttributes(uint8_t* knownType, uint8_t id[5], uint32_t* nomBytesPerPage, uint32_t* spareBytesPerPage, uint32_t* pagesPerBlock, uint32_t* blocksPerDie, uint32_t* diesPerCE, uint32_t* cEsPerDevice);
```

#### **Parameters**

Parameters	Description
uint8_t* knownType	[out] 0this is know type, 1this is unknown device
uint8_t id[5]	[out] these five bytes contain the hardware id
uint32_t* nomBytesPerPage	[out] number of bytes per page
uint32_t* spareBytesPerPage	[out] number of spare bytes per page
uint32_t* pagesPerBlock	[out] pages per block
uint32_t* blocksPerDie	[out] blocks per die
uint32_t* diesPerCE	[out] dies per CE
uint32_t* cEsPerDevice	[out] CEs per device

#### Remarks

This function reads hardware attributes of log flash device.

# 2.1.40.7.30 SecurityAccess::readRtcReg Method

#### C++

```
TMmRc readRtcReg(uint32_t offset, uint8_t* value, bool askSm);
```

#### Remarks

This is readRtcReg, a member of class SecurityAccess.

# 2.1.40.7.31 SecurityAccess::readRtcSram Method

## C++

```
TMmRc readRtcSram(uint32_t offset, uint8_t* value, bool askSm);
```

## Remarks

This is readRtcSram, a member of class SecurityAccess.

# 2.1.40.7.32 SecurityAccess::readSecurityInterface Method

Reads data from security interface.

#### C++

```
TMmRc readSecurityInterface(uint8_t address, uint8_t* readData, uint32_t readDataSize,
uint32_t* readSize);
```

#### **Parameters**

Parameters	Description
uint8_t address	address to read from (0255)
uint8_t* readData	buffer to read data to
uint32_t readDataSize	number of bytes to read

uint32\_t\* readSize

the number of read bytes

#### Returns

MMRC\_Ok on success

# 2.1.40.7.33 SecurityAccess::resetPowerfailBit Method

#### C++

```
TMmRc resetPowerfailBit();
```

#### Remarks

This is resetPowerfailBit, a member of class SecurityAccess.

# 2.1.40.7.34 SecurityAccess::rsaPrivateKeyDecCmp Method

#### C++

```
TMmRc rsaPrivateKeyDecCmp(char* plainData, uint32_t plainLen, char* cipherData, uint16_t&
isoStatus);
```

#### Remarks

This is rsaPrivateKeyDecCmp, a member of class SecurityAccess.

# 2.1.40.7.35 SecurityAccess::selfTest Method

#### C++

```
TMmRc selfTest(int32_t& status, uint16_t& isoStatus);
```

#### Remarks

This is selfTest, a member of class SecurityAccess.

# 2.1.40.7.36 SecurityAccess::setCineLink2Keys Method

#### C++

```
TMmRc setCineLink2Keys(CineLink2Para keys[4]);
```

## Remarks

This is setCineLink2Keys, a member of class SecurityAccess.

# 2.1.40.7.37 SecurityAccess::setForensicMarkingId Method

#### C++

```
TMmRc setForensicMarkingId(uint8_t vendorID, uint16_t serial);
```

## Remarks

This is setForensicMarkingId, a member of class SecurityAccess.

# 2.1.40.7.38 SecurityAccess::setForensicMarkingIndicator Method

#### C++

```
TMmRc setForensicMarkingIndicator(const VideoOutput & output, bool enable);
```

#### Remarks

This is setForensicMarkingIndicator, a member of class SecurityAccess.

# 2.1.40.7.39 SecurityAccess::setHardwareInfo Method

#### C++

TMmRc setHardwareInfo(uint32\_t index, uint32\_t mask, uint32\_t value);

#### Remarks

This is setHardwareInfo, a member of class SecurityAccess.

# 2.1.40.7.40 SecurityAccess::setHashData Method

Transfers data to target for sha256 hash calculation.

## C++

```
TMmRc setHashData(mvc2::SecurityAccess::SoftwareId, mvc2::SecurityAccess::HashAlgorithm eHashAlg, const int32_t currentBufferIndex, const int8_t* buffer, int32_t bufferSize, int32_t bufferLeftSize);
```

#### **Parameters**

Parameters	Description
mvc2::SecurityAccess::SoftwareId softwareId	identfies the software that should calculate the hash value
mvc2::SecurityAccess::HashAlgorithm eHashAlg	identifies the hash algorithm that should be used
const int32_t currentBufferIndex	the index of current buffer that should be send to target, 0 is the index of the first buffer, it initializes the algorithm at target
const int8_t* buffer	the data to be hashed, max buffer size depends on max packet size of underlying transport, fragmentation is not supported
int32_t bufferSize	the size of data

## Returns

should be MMRC\_Ok or MMRC\_NotImplemented

#### Description

This function is used to transfer data for sha256 hash calculation to target. Consecutive calls are possible before a call to getSha256HashValue() retrieves the output of the calculation, a sha256 hash value

# 2.1.40.7.41 SecurityAccess::setMediaDecryptorKey Method

Set media decryption key.

#### C++

```
TMmRc setMediaDecryptorKey(const uint16_t keyIndex, const uint8_t * key, const MediaType type, bool enableFM);
```

## **Parameters**

Parameters	Description
const uint16_t keyIndex	key index inside the key storage
const uint8_t * key	128 bit (16 byte) AES 128 key (big endian)
const MediaType type	media type, which will be decrypted with the key (needed for forensic marking)
bool enableFM	enable the forensic marking if this key is used for decryption

#### Returns

Returns MMRC\_Ok if the key was successful stored.

## **Return Values**

Return Values	Description
MMRC_Ok	successful operation

#### Description

The method is used to address the key storage for media decryption. The MVC card can decrypt input data using a AES128 engine. It has a space for 256 keys.

See also DataBuffer::setKeyIndex().

# 2.1.40.7.42 SecurityAccess::symmetricCryptoFinish Method

#### C++

```
TMmRc symmetricCryptoFinish(mvc2::SecurityAccess::SoftwareId softwareId);
```

#### Remarks

This is symmetricCryptoFinish, a member of class SecurityAccess.

# 2.1.40.7.43 SecurityAccess::symmetricCryptoInit Method

initializes the calculation of symmetric encryption/decryption

#### C++

```
TMmRc symmetricCryptoInit(mvc2::SecurityAccess::SoftwareId softwareId, mvc2::SecurityAccess::SymmetricCryptoOperation symmetricCryptoOp, const uint8_t* key, const uint8_t* iv);
```

#### **Parameters**

Parameters	Description
mvc2::SecurityAccess::SoftwareId softwareId	identifies the software that should encrypt or decrypt
mvc2::SecurityAccess::SymmetricCryptoOperation symmetricCryptoOp	determines the crypto operation
const uint8_t* key	the key that should be used, valid key size depends on algorithm
const uint8_t* iv	the iv that should be used, valid iv size depends on algorithm, on those without iv, this parameter is ignored

#### **Returns**

should be MMRC\_Ok on success

# 2.1.40.7.44 SecurityAccess::symmetricCryptoProcess Method

This function processes input data and gives the encrypted or decrypted result.

#### C++

```
TMmRc symmetricCryptoProcess(mvc2::SecurityAccess::SoftwareId softwareId, int32_t index,
const uint8_t* inBuf, uint32_t inLen, uint8_t* outBuf);
```

## **Parameters**

Parameters	Description
mvc2::SecurityAccess::SoftwareId softwareId	the software that should process the input data
const uint8_t* inBuf	input data (plain-text resp. cipher-text)
uint32_t inLen	number of bytes that should be processed, this must not exceed 2048 and should be a multiple of according blocksize of algorithm, no padding inside

uint8_t* outBuf	this contains the processed data, output size is equal to input size
counter	of operation since last call to symmetricCryptolnit, should
	begin with 0

# 2.1.40.7.45 SecurityAccess::validateHardwareInfo Method

#### C++

```
TMmRc validateHardwareInfo(uint32_t index, uint32_t mask);
```

#### Remarks

This is validateHardwareInfo, a member of class SecurityAccess.

# 2.1.40.7.46 SecurityAccess::verify Method

#### C++

```
TMmRc verify(mvc2::SecurityAccess::HashAlgorithm eHashAlg, uint8_t* signature, uint32_t u32SignatureSize, uint8_t* hashValue, uint32_t* readSize);
```

#### Remarks

This is verify, a member of class SecurityAccess.

# 2.1.40.7.47 SecurityAccess::writeRtcReg Method

#### C++

```
TMmRc writeRtcReg(uint32_t offset, uint8_t value, bool askSm);
```

#### Remarks

This is writeRtcReg, a member of class SecurityAccess.

# 2.1.40.7.48 SecurityAccess::writeSecurityInterface Method

Writes data to security interface.

### C++

```
TMmRc writeSecurityInterface(uint8_t address, uint8_t* writeData, uint32_t writeDataSize);
```

#### **Parameters**

Parameters	Description
uint8_t address	address to write to (0255)
uint8_t* writeData	buffer to with data to write
uint32 t writeDataSize	number of bytes to write

## Returns

MMRC\_Ok on success

# 2.1.40.8 SecurityAccess Operators

# 2.1.40.8.1 SecurityAccess::= Operator

Default assignment operator.

#### C++

SecurityAccess& operator =(const SecurityAccess& other);

#### **Parameters**

Parameters	Description
const SecurityAccess& other	source object

#### Returns

Pointer to itself.

## Description

Copies on all information from one SecurityAccess object.

# 2.1.40.8.2 SecurityAccess::bool Operator

Check for object validity.

C++

operator bool() const;

#### Description

This method is used to check if a object is filled with valid data and has there for a connection to a MvcDevice.

# 2.1.41 SecurityManager Class



Security manager API implementation.

## **Inheritance Hierarchy**

C++

class SecurityManager;

File

mvc2api\_securitymanager.h

## Description

This class provides an interface to the MediaBlock internal security manager. The connection to the security manager must be save, that's why it implements a TLS connection for the communication. Therefore a simple object create is not enough here and a TLS connection handshaking must be done before sending any security manager commands. To establish the connection a SMS certification chain is needed which fits to the DCI certification chain installed on the IMB. First setup the chain using addCertificate()/loadCertificateChainFile() and usePrivateKey()/loadPrivateKeyFile(). After that call connect() and the TLS connection will be established. Any security manager function can be used now. The TLS connection will automatically be closed on deletion of the object.

## Members

## SecurityManager Data Members

	Name	Description
•	c_ProjectorCertificate	This is c_ProjectorCertificate, a member of class SecurityManager.

#### SecurityManager Enumerations

Name	Description
CPL_ASSET_EXCEPTION	This is record mvc2::SecurityManager::CPL_ASSET_EXCEPTION.

<b>1</b>	SM_OPERATION	This is record mvc2::SecurityManager::SM_OPERATION.
<b>a</b>	SUITE_STATUS	This is record mvc2::SecurityManager::SUITE_STATUS.

# SecurityManager Friends

Name	Description
class MvcDecoder	This is friend friend class MvcDecoder.

# SecurityManager Methods

	Name	Description
<b>=</b> ♦	addCertificate	Build a SMS certification chain.
<b>=</b> ♦	adjustTime	This is adjustTime, a member of class SecurityManager.
<b>=♦</b>	checkShow	This is checkShow, a member of class SecurityManager.
<b>≡</b>	connect	Setup security manager connection.
<b>=</b> ♦	getCertificate	This is getCertificate, a member of class SecurityManager.
<b>=</b> ♦	getCplList	This is getCplList, a member of class SecurityManager.
<b>=♦</b>	getKdmList	This is getKdmList, a member of class SecurityManager.
<b>=♦</b>	getKeyMap	This is getKeyMap, a member of class SecurityManager.
<b>=♦</b>	getLogReport	This is getLogReport, a member of class SecurityManager.
<b>=</b>	initiateMarriage	This is initiateMarriage, a member of class SecurityManager.
<b>≞♦</b>	loadCertificateChainFile	This is loadCertificateChainFile, a member of class SecurityManager.
<b>≡♦</b>	loadPrivateKeyFile	This is loadPrivateKeyFile, a member of class SecurityManager.
<b>≡</b>	playShow	Prepare and activate a show.
<b>≡</b>	purgeCpl	This is purgeCpl, a member of class SecurityManager.
<b>=♦</b>	queryStatus	This is queryStatus, a member of class SecurityManager.
<b>=♦</b>	serviceDoorTamperTermination	This is serviceDoorTamperTermination, a member of class SecurityManager.
<b>≡</b>	startSuite	Start the security manager suite.
<b>≡</b>	startSuite	Start the security manager suite.
<b>=♦</b>	stopShow	This is stopShow, a member of class SecurityManager.
<b>≡</b>	stopSuite	This is stopSuite, a member of class SecurityManager.
<b>≡</b>	uploadCplBuffer	This is uploadCplBuffer, a member of class SecurityManager.
<b>≡</b>	uploadCplFile	This is uploadCplFile, a member of class SecurityManager.
<b>≡</b>	uploadKdmBuffer	This is uploadKdmBuffer, a member of class SecurityManager.
<b>≡</b>	uploadKdmFile	This is uploadKdmFile, a member of class SecurityManager.
<b>≡</b>	usePrivateKey	Set private to use within the TLS connection.
<b>=</b>	zeroize	This is zeroize, a member of class SecurityManager.

# SecurityManager Nested Types

Name	Description
AesKeyMap	This is nested type mvc2::SecurityManager::AesKeyMap.
КеуМар	This is nested type mvc2::SecurityManager::KeyMap.
SpblpAddress	This is nested type mvc2::SecurityManager::SpblpAddress.

# SecurityManager Operators

	Name	Description
=+)	=	This is =, a member of class SecurityManager.
=+)	bool	Check for object validity.

## Methods

	Name	Description
<b>=♦</b> ₩	~SecurityManager	This is ~SecurityManager, a member of class SecurityManager.
<b>≡</b>	SecurityManager	This is SecurityManager, a member of class SecurityManager.
<b>≡♦</b>	SecurityManager	This is SecurityManager, a member of class SecurityManager.
<b>≡</b> ♦	SecurityManager	This is SecurityManager, a member of class SecurityManager.

# SecurityManager Data Members

	Name	Description
•	c_ProjectorCertificate	This is c_ProjectorCertificate, a member of class
		SecurityManager.

# SecurityManager Enumerations

	Name	Description
<b>a</b>	CPL_ASSET_EXCEPTION	This is record mvc2::SecurityManager::CPL_ASSET_EXCEPTION.
<b>a</b>	SM_OPERATION	This is record mvc2::SecurityManager::SM_OPERATION.
<b>a</b>	SUITE_STATUS	This is record mvc2::SecurityManager::SUITE_STATUS.

# SecurityManager Friends

Name	Description
class MvcDecoder	This is friend friend class MvcDecoder.

# **SecurityManager Methods**

	Name	Description
<b>=♦</b>	addCertificate	Build a SMS certification chain.
<b>=♦</b>	adjustTime	This is adjustTime, a member of class SecurityManager.
<b>=♦</b>	checkShow	This is checkShow, a member of class SecurityManager.
<b>=♦</b>	connect	Setup security manager connection.
<b>=♦</b>	getCertificate	This is getCertificate, a member of class SecurityManager.
<b>=♦</b>	getCplList	This is getCplList, a member of class SecurityManager.
<b>=♦</b>	getKdmList	This is getKdmList, a member of class SecurityManager.
<b>=♦</b>	getKeyMap	This is getKeyMap, a member of class SecurityManager.
<b>≡</b>	getLogReport	This is getLogReport, a member of class SecurityManager.
<b>≡</b>	initiateMarriage	This is initiateMarriage, a member of class SecurityManager.
<b>∉</b>	IoadCertificateChainFile	This is loadCertificateChainFile, a member of class SecurityManager.
<b>∉</b> ∳	loadPrivateKeyFile	This is loadPrivateKeyFile, a member of class SecurityManager.
<b>≡</b>	playShow	Prepare and activate a show.
<b>≡</b>	purgeCpl	This is purgeCpl, a member of class SecurityManager.
<b>=♦</b>	queryStatus	This is queryStatus, a member of class SecurityManager.
<b>≡</b>	serviceDoorTamperTermination	This is serviceDoorTamperTermination, a member of class SecurityManager.
<b>≡</b>	startSuite	Start the security manager suite.
<b>=♦</b>	startSuite	Start the security manager suite.
<b>=♦</b>	stopShow	This is stopShow, a member of class SecurityManager.
<b>=♦</b>	stopSuite	This is stopSuite, a member of class SecurityManager.

<b>=♦</b>	uploadCplBuffer	This is uploadCplBuffer, a member of class SecurityManager.
<b>=♦</b>	uploadCplFile	This is uploadCplFile, a member of class SecurityManager.
=•	uploadKdmBuffer	This is uploadKdmBuffer, a member of class SecurityManager.
<b>=♦</b>	uploadKdmFile	This is uploadKdmFile, a member of class SecurityManager.
<b>=♦</b>	usePrivateKey	Set private to use within the TLS connection.
<b>≡♦</b>	zeroize	This is zeroize, a member of class SecurityManager.

## **SecurityManager Nested Types**

Name	Description
AesKeyMap	This is nested type mvc2::SecurityManager::AesKeyMap.
KeyMap	This is nested type mvc2::SecurityManager::KeyMap.
SpblpAddress	This is nested type mvc2::SecurityManager::SpblpAddress.

#### **SecurityManager Operators**

	Name	Description
=+)	=	This is =, a member of class SecurityManager.
=+)	bool	Check for object validity.

# 2.1.41.1 SecurityManager Data Members

# 2.1.41.1.1 SecurityManager::c\_ProjectorCertificate Data Member

#### C++

```
const uint32_t c_ProjectorCertificate = 10000;
```

#### Remarks

This is c\_ProjectorCertificate, a member of class SecurityManager.

# 2.1.41.2 SecurityManager Enumerations

# 2.1.41.2.1 mvc2::SecurityManager::CPL\_ASSET\_EXCEPTION Enumeration

### C++

```
enum CPL_ASSET_EXCEPTION {
  CplException_None = 0,
  CplException_Hash = 1,
  CplException_Missing = 2
};
```

## File

mvc2api\_securitymanager.h

#### Remarks

This is record mvc2::SecurityManager::CPL\_ASSET\_EXCEPTION.

# 2.1.41.2.2 mvc2::SecurityManager::SM\_OPERATION Enumeration

## C++

```
enum SM_OPERATION {
  Operation_Unknown = 0,
```

192

```
Operation_NotPlaying = 1,
Operation_Playing = 2
};
```

### File

mvc2api\_securitymanager.h

### **Members**

Members	Description
Operation_NotPlaying = 1	idle

### Remarks

This is record mvc2::SecurityManager::SM\_OPERATION.

# 2.1.41.2.3 mvc2::SecurityManager::SUITE\_STATUS Enumeration

#### $C^{++}$

```
enum SUITE_STATUS {
   Suite_Started = 1,
   Suite_Stopped = 2,
   Suite_Progress = 3,
   Suite_Restarted = 4,
   Suite_Lost = 5,
   Suite_Error = 6
};
```

### File

mvc2api\_securitymanager.h

### Remarks

This is record mvc2::SecurityManager::SUITE\_STATUS.

# 2.1.41.3 SecurityManager Friends

# 2.1.41.3.1 friend class MvcDecoder Friend

## C++

friend class MvcDecoder;

# Remarks

This is friend friend class MvcDecoder.

# 2.1.41.4 SecurityManager Methods

# 2.1.41.4.1 SecurityManager::addCertificate Method

Build a SMS certification chain.

# C++

```
TMmRc addCertificate(void * cert);
```

## **Parameters**

Parameters	Description
void * cert	pointer to a X509 structure with the certificate

## **Returns**

MMRC\_Ok on success.

## Description

This method adds a new certificate to create a complete chain which will be used to open the TLS connection to the security manager. The first certificate is the SMS certificate, where a private key is also needed (see usePrivateKey). The second certificate is an intermediate and the last one the root certificate.

# 2.1.41.4.2 SecurityManager::adjustTime Method

## C++

```
TMmRc adjustTime(const char * authId, int32_t diff);
```

## Remarks

This is adjustTime, a member of class SecurityManager.

# 2.1.41.4.3 SecurityManager::checkShow Method

### C++

```
TMmRc checkShow(const UuidValue * cplUuidArray, uint32_t arrayLen, uint64_t checkTime,
uint64_t * keyExpTime = 0);
```

### Remarks

This is checkShow, a member of class SecurityManager.

# 2.1.41.4.4 SecurityManager::connect Method

Setup security manager connection.

# C++

```
TMmRc connect();
```

## Returns

MMRC\_Ok on success.

## **Description**

After certificate chain and private key setup a TLS connection can be established. Connect() will do all the necessary work with the handshaking and result ideally in a successful established TLS connection.

# 2.1.41.4.5 SecurityManager::getCertificate Method

# C++

```
TMmRc getCertificate(uint32_t which, uint8_t * buffer, uint32_t * buffersize);
```

## Remarks

This is getCertificate, a member of class SecurityManager.

# 2.1.41.4.6 SecurityManager::getCplList Method

## C++

```
TMmRc getCplList(UuidValue * cplUuidArray, uint32_t * arrayLen);
```

# Remarks

This is getCplList, a member of class SecurityManager.

# 2.1.41.4.7 SecurityManager::getKdmList Method

#### C++

```
TMmRc getKdmList(const UuidValue & cplUuid, UuidValue * kdmUuidArray, uint32_t * arrayLen);
```

#### Remarks

This is getKdmList, a member of class SecurityManager.

# 2.1.41.4.8 SecurityManager::getKeyMap Method

### C++

```
TMmRc getKeyMap(KeyMap * keymapArray, uint32_t * numberOfEntries);
```

## Remarks

This is getKeyMap, a member of class SecurityManager.

# 2.1.41.4.9 SecurityManager::getLogReport Method

#### C++

```
TMmRc getLogReport(uint8_t * buffer, uint32_t * buffersize, uint64_t startTime, uint64_t
endTime, uint64_t * lastLogTime);
```

### Remarks

This is getLogReport, a member of class SecurityManager.

# 2.1.41.4.10 SecurityManager::initiateMarriage Method

## C++

```
TMmRc initiateMarriage(const char * authId);
```

## Remarks

This is initiateMarriage, a member of class SecurityManager.

# 2.1.41.4.11 SecurityManager::loadCertificateChainFile Method

## C++

```
TMmRc loadCertificateChainFile(const char * filename);
```

## Remarks

This is loadCertificateChainFile, a member of class SecurityManager.

# 2.1.41.4.12 SecurityManager::loadPrivateKeyFile Method

## C++

```
TMmRc loadPrivateKeyFile(const char * filename);
```

# Remarks

This is loadPrivateKeyFile, a member of class SecurityManager.

# 2.1.41.4.13 SecurityManager::playShow Method

Prepare and activate a show.

2.1 mvc2 Namespace MVC20x API SecurityManager Class

### C++

```
TMmRc playShow(const char * authId, const UuidValue * cplUuidArray, uint32_t arrayLen,
uint64_t * keyExpTime = 0);
```

#### **Parameters**

Parameters	Description
const char * authld	authentication identifier used for logs
const UuidValue * cplUuidArray	array of CPL UUIDs used by the show
uint32_t arrayLen	number of entries of cplUuidArray
uint64_t * keyExpTime = 0	pointer to a variable which receives the minimum expire time of the show

## **Returns**

MMRC\_Ok if the show was activated, otherwise an error.

# Description

This function will prepare and activate a should which consists of multi CPLs. All CPLs and the according KDMs must be uploaded before using this function. The keys will be activated and the key uuid to key index translation table will be saves inside the security manager object. The automatic key index translation is used if the security manager object is set in the decoder (using MvcDecoder::setSecurityManager Method). The function also returns the maximum time the show is valid. This valid has only a informational purpose and can be used by the SMS to inform the user.

# 2.1.41.4.14 SecurityManager::purgeCpl Method

#### C++

```
TMmRc purgeCpl(const UuidValue & cplUuid);
```

## Remarks

This is purgeCpl, a member of class SecurityManager.

# 2.1.41.4.15 SecurityManager::queryStatus Method

## C++

