

MPEG CDVA Experimentation Model (CXM)

1.0

Generated by Doxygen 1.8.13

Contents

1	Documentation	1
2	Namespace Index	3
2.1	Namespace List	3
3	Hierarchical Index	5
3.1	Class Hierarchy	5
4	Data Structure Index	7
4.1	Data Structures	7
5	File Index	9
5.1	File List	9
6	Namespace Documentation	11
6.1	evx Namespace Reference	11
6.1.1	Typedef Documentation	12
6.1.1.1	evx_status	12
6.1.1.2	float32	12
6.1.1.3	float64	12
6.1.1.4	int16	12
6.1.1.5	int32	12
6.1.1.6	int64	12
6.1.1.7	int8	13
6.1.1.8	uint16	13
6.1.1.9	uint32	13

6.1.1.10	uint64	13
6.1.1.11	uint8	13
6.1.1.12	wchar	13
6.1.2	Function Documentation	13
6.1.2.1	abs() [1/3]	13
6.1.2.2	abs() [2/3]	14
6.1.2.3	abs() [3/3]	14
6.1.2.4	align()	14
6.1.2.5	aligned_bit_copy()	14
6.1.2.6	clip_range()	14
6.1.2.7	greater_multiple()	14
6.1.2.8	log2() [1/3]	15
6.1.2.9	log2() [2/3]	15
6.1.2.10	log2() [3/3]	15
6.1.2.11	post_error_i()	15
6.1.2.12	unaligned_bit_copy()	15
6.1.3	Variable Documentation	16
6.1.3.1	log2_byte_lut	16
6.2	mpeg7cdva Namespace Reference	16
6.2.1	Detailed Description	17
6.2.2	Typedef Documentation	17
6.2.2.1	ShotDescriptorList	17
6.2.3	Enumeration Type Documentation	17
6.2.3.1	LogFormat	17
6.2.3.2	OPERATION	18

7 Data Structure Documentation	19
7.1 evx::bitstream Class Reference	19
7.1.1 Constructor & Destructor Documentation	19
7.1.1.1 bitstream() [1/3]	20
7.1.1.2 bitstream() [2/3]	20
7.1.1.3 bitstream() [3/3]	20
7.1.1.4 ~bitstream()	20
7.1.2 Member Function Documentation	20
7.1.2.1 assign() [1/2]	20
7.1.2.2 assign() [2/2]	20
7.1.2.3 clear()	21
7.1.2.4 empty()	21
7.1.2.5 is_empty()	21
7.1.2.6 is_full()	21
7.1.2.7 query_byte_occupancy()	21
7.1.2.8 query_capacity()	21
7.1.2.9 query_data()	21
7.1.2.10 query_occupancy()	21
7.1.2.11 read_bit()	22
7.1.2.12 read_bits()	22
7.1.2.13 read_byte()	22
7.1.2.14 read_bytes()	22
7.1.2.15 resize_capacity()	22
7.1.2.16 seek()	22
7.1.2.17 write_bit()	22
7.1.2.18 write_bits()	23
7.1.2.19 write_byte()	23
7.1.2.20 write_bytes()	23
7.2 mpeg7cdva::Buffer Class Reference	23
7.2.1 Detailed Description	24

7.2.2	Constructor & Destructor Documentation	24
7.2.2.1	Buffer() [1/4]	25
7.2.2.2	~Buffer()	25
7.2.2.3	Buffer() [2/4]	25
7.2.2.4	Buffer() [3/4]	25
7.2.2.5	Buffer() [4/4]	25
7.2.3	Member Function Documentation	25
7.2.3.1	assign()	25
7.2.3.2	clear()	26
7.2.3.3	compare()	26
7.2.3.4	data() [1/2]	26
7.2.3.5	data() [2/2]	26
7.2.3.6	empty()	26
7.2.3.7	equals()	27
7.2.3.8	fill()	27
7.2.3.9	operator=()	27
7.2.3.10	operator==()	27
7.2.3.11	read()	27
7.2.3.12	resize()	27
7.2.3.13	sdata() [1/2]	28
7.2.3.14	sdata() [2/2]	28
7.2.3.15	size()	28
7.2.3.16	swap()	28
7.2.3.17	write()	28
7.3	mpeg7cdva::CdvaException Class Reference	29
7.3.1	Detailed Description	29
7.3.2	Constructor & Destructor Documentation	29
7.3.2.1	CdvaException()	29
7.3.2.2	~CdvaException()	29
7.3.3	Member Function Documentation	29