```
TMmRc queryStatus(SM_OPERATION * smOp, SUITE_STATUS * suiteStat, uint64_t * secureTime, int32_t * time_adjust);
```

## Remarks

This is queryStatus, a member of class SecurityManager.

# 2.1.41.4.16 SecurityManager::serviceDoorTamperTermination Method

# C++

```
TMmRc serviceDoorTamperTermination(const char * authId);
```

## Remarks

This is serviceDoorTamperTermination, a member of class SecurityManager.

# 2.1.41.4.17 SecurityManager::startSuite Method (SpblpAddress \*, uint32\_t)

Start the security manager suite.

## C++

```
TMmRc startSuite(SpbIpAddress * spbList, uint32_t spbCount);
```

### **Parameters**

Parameters	Description
SpblpAddress * spbList	pointer to a array of ip addresses
uint32_t spbCount	number of entries of spbList

#### Returns

MMRC\_Ok will be return if the suite was started successfully, else an error.

## Description

The security manager suite will be started and all connected SPBs will be saved. The SPBs will be used later to check against the trusted device list. Normally you should provide one or more projector ip addresses here. In case of a MVC201 the list can be empty, then the MediaBlock will use the internal projector connection.

# 2.1.41.4.18 SecurityManager::startSuite Method (char \*)

Start the security manager suite.

# C++

```
TMmRc startSuite(const char * spbList = 0);
```

#### **Parameters**

Parameters	Description
const char * spbList = 0	string with multiple ip addresses separated by ',' or '/'

## Returns

MMRC\_Ok on successful operation.

# Description

The security manager suite will be started and all connected SPBs will be saved. The SPBs will be used later to check against the trusted device list. Normally you should provide one or more projector ip addresses here. In case of a MVC201 the list can be empty, then the MediaBlock will use the internal projector connection.

# 2.1.41.4.19 SecurityManager::stopShow Method

# C++

```
TMmRc stopShow(const char * authId);
```

## Remarks

This is stopShow, a member of class SecurityManager.

# 2.1.41.4.20 SecurityManager::stopSuite Method

## C++

```
TMmRc stopSuite();
```

# Remarks

This is stopSuite, a member of class SecurityManager.

# 2.1.41.4.21 SecurityManager::uploadCplBuffer Method

## C++

```
TMmRc uploadCplBuffer(const uint8_t * cplBuffer, uint32_t bufferSize, CPL_ASSET_EXCEPTION
assetExc = CplException_None);
```

### Remarks

This is uploadCplBuffer, a member of class SecurityManager.

# 2.1.41.4.22 SecurityManager::uploadCplFile Method

#### C++

```
TMmRc uploadCplFile(const char * cplPath, CPL_ASSET_EXCEPTION assetExc = CplException_None);
```

## Remarks

This is uploadCplFile, a member of class SecurityManager.

# 2.1.41.4.23 SecurityManager::uploadKdmBuffer Method

## C++

```
TMmRc uploadKdmBuffer(const uint8_t * kdmBuffer, uint32_t bufferSize, AesKeyMap *
subKeyArray = 0, uint32_t * numberOfEntries = 0);
```

## Remarks

This is uploadKdmBuffer, a member of class SecurityManager.

# 2.1.41.4.24 SecurityManager::uploadKdmFile Method

### C++

```
TMmRc uploadKdmFile(const char * kdmPath, AesKeyMap * subKeyArray = 0, uint32_t *
numberOfEntries = 0);
```

#### Remarks

This is uploadKdmFile, a member of class SecurityManager.

# 2.1.41.4.25 SecurityManager::usePrivateKey Method

Set private to use within the TLS connection.

## C++

```
TMmRc usePrivateKey(void * pkey);
```

# **Parameters**

Parameters	Description
void * pkey	pointer to a EVP_PKEY with the private key

## Returns

MMRC\_Ok on success.

# Description

After setting up the certificate chain for the TLS connection, the private key for the SMS is needed. This function is used to set it.

# 2.1.41.4.26 SecurityManager::zeroize Method

## C++

```
TMmRc zeroize(const char * authId);
```

# Remarks

This is zeroize, a member of class SecurityManager.

# 2.1.41.5 SecurityManager Nested Types

# 2.1.41.5.1 SecurityManager::AesKeyMap Nested Type

#### C++

```
typedef struct {
   UuidValue key_id;
   uint8_t aes_key[16];
} AesKeyMap;
```

#### Remarks

This is nested type mvc2::SecurityManager::AesKeyMap.

# 2.1.41.5.2 SecurityManager::KeyMap Nested Type

#### C++

```
typedef struct {
   UuidValue key_id;
   UuidValue kdm_id;
   uint16_t key_index;
   uint8_t key_type;
}
```

## Remarks

This is nested type mvc2::SecurityManager::KeyMap.

# 2.1.41.5.3 SecurityManager::SpblpAddress Nested Type

## C++

```
typedef struct {
  uint8_t b3;
  uint8_t b2;
  uint8_t b1;
  uint8_t b0;
} SpbIpAddress;
```

## Members

Members		Description
	uint8_t b0;	as in b0.b1.b2.b3

## Remarks

This is nested type mvc2::SecurityManager::SpblpAddress.

# 2.1.41.6 SecurityManager Operators

# 2.1.41.6.1 SecurityManager::= Operator

# C++

```
SecurityManager& operator =(const SecurityManager& other);
```

## Remarks

This is =, a member of class SecurityManager.

# 2.1.41.6.2 SecurityManager::bool Operator

Check for object validity.

C++

```
operator bool() const;
```

## Description

This method is used to check if a object is filled with valid data and has there for a connection to a MvcDevice.

# 2.1.41.7 SecurityManager::~SecurityManager Destructor

C++

```
virtual ~SecurityManager();
```

#### Remarks

This is ~SecurityManager, a member of class SecurityManager.

# 2.1.41.8 SecurityManager::SecurityManager Constructor ()

C++

```
SecurityManager();
```

#### Remarks

This is SecurityManager, a member of class SecurityManager.

# 2.1.41.9 SecurityManager::SecurityManager Constructor (SecurityManager&)

C++

```
SecurityManager(const SecurityManager& other);
```

## Remarks

This is SecurityManager, a member of class SecurityManager.

# 2.1.41.10 SecurityManager::SecurityManager Constructor (TMmRc \*, MvcDevice &)

C++

```
SecurityManager(TMmRc * resultPointer, const MvcDevice & mvcDevice);
```

## Remarks

This is SecurityManager, a member of class SecurityManager.

# 2.1.42 SubtitleDecoder Class

Subtitle overlay renderer.

# **Inheritance Hierarchy**

## C++

class SubtitleDecoder : public MvcDecoder;

### File

mvc2api\_overlay.h

# Description

This class is a special overlay decoder which renders CineCanvas<sup>™</sup> and SMPTE 428-7 into the overlay buffer and handles the output timing.

# **Members**

## Methods

	Name	Description
<b>=</b> ♦ ₩	~MvcDecoder	This is ~MvcDecoder, a member of class MvcDecoder.
<b>≡</b>	MvcDecoder	This is MvcDecoder, a member of class MvcDecoder.
<b>=♦</b>	MvcDecoder	This is MvcDecoder, a member of class MvcDecoder.
<b>=Q</b> <sub><b>Q</b></sub>	MvcDecoder	This is MvcDecoder, a member of class MvcDecoder.

## **SubtitleDecoder Class**

	Name	Description
<b>=</b> ♦ ₩	~SubtitleDecoder	This is ~SubtitleDecoder, a member of class SubtitleDecoder.
<b>=</b> ♦	SubtitleDecoder	This is SubtitleDecoder, a member of class SubtitleDecoder.
<b>≡♦</b>	SubtitleDecoder	This is SubtitleDecoder, a member of class SubtitleDecoder.
<b>≡♦</b>	SubtitleDecoder	This is SubtitleDecoder, a member of class SubtitleDecoder.

## **MvcDecoder Enumerations**

	Name	Description
<b>a</b>	DecoderType	This is record mvc2::MvcDecoder::DecoderType.
<b>a</b>	TIMEOUT	This is record mvc2::MvcDecoder::TIMEOUT.

# **SubtitleDecoder Class**

	Name	Description
<b>a</b>	RENDERFLAGS	This is record mvc2::SubtitleDecoder::RENDERFLAGS.

# **MvcDecoder Methods**

	Name	Description
<b>≡∳</b>	flush	This is flush, a member of class MvcDecoder.
<b>≡</b>	getBufferStatus	This is getBufferStatus, a member of class MvcDecoder.
<b>≡</b>	getChannel	Gets the used channel number.
<b>=</b> ♦ ₩	getDataBuffer	Return a buffer for data transfer.
<b>=♦</b>	getDecoderErrors	This is getDecoderErrors, a member of class MvcDecoder.
<b>≡♦</b>	getDecoderStatus	This is getDecoderStatus, a member of class MvcDecoder.
<b>≡♦</b>	getDecoderType	This is getDecoderType, a member of class MvcDecoder.
<b>≡\$</b> }	getPrivate	This is getPrivate, a member of class MvcDecoder.
<b>∉</b> ∳	setBackwardPlayback	This is setBackwardPlayback, a member of class MvcDecoder.
<b>≡♦</b>	setCplUid	This is setCplUid, a member of class MvcDecoder.
<b>=♦</b>	setEndOfStream	Go to end of stream state.
<b>=♦</b>	setSecurityManager	This is setSecurityManager, a member of class MvcDecoder.
<b>=♦</b>	setStartDelay	Setup start delay for the decoder.
<b>=♦</b>	waitDecoderStatus	This is waitDecoderStatus, a member of class MvcDecoder.

<b>≡♦</b>	waitForTimeStamp	This is waitForTimeStamp, a member of class MvcDecoder.
<b>=♦</b>	waitForTransferFinish	This is waitForTransferFinish, a member of class MvcDecoder.
<b>≡♦</b>	waitForUserData	This is waitForUserData, a member of class MvcDecoder.

# **SubtitleDecoder Class**

	Name	Description
<b>=</b> ♦	connectOutput	Connects an output object to the decoder.
<b>≡♦</b>	disableSubtitles	This is disableSubtitles, a member of class SubtitleDecoder.
<b>=</b> ♦	disconnectOutput	Disconnects the output.
<b>=</b> ♦	enableSubtitles	This is enableSubtitles, a member of class SubtitleDecoder.
<b>=♦</b> ₩	getDataBuffer	This is getDataBuffer, a member of class SubtitleDecoder.
<b>=</b> ♦ ₩	getDataBuffer	This is getDataBuffer, a member of class SubtitleDecoder.
=•	getOverlayElementRequest	This is getOverlayElementRequest, a member of class SubtitleDecoder.
=♦	sendOverlayElement	This is sendOverlayElement, a member of class SubtitleDecoder.
<b>=♦</b>	sendSubtitleFile	This is sendSubtitleFile, a member of class SubtitleDecoder.
<b>=</b> ♦	setOutputResolution	This is setOutputResolution, a member of class SubtitleDecoder.
<b>=♦</b>	setZOffset	This is setZOffset, a member of class SubtitleDecoder.

# **MvcDecoder Operators**

	Name	Description
( <u>/-</u> =+)	=	This is =, a member of class MvcDecoder.
=+)	bool	Returns if the object is valid.

# **SubtitleDecoder Class**

		Name	Description
=	<u>/-</u> =+)	=	This is =, a member of class SubtitleDecoder.

# **MvcDecoder Enumerations**

	Name	Description
e <sup>a</sup>	DecoderType	This is record mvc2::MvcDecoder::DecoderType.
<b>=</b>	TIMEOUT	This is record mvc2::MvcDecoder::TIMEOUT.

# SubtitleDecoder Class

	Name	Description
<b>a</b>	RENDERFLAGS	This is record mvc2::SubtitleDecoder::RENDERFLAGS.

# **MvcDecoder Methods**

	Name	Description
<b>≡</b>	flush	This is flush, a member of class MvcDecoder.
<b>≡</b>	getBufferStatus	This is getBufferStatus, a member of class MvcDecoder.
<b>≡</b>	getChannel	Gets the used channel number.
<b>=♦</b> ₩	getDataBuffer	Return a buffer for data transfer.
<b>≡∳</b>	getDecoderErrors	This is getDecoderErrors, a member of class MvcDecoder.
<b>≡</b>	getDecoderStatus	This is getDecoderStatus, a member of class MvcDecoder.
<b>≡</b>	getDecoderType	This is getDecoderType, a member of class MvcDecoder.
<b>≡</b> ∳ <sub></sub>	getPrivate	This is getPrivate, a member of class MvcDecoder.
<b>≡♦</b>	setBackwardPlayback	This is setBackwardPlayback, a member of class MvcDecoder.

<b>≡</b>	setCplUid	This is setCplUid, a member of class MvcDecoder.
<b>≡♦</b>	setEndOfStream	Go to end of stream state.
<b>=</b> ♦	setSecurityManager	This is setSecurityManager, a member of class MvcDecoder.
<b>=♦</b>	setStartDelay	Setup start delay for the decoder.
<b>=♦</b>	waitDecoderStatus	This is waitDecoderStatus, a member of class MvcDecoder.
<b>≡</b>	waitForTimeStamp	This is waitForTimeStamp, a member of class MvcDecoder.
<b>≡♦</b>	waitForTransferFinish	This is waitForTransferFinish, a member of class MvcDecoder.
<b>=♦</b>	waitForUserData	This is waitForUserData, a member of class MvcDecoder.

### **SubtitleDecoder Class**

	Name	Description
<b>≡♦</b>	connectOutput	Connects an output object to the decoder.
<b>=♦</b>	disableSubtitles	This is disableSubtitles, a member of class SubtitleDecoder.
<b>=♦</b>	disconnectOutput	Disconnects the output.
<b>=♦</b>	enableSubtitles	This is enableSubtitles, a member of class SubtitleDecoder.
<b>=</b> ♦ ₩	getDataBuffer	This is getDataBuffer, a member of class SubtitleDecoder.
<b>=♦</b> ₩	getDataBuffer	This is getDataBuffer, a member of class SubtitleDecoder.
<b>∉</b> ∳	getOverlayElementRequest	This is getOverlayElementRequest, a member of class SubtitleDecoder.
<b>∉</b> ∳	sendOverlayElement	This is sendOverlayElement, a member of class SubtitleDecoder.
<b>≡♦</b>	sendSubtitleFile	This is sendSubtitleFile, a member of class SubtitleDecoder.
<b>≡</b>	setOutputResolution	This is setOutputResolution, a member of class SubtitleDecoder.
<b>≡♦</b>	setZOffset	This is setZOffset, a member of class SubtitleDecoder.

# **MvcDecoder Operators**

	Name	Description
( <u>/-</u>	=	This is =, a member of class MvcDecoder.
( <u>/-</u> =+)	bool	Returns if the object is valid.

## **SubtitleDecoder Class**

	Name	Description
<u>(/-</u> =+)	=	This is =, a member of class SubtitleDecoder.

# 2.1.42.1 SubtitleDecoder::~SubtitleDecoder Destructor

## C++

```
virtual ~SubtitleDecoder();
```

# Remarks

This is ~SubtitleDecoder, a member of class SubtitleDecoder.

# 2.1.42.2 SubtitleDecoder::SubtitleDecoder Constructor ()

## $C^{++}$

SubtitleDecoder();

# Remarks

This is SubtitleDecoder, a member of class SubtitleDecoder.

# 2.1.42.3 SubtitleDecoder::SubtitleDecoder Constructor (SubtitleDecoder&)

C++

```
SubtitleDecoder(const SubtitleDecoder& other);
```

### Remarks

This is SubtitleDecoder, a member of class SubtitleDecoder.

# 2.1.42.4 SubtitleDecoder::SubtitleDecoder Constructor (TMmRc \*, MvcDevice &)

C++

```
SubtitleDecoder(TMmRc * resultPointer, const MvcDevice & mvcdev);
```

## Remarks

This is SubtitleDecoder, a member of class SubtitleDecoder.

# 2.1.42.5 SubtitleDecoder Enumerations

# 2.1.42.5.1 mvc2::SubtitleDecoder::RENDERFLAGS Enumeration

C++

```
enum RENDERFLAGS {
   Render_Force_Shadow = (1<<0),
   Render_Force_Border = (1<<1),
   Render_Soft_Shadows = (1<<2),
   Render_3D_Mode = (1<<3),
   Render_BaseResolution_2K = (0<<4),
   Render_BaseResolution_4K = (1<<4)
};

File
   mvc2api_overlay.h</pre>
```

## Remarks

This is record mvc2::SubtitleDecoder::RENDERFLAGS.

# 2.1.42.6 SubtitleDecoder Methods

# 2.1.42.6.1 SubtitleDecoder::connectOutput Method

Connects an output object to the decoder.

C++

```
TMmRc connectOutput(const VideoOutput& VideoOut);
```

## **Parameters**

Parameters	Description
const VideoOutput& VideoOut	dedicated VideoOutput object.

## **Returns**

MMRC\_Ok or an error if the connection fails.

## **Return Values**

Return Values	Description
MMRC_Ok	Output successfully connected
MMRC_MVC2_ConnectionError	Output could not be connected, it may not be compatible

## Description

This method is used to assign an output to the decoder which is used to display the decoder frames.

# 2.1.42.6.2 SubtitleDecoder::disableSubtitles Method

### C++

```
TMmRc disableSubtitles();
```

#### Remarks

This is disableSubtitles, a member of class SubtitleDecoder.

# 2.1.42.6.3 SubtitleDecoder::disconnectOutput Method

Disconnects the output.

#### C++

TMmRc disconnectOutput(const VideoOutput& VideoOut);

### **Parameters**

Parameters	Description
const VideoOutput& VideoOut	previously connected output

# Returns

MMRC\_Ok if the disconnection was successful, else an error.

## **Return Values**

Return Values	Description
MMRC_Ok	successful disconnections
MMRC_MVC2_ConnectionError	object is not connected to the decoder

# Description

Disconnects a previously connected output from the decoder, so it can be reused for another decoder.

# 2.1.42.6.4 SubtitleDecoder::enableSubtitles Method

## C++

```
TMmRc enableSubtitles(int32_t localTimeOffset = 0, uint32_t flags = 0);
```

# Remarks

This is enableSubtitles, a member of class SubtitleDecoder.

# 2.1.42.6.5 SubtitleDecoder::getDataBuffer Method (OverlayElementDataBuffer &, uint32\_t, uint32\_t)

#### C++

```
virtual TMmRc getDataBuffer(OverlayElementDataBuffer & databuf, uint32_t minsize = 0,
uint32_t timeout = WaitInfinite);
```

#### Remarks

This is getDataBuffer, a member of class SubtitleDecoder.

# 2.1.42.6.6 SubtitleDecoder::getDataBuffer Method (OverlaySubtitleDataBuffer &, uint32\_t, uint32\_t)

### C++

```
virtual TMmRc getDataBuffer(OverlaySubtitleDataBuffer & databuf, uint32_t minsize = 0,
uint32_t timeout = WaitInfinite);
```

#### Remarks

This is getDataBuffer, a member of class SubtitleDecoder.

# 2.1.42.6.7 SubtitleDecoder::getOverlayElementRequest Method

### C++

```
TMmRc getOverlayElementRequest(OverlayElementDataBuffer & databuf, uint32_t timeout =
WaitInfinite);
```

## Remarks

This is getOverlayElementRequest, a member of class SubtitleDecoder.

# 2.1.42.6.8 SubtitleDecoder::sendOverlayElement Method

# C++

```
TMmRc sendOverlayElement(const char * elementName, const uint8_t * data, uint32_t dataSize,
uint32_t resourceId, uint16_t keyIndex = DataBuffer::NoDecryption, uint8_t * iv = 0,
uint8_t * cv = 0, uint32_t plainOffset = 0, uint32_t sourceLength = 0);
```

## Remarks

This is sendOverlayElement, a member of class SubtitleDecoder.

# 2.1.42.6.9 SubtitleDecoder::sendSubtitleFile Method

## C++

```
TMmRc sendSubtitleFile(const uint8_t * data, uint32_t dataSize, uint32_t * resourceId = 0,
int32_t timeOffset = 0, uint32_t startTime = 0, uint32_t endTime = UINT32_MAX);
```

## Romarks

This is sendSubtitleFile, a member of class SubtitleDecoder.

# 2.1.42.6.10 SubtitleDecoder::setOutputResolution Method

# C++

```
TMmRc setOutputResolution(uint32_t width, uint32_t height, uint32_t flags =
OverlayDecoder::Resolution_Absolute, uint32_t resourceId = 0);
```

## Remarks

This is setOutputResolution, a member of class SubtitleDecoder.

# 2.1.42.6.11 SubtitleDecoder::setZOffset Method

#### C++

```
TMmRc setZOffset(float offset);
```

### Remarks

This is setZOffset, a member of class SubtitleDecoder.

# 2.1.42.7 SubtitleDecoder Operators

# 2.1.42.7.1 SubtitleDecoder::= Operator

## C++

```
SubtitleDecoder& operator =(const SubtitleDecoder& other);
```

#### Remarks

This is =, a member of class SubtitleDecoder.

# 2.1.43 SystemStatus Class

# Inheritance Hierarchy

# C++

class SystemStatus;

## File

mvc2api\_device.h

# Remarks

This class holds the collected information about the current system status. It is the result of the MvcDevice::getSystemStatus() method.

## **Members**

## Methods

	Name	Description
<b>≡</b>	~SystemStatus	This is ~SystemStatus, a member of class SystemStatus.
<b>≡</b>	SystemStatus	Default constructor.
<b>=♦</b>	SystemStatus	Default constructor.
<b>=♦</b>	SystemStatus	Default constructor.

## **SystemStatus Enumerations**

	Name	Description
<b>.</b>	CPULOAD	CPU load enumeration used for getCPULoad() method.

# **SystemStatus Methods**

	Name	Description
<b>≡∳</b>	getBoardTemperature	This is getBoardTemperature, a member of class SystemStatus.
<b>≡</b>	getCPULoad	Gets the CPU load information.
<b>≡∳</b>	getDecoderTemperature	This is getDecoderTemperature, a member of class SystemStatus.
<b>≡</b>	getFanSpeed	This is getFanSpeed, a member of class SystemStatus.
•	getFreeMemory	Gets free memory information from the firmware.
•	getMaxMemory	Gets the memory information of the firmware.
<b>≡∳</b>	getOutputTemperature	This is getOutputTemperature, a member of class SystemStatus.
<b>≡∳</b>	getUsedMemory	Gets used memory information from the firmware.

# **SystemStatus Operators**

	Name	Description
<del>(/=</del>	=	Assignment operator.
( <u>/−</u> =+)	bool	This is bool, a member of class SystemStatus.

# **SystemStatus Enumerations**

	Name	Description
<b>a</b>	CPULOAD	CPU load enumeration used for getCPULoad() method.

# **SystemStatus Methods**

	Name	Description
<b>∉</b> ∳	getBoardTemperature	This is getBoardTemperature, a member of class SystemStatus.
<b>=♦</b>	getCPULoad	Gets the CPU load information.
<b>∉</b> ∳	getDecoderTemperature	This is getDecoderTemperature, a member of class SystemStatus.
<b>=♦</b>	getFanSpeed	This is getFanSpeed, a member of class SystemStatus.
<b>≡</b>	getFreeMemory	Gets free memory information from the firmware.
<b>≡</b>	getMaxMemory	Gets the memory information of the firmware.
<b>∉</b> ∳	getOutputTemperature	This is getOutputTemperature, a member of class SystemStatus.
<b>=</b>	getUsedMemory	Gets used memory information from the firmware.

# **SystemStatus Operators**

	Name	Description
=+)	=	Assignment operator.
<u>(/-</u> =+)	bool	This is bool, a member of class SystemStatus.

# 2.1.43.1 SystemStatus::~SystemStatus Destructor

# C++

~SystemStatus();

## Remarks

This is ~SystemStatus, a member of class SystemStatus.

# 2.1.43.2 SystemStatus::SystemStatus Constructor ()

Default constructor.

### C++

```
SystemStatus();
SystemStatus(const SystemStatus & other);
SystemStatus(MvcDevicePrivate * mvcdev, bool readTemperatures);
```

# Description

This constructor produces a empty SystemStatus object with any value set to zero.

# 2.1.43.3 SystemStatus Enumerations

# 2.1.43.3.1 mvc2::SystemStatus::CPULOAD Enumeration

## C++

```
enum CPULOAD {
  Load_100ms,
  Load_1s,
  Load_10s
};
```

#### File

mvc2api device.h

### **Members**

Members	Description
Load_100ms	CPU load calculated over a 100 ms period of time.
Load_1s	CPU load calculated over a 1 second period of time.
Load_10s	CPU load calculated over a 10 seconds period of time.

# Remarks

CPU load enumeration used for getCPULoad() method.

# 2.1.43.4 SystemStatus Methods

# 2.1.43.4.1 SystemStatus::getBoardTemperature Method

## $C^{++}$

```
float getBoardTemperature() const;
```

# Remarks

This is getBoardTemperature, a member of class SystemStatus.

# 2.1.43.4.2 SystemStatus::getCPULoad Method

Gets the CPU load information.

# C++

```
uint32_t getCPULoad(CPULOAD load) const;
```

## **Parameters**

Parameters	Description
CPULOAD load	period of time (see enumerations) of the CPU load
	calculation

#### Returns

CPU load in percent (value between 0 and 100, where 100 means maximal CPU load)

## **Description**

This method will return the CPU load of the firmware for a given time span.

# 2.1.43.4.3 SystemStatus::getDecoderTemperature Method

## C++

```
float getDecoderTemperature() const;
```

### Remarks

This is getDecoderTemperature, a member of class SystemStatus.

# 2.1.43.4.4 SystemStatus::getFanSpeed Method

## C++

```
uint32_t getFanSpeed() const;
```

## Remarks

This is getFanSpeed, a member of class SystemStatus.

# 2.1.43.4.5 SystemStatus::getFreeMemory Method

Gets free memory information from the firmware.

## C++

```
uint32_t getFreeMemory() const;
```

# Returns

Number of bytes free on the firmware.

## Description

This method returns the number of bytes used on the firmware.

# 2.1.43.4.6 SystemStatus::getMaxMemory Method

Gets the memory information of the firmware.

## C++

```
uint32_t getMaxMemory() const;
```

## Returns

Total number of memory bytes.

# Description

This method will return the total bytes of memory from the firmware.

# 2.1.43.4.7 SystemStatus::getOutputTemperature Method

#### C++

```
float getOutputTemperature() const;
```

### Remarks

This is getOutputTemperature, a member of class SystemStatus.

# 2.1.43.4.8 SystemStatus::getUsedMemory Method

Gets used memory information from the firmware.

### C++

```
uint32_t getUsedMemory() const;
```

#### Returns

Number of bytes used in the firmware.

# Description

This method returns the number of bytes used on the firmware.

# 2.1.43.5 SystemStatus Operators

# 2.1.43.5.1 SystemStatus::= Operator

Assignment operator.

# C++

```
SystemStatus & operator =(const SystemStatus & inbuf);
```

## **Parameters**

Parameters	Description
const SystemStatus & inbuf	source SystemStatus object

## Returns

Returns true if the new SystemStatus object is not empty.

## Description

The operator copies all data from a source object.

# 2.1.43.5.2 SystemStatus::bool Operator

## C++

```
operator bool() const;
```

# Remarks

This is bool, a member of class SystemStatus.

# 2.1.44 TimeCode Class

TimeCode value type.

# **Inheritance Hierarchy**

C++

class TimeCode;

File

mvc2api\_decoder.h

# Description

Objects of this call hold a time code and make manipulations very easy.

# Members

# Methods

	Name	Description
<b>=♦</b>	TimeCode	TimeCode default constructor.
<b>=</b> ♦	TimeCode	Copy constructor.
<b>=</b> ♦	TimeCode	Constructs a TimeCode with a given value.
<b>=♦</b>	TimeCode	Constructs a TimeCode object with a given value.

## **TimeCode Methods**

	Name	Description
<b>=♦</b>	getFrames	Gets frames info.
<b>=</b> ♦	getHours	Gets hours info.
<b>=♦</b>	getMinutes	Gets minutes info.
<b>≡</b>	getSeconds	Gets seconds info.
<b>=♦</b>	getTimeCodePacked	Gets the time code in packed format.
<b>=♦</b>	isDropFrame	Gets drop frame flag.

# **TimeCode Operators**

	Name	Description
<u>(/-</u> =+)	=	Assignment operator.

# **TimeCode Methods**

	Name	Description
<b>=</b> ♦	getFrames	Gets frames info.
<b>=♦</b>	getHours	Gets hours info.
<b>≡</b>	getMinutes	Gets minutes info.
<b>=</b>	getSeconds	Gets seconds info.
<b>=♦</b>	getTimeCodePacked	Gets the time code in packed format.
<b>≡∳</b>	isDropFrame	Gets drop frame flag.

# **TimeCode Operators**

	Name	Description
( <u>/-</u> =+)	=	Assignment operator.

# 2.1.44.1 TimeCode::TimeCode Constructor ()

TimeCode default constructor.

C++

TimeCode();

# Description

A object with time code 00:00:00.000 without drop frame will be created.

# 2.1.44.2 TimeCode::TimeCode Constructor (TimeCode &)

Copy constructor.

#### C++

TimeCode(const TimeCode & other);

## **Parameters**

Parameters	Description
const TimeCode & other	TimeCode object to copy the value from.

## Description

This constructor creates a new object with the value of a source object.

# 2.1.44.3 TimeCode::TimeCode Constructor (uint32\_t)

Constructs a TimeCode with a given value.

### C++

TimeCode(uint32\_t packedtc);

### **Parameters**

Parameters	Description
uint32_t packedtc	Time code represented in a 32 bit integer.

# Description

A new TimeCode object will be created with the value of packettc. See getTimeCodePacked().

# 2.1.44.4 TimeCode::TimeCode Constructor (uint8\_t, uint8\_t, uint8\_t, uint8\_t, bool)

Constructs a TimeCode object with a given value.

C++

TimeCode(uint8\_t hours, uint8\_t mins, uint8\_t secs, uint8\_t frames, bool dropframe);

## **Parameters**

Parameters	Description
uint8_t hours	hours (0-23)
uint8_t mins	minutes (0-59)
uint8_t secs	seconds (0-59)
uint8_t frames	frames (0-255)
bool dropframe	drop frame flag

# Description

A new TimeCode object will be created and initialized.

# 2.1.44.5 TimeCode Methods

# 2.1.44.5.1 TimeCode::getFrames Method

Gets frames info.

#### C++

```
uint8_t getFrames() const;
```

## Returns

8 bit frames value.

## Description

This method returns the frames value from the TimeCode.

# 2.1.44.5.2 TimeCode::getHours Method

Gets hours info.

### C++

```
uint8_t getHours() const;
```

## **Returns**

8 bit hours value.

## **Description**

This method returns the hours value from the TimeCode.

# 2.1.44.5.3 TimeCode::getMinutes Method

Gets minutes info.

## C++

```
uint8_t getMinutes() const;
```

## Returns

8 bit minutes value.

# **Description**

This method returns the minutes value from the TimeCode.

# 2.1.44.5.4 TimeCode::getSeconds Method

Gets seconds info.

# C++

```
uint8_t getSeconds() const;
```

## Returns

8 bit seconds value.

## Description

This method returns the seconds value from the TimeCode.

# 2.1.44.5.5 TimeCode::getTimeCodePacked Method

Gets the time code in packed format.

## C++

```
uint32_t getTimeCodePacked() const;
```

## **Returns**

A 32 bit value with the hold time code information.

## Description

This method returns the time code in a 32 bit representation.

bits	description
7-0	frames
15-8	seconds
23-16	minutes
30-24	hours
31	drop frame flag

# 2.1.44.5.6 TimeCode::isDropFrame Method

Gets drop frame flag.

### C++

```
bool isDropFrame() const;
```

## Returns

True if drop frame flag is set, otherwise false.

# Description

This method is used to check if drop frame flag is set.

# 2.1.44.6 TimeCode Operators

# 2.1.44.6.1 TimeCode::= Operator

Assignment operator.

# C++

```
TimeCode & operator =(const TimeCode & other);
```

## **Parameters**

Parameters	Description	
const TimeCode & other	source TimeCode object.	

## Returns

A reference to the destination TimeCode object.

## Description

This operator copies one TimeCode over another.

# 2.1.45 UuidValue Class

Helper class for UUIDs.

# **Inheritance Hierarchy**

### C++

# class UuidValue;

# File

mvc2api\_securitymanager.h

# Description

This class provides an easy interface to work with 128 bit UUIDs.

## **Members**

# Methods

	Name	Description
<b>=</b> ♦	UuidValue	This is UuidValue, a member of class UuidValue.
<b>=</b> ♦	UuidValue	This is UuidValue, a member of class UuidValue.
<b>≡♦</b>	UuidValue	This is UuidValue, a member of class UuidValue.
<b>≡♦</b>	UuidValue	This is UuidValue, a member of class UuidValue.
<b>=♦</b>	UuidValue	This is UuidValue, a member of class UuidValue.

## **UuidValue Data Members**

	Name	Description
<b>₽</b> ₽	m_key	This is m_key, a member of class UuidValue.

# **UuidValue Methods**

	Name	Description
<b>≡♦</b>	toArray	This is toArray, a member of class UuidValue.
<b>=</b> ♦	toString	This is toString, a member of class UuidValue.

# **UuidValue Operators**

	Name	Description
=+)	!=	This is !=, a member of class UuidValue.
=+)		This is [], a member of class UuidValue.
=+)		This is [], a member of class UuidValue.
=+)	=	This is =, a member of class UuidValue.
=+)	==	This is ==, a member of class UuidValue.
=+)	bool	This is bool, a member of class UuidValue.

# **UuidValue Data Members**

	Name	Description
<b>₽</b> *	m_key	This is m_key, a member of class UuidValue.

# **UuidValue Methods**

	Name	Description
<b>≡♦</b>	toArray	This is toArray, a member of class UuidValue.
<b>=</b> ♦	toString	This is toString, a member of class UuidValue.

# **UuidValue Operators**

	Name	Description
<u>(/-</u> =+)	!=	This is !=, a member of class UuidValue.
=+)	[]	This is [], a member of class UuidValue.
=+)	[]	This is [], a member of class UuidValue.
=+)	=	This is =, a member of class UuidValue.
<u>(&gt;</u> =+)	==	This is ==, a member of class UuidValue.
( <u>/-</u> =+)	bool	This is bool, a member of class UuidValue.

# 2.1.45.1 UuidValue::UuidValue Constructor (UuidValue &)

#### C++

```
UuidValue(const UuidValue & inkey);
```

#### Remarks

This is UuidValue, a member of class UuidValue.

# 2.1.45.2 UuidValue::UuidValue Constructor (char \*)

#### C++

```
UuidValue(const char * str);
```

#### Remarks

This is UuidValue, a member of class UuidValue.

# 2.1.45.3 UuidValue::UuidValue Constructor (uint8\_t)

# C++

```
UuidValue(uint8_t fill = 0);
```

## Remarks

This is UuidValue, a member of class UuidValue.

# 2.1.45.4 UuidValue::UuidValue Constructor (uint8\_t, uint8\_t, uint8

## C++

```
UuidValue(uint8_t k0, uint8_t k1, uint8_t k2, uint8_t k3, uint8_t k4, uint8_t k5, uint8_t
k6, uint8_t k7, uint8_t k8, uint8_t k9, uint8_t k10, uint8_t k11, uint8_t k12, uint8_t k13,
uint8_t k14, uint8_t k15);
```

# Remarks

This is UuidValue, a member of class UuidValue.

# 2.1.45.5 UuidValue::UuidValue Constructor (uint8\_t \*)

#### C++

```
UuidValue(const uint8_t * key);
```

### Remarks

This is UuidValue, a member of class UuidValue.

# 2.1.45.6 UuidValue Data Members

# 2.1.45.6.1 UuidValue::m\_key Data Member

## C++

```
uint8_t m_key[16];
```

## Remarks

This is m\_key, a member of class UuidValue.

# 2.1.45.7 UuidValue Methods

# 2.1.45.7.1 UuidValue::toArray Method

## C++

```
void toArray(uint8_t * array) const;
```

# Remarks

This is to Array, a member of class Uuid Value.

# 2.1.45.7.2 UuidValue::toString Method

## C++

```
char * toString(char * stringBuffer) const;
```

## Remarks

This is toString, a member of class UuidValue.

# 2.1.45.8 UuidValue Operators

# 2.1.45.8.1 UuidValue::!= Operator

## C++

```
bool operator !=(const UuidValue & inkey) const;
```

# Remarks

This is !=, a member of class UuidValue.

# 2.1.45.8.2 UuidValue::[] Operator (uint8\_t)

#### C++

```
uint8_t & operator [](uint8_t index);
```

### Remarks

This is [], a member of class UuidValue.

# 2.1.45.8.3 UuidValue::[] Operator (uint8\_t)

### C++

```
uint8_t operator [](uint8_t index) const;
```

## Remarks

This is [], a member of class UuidValue.

# 2.1.45.8.4 UuidValue::= Operator

### C++

```
UuidValue & operator =(const UuidValue & inkey);
```

## Remarks

This is =, a member of class UuidValue.

# 2.1.45.8.5 UuidValue::== Operator

#### C++

```
bool operator ==(const UuidValue & inkey) const;
```

## Remarks

This is ==, a member of class UuidValue.

# 2.1.45.8.6 UuidValue::bool Operator

# C++

```
operator bool() const;
```

# Remarks

This is bool, a member of class UuidValue.

# 2.1.46 TamperStatus Class



# **Inheritance Hierarchy**

## C++

```
class TamperStatus;
```

## File

mvc2api\_security.h

## Remarks

This is class mvc2::TamperStatus.

## **Members**

## Methods

	Name	Description
<b>=</b> ♦	TamperStatus	This is TamperStatus, a member of class TamperStatus.

# **TamperStatus Enumerations**

	Name	Description
a <sup>p</sup>	TAMPER_EVENTS	This is record mvc2::TamperStatus::TAMPER_EVENTS.
<b>a</b>	TAMPER_SWITCHES	This is record mvc2::TamperStatus::TAMPER_SWITCHES.

# **TamperStatus Methods**

	Name	Description
<b>≡♦</b>	getSwitchStatus	This is getSwitchStatus, a member of class TamperStatus.
<b>≡♦</b>	getTamperEvent	This is getTamperEvent, a member of class TamperStatus.

# **TamperStatus Friends**

Name	Description
class SecurityAccess	This is friend friend class SecurityAccess.

# **TamperStatus Enumerations**

	Name	Description
<b>a</b>	TAMPER_EVENTS	This is record mvc2::TamperStatus::TAMPER_EVENTS.
	TAMPER_SWITCHES	This is record mvc2::TamperStatus::TAMPER_SWITCHES.

# **TamperStatus Friends**

Name	Description
class SecurityAccess	This is friend friend class SecurityAccess.

# **TamperStatus Methods**

	Name	Description
<b>≡♦</b>	getSwitchStatus	This is getSwitchStatus, a member of class TamperStatus.
<b>≡♦</b>	getTamperEvent	This is getTamperEvent, a member of class TamperStatus.

# 2.1.46.1 TamperStatus::TamperStatus Constructor

C++

TamperStatus();

## Remarks

This is TamperStatus, a member of class TamperStatus.

# 2.1.46.2 TamperStatus Enumerations

# 2.1.46.2.1 mvc2::TamperStatus::TAMPER\_EVENTS Enumeration

## C++

```
enum TAMPER_EVENTS {
  EVENT_DOOR_OPEN_1 = 0,
  EVENT_DOOR_OPEN_2 = 1,
  EVENT_DOOR_CLOSE_1 = 2,
  EVENT_DOOR_CLOSE_2 = 3,
```

```
EVENT_DIVORCE_1 = 4,
EVENT_DIVORCE_2 = 5,
EVENT_TAMPER_1 = 6,
EVENT_TAMPER_2 = 7,
EVENT_POWER_LOSS = 8,
EVENT_POWER_OV = 9,
EVENT_BAT_REMOVE = 10,
EVENT_BAT_NEW = 11
};

File
mvc2api_security.h
```

## Remarks

This is record mvc2::TamperStatus::TAMPER\_EVENTS.

# 2.1.46.2.2 mvc2::TamperStatus::TAMPER\_SWITCHES Enumeration

### C++

```
enum TAMPER_SWITCHES {
    SWITCH_TAMPER_1 = (1<<0),
    SWITCH_TAMPER_2 = (1<<1),
    SWITCH_MARRIAGE_1 = (1<<2),
    SWITCH_MARRIAGE_2 = (1<<3),
    SWITCH_DOOR_1 = (1<<4),
    SWITCH_DOOR_2 = (1<<5)
};</pre>
File
```

mvc2api\_security.h

## Remarks

This is record mvc2::TamperStatus::TAMPER\_SWITCHES.

# 2.1.46.3 TamperStatus Methods

# 2.1.46.3.1 TamperStatus::getSwitchStatus Method

# C++

```
uint32_t getSwitchStatus() const;
```

## Remarks

This is getSwitchStatus, a member of class TamperStatus.

# 2.1.46.3.2 TamperStatus::getTamperEvent Method

## C++

```
bool getTamperEvent(uint32_t index, uint32_t * tamperEvent, time_t * tamperTimer) const;
```

## Remarks

This is getTamperEvent, a member of class TamperStatus.

# 2.1.46.4 TamperStatus Friends

# 2.1.46.4.1 friend class SecurityAccess Friend

#### C++

friend class SecurityAccess;

## Remarks

This is friend friend class SecurityAccess.

# 2.1.47 VersionValue Class

# **Inheritance Hierarchy**

## C++

class VersionValue;

## File

mvc2api\_device.h

# Remarks

Class to hold version numbers

## **Members**

## Methods

	Name	Description
<b>≡</b>	VersionValue	Copy constructor.
<b>≡</b>	VersionValue	Default constructor
<b>=♦</b>	VersionValue	Constructor with version number as bytes

# **VersionValue Methods**

	Name	Description
<b>=♦</b>	getBuildRevision	Return build revision.
<b>=♦</b>	getBuildVersion	Return build version.
<b>=♦</b>	getRevision	Return revision
<b>=♦</b>	getVersion	Return version

# **VersionValue Operators**

	Name	Description
=+)	=	Default assignment operator
=+0	bool	Bool operator.

# **VersionValue Methods**

	Name	Description
<b>≡</b> ♦	getBuildRevision	Return build revision.
<b>≡♦</b>	getBuildVersion	Return build version.
<b>≡</b> ♦	getRevision	Return revision
<b>≡♦</b>	getVersion	Return version

# **VersionValue Operators**

	Name	Description
<del>V=</del> =+)	=	Default assignment operator
<del>(/_</del> =+)	bool	Bool operator.

# 2.1.47.1 VersionValue::VersionValue Constructor (VersionValue &)

Copy constructor.

#### C++

VersionValue(const VersionValue & inver);

## **Parameters**

Parameters	Description
const VersionValue & inver	source VersionValue object

## Description

This constructor creates a new VersionValue object with a value of the source object.

# 2.1.47.2 VersionValue::VersionValue Constructor (uint32\_t)

Default constructor

#### C++

VersionValue(uint32\_t version = 0);

#### **Parameters**

Parameters	Description
uint32_t version = 0	version number, big endian (upper byte is version, lower byte is build revision)

# Description

Constructor which creates a VersionValue object with the version number given as 32bit value.

# 2.1.47.3 VersionValue::VersionValue Constructor (uint32\_t, uint32\_t, uint32\_t, uint32\_t)

Constructor with version number as bytes

C++

VersionValue(uint32\_t ver, uint32\_t rev, uint32\_t buildver, uint32\_t buildrev);

## **Parameters**

Parameters	Description
uint32_t ver	version number
uint32_t rev	version revision
uint32_t buildver	build number
uint32_t buildrev	build revision

## Description

Create a VersionValue object based on given version information.

# 2.1.47.4 VersionValue Methods

# 2.1.47.4.1 VersionValue::getBuildRevision Method

Return build revision.

### C++

```
uint32_t getBuildRevision() const;
```

## Returns

Build revision.

## **Description**

Gets build revision number.

# 2.1.47.4.2 VersionValue::getBuildVersion Method

Return build version.

## C++

```
uint32_t getBuildVersion() const;
```

### Returns

Build number.

# Description

Gets build number.

# 2.1.47.4.3 VersionValue::getRevision Method

Return revision

## C++

```
uint32_t getRevision() const;
```

# Returns

Revision number.

## Description

Gets revision number.

# 2.1.47.4.4 VersionValue::getVersion Method

Return version

## C++

```
uint32_t getVersion() const;
```

# Returns

Version number.

# Description

Gets version number.

# 2.1.47.5 VersionValue Operators

# 2.1.47.5.1 VersionValue::= Operator

Default assignment operator

## C++

VersionValue & operator =(const VersionValue & inver);

### **Parameters**

Parameters		Description
const VersionValue & inver	0,	input VersionValue object

## **Returns**

True of the new VersionValue object is not empty.

# Description

Copies all information from a source VersionValue object to a new one.

# 2.1.47.5.2 VersionValue::bool Operator

Bool operator.

### C++

operator bool() const;

## Description

This operator is used to check if a VersionValue is not zero.

# 2.1.48 VideoDataBuffer Class

# Inheritance Hierarchy

# C++

class VideoDataBuffer : public DataBuffer;

## File

mvc2api\_decoder.h

# Remarks

This is class mvc2::VideoDataBuffer.

## **Members**

## Methods

	Name	Description
<b>=</b> ♦ <b>W</b>	~DataBuffer	Destructor
<b>≡⋄</b>	DataBuffer	This is DataBuffer, a member of class DataBuffer.
<b>≡⋄</b>	DataBuffer	This is DataBuffer, a member of class DataBuffer.
<b>≡∳</b> ?	DataBuffer	This is DataBuffer, a member of class DataBuffer.

## VideoDataBuffer Class

	Name	Description
<b>=</b> ♦ <b>W</b>	~VideoDataBuffer	This is ~VideoDataBuffer, a member of class VideoDataBuffer.
<b>≟∳</b>	VideoDataBuffer	This is VideoDataBuffer, a member of class VideoDataBuffer.

<b>=\$9</b>	VideoDataBuffer	This is VideoDataBuffer, a member of class VideoDataBuffer.
-------------	-----------------	---

# **DataBuffer Methods**

	Name	Description
<b>≡</b>	сору	Copy a data block to DataBuffer.
<b>≡</b>	getBufferAddress	Gets the buffer memory address.
<b>=♦</b>	getFreeSize	Get the available buffer size.
<b>≡</b>	getInSize	Get the transfer size.
<b>≡</b>	getTimeStamp	Get current timestamp of the DataBuffer.
<b>≡</b>	send	Send buffer to MVC card.
<b>≡</b>	send	Send DataBuffer to the MVC card.
<b>≡</b>	setDecryptionSize	Set decryption size.
<b>≡</b>	setInSize	Sets the transfer size.
<b>≡</b>	setKeyId	Set decryption information.
<b>≡</b>	setKeyIndex	Set decryption information.
<b>=♦</b>	setMicValue	Set the MIC value extracted from the MXF file.
<b>≡</b>	setTimeStamp	Sets the presentation time stamp.
<b>≡</b>	setUserData	Sets user data.
<b>≡</b>	wait	Wait for transfer completion

# VideoDataBuffer Class

	Name	Description
<b>≡♦</b>	setTimeCode	This is setTimeCode, a member of class VideoDataBuffer.

# **DataBuffer Operators**

	Name	Description
<u>(/-</u> =+)	=	Assignment operator.
<u>(&gt;−</u> <del>=+)</del>	bool	This is bool, a member of class DataBuffer.

# VideoDataBuffer Friends

# VideoDataBuffer Class

Name	Description
class VideoDecoder	This is friend friend class VideoDecoder.

# VideoDataBuffer Friends

# VideoDataBuffer Class

Name	Description
class VideoDecoder	This is friend friend class VideoDecoder.

# **DataBuffer Methods**

	Name	Description
<b>=</b> ♦	сору	Copy a data block to DataBuffer.
<b>=</b> ♦	getBufferAddress	Gets the buffer memory address.
<b>=</b> ♦	getFreeSize	Get the available buffer size.
<b>≡♦</b>	getInSize	Get the transfer size.
<b>≡♦</b>	getTimeStamp	Get current timestamp of the DataBuffer.
<b>≡♦</b>	send	Send buffer to MVC card.
<b>≡♦</b>	send	Send DataBuffer to the MVC card.
<b>≡</b>	setDecryptionSize	Set decryption size.
<b>≡♦</b>	setInSize	Sets the transfer size.
<b>≡</b>	setKeyId	Set decryption information.

<b>≡♦</b>	setKeyIndex	Set decryption information.
<b>=</b> ♦	setMicValue	Set the MIC value extracted from the MXF file.
<b>=</b> ♦	setTimeStamp	Sets the presentation time stamp.
<b>=♦</b>	setUserData	Sets user data.
<b>=</b> ♦	wait	Wait for transfer completion

## VideoDataBuffer Class

	Name	Description
<b>≡♦</b>	setTimeCode	This is setTimeCode, a member of class VideoDataBuffer.

# **DataBuffer Operators**

	Name	Description
=+)	=	Assignment operator.
( <u>/-</u> =+)	bool	This is bool, a member of class DataBuffer.

# 2.1.48.1 VideoDataBuffer::~VideoDataBuffer Destructor

#### C++

virtual ~VideoDataBuffer();

#### Remarks

This is ~VideoDataBuffer, a member of class VideoDataBuffer.

# 2.1.48.2 VideoDataBuffer::VideoDataBuffer Constructor ()

## C++

VideoDataBuffer();

## Remarks

This is VideoDataBuffer, a member of class VideoDataBuffer.

# 2.1.48.3 VideoDataBuffer::VideoDataBuffer Constructor (MvcDecoderPrivate \*, MemoryBuffer \*)

C++