7.3.3.1	what()	30
7.4	mpeg7cdva::CdvalImpl Class Reference	30
7.4.1	Detailed Description	32
7.4.2	Constructor & Destructor Documentation	32
7.4.2.1	CdvalImpl()	32
7.4.2.2	~CdvalImpl()	32
7.4.3	Member Function Documentation	33
7.4.3.1	byDescendingScore()	33
7.4.3.2	checkBitrate()	33
7.4.3.3	close()	33
7.4.3.4	commitDB()	33
7.4.3.5	encodeCoordinates()	34
7.4.3.6	encodeShot()	34
7.4.3.7	extract()	34
7.4.3.8	generateDTM()	34
7.4.3.9	getDescriptorExt()	35
7.4.3.10	getDiffSignature()	35
7.4.3.11	getExt()	35
7.4.3.12	init()	36
7.4.3.13	localDescCodingAbs()	36
7.4.3.14	makeindex()	37
7.4.3.15	match() [1/2]	37
7.4.3.16	match() [2/2]	37
7.4.3.17	match_med1()	38
7.4.3.18	match_med2()	38
7.4.3.19	nBitsSet() [1/2]	38
7.4.3.20	nBitsSet() [2/2]	38
7.4.3.21	parse()	38
7.4.3.22	retrieve()	38
7.4.4	Field Documentation	39

7.4.4.1	calc_desc_sizes	39
7.4.4.2	cdvsclient	39
7.4.4.3	cdvsconfig	39
7.4.4.4	cdvsMode	39
7.4.4.5	cdvsserver	39
7.4.4.6	current_op	40
7.4.4.7	drop_frame_th	40
7.4.4.8	encode_th	40
7.4.4.9	forceSampleMs	40
7.4.4.10	max_retrieved	40
7.4.4.11	minLocalDiff	40
7.4.4.12	minShotLen	41
7.4.4.13	optMatch	41
7.4.4.14	optMatch_b	41
7.4.4.15	optMatch_mode	41
7.4.4.16	optMatch_tau	41
7.4.4.17	readWriteCompressed	41
7.4.4.18	shot_cut_th	42
7.4.4.19	shot_ver_th	42
7.4.4.20	skip_after	42
7.4.4.21	skip_before	42
7.4.4.22	verboseMode	42
7.5	mpeg7cdva::CoordinateCoding::CircularSumContext Struct Reference	42
7.5.1	Detailed Description	43
7.5.2	Field Documentation	43
7.5.2.1	vCount	43
7.5.2.2	vInitialMap	43
7.5.2.3	vMap	43
7.6	mpeg7cdva::CompressedFeatureList Class Reference	44
7.6.1	Constructor & Destructor Documentation	44

7.6.1.1	CompressedFeatureList() [1/3]	44
7.6.1.2	CompressedFeatureList() [2/3]	44
7.6.1.3	CompressedFeatureList() [3/3]	45
7.6.2	Member Function Documentation	45
7.6.2.1	allocate()	45
7.6.2.2	getFeatureList()	45
7.6.2.3	matchDescriptors_oneWay()	45
7.6.2.4	matchDescriptors_twoWay()	45
7.6.2.5	operator=() [1/2]	46
7.6.2.6	operator=() [2/2]	46
7.6.2.7	toCDVScfl()	46
7.6.3	Field Documentation	46
7.6.3.1	relevances	46
7.7	mpeg7cdva::CompressedSegmentDescriptor Class Reference	46
7.7.1	Detailed Description	48
7.7.2	Constructor & Destructor Documentation	48
7.7.2.1	CompressedSegmentDescriptor()	48
7.7.2.2	~CompressedSegmentDescriptor()	48
7.7.3	Member Function Documentation	48
7.7.3.1	allocateGlobalBuffer()	48
7.7.3.2	allocateHistoBuffer()	49
7.7.3.3	allocateLocalBuffer()	49
7.7.3.4	clear()	49
7.7.3.5	decode()	49
7.7.3.6	getDtm()	49
7.7.3.7	getGlobalBuffer()	50
7.7.3.8	getGlobalBufSz()	50
7.7.3.9	getHistoBuffer()	50
7.7.3.10	getHistoBufSz()	50
7.7.3.11	getLocalBuffer()	50

7.7.3.12	getLocalBufSz()	50
7.7.3.13	nBitsSet()	50
7.7.3.14	numGlobFctPresent()	51
7.7.3.15	read()	51
7.7.3.16	readFeatureListFromBinaryAbs()	51
7.7.3.17	reconstructGlobalDifferences()	51
7.7.3.18	setGlobalUncompressedBufSz()	51
7.7.3.19	setHistoBufSz()	52
7.7.3.20	setLocalUncompressedBufSz()	52
7.7.3.21	write()	52
7.7.4	Field Documentation	52
7.7.4.1	dtm	52
7.7.4.2	globalBuf	52
7.7.4.3	globalBufSz	52
7.7.4.4	globalUncompressedBufSz	53
7.7.4.5	histoBuf	53
7.7.4.6	histoBufSz	53
7.7.4.7	localBuf	53
7.7.4.8	localBufSz	53
7.7.4.9	localUncompressedBufSz	53
7.7.4.10	params	54
7.7.4.11	SHOT_HEADER_SIZE	54
7.8	mpeg7cdva::CoordinateCoding Class Reference	54
7.8.1	Constructor & Destructor Documentation	55
7.8.1.1	CoordinateCoding()	55
7.8.1.2	~CoordinateCoding()	55
7.8.2	Member Function Documentation	56
7.8.2.1	AddImageSample()	56
7.8.2.2	compare()	56
7.8.2.3	difference()	56