```
VideoDataBuffer(MvcDecoderPrivate * mvcdec, MemoryBuffer * mem);
```

## Remarks

This is VideoDataBuffer, a member of class VideoDataBuffer.

# 2.1.48.4 VideoDataBuffer Methods

# 2.1.48.4.1 VideoDataBuffer::setTimeCode Method



# C++

```
void setTimeCode(const TimeCode & tc);
```

## Remarks

This is setTimeCode, a member of class VideoDataBuffer.

# 2.1.48.5 VideoDataBuffer Friends

# 2.1.48.5.1 friend class VideoDecoder Friend

C++

friend class VideoDecoder;

## Remarks

This is friend friend class VideoDecoder.

# 2.1.49 VideoDecoder Class

# Inheritance Hierarchy

C++

class VideoDecoder : public MvcDecoder;

File

mvc2api\_decoder.h

# Remarks

This is class mvc2::VideoDecoder.

# **Members**

# Methods

	Name	Description
<b>=♦</b> ₩	~MvcDecoder	This is ~MvcDecoder, a member of class MvcDecoder.
<b>=</b> ♦	MvcDecoder	This is MvcDecoder, a member of class MvcDecoder.
<b>=♦</b>	MvcDecoder	This is MvcDecoder, a member of class MvcDecoder.
<b>=♦</b> •	MvcDecoder	This is MvcDecoder, a member of class MvcDecoder.

# **VideoDecoder Class**

	Name	Description
<b>≟</b> ∳9	VideoDecoder	This is VideoDecoder, a member of class VideoDecoder.
<b>=</b> ♦•	VideoDecoder	This is VideoDecoder, a member of class VideoDecoder.

## **MvcDecoder Enumerations**

	Name	Description
<b>a</b>	DecoderType	This is record mvc2::MvcDecoder::DecoderType.
<b>a</b>	TIMEOUT	This is record mvc2::MvcDecoder::TIMEOUT.

# **MvcDecoder Methods**

	Name	Description
<b>≡</b> ♦	flush	This is flush, a member of class MvcDecoder.
<b>=</b> ♦	getBufferStatus	This is getBufferStatus, a member of class MvcDecoder.
<b>≡♦</b>	getChannel	Gets the used channel number.
<b>=</b> ♦ ₩	getDataBuffer	Return a buffer for data transfer.
<b>=</b> ♦	getDecoderErrors	This is getDecoderErrors, a member of class MvcDecoder.
<b>≡♦</b>	getDecoderStatus	This is getDecoderStatus, a member of class MvcDecoder.

<b>≡∳</b>	getDecoderType	This is getDecoderType, a member of class MvcDecoder.
<b>≡\$</b> 9	getPrivate	This is getPrivate, a member of class MvcDecoder.
<b>∉</b> ∳	setBackwardPlayback	This is setBackwardPlayback, a member of class MvcDecoder.
<b>=♦</b>	setCplUid	This is setCplUid, a member of class MvcDecoder.
<b>=♦</b>	setEndOfStream	Go to end of stream state.
<b>=♦</b>	setSecurityManager	This is setSecurityManager, a member of class MvcDecoder.
<b>=♦</b>	setStartDelay	Setup start delay for the decoder.
<b>=♦</b>	waitDecoderStatus	This is waitDecoderStatus, a member of class MvcDecoder.
<b>=♦</b>	waitForTimeStamp	This is waitForTimeStamp, a member of class MvcDecoder.
<b>∉∳</b>	waitForTransferFinish	This is waitForTransferFinish, a member of class MvcDecoder.
<b>=</b>	waitForUserData	This is waitForUserData, a member of class MvcDecoder.

#### VideoDecoder Class

	Name	Description
<b>≡</b>	calcDolby3DMatrix	This is calcDolby3DMatrix, a member of class VideoDecoder.
<b>=♦</b>	connectOutput	Connects an output object to the decoder.
<b>=</b>	disconnectOutput	Disconnects the output.
<b>=♦</b>	getDecoderErrors	This is getDecoderErrors, a member of class VideoDecoder.
<b>=♦</b>	getDecoderStatus	This is getDecoderStatus, a member of class VideoDecoder.
<b>∉</b> ∳	getVideoDataBuffer	This is getVideoDataBuffer, a member of class VideoDecoder.
<b>=♦</b>	setCloneVideo	This is setCloneVideo, a member of class VideoDecoder.
<b>∉</b> ∳	setDolby3DChroma	This is setDolby3DChroma, a member of class VideoDecoder.
<b>≡</b>	setDolby3DMatrix	This is setDolby3DMatrix, a member of class VideoDecoder.
<b>≡∳</b>	waitDecoderStatus	This is waitDecoderStatus, a member of class VideoDecoder.

#### **MvcDecoder Operators**

	Name	Description
=+)	=	This is =, a member of class MvcDecoder.
( <u>/-</u> =+)	bool	Returns if the object is valid.

### **MvcDecoder Enumerations**

	Name	Description
<b>₽</b>	DecoderType	This is record mvc2::MvcDecoder::DecoderType.
<b>a</b>	TIMEOUT	This is record mvc2::MvcDecoder::TIMEOUT.

#### **MvcDecoder Methods**

	Name	Description
<b>≡♦</b>	flush	This is flush, a member of class MvcDecoder.
<b>≡</b>	getBufferStatus	This is getBufferStatus, a member of class MvcDecoder.
<b>≡</b>	getChannel	Gets the used channel number.
<b>■</b>	getDataBuffer	Return a buffer for data transfer.
<b>≡</b>	getDecoderErrors	This is getDecoderErrors, a member of class MvcDecoder.
<b>≡</b>	getDecoderStatus	This is getDecoderStatus, a member of class MvcDecoder.
<b>≡</b>	getDecoderType	This is getDecoderType, a member of class MvcDecoder.
<b>≡∳</b> ?	getPrivate	This is getPrivate, a member of class MvcDecoder.
<b>≡♦</b>	setBackwardPlayback	This is setBackwardPlayback, a member of class MvcDecoder.
<b>≡</b>	setCplUid	This is setCplUid, a member of class MvcDecoder.

<b>=♦</b>	setEndOfStream	Go to end of stream state.
<b>=♦</b>	setSecurityManager	This is setSecurityManager, a member of class MvcDecoder.
<b>=♦</b>	setStartDelay	Setup start delay for the decoder.
<b>=</b> ♦	waitDecoderStatus	This is waitDecoderStatus, a member of class MvcDecoder.
<b>=</b> ♦	waitForTimeStamp	This is waitForTimeStamp, a member of class MvcDecoder.
<b>=</b> ♦	waitForTransferFinish	This is waitForTransferFinish, a member of class MvcDecoder.
<b>=♦</b>	waitForUserData	This is waitForUserData, a member of class MvcDecoder.

#### **VideoDecoder Class**

	Name	Description
<b>≡</b>	calcDolby3DMatrix	This is calcDolby3DMatrix, a member of class VideoDecoder.
<b>=</b> ♦	connectOutput	Connects an output object to the decoder.
<b>≡∳</b>	disconnectOutput	Disconnects the output.
<b>≡∳</b>	getDecoderErrors	This is getDecoderErrors, a member of class VideoDecoder.
<b>≡∳</b>	getDecoderStatus	This is getDecoderStatus, a member of class VideoDecoder.
<b>=♦</b>	getVideoDataBuffer	This is getVideoDataBuffer, a member of class VideoDecoder.
<b>=</b> ♦	setCloneVideo	This is setCloneVideo, a member of class VideoDecoder.
<b>=</b> ♦	setDolby3DChroma	This is setDolby3DChroma, a member of class VideoDecoder.
<b>=♦</b>	setDolby3DMatrix	This is setDolby3DMatrix, a member of class VideoDecoder.
<b>=♦</b>	waitDecoderStatus	This is waitDecoderStatus, a member of class VideoDecoder.

#### **MvcDecoder Operators**

	Name	Description
( <u>/-</u> =+)	=	This is =, a member of class MvcDecoder.
( <u>/-</u> =+)	bool	Returns if the object is valid.

# 2.1.49.1 VideoDecoder::VideoDecoder Constructor ()

#### C++

VideoDecoder();

#### Remarks

This is VideoDecoder, a member of class VideoDecoder.

# 2.1.49.2 VideoDecoder::VideoDecoder Constructor (MvcDecoderPrivate \*)

#### C++

VideoDecoder(MvcDecoderPrivate \* mvcdec);

#### Remarks

This is VideoDecoder, a member of class VideoDecoder.

## 2.1.49.3 VideoDecoder Methods

# 2.1.49.3.1 VideoDecoder::calcDolby3DMatrix Method



#### C++

static TMmRc calcDolby3DMatrix(const Dolby3Dchromaticity & dolby3d, float \* matrix);

#### Remarks

This is calcDolby3DMatrix, a member of class VideoDecoder.

# 2.1.49.3.2 VideoDecoder::connectOutput Method

Connects an output object to the decoder.



#### C++

TMmRc connectOutput(const VideoOutput& VideoOut);

#### **Parameters**

Parameters	Description
const VideoOutput& VideoOut	dedicated VideoOutput object.

#### Returns

MMRC\_Ok or an error if the connection fails.

#### **Return Values**

Return Values	Description
MMRC_Ok	Output successfully connected
MMRC_MVC2_ConnectionError	Output could not be connected, it may not be compatible

#### Description

This method is used to assign an output to the decoder which is used to display the decoder frames.

# 2.1.49.3.3 VideoDecoder::disconnectOutput Method

Disconnects the output.

#### C++

TMmRc disconnectOutput(const VideoOutput& VideoOut);

#### **Parameters**

Parameters	Description
const VideoOutput& VideoOut	previously connected output

#### Returns

MMRC\_Ok if the disconnection was successful, else an error.

#### **Return Values**

Return Values	Description
MMRC_Ok	successful disconnections
MMRC_MVC2_ConnectionError	object is not connected to the decoder

#### Description

Disconnects a previously connected output from the decoder, so it can be reused for another decoder.

## 2.1.49.3.4 VideoDecoder::getDecoderErrors Method

#### C++

TMmRc getDecoderErrors(VideoDecoderErrors & errors) const;

#### Remarks

This is getDecoderErrors, a member of class VideoDecoder.

# 2.1.49.3.5 VideoDecoder::getDecoderStatus Method

#### C++

TMmRc getDecoderStatus(VideoDecoderStatus & status) const;

#### Remarks

This is getDecoderStatus, a member of class VideoDecoder.

### 2.1.49.3.6 VideoDecoder::getVideoDataBuffer Method

#### C++

TMmRc getVideoDataBuffer(VideoDataBuffer & databuf, uint32\_t minsize = 0, uint32\_t timeout = WaitInfinite);

#### Remarks

This is getVideoDataBuffer, a member of class VideoDecoder.

## 2.1.49.3.7 VideoDecoder: setCloneVideo Method

#### C++

TMmRc setCloneVideo(bool clone);

#### Remarks

This is setCloneVideo, a member of class VideoDecoder.

# 2.1.49.3.8 VideoDecoder::setDolby3DChroma Method



#### C++

TMmRc setDolby3DChroma(const Dolby3Dchromaticity & dolby3d);

#### Remarks

This is setDolby3DChroma, a member of class VideoDecoder.

# 2.1.49.3.9 VideoDecoder::setDolby3DMatrix Method



#### C++

TMmRc setDolby3DMatrix(const float \* matrix);

#### Remarks

This is setDolby3DMatrix, a member of class VideoDecoder.

## 2.1.49.3.10 VideoDecoder::waitDecoderStatus Method

#### C++

TMmRc waitDecoderStatus(VideoDecoderStatus & status) const;

#### Remarks

This is waitDecoderStatus, a member of class VideoDecoder.

# 2.1.50 VideoDecoderErrors Class

#### **Inheritance Hierarchy**

#### C++

class VideoDecoderErrors : public DecoderErrors;

#### File

mvc2api\_decoder.h

#### Remarks

These objects collect a couple of error counters of a specific decoder.

#### **Members**

#### Methods

	Name	Description
<b>■♦ ₩</b>	~DecoderErrors	This is ~DecoderErrors, a member of class DecoderErrors.
<b>=♦</b>	DecoderErrors	This is DecoderErrors, a member of class DecoderErrors.
<b>=♦</b>	DecoderErrors	This is DecoderErrors, a member of class DecoderErrors.
<b>≡♦</b>	DecoderErrors	This is DecoderErrors, a member of class DecoderErrors.
<b>=♦</b>	DecoderErrors	This is DecoderErrors, a member of class DecoderErrors.

#### VideoDecoderErrors Class

	Name	Description
<b>=</b> ♦ ₩	~VideoDecoderErrors	This is ~VideoDecoderErrors, a member of class VideoDecoderErrors.
<b>≡</b>	VideoDecoderErrors	This is VideoDecoderErrors, a member of class VideoDecoderErrors.
<b>=♦</b>	VideoDecoderErrors	This is VideoDecoderErrors, a member of class VideoDecoderErrors.
<b>∉</b> ∳	VideoDecoderErrors	This is VideoDecoderErrors, a member of class VideoDecoderErrors.
<b>∉</b> ∳	VideoDecoderErrors	This is VideoDecoderErrors, a member of class VideoDecoderErrors.

#### **DecoderErrors Methods**

	Name	Description
<b>≡♦</b>	getReturnCode	This is getReturnCode, a member of class DecoderErrors.

#### VideoDecoderErrors Class

	Name	Description
<b>≡</b>	getInputUnderruns	This is getInputUnderruns, a member of class VideoDecoderErrors.

<b>≡</b> ∳	getPartialFrames	This is getPartialFrames, a member of class VideoDecoderErrors.
<b>≡</b> •	getPlayedFramesError	This is getPlayedFramesError, a member of class VideoDecoderErrors.
<b>≡</b> •	getPlayedFramesOk	This is getPlayedFramesOk, a member of class VideoDecoderErrors.
<b>≡∳</b>	getRepeatedFramesAVSync	This is getRepeatedFramesAVSync, a member of class VideoDecoderErrors.
<b>≡</b> •	getRepeatedFramesUnderrun	This is getRepeatedFramesUnderrun, a member of class VideoDecoderErrors.
<b>≡</b> •	getSkippedFramesAVSync	This is getSkippedFramesAVSync, a member of class VideoDecoderErrors.
<b>≡</b>	getSkippedFramesError	This is getSkippedFramesError, a member of class VideoDecoderErrors.

#### **DecoderErrors Operators**

	Name	Description
( <u>/-</u> =+)	=	This is =, a member of class DecoderErrors.
<u>(/−</u> =+)	bool	Returns if the object is valid.

#### VideoDecoderErrors Class

	Name	Description
( <u>/-</u> =+)	=	This is =, a member of class VideoDecoderErrors.

#### **DecoderErrors Methods**

	Name	Description
<b>≡♦</b>	getReturnCode	This is getReturnCode, a member of class DecoderErrors.

#### VideoDecoderErrors Class

	Name	Description
<b>≡∳</b>	getInputUnderruns	This is getInputUnderruns, a member of class VideoDecoderErrors.
<b>≡∳</b>	getPartialFrames	This is getPartialFrames, a member of class VideoDecoderErrors.
<b>≡∳</b>	getPlayedFramesError	This is getPlayedFramesError, a member of class VideoDecoderErrors.
<b>≡∳</b>	getPlayedFramesOk	This is getPlayedFramesOk, a member of class VideoDecoderErrors.
<b>∉</b> ∳	getRepeatedFramesAVSync	This is getRepeatedFramesAVSync, a member of class VideoDecoderErrors.
<b>≟</b> ∳	getRepeatedFramesUnderrun	This is getRepeatedFramesUnderrun, a member of class VideoDecoderErrors.
<b>≟</b> ∳	getSkippedFramesAVSync	This is getSkippedFramesAVSync, a member of class VideoDecoderErrors.
<b>≡♦</b>	getSkippedFramesError	This is getSkippedFramesError, a member of class VideoDecoderErrors.

#### **DecoderErrors Operators**

	Name	Description
<u>(/−</u> =+)	=	This is =, a member of class DecoderErrors.
(/ <del>_</del> =+)	bool	Returns if the object is valid.

#### VideoDecoderErrors Class

	Name	Description
=+)	=	This is =, a member of class VideoDecoderErrors.

### 2.1.50.1 VideoDecoderErrors::~VideoDecoderErrors Destructor

#### C++

```
virtual ~VideoDecoderErrors();
```

#### Remarks

This is ~VideoDecoderErrors, a member of class VideoDecoderErrors.

## 2.1.50.2 VideoDecoderErrors::VideoDecoderErrors Constructor ()

#### C++

VideoDecoderErrors();

#### Remarks

This is VideoDecoderErrors, a member of class VideoDecoderErrors.

# 2.1.50.3 VideoDecoderErrors::VideoDecoderErrors Constructor (MvcDecoderPrivate \*)

#### C++

VideoDecoderErrors(MvcDecoderPrivate \* mvcdec);

#### Remarks

This is VideoDecoderErrors, a member of class VideoDecoderErrors.

# 2.1.50.4 VideoDecoderErrors::VideoDecoderErrors Constructor (MvcDevice &, uint32 t)

C++

VideoDecoderErrors(const MvcDevice & mvcdev, uint32\_t channel);

#### Remarks

This is VideoDecoderErrors, a member of class VideoDecoderErrors.

# 2.1.50.5 VideoDecoderErrors::VideoDecoderErrors Constructor (VideoDecoderErrors &)

#### C++

VideoDecoderErrors(const VideoDecoderErrors & other);

#### Remarks

This is VideoDecoderErrors, a member of class VideoDecoderErrors.

## 2.1.50.6 VideoDecoderErrors Methods

## 2.1.50.6.1 VideoDecoderErrors::getInputUnderruns Method

#### C++

```
uint32_t getInputUnderruns() const;
```

#### Remarks

This is getInputUnderruns, a member of class VideoDecoderErrors.

### 2.1.50.6.2 VideoDecoderErrors::getPartialFrames Method

#### C++

```
uint32_t getPartialFrames() const;
```

#### Remarks

This is getPartialFrames, a member of class VideoDecoderErrors.

### 2.1.50.6.3 VideoDecoderErrors::getPlayedFramesError Method

#### C++

```
uint32_t getPlayedFramesError() const;
```

#### Remarks

This is getPlayedFramesError, a member of class VideoDecoderErrors.

### 2.1.50.6.4 VideoDecoderErrors::getPlayedFramesOk Method

#### C++

```
uint32_t getPlayedFramesOk() const;
```

#### Remarks

This is getPlayedFramesOk, a member of class VideoDecoderErrors.

## 2.1.50.6.5 VideoDecoderErrors::getRepeatedFramesAVSync Method

#### C++

```
uint32_t getRepeatedFramesAVSync() const;
```

#### Remarks

This is getRepeatedFramesAVSync, a member of class VideoDecoderErrors.

# 2.1.50.6.6 VideoDecoderErrors::getRepeatedFramesUnderrun Method

#### C++

```
uint32_t getRepeatedFramesUnderrun() const;
```

#### Remarks

This is getRepeatedFramesUnderrun, a member of class VideoDecoderErrors.

# 2.1.50.6.7 VideoDecoderErrors::getSkippedFramesAVSync Method

#### C++

```
uint32_t getSkippedFramesAVSync() const;
```

#### Remarks

This is getSkippedFramesAVSync, a member of class VideoDecoderErrors.

# 2.1.50.6.8 VideoDecoderErrors::getSkippedFramesError Method

#### C++

```
uint32_t getSkippedFramesError() const;
```

#### Remarks

This is getSkippedFramesError, a member of class VideoDecoderErrors.

# 2.1.50.7 VideoDecoderErrors Operators

# 2.1.50.7.1 VideoDecoderErrors::= Operator

#### C++

```
VideoDecoderErrors & operator =(const VideoDecoderErrors & other);
```

#### Remarks

This is =, a member of class VideoDecoderErrors.

# 2.1.51 VideoDecoderStatus Class

#### Inheritance Hierarchy

#### C++

```
class VideoDecoderStatus : public DecoderStatus;
```

#### File

mvc2api\_decoder.h

#### Remarks

This is class mvc2::VideoDecoderStatus.

#### **Members**

#### Methods

	Name	Description
<b>=</b> ♦ <b>W</b>	~DecoderStatus	This is ~DecoderStatus, a member of class DecoderStatus.
<b>≡♦</b>	DecoderStatus	This is DecoderStatus, a member of class DecoderStatus.
<b>≡♦</b>	DecoderStatus	This is DecoderStatus, a member of class DecoderStatus.
<b>=</b>	DecoderStatus	This is DecoderStatus, a member of class DecoderStatus.
<b>≡♦</b>	DecoderStatus	This is DecoderStatus, a member of class DecoderStatus.
<b>≡♦</b>	DecoderStatus	This is DecoderStatus, a member of class DecoderStatus.
<b>≡♦</b>	DecoderStatus	This is DecoderStatus, a member of class DecoderStatus.

#### VideoDecoderStatus Class

	Name	Description
<b>≡</b>	VideoDecoderStatus	This is VideoDecoderStatus, a member of class VideoDecoderStatus.

<b>=</b> ♦	VideoDecoderStatus	This is VideoDecoderStatus, a member of class VideoDecoderStatus.
=•	VideoDecoderStatus	This is VideoDecoderStatus, a member of class VideoDecoderStatus.
<b>≡♦</b>	VideoDecoderStatus	This is VideoDecoderStatus, a member of class VideoDecoderStatus.
=•	VideoDecoderStatus	This is VideoDecoderStatus, a member of class VideoDecoderStatus.

#### **DecoderStatus Methods**

	Name	Description
<b>=♦</b>	getReturnCode	This is getReturnCode, a member of class DecoderStatus.
<b>≡</b>	getTimeStamp	This is getTimeStamp, a member of class DecoderStatus.
<b>=♦</b>	getUserData	This is getUserData, a member of class DecoderStatus.

#### VideoDecoderStatus Class

	Name	Description
<b>≡</b>	getPictureHeight	This is getPictureHeight, a member of class VideoDecoderStatus.
<b>≅∳</b>	getPictureWidth	This is getPictureWidth, a member of class VideoDecoderStatus.
<b>≡♦</b>	getTimeCode	This is getTimeCode, a member of class VideoDecoderStatus.

#### **DecoderStatus Operators**

	Name	Description
<del>(/=</del>	=	This is =, a member of class DecoderStatus.
(/ <del>_</del> =+)	bool	Returns if the object is valid.

#### VideoDecoderStatus Class

	Name	Description
<del>(/-</del> =+)	=	This is =, a member of class VideoDecoderStatus.

#### **DecoderStatus Methods**

	Name	Description
<b>≡♦</b>	getReturnCode	This is getReturnCode, a member of class DecoderStatus.
<b>=♦</b>	getTimeStamp	This is getTimeStamp, a member of class DecoderStatus.
<b>≡</b>	getUserData	This is getUserData, a member of class DecoderStatus.

#### VideoDecoderStatus Class

	Name	Description
<b>∉</b> ∳	getPictureHeight	This is getPictureHeight, a member of class VideoDecoderStatus.
<b>≡∳</b>	getPictureWidth	This is getPictureWidth, a member of class VideoDecoderStatus.
<b>∉</b> ∳	getTimeCode	This is getTimeCode, a member of class VideoDecoderStatus.

#### **DecoderStatus Operators**

	Name	Description
( <u>/-</u> =+)	=	This is =, a member of class DecoderStatus.
=+0	bool	Returns if the object is valid.

#### VideoDecoderStatus Class

		Name	Description
Ì	<u>(/_</u> =+)	=	This is =, a member of class VideoDecoderStatus.

# 2.1.51.1 VideoDecoderStatus::VideoDecoderStatus Constructor ()

C++

VideoDecoderStatus();

#### **Remarks**

This is VideoDecoderStatus, a member of class VideoDecoderStatus.

# 2.1.51.2 VideoDecoderStatus::VideoDecoderStatus Constructor (MvcDecoderPrivate \*)

C++

VideoDecoderStatus(MvcDecoderPrivate \* mvcdec);

#### Remarks

This is VideoDecoderStatus, a member of class VideoDecoderStatus.

# 2.1.51.3 VideoDecoderStatus::VideoDecoderStatus Constructor (MvcDecoderPrivate \*, VideoDecoderStatus &)

C++

VideoDecoderStatus(MvcDecoderPrivate \* mvcdec, const VideoDecoderStatus & other);

#### Remarks

This is VideoDecoderStatus, a member of class VideoDecoderStatus.

# 2.1.51.4 VideoDecoderStatus::VideoDecoderStatus Constructor (MvcDevice &, uint32\_t)

C++

VideoDecoderStatus(const MvcDevice & mvcdev, uint32\_t channel);

#### Remarks

This is VideoDecoderStatus, a member of class VideoDecoderStatus.

# 2.1.51.5 VideoDecoderStatus::VideoDecoderStatus Constructor (VideoDecoderStatus &)

C++

VideoDecoderStatus(const VideoDecoderStatus & other);

#### Remarks

This is VideoDecoderStatus, a member of class VideoDecoderStatus.

239

# 2.1.51.6 VideoDecoderStatus Methods

# 2.1.51.6.1 VideoDecoderStatus::getPictureHeight Method

#### C++