7.8.2.4	EndTrainingMode()	56
7.8.2.5	exportVars()	56
7.8.2.6	fromBinary() [1/2]	57
7.8.2.7	fromBinary() [2/2]	57
7.8.2.8	generateFeatureList()	57
7.8.2.9	generateHistogramMap() [1/2]	58
7.8.2.10	generateHistogramMap() [2/2]	58
7.8.2.11	generateHistogramMap2()	58
7.8.2.12	readSeparateContext() [1/2]	59
7.8.2.13	readSeparateContext() [2/2]	59
7.8.2.14	setIsOK()	59
7.8.2.15	setVerbose()	59
7.8.2.16	StartTrainingMode()	59
7.8.2.17	toBinary()	59
7.8.2.18	writeSeparateContext()	60
7.8.3	Field Documentation	60
7.8.3.1	CONTEXT_RANGE	60
7.8.3.2	MAXIMUM_SUM_CONTEXT	60
7.8.3.3	SUM_HIST_COUNT_SIZE	60
7.9	mpeg7cdva::CoordList Class Reference	60
7.9.1	Constructor & Destructor Documentation	61
7.9.1.1	CoordList() [1/2]	61
7.9.1.2	CoordList() [2/2]	61
7.9.2	Member Function Documentation	61
7.9.2.1	getX()	61
7.9.2.2	getY()	61
7.10	CsscCoordinateCoding Class Reference	62
7.10.1	Detailed Description	62
7.11	mpeg7cdva::DescriptorTimeMap Class Reference	62
7.11.1	Constructor & Destructor Documentation	63

7.11.1.1	DescriptorTimeMap() [1/4]	63
7.11.1.2	DescriptorTimeMap() [2/4]	63
7.11.1.3	DescriptorTimeMap() [3/4]	63
7.11.1.4	DescriptorTimeMap() [4/4]	63
7.11.1.5	~DescriptorTimeMap()	64
7.11.2	Member Function Documentation	64
7.11.2.1	add()	64
7.11.2.2	addDescIndex()	64
7.11.2.3	clear()	64
7.11.2.4	get()	64
7.11.2.5	getBitSize()	65
7.11.2.6	getBuffer()	65
7.11.2.7	getCount()	65
7.11.2.8	getDescIndex()	65
7.11.2.9	getFeatureStartIndex()	65
7.11.2.10	getNAbsDescr()	65
7.11.2.11	getNDescr()	66
7.11.2.12	getNFrames()	66
7.11.2.13	init()	66
7.11.2.14	operator=()	66
7.11.2.15	parseDescIndex()	66
7.11.2.16	writeDescIndex()	66
7.11.3	Field Documentation	67
7.11.3.1	descIndex	67
7.11.3.2	ldOffsets	67
7.11.3.3	m_buffer	67
7.11.3.4	m_nAbsDescr	67
7.11.3.5	m_nbits	67
7.11.3.6	m_nDescr	67
7.11.3.7	m_nFrames	68

7.12	evx::entropy_coder Class Reference	68
7.12.1	Constructor & Destructor Documentation	68
7.12.1.1	entropy_coder() [1/2]	68
7.12.1.2	entropy_coder() [2/2]	68
7.12.2	Member Function Documentation	68
7.12.2.1	clear()	69
7.12.2.2	decode()	69
7.12.2.3	encode()	69
7.12.2.4	finish_encode()	69
7.12.2.5	start_decode()	69
7.13	mpeg7cdva::ExtractData Class Reference	69
7.13.1	Detailed Description	70
7.13.2	Constructor & Destructor Documentation	70
7.13.2.1	ExtractData()	70
7.13.3	Member Function Documentation	70
7.13.3.1	setDescriptorLength()	70
7.13.3.2	setNumFrames()	71
7.13.3.3	setNumShots()	71
7.13.3.4	setVideoDuration()	71
7.13.4	Field Documentation	71
7.13.4.1	clip_duration	72
7.13.4.2	coordinate_bit_count	72
7.13.4.3	descriptorlength	72
7.13.4.4	global_bit_count	72
7.13.4.5	header_bit_count	72
7.13.4.6	local_bit_count	72
7.13.4.7	n_keyframes	72
7.13.4.8	numframes	72
7.13.4.9	numshots	73
7.14	mpeg7cdva::FileManager Class Reference	73

7.14.1 Detailed Description	74
7.14.2 Constructor & Destructor Documentation	74
7.14.2.1 FileManager()	74
7.14.2.2 ~FileManager()	74
7.14.3 Member Function Documentation	74
7.14.3.1 countNames()	74
7.14.3.2 getDatasetName()	75
7.14.3.3 getDatasetPath()	75
7.14.3.4 getDatasetPathName()	75
7.14.3.5 getDatasetSize()	76
7.14.3.6 getQueryName()	76
7.14.3.7 getReferenceName()	76
7.14.3.8 getWorkspaceDir()	77
7.14.3.9 replaceExt()	77
7.14.3.10 replacePath()	77
7.14.3.11 setWorkspaceDir()	78
7.15 mpeg7cdva::LeanCoordList Class Reference	78
7.15.1 Constructor & Destructor Documentation	79
7.15.1.1 LeanCoordList() [1/4]	79
7.15.1.2 LeanCoordList() [2/4]	79
7.15.1.3 LeanCoordList() [3/4]	79
7.15.1.4 LeanCoordList() [4/4]	79
7.15.1.5 ~LeanCoordList()	79
7.15.2 Member Function Documentation	79
7.15.2.1 getFeatureList() [1/2]	80
7.15.2.2 getFeatureList() [2/2]	80
7.15.2.3 getLength()	80
7.15.2.4 getX()	80
7.15.2.5 getY()	80
7.15.2.6 operator=()	80