```
uint16_t getPictureHeight() const;
```

#### Remarks

This is getPictureHeight, a member of class VideoDecoderStatus.

# 2.1.51.6.2 VideoDecoderStatus::getPictureWidth Method

#### C++

```
uint16_t getPictureWidth() const;
```

#### Remarks

This is getPictureWidth, a member of class VideoDecoderStatus.

### 2.1.51.6.3 VideoDecoderStatus::getTimeCode Method

#### C++

```
TimeCode getTimeCode();
```

#### Remarks

This is getTimeCode, a member of class VideoDecoderStatus.

# 2.1.51.7 VideoDecoderStatus Operators

# 2.1.51.7.1 VideoDecoderStatus::= Operator

#### C++

```
VideoDecoderStatus & operator =(const VideoDecoderStatus & instat);
```

#### Remarks

This is =, a member of class VideoDecoderStatus.

# 2.1.52 VideoMode Class

#### Inheritance Hierarchy

#### C++

class VideoMode;

#### File

mvc2api\_output.h

#### Remarks

Video mode definition class.

#### **Members**

#### Methods

	Name	Description
<b>≟</b>	VideoMode	This is VideoMode, a member of class VideoMode.

#### VideoMode Enumerations

	Name	Description
	Mode	This is record mvc2::VideoMode::Mode.

#### **VideoMode Methods**

	Name	Description
<b>≡♦</b>	getMode	Gets video mode value.

#### **VideoMode Operators**

		Name	Description
=+)	ı	=	This is =, a member of class VideoMode.

#### **VideoMode Enumerations**

	Name	Description
<b>a</b>	Mode	This is record mvc2::VideoMode::Mode.

#### **VideoMode Methods**

	Name	Description
<b>≡♦</b>	getMode	Gets video mode value.

#### **VideoMode Operators**

	Name	Description
( <u>/-</u> =+)	=	This is =, a member of class VideoMode.

# 2.1.52.1 VideoMode::VideoMode Constructor

#### C++

```
VideoMode(VideoMode::Mode mode = Mode_None);
```

#### Remarks

This is VideoMode, a member of class VideoMode.

### 2.1.52.2 VideoMode Enumerations

#### 2.1.52.2.1 mvc2::VideoMode::Mode Enumeration

#### C++

```
enum Mode {
   Mode_None = 0,
   Mode_1920_1080_6000_p = 1,
   Mode_1920_1080_5994_p = 2,
   Mode_1920_1080_5000_p = 3,
   Mode_1920_1080_4800_p = 31,
   Mode_1920_1080_3000_p = 4,
   Mode_1920_1080_2997_p = 5,
   Mode_1920_1080_2500_p = 6,
   Mode_1920_1080_6000_i = 7,
   Mode_1920_1080_5994_i = 8,
```

```
Mode_1920_1080_5000_i = 9
Mode_1920_1080_5000_i_1250 = 10,
Mode_{1920_{1080_{2400_p}} = 11,
Mode_{1920_{1080_{2398_p}} = 12}
Mode_1920_1080_2400_psf = 13,
Mode_{1920_{1080_{2398_{psf}}} = 14,
Mode_1920_1080_2400_psf_1250 = 15,
Mode_1920_1080_2398_psf_1250 = 16,
Mode_1280_720_6000_p = 17,
Mode 1280 720 5994 p = 18,
Mode_{1280_{720_{5000_p}} = 19}
Mode_{1280_{720_{3000_{p}}} = 20,
Mode_{1280_{720_{2997_p}} = 21,
Mode_{1280_{720_{2500_p}} = 22}
Mode_1280_720_2400_p = 23,
Mode_1280_720_2398_p = 24,
Mode_720_576_5000_i = 25,
Mode_720_576_2500_p = 26,
Mode_{720_{576_{5000_{p}}} = 27}
Mode_{720_{480_{5994_i}} = 28,}
Mode_{720}_{480}_{2997}_{p} = 29, Mode_{720}_{480}_{5994}_{p} = 30,
Mode_2048_1080_6000_p = 32,
Mode_2048_1080_5994_p = 33,
Mode_2048_1080_5000_p = 34,
Mode_2048_1080_4800_p = 35,
Mode_{2048_{1080_{3000_p}}} = 36,
Mode_2048_1080_2997_p = 37,
Mode_2048_1080_2500_p = 38,
Mode_2048_1080_2400_p = 39,
Mode_2048_1080_2398_p = 40, Mode_2048_1080_6000_i = 41,
Mode_2048_1080_5994_i = 42
Mode_2048_1080_5000_i = 43,
Mode_2048_1080_2400_psf = 44,
Mode_2048_1080_2398_psf = 45,
Mode_4096_2160_3000_p = 50,
Mode_4096_2160_2997_p = 51,
Mode_{4096_{2160_{2500_p}} = 52,
Mode_{4096_{2160_{2400_p}} = 53,}
Mode_{4096_{2160_{2398_p}} = 54,}
Mode_1920_1080_12000_p = 55
Mode_{1920_{1080_{11988_p}} = 56,}
Mode_1920_1080_10000_p = 57,
Mode_{1920_{1080_{9600_p}} = 58,}
Mode_2048_1080_12000_p = 59
Mode_2048_1080_11988_p = 60,
Mode_2048_1080_10000_p = 61,
Mode_2048_1080_9600_p = 62,
Mode_Max
```

File

mvc2api\_output.h

#### Remarks

This is record mvc2::VideoMode::Mode.

## 2.1.52.3 VideoMode Methods

# 2.1.52.3.1 VideoMode::getMode Method

Gets video mode value.

#### C++

VideoMode::Mode getMode() const;

#### Returns

VideoMode::Mode or VideoMode::Mode\_None on failure.

#### Description

This function returns the actual VideoMode::Mode value from the object.

# 2.1.52.4 VideoMode Operators

# 2.1.52.4.1 VideoMode::= Operator

#### C++

```
VideoMode & operator =(const VideoMode & mode);
```

#### Remarks

This is =, a member of class VideoMode.

# 2.1.53 VideoOutput Class

#### **Inheritance Hierarchy**

#### C++

class VideoOutput;

#### File

mvc2api\_output.h

#### Remarks

The VideoOutput class represents physical or virtual video outputs.

#### Members

#### **Methods**

	Name	Description
<b>=</b> ♦ ₩	~VideoOutput	Destructs a VideoOutput.
<b>=</b> ♦	VideoOutput	Default constructor for a VideoOutput.
<b>≡♦</b>	VideoOutput	Creates a VideoOutput object.
<b>≡♦</b>	VideoOutput	Constructs a linked video output.
<b>≡♦</b>	VideoOutput	VideoOutput copy constructor.

#### **VideoOutput Enumerations**

	Name	Description
e <sup>20</sup>	EFFECTSPEED	Effect speed e.g. use by setPosition().
a <sup>2</sup>	MOVEEFFECT	Moving effect e.g. use by setPosition().
<b>a</b>	OUTPUTACTIVITY	Possible values for setOutputActivity().
<b>a</b>	VIDEOPROPERTY	Possible properties of a video output.

#### **VideoOutput Methods**

	Name	Description
<b>≡♦</b>	disableDeGhosting	This is disableDeGhosting, a member of class VideoOutput.
<b>≡⋄</b>	enableDeGhosting	This is enableDeGhosting, a member of class VideoOutput.
<b>≡⋄</b>	getActiveVideoMode	Get the active video mode.
<b>≡⋄</b>	setAudioEmbedding	Enables audio embedding.
<b>≡♦</b>	setOutputActivity	Select the behavior of a VideoOutput.
<b>=♦</b>	setPosition	Sets the position of the video.
<b>≡♦</b>	setVideoMode	Activate a video mode.

#### **VideoOutput Operators**

	Name	Description
( <u>/-</u> =+)	=	Assignment operator.
( <u>/-</u> =+)	bool	Bool operator.

#### **VideoOutput Enumerations**

	Name	Description
<b>a</b>	EFFECTSPEED	Effect speed e.g. use by setPosition().
<b>a</b>	MOVEEFFECT	Moving effect e.g. use by setPosition().
a <sup>a</sup>	OUTPUTACTIVITY	Possible values for setOutputActivity().
<b>a</b>	VIDEOPROPERTY	Possible properties of a video output.

#### **VideoOutput Methods**

	Name	Description
<b>=♦</b>	disableDeGhosting	This is disableDeGhosting, a member of class VideoOutput.
<b>=♦</b>	enableDeGhosting	This is enableDeGhosting, a member of class VideoOutput.
<b>=♦</b>	getActiveVideoMode	Get the active video mode.
<b>=♦</b>	setAudioEmbedding	Enables audio embedding.
<b>=♦</b>	setOutputActivity	Select the behavior of a VideoOutput.
<b>=♦</b>	setPosition	Sets the position of the video.
<b>=♦</b>	setVideoMode	Activate a video mode.

#### **VideoOutput Operators**

	Name	Description
( <u>/-</u> =+)	=	Assignment operator.
( <u>/-</u> =+)	bool	Bool operator.

# 2.1.53.1 VideoOutput::~VideoOutput Destructor

Destructs a VideoOutput.

#### C++

virtual ~VideoOutput();

#### Description

A VideoOutput object will be destruct if no more references exist. All decoders will automatically disconnect and the screen goes to black by stays in VideoMode.

# 2.1.53.2 VideoOutput::VideoOutput Constructor ()

Default constructor for a VideoOutput.

#### C++

VideoOutput();

#### Description

A empty VideoOutput object will be created.

# 2.1.53.3 VideoOutput::VideoOutput Constructor (TMmRc \*, MvcDevice &, uint32\_t)

Creates a VideoOutput object.

#### C++

VideoOutput(TMmRc \* resultPointer, const MvcDevice & mvcdev, uint32\_t VideoProperty = 0);

#### **Parameters**

Parameters	Description
TMmRc * resultPointer	Pointer where to store the result in. if 0 no result will be written.
const MvcDevice & mvcdev	MvcDevice to create the output on
uint32_t VideoProperty = 0	optional properties to influence the output selection

#### Description

This method creates a object for a physical or virtual video output. Before the object will be created the list of available outputs will be parsed to find an output which fits best to the VideoProperty.

# 2.1.53.4 VideoOutput::VideoOutput Constructor (TMmRc \*, VideoOutput &)

Constructs a linked video output.

#### C++

VideoOutput(TMmRc \* resultPointer, const VideoOutput & linked\_output);

#### **Parameters**

Parameters	Description
TMmRc * resultPointer	Return code of the creation, MMRC_Ok if anything was successful
const VideoOutput & linked_output	reference to the main VideoOutput

#### Description

A linked video output shares some parameter with another output. Due to hardware limitations only one real physical video output can be created and all others have to be linked to it. Linked outputs share the same video mode, but can have different video decoder connected. They are mainly used to output 3D videos. Second usage is to created virtual linked outputs where the output video can be selected with setOutputActivity() on the main VideoOutput.

# 2.1.53.5 VideoOutput::VideoOutput Constructor (VideoOutput &)

VideoOutput copy constructor.

#### C++

VideoOutput(const VideoOutput & vout);

#### **Parameters**

Parameters	Description
const VideoOutput & vout	VideoOutput object to copy the reference from

#### Description

Creates a new VideoOutput object from another VideoOutput. Both the old and the new VideoOutput object reference the same physical VideoOutput.

# 2.1.53.6 VideoOutput Enumerations

## 2.1.53.6.1 mvc2::VideoOutput::EFFECTSPEED Enumeration

#### C++

```
enum EFFECTSPEED {
   Speed_Immediately = 0,
   Speed_ExtremelyFast = 1,
   Speed_VeryFast = 4,
   Speed_Fast = 8,
   Speed_Normal = 16,
   Speed_Slow = 32,
   Speed_VerySlow = 48
};
```

#### File

mvc2api\_output.h

#### **Members**

Members	Description
Speed_Immediately = 0	no transition effect
Speed_ExtremelyFast = 1	approx. 1/16 second
Speed_VeryFast = 4	approx. 1/4 second
Speed_Fast = 8	approx. 1/2 second
Speed_Normal = 16	approx. 1 second
Speed_Slow = 32	approx. 2 seconds
Speed_VerySlow = 48	approx. 3 seconds

#### Remarks

Effect speed e.g. use by setPosition().

# 2.1.53.6.2 mvc2::VideoOutput::MOVEEFFECT Enumeration

#### C++

```
enum MOVEEFFECT {
   MoveEffect_Linear = 0,
   MoveEffect_SlowDown = 1,
   MoveEffect_SpeedUp = 2
};
```

#### File

mvc2api\_output.h

#### Members

Members	Description
MoveEffect_Linear = 0	The movement speed is linear from start to end.

MoveEffect_SlowDown = 1	The movement starts fast and slows down when it comes to end position.
MoveEffect_SpeedUp = 2	The movement starts slow and will speed up constantly.

#### Remarks

Moving effect e.g. use by setPosition().

# 2.1.53.6.3 mvc2::VideoOutput::OUTPUTACTIVITY Enumeration

#### C++

```
enum OUTPUTACTIVITY {
   Activity_Primary_Only = 0,
   Activity_Secondary_Only = 1,
   Activity_Primary_Secondary_as_Overlay = 2,
   Activity_Secondary_Primary_as_Overlay = 3,
   Activity_Primary_Secondary = 4
};
```

#### File

mvc2api\_output.h

#### **Members**

Members	Description
Activity_Primary_Only = 0	Only the primary video from the main VideoOutput is displayed.
Activity_Secondary_Only = 1	Only the secondary video from the linked VideoOutput is displayed.
Activity_Primary_Secondary_as_Overlay = 2	Secondary video is blended over the primary video.
Activity_Secondary_Primary_as_Overlay = 3	Primary video blended over secondary video.
Activity_Primary_Secondary = 4	Both videos are displayed, standard 3D mode without virtual VideoOutputs.

#### Remarks

Possible values for setOutputActivity().

# 2.1.53.6.4 mvc2::VideoOutput::VIDEOPROPERTY Enumeration

#### C++

```
enum VIDEOPROPERTY {
   VideoProperty_SDTV = (1<<0),
   VideoProperty_SDTV_Progressive = (1<<1),
   VideoProperty_HDTV = (1<<2),
   VideoProperty_Dual_HDTV = (1<<3),
   VideoProperty_VirtualLinked = (1<<4)
};</pre>
```

#### File

mvc2api\_output.h

#### **Members**

Members	Description
VideoProperty_SDTV = (1<<0)	Output support standard definition TV frequencies.
VideoProperty_SDTV_Progressive = (1<<1)	Output supports standard definition TV frequencies with doubled frame rate as progressive. This includes 480p and 576p.
VideoProperty_HDTV = (1<<2)	Output supports high definition TV frequencies.

VideoProperty_Dual_HDTV = (1<<3)	Output supports high definition TV frequencies with doubled frame rate, like 1080p60.
VideoProperty_VirtualLinked = (1<<4)	Linked outputs are only virtual and not physical available.

#### Remarks

Possible properties of a video output.

# 2.1.53.7 VideoOutput Methods

# 2.1.53.7.1 VideoOutput::disableDeGhosting Method

#### C++

```
TMmRc disableDeGhosting();
```

#### Remarks

This is disableDeGhosting, a member of class VideoOutput.

## 2.1.53.7.2 VideoOutput::enableDeGhosting Method

#### C++

```
TMmRc enableDeGhosting(const int32_t * parameter = 0, uint32_t size = 0, uint32_t type = 0);
```

#### Remarks

This is enableDeGhosting, a member of class VideoOutput.

### 2.1.53.7.3 VideoOutput::getActiveVideoMode Method

Get the active video mode.

#### C++

VideoMode getActiveVideoMode();

#### Returns

Returns MMRC\_Ok on successful retrieval of the video mode.

#### **Return Values**

Return Values	Description
MMRC_Ok	successful operation

#### Description

This method returns the currently active video mode.

# 2.1.53.7.4 VideoOutput::setAudioEmbedding Method

Enables audio embedding.

#### C++

```
TMmRc setAudioEmbedding(bool enable);
```

#### **Parameters**

Parameters	Description
bool enable	true to enable embedding else false

#### **Returns**

MMRC\_Ok if the audio embedding was successfully enabled.

#### **Return Values**

Return Values	Description
MMRC_Ok	success
MMRC_MVC2_InvalidObject	if the VideoOutput object is empty

#### Description

This method controls if embedded audio into HD-SDI is used or not. All 16 channels will be embedded into channel A.

# 2.1.53.7.5 VideoOutput::setOutputActivity Method

Select the behavior of a VideoOutput.

#### C++

TMmRc setOutputActivity(OUTPUTACTIVITY activity, float secondaryAlpha = 1.0f);

#### **Parameters**

Parameters	Description
OUTPUTACTIVITY activity	One of the VideoOutput::OUTPUTACTIVITY values
float secondaryAlpha = 1.0f	For Activity_Primary_Secondary_as_Overlay or Activity_Secondary_Primary_as_Overlay a alpha value between 1.0 and 0.0 can be given to use for the overlay

#### Returns

MMRC\_Ok if the setup was successfully completed, otherwise an error code.

#### **Return Values**

Return Values	Description
MMRC_Ok	success
MMRC_MVC2_InvalidObject	if the VideoOutput object is empty

#### Description

This method selects the video which should be output on the primary channel. It's used in conjunction with two linked VideoOutputs if the secondary output is only virtual. It is possible to select the video from decoder at VideoOutput one or two, or blend the video from both outputs together. The main functionality is to dynamically switch between two channels of a 3D video by having only one projector/monitor connected.

See VideoOutput::OUTPUTACTIVITY for possible settings.

# 2.1.53.7.6 VideoOutput::setPosition Method

Sets the position of the video.

#### C++

```
TMmRc setPosition(const FramePosition & pos, EFFECTSPEED transistionSpeed =
Speed_Immediately, MOVEEFFECT transisitionEffect = MoveEffect_Linear);
```

#### **Parameters**

Parameters	Description
const FramePosition & pos	destination position of the video
EFFECTSPEED transistionSpeed = Speed_Immediately	VideoOutput::EFFECTSPEED how fast the transition from current to destination position is completed

MOVEEFFECT transisitionEffect = MoveEffect_Linear	VideoOutput::MOVEEFFECT selects the movement effect
	for the transition

#### Returns

MMRC\_Ok on success, otherwise an error code.

#### **Return Values**

Return Values	Description
MMRC_Ok	success
MMRC_MVC2_InvalidObject	if the VideoOutput object is empty

#### Description

A video can be freely positioned on screen. This is especially useful if the input video is smaller or bigger than the output video mode requires it. See FramePosition for greater detail on the exact position. The default position is centered in both directions.

# 2.1.53.7.7 VideoOutput::setVideoMode Method

Activate a video mode.

#### C++

TMmRc setVideoMode(const VideoMode mode, const ColorSpace color =
ColorSpace(ColorSpace::Space\_Default));

#### **Parameters**

Parameters	Description
const VideoMode mode	VideoMode
const ColorSpace color = ColorSpace(ColorSpace::Space_Default)	ColorSpace or default for the default color space of the video mode.

#### Returns

Returns MMRC\_Ok on successful activation of the video mode.

#### **Return Values**

Return Values	Description
MMRC_Ok	successful operation
MMRC_MVC2_OutputNotFound	the video output could not be location on the card
MMRC_MVC2_VideoModeNotSupported	the video mode is not supported

#### Description

Select a video mode and color space for a VideoOutput. Some video mode may support different color spaces, that makes it important to select the color space you want to output. For example dual HD-SDI may support YCbCr and RGB. It's an important information to give the firmware the chance to convert the input video to the correct output color space.

Note: The current firmware version does not support a color conversion. The color space selection will only effect the color for inactive areas inside a frame or if no video is displayed.

# 2.1.53.8 VideoOutput Operators

# 2.1.53.8.1 VideoOutput::= Operator

Assignment operator.

#### C++

VideoOutput & operator =(const VideoOutput & vout);

#### **Parameters**

Parameters	Description
const VideoOutput & vout	source VideoOutput object

#### Returns

Reference to the destination VideoOutput object.

#### Description

This operator copies the reference of a VideoOutput to a new object.

## 2.1.53.8.2 VideoOutput::bool Operator

Bool operator.

#### C++

operator bool() const;

#### Description

Check if a VideoOutput object is empty or not. Returns false for an empty object.

# 2.1.54 VSCConnector Class



#### **Inheritance Hierarchy**

#### C++

class VSCConnector : public ExternalConnector;

#### File

mvc2api\_output.h

#### Remarks

This is class mvc2::VSCConnector.

#### **Members**

#### Methods

	Name	Description
<b>=♦</b> ₩	~ExternalConnector	This is ~ExternalConnector, a member of class ExternalConnector.
<b>≡</b>	ExternalConnector	This is ExternalConnector, a member of class ExternalConnector.
<b>≡♦</b>	ExternalConnector	This is ExternalConnector, a member of class ExternalConnector.
<b>=♦</b> ?	ExternalConnector	This is ExternalConnector, a member of class ExternalConnector.
<b>≡∳</b>	ExternalConnector	This is ExternalConnector, a member of class ExternalConnector.

#### **VSCConnector Class**

	Name	Description
<b>=</b> ♦ <b>W</b>	~VSCConnector	This is ~VSCConnector, a member of class VSCConnector.

251

<b>=</b> ♦	VSCConnector	This is VSCConnector, a member of class VSCConnector.
<b>=♦9</b>	VSCConnector	This is VSCConnector, a member of class VSCConnector.
<b>=♦</b>	VSCConnector	This is VSCConnector, a member of class VSCConnector.
<b>=♦</b>	VSCConnector	This is VSCConnector, a member of class VSCConnector.

#### **ExternalConnector Data Members**

	Name	Description
<b>₽</b> ₽	m_ConnectorPrivate	This is m_ConnectorPrivate, a member of class
		ExternalConnector.

#### **ExternalConnector Methods**

	Name	Description
<b>=</b> ♦	readI2C	This is readl2C, a member of class ExternalConnector.
<b>≡♦</b>	readI2C	This is readl2C, a member of class ExternalConnector.
<b>≡♦</b>	writel2C	This is writel2C, a member of class ExternalConnector.

#### **VSCConnector Class**

	Name	Description
<b>≅♦</b>	abortGPIOInterrupt	This is abortGPIOInterrupt, a member of class VSCConnector.
<b>=</b> ♦	getGPIOValue	This is getGPIOValue, a member of class VSCConnector.
<b>≡</b>	setGPIODirection	preliminary functions:
<b>≡</b>	setGPIOInterrupt	This is setGPIOInterrupt, a member of class VSCConnector.
<b>≡</b>	setGPIOValue	This is setGPIOValue, a member of class VSCConnector.
<b>≡♦</b>	waitForGPIOInterrupt	This is waitForGPIOInterrupt, a member of class VSCConnector.

#### **ExternalConnector Operators**

	Name	Description
=+)	=	This is =, a member of class ExternalConnector.
<u>(/-</u> =+)	bool	Returns if the object is valid.

#### **ExternalConnector Data Members**

	Name	Description
<b>∳</b> ∳	m_ConnectorPrivate	This is m_ConnectorPrivate, a member of class
		ExternalConnector.

#### **ExternalConnector Methods**

	Name	Description
<b>≡</b>	readI2C	This is readI2C, a member of class ExternalConnector.
<b>=</b> ♦	readl2C	This is readI2C, a member of class ExternalConnector.
<b>≡</b>	writel2C	This is writel2C, a member of class ExternalConnector.

#### **VSCConnector Class**

	Name	Description
<b>=</b> ♦	abortGPIOInterrupt	This is abortGPIOInterrupt, a member of class VSCConnector.
<b>≡</b>	getGPIOValue	This is getGPIOValue, a member of class VSCConnector.
<b>≡</b>	setGPIODirection	preliminary functions:
<b>≡</b>	setGPIOInterrupt	This is setGPIOInterrupt, a member of class VSCConnector.
<b>≡</b>	setGPIOValue	This is setGPIOValue, a member of class VSCConnector.
<b>≟</b>	waitForGPIOInterrupt	This is waitForGPIOInterrupt, a member of class VSCConnector.

#### **ExternalConnector Operators**

	Name	Description
=+)	=	This is =, a member of class ExternalConnector.
<u>(/-</u> =+)	bool	Returns if the object is valid.

## 2.1.54.1 VSCConnector::~VSCConnector Destructor

#### C++