7.15.2.7	setX()	81
7.15.2.8	setY()	81
7.15.3	Field Documentation	81
7.15.3.1	nFeatures	81
7.15.3.2	xCoord	81
7.15.3.3	yCoord	81
7.16	mpeg7cdva::LogManager Class Reference	81
7.16.1	Detailed Description	82
7.16.2	Constructor & Destructor Documentation	82
7.16.2.1	LogManager()	82
7.16.2.2	~LogManager()	83
7.16.3	Member Function Documentation	83
7.16.3.1	close()	83
7.16.3.2	init()	83
7.16.3.3	printExtractData()	83
7.16.3.4	printExtractHeader()	83
7.16.3.5	printMatchData()	84
7.16.3.6	printMatchHeader()	84
7.16.3.7	printRetrievalData()	84
7.16.3.8	printRetrievalHeader()	84
7.17	mpeg7cdva::MatchData Class Reference	84
7.17.1	Detailed Description	85
7.17.2	Constructor & Destructor Documentation	85
7.17.2.1	MatchData()	85
7.17.2.2	~MatchData()	85
7.17.3	Member Function Documentation	85
7.17.3.1	getFirstMatchingTime()	86
7.17.3.2	getLastMatchingTime()	86
7.17.3.3	getReferenceId()	86
7.17.3.4	getScore()	86

7.17.3.5	setMatchingScore()	86
7.17.3.6	setMatchingTime()	87
7.17.3.7	setReferenceID()	87
7.18	mpeg7cdva::RefFeature Class Reference	87
7.18.1	Constructor & Destructor Documentation	88
7.18.1.1	RefFeature()	88
7.18.2	Member Function Documentation	88
7.18.2.1	compress()	88
7.18.3	Field Documentation	88
7.18.3.1	compressed	88
7.18.3.2	descrDiff	89
7.18.3.3	diff	89
7.18.3.4	refFeatureIndex	89
7.18.3.5	refFrame	89
7.18.3.6	refOnly	89
7.18.3.7	timesPresent	89
7.19	mpeg7cdva::SegmentDescriptor Class Reference	89
7.19.1	Detailed Description	91
7.19.2	Constructor & Destructor Documentation	91
7.19.2.1	SegmentDescriptor()	91
7.19.2.2	~SegmentDescriptor()	91
7.19.3	Member Function Documentation	91
7.19.3.1	addSegmentToDB()	91
7.19.3.2	clear()	91
7.19.3.3	empty()	92
7.19.3.4	getEndTimeMs()	92
7.19.3.5	getParity()	92
7.19.3.6	getSize()	92
7.19.3.7	getStartTimeMs()	92
7.19.3.8	read()	92

7.19.3.9	setEndTimeMs()	93
7.19.3.10	setParity()	93
7.19.3.11	setStartTimeMs()	93
7.19.3.12	write()	93
7.19.4	Field Documentation	93
7.19.4.1	endTime	93
7.19.4.2	keyframes	94
7.19.4.3	parity	94
7.19.4.4	SHOT_HEADER_SIZE	94
7.19.4.5	startTime	94
7.20	mpeg7cdva::WrappedCoordList Class Reference	94
7.20.1	Constructor & Destructor Documentation	95
7.20.1.1	WrappedCoordList()	95
7.20.2	Member Function Documentation	95
7.20.2.1	clampIdx()	95
7.20.2.2	getX()	95
7.20.2.3	getY()	95
7.20.3	Field Documentation	95
7.20.3.1	mfl	95
8	File Documentation	97
8.1	base.h File Reference	97
8.2	bitstream.h File Reference	97
8.2.1	Macro Definition Documentation	97
8.2.1.1	EVX_READ_BIT	98
8.2.1.2	EVX_WRITE_BIT	98
8.3	Buffer.h File Reference	98
8.4	cabac.h File Reference	98
8.5	cdva.h File Reference	99
8.6	CdvaException.h File Reference	99
8.7	Cdvalmpl.h File Reference	100