```
virtual ~VSCConnector();
```

#### Remarks

This is ~VSCConnector, a member of class VSCConnector.

# 2.1.54.2 VSCConnector::VSCConnector Constructor ()

#### C++

VSCConnector();

#### **Remarks**

This is VSCConnector, a member of class VSCConnector.

# 2.1.54.3 VSCConnector::VSCConnector Constructor (ExternalConnectorPrivate \*)

#### C++

VSCConnector(ExternalConnectorPrivate \* dev);

#### Remarks

This is VSCConnector, a member of class VSCConnector.

# 2.1.54.4 VSCConnector::VSCConnector Constructor (TMmRc \*, MvcDevice &)

#### C++

```
VSCConnector(TMmRc * resultPointer, const MvcDevice & mvcDevice);
```

#### Remarks

This is VSCConnector, a member of class VSCConnector.

# 2.1.54.5 VSCConnector::VSCConnector Constructor (VSCConnector &)

#### C++

```
VSCConnector (const VSCConnector & other);
```

#### Remarks

This is VSCConnector, a member of class VSCConnector.

# 2.1.54.6 VSCConnector Methods

## 2.1.54.6.1 VSCConnector::abortGPIOInterrupt Method

#### C++

```
TMmRc abortGPIOInterrupt();
```

#### Remarks

This is abortGPIOInterrupt, a member of class VSCConnector.

### 2.1.54.6.2 VSCConnector::getGPIOValue Method

#### C++

```
TMmRc getGPIOValue(uint32_t * val);
```

#### Remarks

This is getGPIOValue, a member of class VSCConnector.

#### 2.1.54.6.3 VSCConnector::setGPIODirection Method

#### C++

```
TMmRc setGPIODirection(uint32_t dir, uint32_t dir_mask);
```

#### Remarks

preliminary functions:

## 2.1.54.6.4 VSCConnector::setGPIOInterrupt Method

#### C++

```
TMmRc setGPIOInterrupt(uint32_t gpioNum, uint32_t triggerMode);
```

#### Remarks

This is setGPIOInterrupt, a member of class VSCConnector.

#### 2.1.54.6.5 VSCConnector::setGPIOValue Method

#### C++

```
TMmRc setGPIOValue(uint32_t val, uint32_t value_mask);
```

#### Remarks

This is setGPIOValue, a member of class VSCConnector.

### 2.1.54.6.6 VSCConnector::waitForGPIOInterrupt Method

#### C++