8.8	CompressedFeatureList.h File Reference	100
8.9	CoordinateCoding.h File Reference	101
8.10	DescriptorTimeMap.h File Reference	101
8.11	FileManager.h File Reference	102
8.12	LogManager.h File Reference	102
8.13	math.h File Reference	103
8.13.1	Macro Definition Documentation	103
8.13.1.1	EVX_GB	104
8.13.1.2	EVX_KB	104
8.13.1.3	evx_max2	104
8.13.1.4	evx_max3	104
8.13.1.5	EVX_MAX_INT16	104
8.13.1.6	EVX_MAX_INT32	104
8.13.1.7	EVX_MAX_INT64	104
8.13.1.8	EVX_MAX_INT8	105
8.13.1.9	EVX_MAX_UINT16	105
8.13.1.10	EVX_MAX_UINT32	105
8.13.1.11	EVX_MAX_UINT64	105
8.13.1.12	EVX_MAX_UINT8	105
8.13.1.13	EVX_MB	105
8.13.1.14	evx_min2	105
8.13.1.15	evx_min3	106
8.13.1.16	EVX_MIN_INT16	106
8.13.1.17	EVX_MIN_INT32	106
8.13.1.18	EVX_MIN_INT64	106
8.13.1.19	EVX_MIN_INT8	106
8.14	memory.h File Reference	106
8.15	RefFeature.h File Reference	107
8.16	version.h File Reference	107
8.16.1	Macro Definition Documentation	107
8.16.1.1	EVX_MAJOR_VERSION	107
8.16.1.2	EVX_MINOR_VERSION	107
8.16.1.3	EVX_VERSION_CHANGELIST	108
8.16.1.4	EVX_VERSION_MAJOR	108
8.16.1.5	EVX_VERSION_MINOR	108
8.16.1.6	EVX_VERSION_WORD	108

Chapter 1

Documentation

This is the documentation of the C++ classes implementing the MPEG CDVA Experimentation Model (CXM). The software implements the recommendations contained in the following documents (at <http://wg11.sc29.org/>):

- N15338: "Evaluation Framework for Compact Descriptors for Video Analysis - Search and Retrieval", July 2015, Warsaw, Poland
- N15729: "Evaluation Framework for Compact Descriptors for Video Analysis - Search and Retrieval – Version 2.0", October 2015, Geneva, CH

Documentation on how to build and install the code is contained in the "CDVA_build_run_instructions" document which can be found in the "docs" directory.

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

evx	11
mpeg7cdva Namespace used to encapsulate all MPEG-7 CDVA declarations	16

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

evx::bitstream	19
mpeg7cdva::Buffer	23
mpeg7cdva::CdvaImpl	30
mpeg7cdva::CoordinateCoding::CircularSumContext	42
CompressedFeatureList	
mpeg7cdva::CompressedFeatureList	44
mpeg7cdva::CoordinateCoding	54
mpeg7cdva::CoordList	60
mpeg7cdva::LeanCoordList	78
mpeg7cdva::WrappedCoordList	94
CsscCoordinateCoding	62
mpeg7cdva::DescriptorTimeMap	62
evx::entropy_coder	68
exception	
mpeg7cdva::CdvaException	29
mpeg7cdva::ExtractData	69
Feature	
mpeg7cdva::RefFeature	87
mpeg7cdva::FileManager	73
mpeg7cdva::LogManager	81
mpeg7cdva::MatchData	84
mpeg7cdva::SegmentDescriptor	89
mpeg7cdva::CompressedSegmentDescriptor	46

Chapter 4

Data Structure Index

4.1 Data Structures

Here are the data structures with brief descriptions:

evx::bitstream	19
mpeg7cdva::Buffer A container class for a byte array, intended to replace all malloc() and new() instructions in the main code	23
mpeg7cdva::CdvaException Class defining a specific exception for CDVA	29
mpeg7cdva::CdvaImpl A CDVA implementation based on multiple CDVS descriptors	30
mpeg7cdva::CoordinateCoding::CircularSumContext Basic structure for Cssc	42
mpeg7cdva::CompressedFeatureList	44
mpeg7cdva::CompressedSegmentDescriptor Extension of segment descriptor container to also hold compressed data for a segment	46
mpeg7cdva::CoordinateCoding	54
mpeg7cdva::CoordList	60
CsscCoordinateCoding Class that converts the coordinates of all descriptors of an image into a bitstream, and vice versa	62
mpeg7cdva::DescriptorTimeMap	62
evx::entropy_coder	68
mpeg7cdva::ExtractData A class containing the results of an extraction operation	69
mpeg7cdva::FileManager Helper class to manage lists of file names	73
mpeg7cdva::LeanCoordList	78
mpeg7cdva::LogManager Helper class to produce log files in various formats (csv, text, XML, etc.)	81
mpeg7cdva::MatchData A class containing the results of a matching or retrieval operation	84
mpeg7cdva::RefFeature	87
mpeg7cdva::SegmentDescriptor A container for CdvsDescriptor instances belonging to the same video segment	89
mpeg7cdva::WrappedCoordList	94

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

base.h	97
bitstream.h	97
Buffer.h	98
cabac.h	98
cdva.h	99
CdvaException.h	99
Cdvalmpl.h	100
CompressedFeatureList.h	100
CoordinateCoding.h	101
DescriptorTimeMap.h	101
FileManager.h	102
LogManager.h	102
math.h	103
memory.h	106
RefFeature.h	107
version.h	107

Chapter 6

Namespace Documentation

6.1 evx Namespace Reference

Data Structures

- class [bitstream](#)
- class [entropy_coder](#)

Typedefs

- typedef int64_t [int64](#)
- typedef int32_t [int32](#)
- typedef int16_t [int16](#)
- typedef int8_t [int8](#)
- typedef u_int64_t [uint64](#)
- typedef u_int32_t [uint32](#)
- typedef u_int16_t [uint16](#)
- typedef u_int8_t [uint8](#)
- typedef float [float32](#)
- typedef double [float64](#)
- typedef wchar_t [wchar](#)
- typedef [uint8](#) [evx_status](#)

Functions

- [uint32 post_error_i](#) ([uint8](#) error, const char *error_string, const char *function, const char *filename, [uint32](#) line)
- [uint8 log2](#) ([uint8](#) value)
- [uint8 log2](#) ([uint16](#) value)
- [uint8 log2](#) ([uint32](#) value)
- [int8 abs](#) ([int8](#) value)
- [int16 abs](#) ([int16](#) value)
- [int32 abs](#) ([int32](#) value)
- [int16 clip_range](#) ([int16](#) value, [int16](#) min, [int16](#) max)
- [uint32 greater_multiple](#) ([uint32](#) value, [uint32](#) multiple)
- [uint32 align](#) ([uint32](#) value, [uint32](#) alignment)
- [uint32 aligned_bit_copy](#) ([uint8](#) *dest, [uint32](#) dest_bit_offset, [uint8](#) *source, [uint32](#) source_bit_offset, [uint32](#) copy_bit_count)
- [uint32 unaligned_bit_copy](#) ([uint8](#) *dest, [uint32](#) dest_offset, [uint8](#) *source, [uint32](#) source_offset, [uint32](#) copy_bit_count)

Variables

- const `uint8 log2_byte_lut []`

6.1.1 Typedef Documentation

6.1.1.1 `evx_status`

```
typedef uint8 evx::evx_status
```

6.1.1.2 `float32`

```
typedef float evx::float32
```

6.1.1.3 `float64`

```
typedef double evx::float64
```

6.1.1.4 `int16`

```
typedef int16_t evx::int16
```

6.1.1.5 `int32`

```
typedef int32_t evx::int32
```

6.1.1.6 `int64`

```
typedef int64_t evx::int64
```

6.1.1.7 int8

```
typedef int8_t evx::int8
```

6.1.1.8 uint16

```
typedef u_int16_t evx::uint16
```

6.1.1.9 uint32

```
typedef u_int32_t evx::uint32
```

6.1.1.10 uint64

```
typedef u_int64_t evx::uint64
```

6.1.1.11 uint8

```
typedef u_int8_t evx::uint8
```

6.1.1.12 wchar

```
typedef wchar_t evx::wchar
```

6.1.2 Function Documentation

6.1.2.1 abs() [1/3]

```
int8 evx::abs (
    int8 value ) [inline]
```

6.1.2.2 abs() [2/3]

```
int16 evx::abs (
    int16 value ) [inline]
```

6.1.2.3 abs() [3/3]

```
int32 evx::abs (
    int32 value ) [inline]
```

6.1.2.4 align()

```
uint32 evx::align (
    uint32 value,
    uint32 alignment ) [inline]
```

References `greater_multiple()`.

Here is the call graph for this function:

6.1.2.5 aligned_bit_copy()

```
uint32 evx::aligned_bit_copy (
    uint8 * dest,
    uint32 dest_bit_offset,
    uint8 * source,
    uint32 source_bit_offset,
    uint32 copy_bit_count )
```

6.1.2.6 clip_range()

```
int16 evx::clip_range (
    int16 value,
    int16 min,
    int16 max ) [inline]
```

6.1.2.7 greater_multiple()

```
uint32 evx::greater_multiple (
    uint32 value,
    uint32 multiple ) [inline]
```

Referenced by `align()`.

6.1.2.8 log2() [1/3]

```
uint8 evx::log2 (
    uint8 value ) [inline]
```

Referenced by log2().

6.1.2.9 log2() [2/3]

```
uint8 evx::log2 (
    uint16 value ) [inline]
```

References log2().

Here is the call graph for this function:

6.1.2.10 log2() [3/3]

```
uint8 evx::log2 (
    uint32 value ) [inline]
```

References log2().

Here is the call graph for this function:

6.1.2.11 post_error_i()

```
uint32 evx::post_error_i (
    uint8 error,
    const char * error_string,
    const char * function,
    const char * filename,
    uint32 line ) [inline]
```

References evx_err.

6.1.2.12 unaligned_bit_copy()

```
uint32 evx::unaligned_bit_copy (
    uint8 * dest,
    uint32 dest_offset,
    uint8 * source,
    uint32 source_offset,
    uint32 copy_bit_count )
```


Typedefs

- typedef std::vector< [SegmentDescriptor](#) > [ShotDescriptorList](#)
vector of shots descriptors of a video

Enumerations

- enum [OPERATION](#) { [UNKNOWN](#), [EXTRACT](#), [MATCH](#), [RETRIEVE](#) }
 - enum [LogFormat](#) { [FORMAT_NONE](#) = 0, [FORMAT_CSV](#) = 1, [FORMAT_TEXT](#) = 2, [FORMAT_HTML](#) = 4 }
- Format of output logs.*

6.2.1 Detailed Description

Namespace used to encapsulate all MPEG-7 CDVA declarations.