```
TMmRc waitForGPIOInterrupt(uint32_t * val);
```

#### Remarks

This is waitForGPIOInterrupt, a member of class VSCConnector.

# Index

AudioDecoderStatus class 13

	AudioDecoderStatus 14, 15
A	getAudioLevel 16
ActiveDecoderInfo class 5	isAudioChannelActive 16
~ActiveDecoderInfo 6	AudioOutput class 16
= 7	~AudioOutput 18
about ActiveDecoderInfo class 5	= 23
ActiveDecoderInfo 6	about AudioOutput class 16
getChannel 6	AudioOutput 18
ActiveDecoderIterator class 7	bool 23
~ActiveDecoderIterator 8	getAudioOutputStatus 19
= 8	getChannelMute 19
about ActiveDecoderIterator class 7	getNumberOfChannels 20
ActiveDecoderIterator 8	getOuputFrequency 20
getNext 8	getOutputDelay 20
Activity_Primary_Only enumeration member 247	setChannelMute 20
Activity_Primary_Secondary enumeration member 247	setOutputDelay 21
Activity_Primary_Secondary_as_Overlay enumeration	setOutputFrequency 21
member 247	setUserData 22
Activity_Secondary_Only enumeration member 247	setVolume 22
Activity_Secondary_Primary_as_Overlay enumeration member 247	setVolumeDB 22
AES128OP_cbc_dec enumeration member 175	AudioOutputStatus class 23  ~AudioOutputStatus 24  = 27  about AudioOutputStatus class 23  AudioOutputStatus 25  bool 27
AES128OP_cbc_dec_fpga enumeration member 175	
AES128OP_cbc_enc enumeration member 175	
AES128OP_ecb_dec enumeration member 175	
AES128OP_ecb_enc enumeration member 175	
AES256OP_cbc_dec enumeration member 175	
AES256OP_cbc_enc enumeration member 175	getOvershootLength 26
AudioDecoderErrors class 9	getOvershootSamples 26
~AudioDecoderErrors 10	getSampleActivity 26
= 13	getSampleLevel 26
about AudioDecoderErrors class 9	getSampleLevelDB 27
AudioDecoderErrors 11	isChannelActive 27
getClippedSamples 11	m_StatusPriv 26
getInputUnderruns 12	AudioProperty_16bit_Samples enumeration member 19
getMaxClippingPeriode 12	AudioProperty_20bit_Samples enumeration member 19
getNumberOfClippings 12	AudioProperty_24bit_Samples enumeration member 19
getPlayedSamples 12	AudioProperty_Bitstream_Support enumeration member
getSilentSamples 12	AudioProperty_Video_Embedding enumeration member 19
getSkippedSamples 12	Authentication 3

= 16

about AudioDecoderStatus class 13

B	about ColorConversionMatrix class 31
Design Toward	ColorConversionMatrix 32
Basic Types 1	ColorSpace class 32
BufferStatus class 27	= 36
~BufferStatus 29	about ColorSpace class 32
= 31	ColorSpace 33, 34
about BufferStatus class 27	getColorSpacePacked 35
bool 31	getSpace 35
BufferStatus 29	ConfigAccess class 36
getBitrateDecoder 30	~ConfigAccess 37
getBitrateDriver 30	= 40
getDecoderBufferMax 30	about ConfigAccess class 36
getDecoderBufferSize 30	bool 41
getDecoderFullness 30	ConfigAccess 38
getFreeDataBuffers 30	getLogFilter 38
getPendingFrames 31	getLogFilterConsole 38
getReturnCode 31	getMessageFifoClients 38
	getMessageFifoSize 39
C	getNetworkDHCP 39
Chroma_blue enumeration member 59	getNetworkIPAddress 39
Chroma_blue_proj enumeration member 59	getNetworkMask 39
Chroma_green enumeration member 59	setLogFilter 39
Chroma_green_proj enumeration member 59	setLogFilterConsole 39
Chroma_red enumeration member 59	setMessageFifoClients 40
Chroma_red_proj enumeration member 59	setMessageFifoSize 40
Chroma_white enumeration member 59	setNetworkDHCP 40
Chroma_white_iter enumeration member 59	setNetworkIPAddress 40
Chroma_white_proj enumeration member 59	setNetworkMask 40
ClockId_Firmware enumeration member 106	CplException_Hash enumeration member 192
ClockId_SM enumeration member 106	CplException_Missing enumeration member 192
ClockId_System enumeration member 106	CplException_None enumeration member 192
Coeff_DCI_P3 enumeration member 34	CPUInfo class 41
Coeff_Default enumeration member 34	~CPUInfo 42
Coeff_FCC enumeration member 34	= 44
Coeff_ITUR_BT_470_BG enumeration member 34	about CPUInfo class 41
Coeff_ITUR_BT_709 enumeration member 34	bool 44
Coeff_RGB enumeration member 34	CPUInfo 42
Coeff_SMPTE_170M enumeration member 34	getCoreFrequency 43
Coeff_SMPTE_240M enumeration member 34	getCoreFrequencykHz 43
Coeff_XYZ enumeration member 34	getCPUType 43
Coeff_YCxCz enumeration member 34	getCPUTypeString 43
ColorConversionMatrix class 31	· · · · · ·

getHardwareVersion 43	getTimeStamp 56
getNumberOfCores 44	getUserData 57
CPUTYPE_PMCSierra_MSP8510 enumeration member 43	Design Guide 1
CPUTYPE_Unknown enumeration member 43	Dolby3Dchromaticity class 57
Create_4K_Decoder enumeration member 77	~Dolby3Dchromaticity 58
Create_HighFrameRate enumeration member 77	about Dolby3Dchromaticity class 57
	Dolby3Dchromaticity 58
D	getChroma 59
DataBuffer class 44	getLuminance 59
~DataBuffer 46	isCorrectionUsed 59
= 51	isWPointIterUsed 59
about DataBuffer class 44	setChroma 60
bool 51	setLuminance 60
copy 46	DonotWait enumeration member 82, 99
DataBuffer 46	
getBufferAddress 47	E
getFreeSize 47	EVENT_BAT_NEW enumeration member 220
getInSize 47	EVENT_BAT_REMOVE enumeration member 220
getTimeStamp 48	EVENT_DIVORCE_1 enumeration member 220
send 48	EVENT_DIVORCE_2 enumeration member 220
setDecryptionSize 48	EVENT_DOOR_CLOSE_1 enumeration member 220
setInSize 49	EVENT_DOOR_CLOSE_2 enumeration member 220
setKeyld 49	EVENT_DOOR_OPEN_1 enumeration member 220
setKeyIndex 49	EVENT_DOOR_OPEN_2 enumeration member 220
setMicValue 50	EVENT_POWER_LOSS enumeration member 220
setTimeStamp 50	EVENT_POWER_OV enumeration member 220
setUserData 50	EVENT_TAMPER_1 enumeration member 220
wait 51	EVENT_TAMPER_2 enumeration member 220
DecoderErrors class 52	ExternalConnector class 60
~DecoderErrors 52	~ExternalConnector 61
= 54	= 63
about DecoderErrors class 52	about ExternalConnector class 60
bool 54	bool 63
DecoderErrors 53	ExternalConnector 61, 62
getReturnCode 53	m_ConnectorPrivate 62
DecoderStatus class 54	readI2C 62
~DecoderStatus 55	writeI2C 63
= 57	
about DecoderStatus class 54	F
bool 57	FeatureAccess class 63
DecoderStatus 55, 56	~FeatureAccess 64
getReturnCode 56	= 65

about FeatureAccess class 63	IdSecurityManager enumeration member 86
bool 65	Interval_100ms enumeration member 29
FeatureAccess 64	Interval_10s enumeration member 29
getNextFeature 65	Interval_1s enumeration member 15, 29
installFeature 65	Interval_50ms enumeration member 15
queryFeature 65	
Find existing MVC cards 2	è J
FramePosition class 66	Jpeg2kDecoder class 72
!= 71	~Jpeg2kDecoder 76
= 72	= 80
== 72	about Jpeg2kDecoder class 72
about FramePosition class 66	getFrameRate 78
fgetX 69	getFrameRateTicks 78
fgetY 69	Jpeg2kDecoder 76, 77
FramePosition 67, 68	set4kMode 78
fsetX 69	setColorSpace 78
fsetY 69	setCutOffLevel 79
getModeX 70	setFrameRate 79
getModeY 70	setFrameRateTicks 79
getX 70	setResolutionDivider 79
getY 70	Schresolation Divider 73
setModeX 70	I
setModeY 71	L 1400
setX 71	Load_100ms enumeration member 209
setY 71	Load_10s enumeration member 209
friend class LogAccess 91	Load_1s enumeration member 209
friend class MvcDecoder 193	LogAccess class 80
friend class OverlayDataBuffer 136	~LogAccess 81
friend class OverlayDecoder 125, 133	= 83
friend class PCMDecoder 143	abortGetMessage 82
friend class SecurityAccess 222	about LogAccess class 80
friend class SubtitleDecoder 133, 139	bool 83
friend class VideoDecoder 228	getMessage 82
	LogAccess 81
G	setMessageFilter 83
Getting started 1	LogMessage class 84
Setting started 1	~LogMessage 85
I	= 90
	about LogMessage class 84
IdApi enumeration member 86	bool 91
IdDriver enumeration member 86	class LogAccess 91
IdFirmware enumeration member 86	getCTime 88
IdNone enumeration member 86	getld 88

getMessageCounter 88
getMessageString 89
getMessageStringLength 89
getMicroSeconds 89
getSeverity 89
getSubId 90
getTimeStamp 90
LogMessage 85, 86
setMessage 90

#### M

Md5 enumeration member 174 MEDIA\_TYPE\_Audio enumeration member 174 MEDIA\_TYPE\_Other enumeration member 174 MEDIA\_TYPE\_Video enumeration member 174 MMTrippleDes\_cbc\_dec enumeration member 175 MMTrippleDes\_cbc\_enc enumeration member 175 Mode\_1280\_720\_2398\_p enumeration member 241 Mode\_1280\_720\_2400\_p enumeration member 241 Mode\_1280\_720\_2500\_p enumeration member 241 Mode\_1280\_720\_2997\_p enumeration member 241 Mode\_1280\_720\_3000\_p enumeration member 241 Mode\_1280\_720\_5000\_p enumeration member 241 Mode\_1280\_720\_5994\_p enumeration member 241 Mode\_1280\_720\_6000\_p enumeration member 241 Mode\_1920\_1080\_10000\_p enumeration member 241 Mode\_1920\_1080\_11988\_p enumeration member 241 Mode\_1920\_1080\_12000\_p enumeration member 241 Mode\_1920\_1080\_2398\_p enumeration member 241 Mode\_1920\_1080\_2398\_psf enumeration member 241 Mode\_1920\_1080\_2398\_psf\_1250 enumeration member 241 Mode\_1920\_1080\_2400\_p enumeration member 241 Mode\_1920\_1080\_2400\_psf enumeration member 241 Mode\_1920\_1080\_2400\_psf\_1250 enumeration member 241 Mode\_1920\_1080\_2500\_p enumeration member 241 Mode\_1920\_1080\_2997\_p enumeration member 241 Mode\_1920\_1080\_3000\_p enumeration member 241 Mode\_1920\_1080\_4800\_p enumeration member 241 Mode\_1920\_1080\_5000\_i enumeration member 241 Mode\_1920\_1080\_5000\_i\_1250 enumeration member 241 Mode\_1920\_1080\_5000\_p enumeration member 241 Mode\_1920\_1080\_5994\_i enumeration member 241

Mode\_1920\_1080\_5994\_p enumeration member 241 Mode\_1920\_1080\_6000\_i enumeration member 241 Mode\_1920\_1080\_6000\_p enumeration member 241 Mode 1920 1080 9600 p enumeration member 241 Mode\_2048\_1080\_10000\_p enumeration member 241 Mode\_2048\_1080\_11988\_p enumeration member 241 Mode\_2048\_1080\_12000\_p enumeration member 241 Mode\_2048\_1080\_2398\_p enumeration member 241 Mode\_2048\_1080\_2398\_psf enumeration member 241 Mode\_2048\_1080\_2400\_p enumeration member 241 Mode\_2048\_1080\_2400\_psf enumeration member 241 Mode\_2048\_1080\_2500\_p enumeration member 241 Mode\_2048\_1080\_2997\_p enumeration member 241 Mode\_2048\_1080\_3000\_p enumeration member 241 Mode\_2048\_1080\_4800\_p enumeration member 241 Mode\_2048\_1080\_5000\_i enumeration member 241 Mode\_2048\_1080\_5000\_p enumeration member 241 Mode\_2048\_1080\_5994\_i enumeration member 241 Mode\_2048\_1080\_5994\_p enumeration member 241 Mode\_2048\_1080\_6000\_i enumeration member 241 Mode\_2048\_1080\_6000\_p enumeration member 241 Mode\_2048\_1080\_9600\_p enumeration member 241 Mode\_4096\_2160\_2398\_p enumeration member 241 Mode\_4096\_2160\_2400\_p enumeration member 241 Mode\_4096\_2160\_2500\_p enumeration member 241 Mode 4096 2160 2997 p enumeration member 241 Mode\_4096\_2160\_3000\_p enumeration member 241 MODE\_4K\_Always enumeration member 77 MODE\_4K\_Automatic enumeration member 77 MODE\_4K\_DropScale enumeration member 77 MODE\_4K\_Off enumeration member 77 Mode\_720\_480\_2997\_p enumeration member 241 Mode 720 480 5994 i enumeration member 241 Mode\_720\_480\_5994\_p enumeration member 241 Mode\_720\_576\_2500\_p enumeration member 241 Mode\_720\_576\_5000\_i enumeration member 241 Mode\_720\_576\_5000\_p enumeration member 241 Mode\_Align\_Bottom enumeration member 68 Mode\_Align\_Center enumeration member 68 Mode\_Align\_Left enumeration member 68 Mode\_Align\_Right enumeration member 68 Mode\_Align\_Top enumeration member 68

mvc2::AudioOutput 16 Mode\_Max enumeration member 241 Mode\_None enumeration member 241 mvc2::AudioOutput::~AudioOutput 18 Mode Percent enumeration member 68 mvc2::AudioOutput::= 23 Mode\_Pixels enumeration member 68 mvc2::AudioOutput::AudioOutput 18 mvc2::AudioOutput::AUDIOPROPERTY 19 MoveEffect Linear enumeration member 123, 246 mvc2::AudioOutput::AUDIOPROPERTY enumeration 19 MoveEffect\_SlowDown enumeration member 123, 246 MoveEffect\_SpeedUp enumeration member 123, 246 mvc2::AudioOutput::bool 23 Mpeg2Decoder class 91 mvc2::AudioOutput::getAudioOutputStatus 19 ~Mpeg2Decoder 94 mvc2::AudioOutput::getChannelMute 19 = 96mvc2::AudioOutput::getNumberOfChannels 20 about Mpeg2Decoder class 91 mvc2::AudioOutput::getOuputFrequency 20 Mpeg2Decoder 94, 95 mvc2::AudioOutput::getOutputDelay 20 mvc2 4 mvc2::AudioOutput::setChannelMute 20 mvc2 namespace 4 mvc2::AudioOutput::setOutputDelay 21 mvc2::ActiveDecoderInfo 5 mvc2::AudioOutput::setOutputFrequency 21 mvc2::ActiveDecoderInfo::~ActiveDecoderInfo 6 mvc2::AudioOutput::setUserData 22 mvc2::ActiveDecoderInfo::= 7 mvc2::AudioOutput::setVolume 22 mvc2::ActiveDecoderInfo::ActiveDecoderInfo 6 mvc2::AudioOutput::setVolumeDB 22 mvc2::ActiveDecoderInfo::getChannel 6 mvc2::AudioOutputStatus 23 mvc2::ActiveDecoderIterator 7 mvc2::AudioOutputStatus::~AudioOutputStatus 24 mvc2::ActiveDecoderIterator::~ActiveDecoderIterator 8 mvc2::AudioOutputStatus::= 27 mvc2::ActiveDecoderIterator::= 8 mvc2::AudioOutputStatus::AudioOutputStatus 25 mvc2::ActiveDecoderIterator::ActiveDecoderIterator 8 mvc2::AudioOutputStatus::bool 27 mvc2::ActiveDecoderIterator::getNext 8 mvc2::AudioOutputStatus::getOvershootLength 26 mvc2::AudioDecoderErrors 9 mvc2::AudioOutputStatus::getOvershootSamples 26 mvc2::AudioDecoderErrors::~AudioDecoderErrors 10 mvc2::AudioOutputStatus::getSampleActivity 26 mvc2::AudioDecoderErrors::= 13 mvc2::AudioOutputStatus::getSampleLevel 26 mvc2::AudioDecoderErrors::AudioDecoderErrors 11 mvc2::AudioOutputStatus::getSampleLevelDB 27 mvc2::AudioDecoderErrors::getClippedSamples 11 mvc2::AudioOutputStatus::isChannelActive 27 mvc2::AudioDecoderErrors::getInputUnderruns 12 mvc2::AudioOutputStatus::m\_StatusPriv 26 mvc2::AudioDecoderErrors::getMaxClippingPeriode 12 mvc2::AudioOutputStatus::TIME\_RESOLUTION 25 mvc2::AudioDecoderErrors::getNumberOfClippings 12 mvc2::AudioOutputStatus::TIME\_RESOLUTION enumeration mvc2::AudioDecoderErrors::getPlayedSamples 12 mvc2::BufferStatus 27 mvc2::AudioDecoderErrors::getSilentSamples 12 mvc2::BufferStatus::~BufferStatus 29 mvc2::AudioDecoderErrors::getSkippedSamples 12 mvc2::BufferStatus::= 31 mvc2::AudioDecoderStatus 13 mvc2::BufferStatus::bool 31 mvc2::AudioDecoderStatus::= 16 mvc2::BufferStatus::BufferStatus 29 mvc2::AudioDecoderStatus::AudioDecoderStatus 14, 15 mvc2::BufferStatus::getBitrateDecoder 30 mvc2::AudioDecoderStatus::getAudioLevel 16 mvc2::BufferStatus::getBitrateDriver 30 mvc2::AudioDecoderStatus::INTERVAL 15 mvc2::BufferStatus::getDecoderBufferMax 30 mvc2::AudioDecoderStatus::INTERVAL enumeration 15 mvc2::BufferStatus::getDecoderBufferSize 30 mvc2::AudioDecoderStatus::isAudioChannelActive 16 mvc2::BufferStatus::getDecoderFullness 30

mvc2::CPUInfo::= 44 mvc2::BufferStatus::getFreeDataBuffers 30 mvc2::CPUInfo::bool 44 mvc2::BufferStatus::getPendingFrames 31 mvc2::CPUInfo::CPUInfo 42 mvc2::BufferStatus::getReturnCode 31 mvc2::BufferStatus::INTERVAL 29 mvc2::CPUInfo::CPUTYPE 43 mvc2::BufferStatus::INTERVAL enumeration 29 mvc2::CPUInfo::CPUTYPE enumeration 43 mvc2::ColorConversionMatrix 31 mvc2::CPUInfo::getCoreFrequency 43 mvc2::ColorConversionMatrix::= 32 mvc2::CPUInfo::getCoreFrequencykHz 43 mvc2::CPUInfo::getCPUType 43 mvc2::ColorConversionMatrix::ColorConversionMatrix 32 mvc2::CPUInfo::getCPUTypeString 43 mvc2::ColorSpace 32 mvc2::ColorSpace::= 36 mvc2::CPUInfo::getHardwareVersion 43 mvc2::ColorSpace::ColorSpace 33, 34 mvc2::CPUInfo::getNumberOfCores 44 mvc2::ColorSpace::ColourPrimaries 34 mvc2::DataBuffer 44 mvc2::DataBuffer::~DataBuffer 46 mvc2::ColorSpace::ColourPrimaries enumeration 34 mvc2::ColorSpace::getColorSpacePacked 35 mvc2::DataBuffer::= 51 mvc2::ColorSpace::getSpace 35 mvc2::DataBuffer::bool 51 mvc2::ColorSpace::MatrixCoefficients 34 mvc2::DataBuffer::copy 46 mvc2::ColorSpace::MatrixCoefficients enumeration 34 mvc2::DataBuffer::DataBuffer 46 mvc2::ColorSpace::Space 35 mvc2::DataBuffer::getBufferAddress 47 mvc2::ColorSpace::Space enumeration 35 mvc2::DataBuffer::getFreeSize 47 mvc2::ColorSpace::TransferCharacteristics 35 mvc2::DataBuffer::getInSize 47 mvc2::ColorSpace::TransferCharacteristics enumeration 35 mvc2::DataBuffer::getTimeStamp 48 mvc2::ConfigAccess 36 mvc2::DataBuffer::send 48 mvc2::ConfigAccess::~ConfigAccess 37 mvc2::DataBuffer::setDecryptionSize 48 mvc2::ConfigAccess::= 40 mvc2::DataBuffer::setInSize 49 mvc2::ConfigAccess::bool 41 mvc2::DataBuffer::setKeyId 49 mvc2::ConfigAccess::ConfigAccess 38 mvc2::DataBuffer::setKeyIndex 49 mvc2::ConfigAccess::getLogFilter 38 mvc2::DataBuffer::setMicValue 50 mvc2::ConfigAccess::getLogFilterConsole 38 mvc2::DataBuffer::setTimeStamp 50 mvc2::ConfigAccess::getMessageFifoClients 38 mvc2::DataBuffer::setUserData 50 mvc2::ConfigAccess::getMessageFifoSize 39 mvc2::DataBuffer::wait 51 mvc2::ConfigAccess::getNetworkDHCP 39 mvc2::DecoderErrors 52 mvc2::ConfigAccess::getNetworkIPAddress 39 mvc2::DecoderErrors::~DecoderErrors 52 mvc2::ConfigAccess::getNetworkMask 39 mvc2::DecoderErrors::= 54 mvc2::ConfigAccess::setLogFilter 39 mvc2::DecoderErrors::bool 54 mvc2::ConfigAccess::setLogFilterConsole 39 mvc2::DecoderErrors::DecoderErrors 53 mvc2::ConfigAccess::setMessageFifoClients 40 mvc2::DecoderErrors::getReturnCode 53 mvc2::ConfigAccess::setMessageFifoSize 40 mvc2::DecoderStatus 54 mvc2::ConfigAccess::setNetworkDHCP 40 mvc2::DecoderStatus::~DecoderStatus 55 mvc2::ConfigAccess::setNetworkIPAddress 40 mvc2::DecoderStatus::= 57 mvc2::ConfigAccess::setNetworkMask 40 mvc2::DecoderStatus::bool 57 mvc2::CPUInfo 41 mvc2::DecoderStatus::DecoderStatus 55, 56 mvc2::CPUInfo::~CPUInfo 42 mvc2::DecoderStatus::getReturnCode 56

mvc2::DecoderStatus::getTimeStamp 56 mvc2::FramePosition::MODE 68 mvc2::DecoderStatus::getUserData 57 mvc2::FramePosition::MODE enumeration 68 mvc2::Dolby3Dchromaticity 57 mvc2::FramePosition::setModeX 70 mvc2::Dolby3Dchromaticity::~Dolby3Dchromaticity 58 mvc2::FramePosition::setModeY 71 mvc2::Dolby3Dchromaticity::Dolby3Dchromaticity 58 mvc2::FramePosition::setX 71 mvc2::Dolby3Dchromaticity::getChroma 59 mvc2::FramePosition::setY 71 mvc2::Dolby3Dchromaticity::getLuminance 59 mvc2::Jpeg2kDecoder 72 mvc2::Dolby3Dchromaticity::isCorrectionUsed 59 mvc2::Jpeg2kDecoder::~Jpeg2kDecoder 76 mvc2::Dolby3Dchromaticity::isWPointIterUsed 59 mvc2::Jpeg2kDecoder::= 80 mvc2::Dolby3Dchromaticity::Measurement 59 mvc2::Jpeg2kDecoder::CREATION\_FLAGS 77 mvc2::Dolby3Dchromaticity::Measurement enumeration 59 mvc2::Jpeg2kDecoder::CREATION\_FLAGS enumeration 77 mvc2::Dolby3Dchromaticity::setChroma 60 mvc2::Jpeg2kDecoder::getFrameRate 78 mvc2::Dolby3Dchromaticity::setLuminance 60 mvc2::Jpeg2kDecoder::getFrameRateTicks 78 mvc2::ExternalConnector 60 mvc2::Jpeg2kDecoder::Jpeg2kDecoder 76, 77 mvc2::ExternalConnector::~ExternalConnector 61 mvc2::Jpeg2kDecoder::MODE\_4K 77 mvc2::ExternalConnector::= 63 mvc2::Jpeg2kDecoder::MODE\_4K enumeration 77 mvc2::ExternalConnector::bool 63 mvc2::Jpeg2kDecoder::set4kMode 78 mvc2::ExternalConnector::ExternalConnector 61, 62 mvc2::Jpeg2kDecoder::setColorSpace 78 mvc2::ExternalConnector::m\_ConnectorPrivate 62 mvc2::Jpeg2kDecoder::setCutOffLevel 79 mvc2::ExternalConnector::readI2C 62 mvc2::Jpeg2kDecoder::setFrameRate 79 mvc2::ExternalConnector::writel2C 63 mvc2::Jpeg2kDecoder::setFrameRateTicks 79 mvc2::FeatureAccess 63 mvc2::Jpeg2kDecoder::setResolutionDivider 79 mvc2::FeatureAccess::~FeatureAccess 64 mvc2::LogAccess 80 mvc2::FeatureAccess::= 65 mvc2::LogAccess::~LogAccess 81 mvc2::FeatureAccess::bool 65 mvc2::LogAccess::= 83 mvc2::FeatureAccess::FeatureAccess 64 mvc2::LogAccess::abortGetMessage 82 mvc2::FeatureAccess::getNextFeature 65 mvc2::LogAccess::bool 83 mvc2::FeatureAccess::installFeature 65 mvc2::LogAccess::getMessage 82 mvc2::FeatureAccess::queryFeature 65 mvc2::LogAccess::LogAccess 81 mvc2::FramePosition 66 mvc2::LogAccess::setMessageFilter 83 mvc2::FramePosition::!= 71 mvc2::LogAccess::TIMEOUT 82 mvc2::FramePosition::= 72 mvc2::LogAccess::TIMEOUT enumeration 82 mvc2::FramePosition::== 72 mvc2::LogMessage 84 mvc2::FramePosition::fgetX 69 mvc2::LogMessage::~LogMessage 85 mvc2::FramePosition::fgetY 69 mvc2::LogMessage::= 90 mvc2::FramePosition::FramePosition 67, 68 mvc2::LogMessage::bool 91 mvc2::FramePosition::fsetX 69 mvc2::LogMessage::getCTime 88 mvc2::FramePosition::fsetY 69 mvc2::LogMessage::getId 88 mvc2::FramePosition::getModeX 70 mvc2::LogMessage::getMessageCounter 88 mvc2::FramePosition::getModeY 70 mvc2::LogMessage::getMessageString 89 mvc2::FramePosition::getX 70 mvc2::LogMessage::getMessageStringLength 89 mvc2::FramePosition::getY 70 mvc2::LogMessage::getMicroSeconds 89

mvc2::MvcDecoder::waitForUserData 102 mvc2::LogMessage::getSeverity 89 mvc2::LogMessage::getSubId 90 mvc2::MvcDevice 103 mvc2::MvcDevice::~MvcDevice 105 mvc2::LogMessage::getTimeStamp 90 mvc2::LogMessage::LogId 86 mvc2::MvcDevice::= 113 mvc2::LogMessage::LogId enumeration 86 mvc2::MvcDevice::bool 113 mvc2::MvcDevice::CLOCKID 106 mvc2::LogMessage::LogMessage 85, 86 mvc2::LogMessage::LogSubIdFirmware 87 mvc2::MvcDevice::CLOCKID enumeration 106 mvc2::LogMessage::LogSubIdFirmware enumeration 87 mvc2::MvcDevice::getAPIVersion 106 mvc2::LogMessage::setMessage 90 mvc2::MvcDevice::getBootloaderVersion 107 mvc2::LogMessage::Severity 87 mvc2::MvcDevice::getCPUInfo 107 mvc2::LogMessage::Severity enumeration 87 mvc2::MvcDevice::getDeviceState 107 mvc2::Mpeg2Decoder 91 mvc2::MvcDevice::getDriverVersion 108 mvc2::Mpeg2Decoder::~Mpeg2Decoder 94 mvc2::MvcDevice::getErrorDescription 108 mvc2::Mpeg2Decoder::= 96 mvc2::MvcDevice::getErrorString 108 mvc2::Mpeg2Decoder::Mpeg2Decoder 94, 95 mvc2::MvcDevice::getFirmwareVersion 108 mvc2::Mpeg2Decoder::PROCESSING\_3D 96 mvc2::MvcDevice::getLastError 108 mvc2::Mpeg2Decoder::PROCESSING\_3D enumeration 96 mvc2::MvcDevice::getNetworkConfiguration 108 mvc2::MvcDecoder 96 mvc2::MvcDevice::getPCIBus 109 mvc2::MvcDecoder::~MvcDecoder 98 mvc2::MvcDevice::getPCISlot 109 mvc2::MvcDecoder::= 103 mvc2::MvcDevice::getPowerDownStatus 109 mvc2::MvcDecoder::bool 103 mvc2::MvcDevice::getProductCode 109 mvc2::MvcDecoder::DecoderType 98 mvc2::MvcDevice::getProductRevision 110 mvc2::MvcDecoder::DecoderType enumeration 98 mvc2::MvcDevice::getSecurityBootloaderVersion 110 mvc2::MvcDecoder::flush 99 mvc2::MvcDevice::getSecurityManagerHwStatus 110 mvc2::MvcDecoder::getBufferStatus 99 mvc2::MvcDevice::getSecurityManagerVersion 110 mvc2::MvcDecoder::getChannel 99 mvc2::MvcDevice::getSystemPosixTime 110 mvc2::MvcDecoder::getDataBuffer 100 mvc2::MvcDevice::getSystemStatus 111 mvc2::MvcDecoder::getDecoderErrors 100 mvc2::MvcDevice::getUID 111 mvc2::MvcDecoder::getDecoderStatus 100 mvc2::MvcDevice::getUptimeMs 111 mvc2::MvcDecoder::getDecoderType 101 mvc2::MvcDevice::MvcDevice 105 mvc2::MvcDevice::OUTPUTMODE 106 mvc2::MvcDecoder::getPrivate 101 mvc2::MvcDecoder::MvcDecoder 98 mvc2::MvcDevice::OUTPUTMODE enumeration 106 mvc2::MvcDecoder::setBackwardPlayback 101 mvc2::MvcDevice::PRODUCTCODE 106 mvc2::MvcDecoder::setCplUid 101 mvc2::MvcDevice::PRODUCTCODE enumeration 106 mvc2::MvcDecoder::setEndOfStream 101 mvc2::MvcDevice::putMessage 111 mvc2::MvcDecoder::setSecurityManager 101 mvc2::MvcDevice::resetCard 112 mvc2::MvcDecoder::setStartDelay 102 mvc2::MvcDevice::setAuthenticationPassword 112 mvc2::MvcDecoder::TIMEOUT 99 mvc2::MvcDevice::setOutputMode 112 mvc2::MvcDecoder::TIMEOUT enumeration 99 mvc2::MvcDevice::setPowerDownTime 112 mvc2::MvcDecoder::waitDecoderStatus 102 mvc2::MvcDevice::setStatusLed 112 mvc2::MvcDecoder::waitForTimeStamp 102 mvc2::MvcDeviceIterator 113 mvc2::MvcDecoder::waitForTransferFinish 102 mvc2::MvcDeviceIterator::~MvcDeviceIterator 114

i

mvc2::OverlayElementDataBuffer::OverlayElementDataBuffer mvc2::MvcDeviceIterator::getFirst 114 mvc2::MvcDeviceIterator::getIndex 115 mvc2::OverlayElementDataBuffer::setElementName 133 mvc2::MvcDeviceIterator::getNext 115 mvc2::OverlayElementDataBuffer::setPartialTransferInfo 133 mvc2::MvcDeviceIterator::MvcDeviceIterator 114 mvc2::OverlayRenderCommand 133 mvc2::MvcNetDeviceIterator 115 mvc2::OverlayRenderCommand::~OverlayRenderCommand mvc2::MvcNetDeviceIterator::~MvcNetDeviceIterator 116 135 mvc2::MvcNetDeviceIterator::getFirst 117 mvc2::OverlayRenderCommand::clearRenderArea 136 mvc2::MvcNetDeviceIterator::getIndex 117 mvc2::OverlayRenderCommand::m\_clearColor 135 mvc2::MvcNetDeviceIterator::getNext 117 mvc2::OverlayRenderCommand::m\_flags 135 mvc2::MvcNetDeviceIterator::MvcNetDeviceIterator 116 mvc2::OverlayRenderCommand::m\_reserved1 135 mvc2::NetworkInterfaceInfo 117 mvc2::OverlayRenderCommand::m\_reserved2 136 mvc2::NetworkInterfaceInfo::= 119 mvc2::OverlayRenderCommand::OverlayRenderCommand mvc2::NetworkInterfaceInfo::bool 119 mvc2::OverlayRenderCommand::renderIntoBuffer 136 mvc2::NetworkInterfaceInfo::getIPAddress 118, 119 mvc2::NetworkInterfaceInfo::getMACAddress 119 mvc2::OverlaySubtitleDataBuffer 136 mvc2::OverlaySubtitleDataBuffer::getResourceId 139 mvc2::NetworkInterfaceInfo::NetworkInterfaceInfo 118 mvc2::OverlaySubtitleDataBuffer::OverlaySubtitleDataBuffer mvc2::OverlayDataBuffer 120 138, 139 mvc2::OverlayDataBuffer::addRenderCommand 123 mvc2::OverlaySubtitleDataBuffer::setTimeValues 139 mvc2::OverlayDataBuffer::clearScreen 124 mvc2::PCMDataBuffer 139 mvc2::OverlayDataBuffer::EFFECTSPEED 123 mvc2::PCMDataBuffer::~PCMDataBuffer 142 mvc2::OverlayDataBuffer::EFFECTSPEED enumeration 123 mvc2::PCMDataBuffer::PCMDataBuffer 142 mvc2::OverlayDataBuffer::MOVEEFFECT 123 mvc2::PCMDataBuffer::setBitsPerSample 142 mvc2::OverlayDataBuffer::MOVEEFFECT enumeration 123 mvc2::PCMDataBuffer::setChannelMapping 142 mvc2::OverlayDataBuffer::OverlayDataBuffer 122 mvc2::PCMDataBuffer::setMixingChannel 142 mvc2::OverlayDataBuffer::setDisplayDuration 124 mvc2::PCMDataBuffer::setNumberOfChannels 143 mvc2::OverlayDataBuffer::setGlobalAlpha 124 mvc2::PCMDataBuffer::setSampleFrequency 143 mvc2::OverlayDataBuffer::setGlobalPosition 124 mvc2::PCMDecoder 143 mvc2::OverlayDataBuffer::setIncompleteRender 124 mvc2::PCMDecoder::~PCMDecoder 146 mvc2::OverlayDataBuffer::setSwapBuffer 124 mvc2::PCMDecoder::= 151 mvc2::OverlayDecoder 125 mvc2::PCMDecoder::CHANNEL\_MAPPING\_FLAGS 147 mvc2::OverlayDecoder::~OverlayDecoder 127 mvc2::PCMDecoder::connectOutput 148 mvc2::OverlayDecoder::= 130 mvc2::PCMDecoder::disconnectOutput 148 mvc2::OverlayDecoder::connectOutput 128 mvc2::PCMDecoder::getBitsPerSample 148 mvc2::OverlayDecoder::deleteOverlayElement 129 mvc2::PCMDecoder::getDecoderErrors 149 mvc2::OverlayDecoder::disconnectOutput 129 mvc2::PCMDecoder::getDecoderStatus 149 mvc2::OverlayDecoder::getDataBuffer 129, 130 mvc2::PCMDecoder::getNumberOfChannels 149 mvc2::OverlayDecoder::OverlayDecoder 127, 128 mvc2::PCMDecoder::getPCMDataBuffer 149 mvc2::OverlayDecoder::RESOLUTION\_FLAGS 128 mvc2::PCMDecoder::getSampleFrequency 149 mvc2::OverlayDecoder::RESOLUTION\_FLAGS enumeration mvc2::PCMDecoder::PCMDecoder 146, 147 128 mvc2::PCMDecoder::setBitsPerSample 150 mvc2::OverlayDecoder::setOutputResolution 130 mvc2::PCMDecoder::setChannelMapping 150 mvc2::OverlayElementDataBuffer 130 mvc2::PCMDecoder::setDolbyPrologicIIChannels 150 mvc2::OverlayElementDataBuffer::getElementName 133