Namespace used to encapsulate all MPEG-7 CDVA declarations that are visible when the CDVA headers (in particular [cdva.h](#)) are included.

6.2.2 Typedef Documentation

6.2.2.1 ShotDescriptorList

```
typedef std::vector<SegmentDescriptor> mpeg7cdva::ShotDescriptorList
```

vector of shots descriptors of a video

6.2.3 Enumeration Type Documentation

6.2.3.1 LogFormat

```
enum mpeg7cdva::LogFormat
```

Format of output logs.

Enumerator

FORMAT_NONE	do not output any data
FORMAT_CSV	output data in CSV format
FORMAT_TEXT	output data as free text
FORMAT_HTML	output data in HTML format

6.2.3.2 OPERATION

enum `mpeg7cdva::OPERATION`

Enumerator

UNKNOWN	
EXTRACT	
MATCH	
RETRIEVE	

Chapter 7

Data Structure Documentation

7.1 evx::bitstream Class Reference

```
#include <bitstream.h>
```

Public Member Functions

- [bitstream](#) ()
- [bitstream](#) (uint32 size)
- [bitstream](#) (void *bytes, uint32 size)
- virtual [~bitstream](#) ()
- [uint8 * query_data](#) () const
- [uint32 query_capacity](#) () const
- [uint32 query_occupancy](#) () const
- [uint32 query_byte_occupancy](#) () const
- [uint32 resize_capacity](#) (uint32 size_in_bits)
- [evx_status seek](#) (uint32 bit_offset)
- [evx_status assign](#) (const [bitstream](#) &rvalue)
- [evx_status assign](#) (void *bytes, uint32 size)
- void [clear](#) ()
- void [empty](#) ()
- bool [is_empty](#) () const
- bool [is_full](#) () const
- [evx_status write_byte](#) (uint8 value)
- [evx_status write_bit](#) (uint8 value)
- [evx_status write_bytes](#) (void *data, uint32 byte_count)
- [evx_status write_bits](#) (void *data, uint32 bit_count)
- [evx_status read_byte](#) (void *data)
- [evx_status read_bit](#) (void *data)
- [evx_status read_bytes](#) (void *data, uint32 *byte_count)
- [evx_status read_bits](#) (void *data, uint32 *bit_count)

7.1.1 Constructor & Destructor Documentation

7.1.1.1 `bitstream()` [1/3]

```
evx::bitstream::bitstream ( )
```

7.1.1.2 `bitstream()` [2/3]

```
evx::bitstream::bitstream (
    uint32 size )
```

7.1.1.3 `bitstream()` [3/3]

```
evx::bitstream::bitstream (
    void * bytes,
    uint32 size )
```

7.1.1.4 `~bitstream()`

```
virtual evx::bitstream::~~bitstream ( ) [virtual]
```

7.1.2 Member Function Documentation

7.1.2.1 `assign()` [1/2]

```
evx_status evx::bitstream::assign (
    const bitstream & rvalue )
```

7.1.2.2 `assign()` [2/2]

```
evx_status evx::bitstream::assign (
    void * bytes,
    uint32 size )
```

7.1.2.3 clear()

```
void evx::bitstream::clear ( )
```

7.1.2.4 empty()

```
void evx::bitstream::empty ( )
```

7.1.2.5 is_empty()

```
bool evx::bitstream::is_empty ( ) const
```

7.1.2.6 is_full()

```
bool evx::bitstream::is_full ( ) const
```

7.1.2.7 query_byte_occupancy()

```
uint32 evx::bitstream::query_byte_occupancy ( ) const
```

7.1.2.8 query_capacity()

```
uint32 evx::bitstream::query_capacity ( ) const
```

7.1.2.9 query_data()

```
uint8* evx::bitstream::query_data ( ) const
```

7.1.2.10 query_occupancy()

```
uint32 evx::bitstream::query_occupancy ( ) const
```

7.1.2.11 read_bit()

```
evx_status evx::bitstream::read_bit (
    void * data )
```

7.1.2.12 read_bits()

```
evx_status evx::bitstream::read_bits (
    void * data,
    uint32 * bit_count )
```

7.1.2.13 read_byte()

```
evx_status evx::bitstream::read_byte (
    void * data )
```

7.1.2.14 read_bytes()

```
evx_status evx::bitstream::read_bytes (
    void * data,
    uint32 * byte_count )
```

7.1.2.15 resize_capacity()

```
uint32 evx::bitstream::resize_capacity (
    uint32 size_in_bits )
```

7.1.2.16 seek()

```
evx_status evx::bitstream::seek (
    uint32 bit_offset )
```

7.1.2.17 write_bit()

```
evx_status evx::bitstream::write_bit (
    uint8 value )
```

7.1.2.18 write_bits()

```
evx_status evx::bitstream::write_bits (
    void * data,
    uint32 bit_count )
```