mvc2::PCMDecoder::setMixingChannel 150 mvc2::RenderText::~RenderText 168 mvc2::PCMDecoder::setNumberOfChannels 150 mvc2::RenderText::renderIntoBuffer 168 mvc2::RenderText::RenderText 168 mvc2::PCMDecoder::setSampleFrequency 150 mvc2::PCMDecoder::waitDecoderStatus 151 mvc2::RenderText::setColor 169 mvc2::PlaybackControl 151 mvc2::RenderText::setFont 169 mvc2::RenderText::setPosition 169 mvc2::PlaybackControl::~PlaybackControl 152 mvc2::PlaybackControl::= 158 mvc2::RenderText::TextFlags 168 mvc2::PlaybackControl::bool 158 mvc2::RenderText::TextFlags enumeration 168 mvc2::PlaybackControl::connect 154 mvc2::SecurityAccess 169 mvc2::PlaybackControl::disconnect 154 mvc2::SecurityAccess::\_CineLink2Para 174 mvc2::PlaybackControl::flush 155 mvc2::SecurityAccess::\_CineLink2Para structure 174 mvc2::SecurityAccess::\_HashAlgorithm 174 mvc2::PlaybackControl::getState 155 mvc2::PlaybackControl::pause 155 mvc2::SecurityAccess::\_HashAlgorithm enumeration 174 mvc2::PlaybackControl::PLAYBACK\_STATE 153 mvc2::SecurityAccess::\_MediaType 174 mvc2::PlaybackControl::PLAYBACK\_STATE enumeration 153 mvc2::SecurityAccess::\_MediaType enumeration 174 mvc2::PlaybackControl::PlaybackControl 152, 153 mvc2::SecurityAccess::\_SoftwareId 175 mvc2::PlaybackControl::run 155 mvc2::SecurityAccess::\_SoftwareId enumeration 175 mvc2::PlaybackControl::runSpeed 156 mvc2::SecurityAccess::\_SymmetricCryptoOperation 175 mvc2::SecurityAccess::\_SymmetricCryptoOperation mvc2::PlaybackControl::setSyncSlave 156 enumeration 175 mvc2::PlaybackControl::singleStep 156 mvc2::SecurityAccess::~SecurityAccess 173 mvc2::PlaybackControl::stop 157 mvc2::SecurityAccess::= 188 mvc2::PlaybackControl::waitForEndOfStream 157 mvc2::SecurityAccess::bool 189 mvc2::ProjectorAccess 160 mvc2::SecurityAccess::deleteCertificateChain 176 mvc2::ProjectorAccess::~ProjectorAccess 161 mvc2::SecurityAccess::disableForensicMarking 176 mvc2::ProjectorAccess::= 163 mvc2::SecurityAccess::disableLLE 177 mvc2::ProjectorAccess::bool 163 mvc2::SecurityAccess::enableCinelink 177 mvc2::ProjectorAccess::commandlo 162 mvc2::SecurityAccess::enableCinelink2 178 mvc2::ProjectorAccess::login 163 mvc2::SecurityAccess::enableForensicMarking 178, 179 mvc2::ProjectorAccess::logout 163 mvc2::SecurityAccess::getAllHardwareInfo 179 mvc2::ProjectorAccess::ProjectorAccess 161, 162 mvc2::SecurityAccess::getAppletStatusCode 180 mvc2::RenderFill 158 mvc2::SecurityAccess::getAppletVersion 180 mvc2::RenderFill::~RenderFill 159 mvc2::SecurityAccess::getBuildTimeStrings 180 mvc2::RenderFill::RenderFill 160 mvc2::SecurityAccess::getCertificateSigningRequest 180 mvc2::RenderFill::renderIntoBuffer 160 mvc2::SecurityAccess::getErrorCode 180 mvc2::RenderFill::setBox 160 mvc2::SecurityAccess::getFipsFirmwareVersion 181 mvc2::RenderFill::setColor 160 mvc2::SecurityAccess::getFirmwareVersion 181 mvc2::RenderPicture 164 mvc2::SecurityAccess::getForensicMarkingId 181 mvc2::RenderPicture::~RenderPicture 165 mvc2::SecurityAccess::getHashValue 181 mvc2::RenderPicture::renderIntoBuffer 165 mvc2::SecurityAccess::getMikromPublicKey 182 mvc2::RenderPicture::RenderPicture 165 mvc2::SecurityAccess::getPublicKey 182 mvc2::RenderPicture::setPosition 166 mvc2::SecurityAccess::getRtcHardwareStatus 182 mvc2::RenderText 166 mvc2::SecurityAccess::getSelftestStatus 182

mvc2::SecurityAccess::getTamperStatus 183 mvc2::SecurityManager::getKeyMap 195 mvc2::SecurityAccess::hashRtcSram 183 mvc2::SecurityManager::getLogReport 195 mvc2::SecurityAccess::initHardwareInfo 183 mvc2::SecurityManager::initiateMarriage 195 mvc2::SecurityAccess::initLogFlash 183 mvc2::SecurityManager::KeyMap 199 mvc2::SecurityAccess::installCertificate 183 mvc2::SecurityManager::loadCertificateChainFile 195 mvc2::SecurityAccess::processSmartcardApdu 183 mvc2::SecurityManager::loadPrivateKeyFile 195 mvc2::SecurityAccess::readLogFlashAttributes 184 mvc2::SecurityManager::playShow 195 mvc2::SecurityAccess::readRtcReg 184 mvc2::SecurityManager::purgeCpl 196 mvc2::SecurityAccess::readRtcSram 184 mvc2::SecurityManager::queryStatus 196 mvc2::SecurityAccess::readSecurityInterface 184 mvc2::SecurityManager::SecurityManager 200 mvc2::SecurityAccess::resetPowerfailBit 185 mvc2::SecurityManager::serviceDoorTamperTermination 196 mvc2::SecurityAccess::rsaPrivateKeyDecCmp 185 mvc2::SecurityManager::SM\_OPERATION 192 mvc2::SecurityManager::SM\_OPERATION enumeration 192 mvc2::SecurityAccess::SecurityAccess 173 mvc2::SecurityAccess::selfTest 185 mvc2::SecurityManager::SpblpAddress 199 mvc2::SecurityAccess::setCineLink2Keys 185 mvc2::SecurityManager::startSuite 196, 197 mvc2::SecurityAccess::setForensicMarkingId 185 mvc2::SecurityManager::stopShow 197 mvc2::SecurityAccess::setForensicMarkingIndicator 185 mvc2::SecurityManager::stopSuite 197 mvc2::SecurityAccess::setHardwareInfo 186 mvc2::SecurityManager::SUITE\_STATUS 193 mvc2::SecurityAccess::setHashData 186 mvc2::SecurityManager::SUITE\_STATUS enumeration 193 mvc2::SecurityAccess::setMediaDecryptorKey 186 mvc2::SecurityManager::uploadCplBuffer 197 mvc2::SecurityAccess::symmetricCryptoFinish 187 mvc2::SecurityManager::uploadCplFile 198 mvc2::SecurityManager::uploadKdmBuffer 198 mvc2::SecurityAccess::symmetricCryptoInit 187 mvc2::SecurityAccess::symmetricCryptoProcess 187 mvc2::SecurityManager::uploadKdmFile 198 mvc2::SecurityAccess::validateHardwareInfo 188 mvc2::SecurityManager::usePrivateKey 198 mvc2::SecurityAccess::verify 188 mvc2::SecurityManager::zeroize 198 mvc2::SecurityAccess::writeRtcReg 188 mvc2::SubtitleDecoder 200 mvc2::SecurityAccess::writeSecurityInterface 188 mvc2::SubtitleDecoder::~SubtitleDecoder 203 mvc2::SecurityManager 189 mvc2::SubtitleDecoder::= 207 mvc2::SecurityManager::~SecurityManager 200 mvc2::SubtitleDecoder::connectOutput 204 mvc2::SecurityManager::= 199 mvc2::SubtitleDecoder::disableSubtitles 205 mvc2::SecurityManager::addCertificate 193 mvc2::SubtitleDecoder::disconnectOutput 205 mvc2::SecurityManager::adjustTime 194 mvc2::SubtitleDecoder::enableSubtitles 205 mvc2::SecurityManager::AesKeyMap 199 mvc2::SubtitleDecoder::getDataBuffer 206 mvc2::SecurityManager::bool 200 mvc2::SubtitleDecoder::getOverlayElementRequest 206 mvc2::SecurityManager::c\_ProjectorCertificate 192 mvc2::SubtitleDecoder::RENDERFLAGS 204 mvc2::SecurityManager::checkShow 194 mvc2::SubtitleDecoder::RENDERFLAGS enumeration 204 mvc2::SecurityManager::connect 194 mvc2::SubtitleDecoder::sendOverlayElement 206 mvc2::SecurityManager::CPL\_ASSET\_EXCEPTION 192 mvc2::SubtitleDecoder::sendSubtitleFile 206 mvc2::SecurityManager::CPL\_ASSET\_EXCEPTION mvc2::SubtitleDecoder::setOutputResolution 206 enumeration 192 mvc2::SubtitleDecoder::setZOffset 207 mvc2::SecurityManager::getCertificate 194 mvc2::SubtitleDecoder::SubtitleDecoder 203, 204 mvc2::SecurityManager::getCplList 194 mvc2::SystemStatus 207 mvc2::SecurityManager::getKdmList 195

Ī

mvc2::SystemStatus::~SystemStatus 208 mvc2::VersionValue::= 225 mvc2::SystemStatus::= 211 mvc2::VersionValue::bool 225 mvc2::VersionValue::getBuildRevision 224 mvc2::SystemStatus::bool 211 mvc2::SystemStatus::CPULOAD 209 mvc2::VersionValue::getBuildVersion 224 mvc2::SystemStatus::CPULOAD enumeration 209 mvc2::VersionValue::getRevision 224 mvc2::SystemStatus::getBoardTemperature 209 mvc2::VersionValue::getVersion 224 mvc2::SystemStatus::getCPULoad 209 mvc2::VersionValue::VersionValue 223 mvc2::SystemStatus::getDecoderTemperature 210 mvc2::VideoDataBuffer 225 mvc2::SystemStatus::getFanSpeed 210 mvc2::VideoDataBuffer::~VideoDataBuffer 227 mvc2::SystemStatus::getFreeMemory 210 mvc2::VideoDataBuffer::setTimeCode 227 mvc2::SystemStatus::getMaxMemory 210 mvc2::VideoDataBuffer::VideoDataBuffer 227 mvc2::SystemStatus::getOutputTemperature 211 mvc2::VideoDecoder 228 mvc2::SystemStatus::getUsedMemory 211 mvc2::VideoDecoder::calcDolby3DMatrix 231 mvc2::SystemStatus::SystemStatus 209 mvc2::VideoDecoder::connectOutput 231 mvc2::TamperStatus 219 mvc2::VideoDecoder::disconnectOutput 231 mvc2::TamperStatus::getSwitchStatus 221 mvc2::VideoDecoder::getDecoderErrors 232 mvc2::TamperStatus::getTamperEvent 221 mvc2::VideoDecoder::getDecoderStatus 232 mvc2::TamperStatus::TAMPER\_EVENTS 220 mvc2::VideoDecoder::getVideoDataBuffer 232 mvc2::TamperStatus::TAMPER\_EVENTS enumeration 220 mvc2::VideoDecoder::setCloneVideo 232 mvc2::TamperStatus::TAMPER\_SWITCHES 221 mvc2::VideoDecoder::setDolby3DChroma 232 mvc2::TamperStatus::TAMPER\_SWITCHES enumeration 221 mvc2::VideoDecoder::setDolby3DMatrix 232 mvc2::TamperStatus::TamperStatus 220 mvc2::VideoDecoder::VideoDecoder 230 mvc2::TimeCode 211 mvc2::VideoDecoder::waitDecoderStatus 233 mvc2::TimeCode::= 215 mvc2::VideoDecoderErrors 233 mvc2::TimeCode::getFrames 214 mvc2::VideoDecoderErrors::~VideoDecoderErrors 235 mvc2::TimeCode::getHours 214 mvc2::VideoDecoderErrors::= 237 mvc2::TimeCode::getMinutes 214 mvc2::VideoDecoderErrors::getInputUnderruns 236 mvc2::TimeCode::getSeconds 214 mvc2::VideoDecoderErrors::getPartialFrames 236 mvc2::TimeCode::getTimeCodePacked 215 mvc2::VideoDecoderErrors::getPlayedFramesError 236 mvc2::TimeCode::isDropFrame 215 mvc2::VideoDecoderErrors::getPlayedFramesOk 236 mvc2::TimeCode::TimeCode 212, 213 mvc2::VideoDecoderErrors::getRepeatedFramesAVSync 236 mvc2::UuidValue 216 mvc2::VideoDecoderErrors::getRepeatedFramesUnderrun 236 mvc2::UuidValue::!= 218 mvc2::VideoDecoderErrors::getSkippedFramesAVSync 236 mvc2::UuidValue::[] 219 mvc2::VideoDecoderErrors::getSkippedFramesError 237 mvc2::UuidValue::= 219 mvc2::VideoDecoderErrors::VideoDecoderErrors 235 mvc2::UuidValue::== 219 mvc2::VideoDecoderStatus 237 mvc2::UuidValue::bool 219 mvc2::VideoDecoderStatus::= 240 mvc2::UuidValue::m\_key 218 mvc2::VideoDecoderStatus::getPictureHeight 240 mvc2::UuidValue::toArray 218 mvc2::VideoDecoderStatus::getPictureWidth 240 mvc2::UuidValue::toString 218 mvc2::VideoDecoderStatus::getTimeCode 240 mvc2::UuidValue::UuidValue 217, 218 mvc2::VideoDecoderStatus::VideoDecoderStatus 239 mvc2::VersionValue 222 mvc2::VideoMode 240

getDataBuffer 100 mvc2::VideoMode::= 243 mvc2::VideoMode::getMode 242 getDecoderErrors 100 mvc2::VideoMode::Mode 241 getDecoderStatus 100 mvc2::VideoMode::Mode enumeration 241 getDecoderType 101 mvc2::VideoMode::VideoMode 241 getPrivate 101 mvc2::VideoOutput 243 MvcDecoder 98 mvc2::VideoOutput::~VideoOutput 244 setBackwardPlayback 101 setCplUid 101 mvc2::VideoOutput::= 250 setEndOfStream 101 mvc2::VideoOutput::bool 251 mvc2::VideoOutput::disableDeGhosting 248 setSecurityManager 101 mvc2::VideoOutput::EFFECTSPEED 246 setStartDelay 102 mvc2::VideoOutput::EFFECTSPEED enumeration 246 waitDecoderStatus 102 waitForTimeStamp 102 mvc2::VideoOutput::enableDeGhosting 248 mvc2::VideoOutput::getActiveVideoMode 248 waitForTransferFinish 102 mvc2::VideoOutput::MOVEEFFECT 246 waitForUserData 102 mvc2::VideoOutput::MOVEEFFECT enumeration 246 MvcDevice class 103 mvc2::VideoOutput::OUTPUTACTIVITY 247 ~MvcDevice 105 mvc2::VideoOutput::OUTPUTACTIVITY enumeration 247 = 113 mvc2::VideoOutput::setAudioEmbedding 248 about MvcDevice class 103 mvc2::VideoOutput::setOutputActivity 249 **bool** 113 mvc2::VideoOutput::setPosition 249 getAPIVersion 106 mvc2::VideoOutput::setVideoMode 250 getBootloaderVersion 107 mvc2::VideoOutput::VideoOutput 244, 245 getCPUInfo 107 getDeviceState 107 mvc2::VideoOutput::VIDEOPROPERTY 247 mvc2::VideoOutput::VIDEOPROPERTY enumeration 247 getDriverVersion 108 mvc2::VSCConnector 251 getErrorDescription 108 mvc2::VSCConnector::~VSCConnector 253 getErrorString 108 mvc2::VSCConnector::abortGPIOInterrupt 254 getFirmwareVersion 108 mvc2::VSCConnector::getGPIOValue 254 getLastError 108 mvc2::VSCConnector::setGPIODirection 254 getNetworkConfiguration 108 mvc2::VSCConnector::setGPIOInterrupt 254 getPCIBus 109 mvc2::VSCConnector::setGPIOValue 254 getPCISlot 109 mvc2::VSCConnector::VSCConnector 253 getPowerDownStatus 109 mvc2::VSCConnector::waitForGPIOInterrupt 254 getProductCode 109 MvcDecoder class 96 getProductRevision 110 ~MvcDecoder 98 getSecurityBootloaderVersion 110 = 103getSecurityManagerHwStatus 110 about MvcDecoder class 96 getSecurityManagerVersion 110 **bool** 103 getSystemPosixTime 110 flush 99 getSystemStatus 111 getUID 111 getBufferStatus 99 getChannel 99 getUptimeMs 111

,	MVC20x API
MvcDevice 105	OverlayDataBuffer 122
putMessage 111	setDisplayDuration 124
resetCard 112	setGlobalAlpha 124
setAuthenticationPassword 112	setGlobalPosition 124
setOutputMode 112	setIncompleteRender 124
setPowerDownTime 112	setSwapBuffer 124
setStatusLed 112	OverlayDecoder class 125
MvcDeviceIterator class 113	~OverlayDecoder 127
~MvcDeviceIterator 114	= 130
about MvcDeviceIterator class 113	about OverlayDecoder class 125
getFirst 114	connectOutput 128
getIndex 115	deleteOverlayElement 129
getNext 115	disconnectOutput 129
MvcDeviceIterator 114	getDataBuffer 129, 130
MvcNetDeviceIterator class 115	OverlayDecoder 127, 128
~MvcNetDeviceIterator 116	setOutputResolution 130
about MvcNetDeviceIterator class 115	OverlayElementDataBuffer class 130
getFirst 117	about OverlayElementDataBuffer class 130
getIndex 117	class OverlayDecoder 133
getNext 117	class SubtitleDecoder 133
MvcNetDeviceIterator 116	getElementName 133
	OverlayElementDataBuffer 132
N	setElementName 133
NetworkInterfaceInfo class 117	setPartialTransferInfo 133
= 119	OverlayRenderCommand class 133
about NetworkInterfaceInfo class 117	~OverlayRenderCommand 135
bool 119	about OverlayRenderCommand class 133
getIPAddress 118, 119	class OverlayDataBuffer 136
getMACAddress 119	clearRenderArea 136
NetworkInterfaceInfo 118	m_clearColor 135
TVOLVOTAINO TAGO	m_flags 135
0	m_reserved1 135
	m_reserved2 136
Operation_NotPlaying enumeration member 192	OverlayRenderCommand 135
Operation_Playing enumeration member 192	renderIntoBuffer 136
Operation_Unknown enumeration member 192	OverlaySubtitleDataBuffer class 136
OutputMode_DVI enumeration member 106	about OverlaySubtitleDataBuffer class 136
OutputMode_SDI enumeration member 106	class SubtitleDecoder 139
OverlayDataBuffer class 120	getResourceld 139
about OverlayDataBuffer class 120	OverlaySubtitleDataBuffer 138, 139
addRenderCommand 123	setTimeValues 139
class OverlayDecoder 125	Overview 1
clearScreen 124	

	getState 155
P	pause 155
	PlaybackControl 152, 153
Paused enumeration member 153	run 155
PCMDataBuffer class 139	runSpeed 156
~PCMDataBuffer 142	setSyncSlave 156
about PCMDataBuffer class 139	singleStep 156
class PCMDecoder 143	stop 157
PCMDataBuffer 142	waitForEndOfStream 157
setBitsPerSample 142	Prim_Default enumeration member 34
setChannelMapping 142	Prim_ITUR_BT_470_BG enumeration member 34
setMixingChannel 142	Prim_ITUR_BT_470_M enumeration member 34
setNumberOfChannels 143	Prim_ITUR_BT_709 enumeration member 34
setSampleFrequency 143	Prim_P3 enumeration member 34
PCMDecoder class 143	Prim_SMPTE_170M enumeration member 34
~PCMDecoder 146	Prim_SMPTE_240M enumeration member 34
= 151	Prim_sRGB enumeration member 34
about PCMDecoder class 143	Prim_XYZ enumeration member 34
CHANNEL_MAPPING_FLAGS 147	Processing_None enumeration member 96
connectOutput 148	Processing_SideBySide enumeration member 96
disconnectOutput 148	Processing_TopBottom enumeration member 96
getBitsPerSample 148	ProductCode_MVC200DC enumeration member 106
getDecoderErrors 149	ProductCode_MVC201 enumeration member 106
getDecoderStatus 149	ProjectorAccess class 160
getNumberOfChannels 149	~ProjectorAccess 161
getPCMDataBuffer 149	= 163
getSampleFrequency 149	about ProjectorAccess class 160
PCMDecoder 146, 147	bool 163
setBitsPerSample 150	commandlo 162
setChannelMapping 150	login 163
setDolbyPrologicIIChannels 150	logout 163
setMixingChannel 150	ProjectorAccess 161, 162
setNumberOfChannels 150	.,
setSampleFrequency 150	R
waitDecoderStatus 151	
PlaybackControl class 151	Render_3D_Mode enumeration member 204
~PlaybackControl 152	Render_BaseResolution_2K enumeration member 204
= 158	Render_BaseResolution_4K enumeration member 204
about PlaybackControl class 151	Render_Force_Border enumeration member 204
bool 158	Render_Force_Shadow enumeration member 204
connect 154	Render_Soft_Shadows enumeration member 204
disconnect 154	RenderFill class 158
flush 155	~RenderFill 159

3	MVC20x API
about RenderFill class 158	getAppletVersion 180
RenderFill 160	getBuildTimeStrings 180
renderIntoBuffer 160	getCertificateSigningRequest 180
setBox 160	getErrorCode 180
setColor 160	getFipsFirmwareVersion 181
RenderPicture class 164	getFirmwareVersion 181
~RenderPicture 165	getForensicMarkingId 181
about RenderPicture class 164	getHashValue 181
renderIntoBuffer 165	getMikromPublicKey 182
RenderPicture 165	getPublicKey 182
setPosition 166	getRtcHardwareStatus 182
RenderText class 166	getSelftestStatus 182
~RenderText 168	getTamperStatus 183
about RenderText class 166	hashRtcSram 183
renderIntoBuffer 168	initHardwareInfo 183
RenderText 168	initLogFlash 183
setColor 169	installCertificate 183
setFont 169	processSmartcardApdu 183
setPosition 169	readLogFlashAttributes 184
Resolution_100ms enumeration member 25	readRtcReg 184
Resolution_1s enumeration member 25	readRtcSram 184
Resolution_Absolute enumeration member 128	readSecurityInterface 184
Resolution_Fix_Height enumeration member 128	resetPowerfailBit 185
Resolution_Fix_Width enumeration member 128	rsaPrivateKeyDecCmp 185
Resolution_Full_Screen enumeration member 128	SecurityAccess 173
Resolution_Relative enumeration member 128	selfTest 185
Running enumeration member 153	setCineLink2Keys 185
	setForensicMarkingId 185
S	setForensicMarkingIndicator 185
SecurityAccess class 169	setHardwareInfo 186
~SecurityAccess 173	setHashData 186
= 188	setMediaDecryptorKey 186
about SecurityAccess class 169	symmetricCryptoFinish 187
bool 189	symmetricCryptoInit 187
deleteCertificateChain 176	symmetricCryptoProcess 187
disableForensicMarking 176	validateHardwareInfo 188
disableLLE 177	verify 188
enableCinelink 177	writeRtcReg 188
enableCinelink 177	writeSecurityInterface 188
	SecurityManager class 189
enableForensicMarking 178, 179	~SecurityManager 200
getAnnletStatusCode 190	= 199
getAppletStatusCode 180	

about SecurityManager class 189 Space\_Max enumeration member 35 addCertificate 193 Space\_RGB enumeration member 35 Space\_XYZ enumeration member 35 adjustTime 194 AesKeyMap 199 Space\_YCbCr enumeration member 35 bool 200 Space\_YCbCr\_P3 enumeration member 35 c\_ProjectorCertificate 192 Space\_YCxCz enumeration member 35 checkShow 194 Speed\_ExtremelyFast enumeration member 123, 246 class MvcDecoder 193 Speed\_Fast enumeration member 123, 246 connect 194 Speed\_Immediately enumeration member 123, 246 getCertificate 194 Speed\_Normal enumeration member 123, 246 getCplList 194 Speed\_Slow enumeration member 123, 246 getKdmList 195 Speed\_VeryFast enumeration member 123, 246 getKeyMap 195 Speed\_VerySlow enumeration member 123, 246 getLogReport 195 Stopped enumeration member 153 initiateMarriage 195 SubIdAudio enumeration member 87 KeyMap 199 SubIdControl enumeration member 87 loadCertificateChainFile 195 SubIdDecoding enumeration member 87 loadPrivateKeyFile 195 SubIdNetwork enumeration member 87 playShow 195 SubIdOutput enumeration member 87 purgeCpl 196 SubIdSecurity enumeration member 87 queryStatus 196 SubIdVideo enumeration member 87 SecurityManager 200 SubtitleDecoder class 200 ~SubtitleDecoder 203 serviceDoorTamperTermination 196 = 207 SpblpAddress 199 startSuite 196, 197 about SubtitleDecoder class 200 stopShow 197 connectOutput 204 stopSuite 197 disableSubtitles 205 uploadCplBuffer 197 disconnectOutput 205 enableSubtitles 205 uploadCplFile 198 uploadKdmBuffer 198 getDataBuffer 206 uploadKdmFile 198 getOverlayElementRequest 206 usePrivateKey 198 sendOverlayElement 206 zeroize 198 sendSubtitleFile 206 SeverityError enumeration member 87 setOutputResolution 206 SeverityInfo enumeration member 87 setZOffset 207 SeverityNone enumeration member 87 SubtitleDecoder 203, 204 SeverityTrace enumeration member 87 Suite Error enumeration member 193 SeverityWarning enumeration member 87 Suite\_Lost enumeration member 193 Sha1 enumeration member 174 Suite\_Progress enumeration member 193 Sha1WithZlib enumeration member 174 Suite\_Restarted enumeration member 193 Sha256 enumeration member 174 Suite\_Started enumeration member 193 Space\_Default enumeration member 35 Suite\_Stopped enumeration member 193

3 MVC20x API

SW ID MainBootloader enumeration member 175 about TimeCode class 211

SW_ID_MainBootloader enumeration member 175	about TimeCode class 211
SW_ID_MainFirmware enumeration member 175	getFrames 214
SW_ID_MainFPGA enumeration member 175	getHours 214
SW_ID_SecApplication enumeration member 175	getMinutes 214
SW_ID_SecBootloader enumeration member 175	getSeconds 214
SWITCH_DOOR_1 enumeration member 221	getTimeCodePacked 215
SWITCH_DOOR_2 enumeration member 221	isDropFrame 215
SWITCH_MARRIAGE_1 enumeration member 221	TimeCode 212, 213
SWITCH_MARRIAGE_2 enumeration member 221	Trans_DCI enumeration member 35
SWITCH_TAMPER_1 enumeration member 221	Trans_Default enumeration member 35
SWITCH_TAMPER_2 enumeration member 221	Trans_ITUR_BT_1361 enumeration member 35
SystemStatus class 207	Trans_ITUR_BT_470_BG enumeration member 35
~SystemStatus 208	Trans_ITUR_BT_470_M enumeration member 35
= 211	Trans_ITUR_BT_709 enumeration member 35
about SystemStatus class 207	Trans_Linear enumeration member 35
bool 211	Trans_SMPTE_170M enumeration member 35
getBoardTemperature 209	Trans_SMPTE_240M enumeration member 35
getCPULoad 209	Trans_sRGB enumeration member 35
getDecoderTemperature 210	TrippleDes_cbc_dec enumeration member 175
getFanSpeed 210	TrippleDes_cbc_enc enumeration member 175
getFreeMemory 210	Type_Jpeg2kDecoder enumeration member 98
getMaxMemory 210	Type_Mpeg2Decoder enumeration member 98
getOutputTemperature 211	Type_OverlayDecoder enumeration member 98
getUsedMemory 211	Type_PCMDecoder enumeration member 98
SystemStatus 209	Type_SubtitleDecoder enumeration member 98
	Type_TestDecoder enumeration member 98
Т	Type_UnknownDecoder enumeration member 98
TamperStatus class 219	
about TamperStatus class 219	U
class SecurityAccess 222	UuidValue class 216
getSwitchStatus 221	!= 218
getTamperEvent 221	[] 219
TamperStatus 220	= 219
TextFlag_Bold enumeration member 168	== 219
TextFlag_Border enumeration member 168	about UuidValue class 216
TextFlag_Italic enumeration member 168	bool 219
TextFlag_Shadow enumeration member 168	m_key 218
TextFlag_SoftShadow enumeration member 168	toArray 218
TextFlag_Strikethrough enumeration member 168	toString 218
TextFlag_Underline enumeration member 168	UuidValue 217, 218
TimeCode class 211	
= 215	

V	VideoDecoderStatus class 237
V : V   1 000	= 240
VersionValue class 222	about VideoDecoderStatus class 237
= 225	getPictureHeight 240
about VersionValue class 222	getPictureWidth 240
bool 225	getTimeCode 240
about VersionValue class 222 bool 225 getBuildRevision 224 getBuildVersion 224	VideoDecoderStatus 239
getBuildVersion 224	VideoMode class 240
getrevision 224	= 243
getVersion 224	about VideoMode class 240
VersionValue 223	getMode 242
VideoDataBuffer class 225	VideoMode 241
~VideoDataBuffer 227	VideoOutput class 243
about VideoDataBuffer class 225	~VideoOutput 244
class VideoDecoder 228	= 250
setTimeCode 227	about VideoOutput class 243
VideoDataBuffer 227	bool 251
VideoDecoder class 228	disableDeGhosting 248
about VideoDecoder class 228	enableDeGhosting 248
calcDolby3DMatrix 231	getActiveVideoMode 248
connectOutput 231	setAudioEmbedding 248
disconnectOutput 231	setOutputActivity 249
getDecoderErrors 232	setPosition 249
getDecoderStatus 232	setVideoMode 250
getVideoDataBuffer 232	VideoOutput 244, 245
setCloneVideo 232	VideoProperty_Dual_HDTV enumeration member 247
setDolby3DChroma 232	VideoProperty_HDTV enumeration member 247
setDolby3DMatrix 232	VideoProperty_SDTV enumeration member 247
VideoDecoder 230	VideoProperty_SDTV_Progressive enumeration member 247
waitDecoderStatus 233	VideoProperty_VirtualLinked enumeration member 247
VideoDecoderErrors class 233	VSCConnector class 251
~VideoDecoderErrors 235	~VSCConnector 253
= 237	abortGPIOInterrupt 254
about VideoDecoderErrors class 233	about VSCConnector class 251
getInputUnderruns 236	getGPIOValue 254
getPartialFrames 236	setGPIODirection 254
getPlayedFramesError 236	setGPIOInterrupt 254
getPlayedFramesOk 236	setGPIOValue 254
getRepeatedFramesAVSync 236	VSCConnector 253
getRepeatedFramesUnderrun 236	waitForGPIOInterrupt 254
getSkippedFramesAVSync 236	mail of of formor apt 201
getSkippedFramesError 237	

VideoDecoderErrors 235

t

W

WaitInfinite enumeration member 82, 99