7.1.2.19 write_byte()

```
evx_status evx::bitstream::write_byte (
    uint8 value )
```

7.1.2.20 write_bytes()

```
evx_status evx::bitstream::write_bytes (
    void * data,
    uint32 byte_count )
```

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

- [bitstream.h](#)

7.2 mpeg7cdva::Buffer Class Reference

A container class for a byte array, intended to replace all malloc() and new() instructions in the main code.

```
#include <Buffer.h>
```

Public Member Functions

- [Buffer](#) ()
- virtual [~Buffer](#) ()
- [Buffer](#) (size_t size)
create a buffer of the given size
- [Buffer](#) (unsigned char *data, size_t size)
copy the given array into this Buffer
- [Buffer](#) (const [Buffer](#) &)
copy the given Buffer into this Buffer
- [Buffer](#) & operator= (const [Buffer](#) &)
assign a Buffer to another
- void [swap](#) ([Buffer](#) &x)
swap the content of two Buffer(s)
- void [fill](#) (unsigned char value=0)
fill a Buffer with the given value
- size_t [size](#) () const

- return the current size of the [Buffer](#)*
- bool [resize](#) (size_t newsize)
 - change buffer size; content is lost if newsize is less than the current size*
- bool [empty](#) () const
 - return true if the [Buffer](#) is empty*
- void [clear](#) ()
 - clear the [Buffer](#)*
- bool [assign](#) (const unsigned char *data, size_t size)
 - assign the given data to [Buffer](#)*
- bool [equals](#) (Buffer &buffer)
 - compare if two Buffer(s) are equal (i.e. if they have the same size and contain the same data)*
- unsigned char * [data](#) ()
 - access to [Buffer](#)'s data as unsigned char (writable)*
- const unsigned char * [data](#) () const
 - access to [Buffer](#)'s data as unsigned char (read only)*
- char * [sdata](#) ()
 - access to [Buffer](#)'s data as signed char (writable)*
- const char * [sdata](#) () const
 - access to [Buffer](#)'s data as signed char (read only)*
- void [read](#) (const char *fname)
 - read [Buffer](#) from a file*
- void [write](#) (const char *fname) const
 - write [Buffer](#) to file*
- int [compare](#) (const Buffer &other) const
 - Compare this buffer with another; return the number of different bytes.*
- bool [operator==](#) (const Buffer &other) const
 - compare if two Buffer(s) are equal (i.e. if they have the same size and contain the same data)*

7.2.1 Detailed Description

A container class for a byte array, intended to replace all malloc() and new() instructions in the main code.

This class properly deallocates memory when an exception is thrown.

Author

Massimo Balestri

Date

2013

7.2.2 Constructor & Destructor Documentation

7.2.2.1 Buffer() [1/4]

```
mpeg7cdva::Buffer::Buffer ( )
```

7.2.2.2 ~Buffer()

```
virtual mpeg7cdva::Buffer::~~Buffer ( ) [virtual]
```

7.2.2.3 Buffer() [2/4]

```
mpeg7cdva::Buffer::Buffer (
    size_t size )
```

create a buffer of the given size

7.2.2.4 Buffer() [3/4]

```
mpeg7cdva::Buffer::Buffer (
    unsigned char * data,
    size_t size )
```

copy the given array into this [Buffer](#)

7.2.2.5 Buffer() [4/4]

```
mpeg7cdva::Buffer::Buffer (
    const Buffer & )
```

copy the given [Buffer](#) into this [Buffer](#)

7.2.3 Member Function Documentation

7.2.3.1 assign()

```
bool mpeg7cdva::Buffer::assign (
    const unsigned char * data,
    size_t size )
```

assign the given data to [Buffer](#)

7.2.3.2 clear()

```
void mpeg7cdva::Buffer::clear ( )
```

clear the [Buffer](#)

7.2.3.3 compare()

```
int mpeg7cdva::Buffer::compare (
    const Buffer & other ) const
```

Compare this buffer with another; return the number of different bytes.

Parameters

<i>other</i>	the other Buffer
--------------	----------------------------------

Returns

the number of differences; zero if no difference is found.

7.2.3.4 data() [1/2]

```
unsigned char* mpeg7cdva::Buffer::data ( )
```

access to [Buffer](#)'s data as unsigned char (writable)

7.2.3.5 data() [2/2]

```
const unsigned char* mpeg7cdva::Buffer::data ( ) const
```

access to [Buffer](#)'s data as unsigned char (read only)

7.2.3.6 empty()

```
bool mpeg7cdva::Buffer::empty ( ) const
```

return true if the [Buffer](#) is empty

7.2.3.7 equals()

```
bool mpeg7cdva::Buffer::equals (
    Buffer & buffer )
```

compare if two Buffer(s) are equal (i.e. if they have the same size and contain the same data)

7.2.3.8 fill()

```
void mpeg7cdva::Buffer::fill (
    unsigned char value = 0 )
```

fill a Buffer with the given value

7.2.3.9 operator=()

```
Buffer& mpeg7cdva::Buffer::operator= (
    const Buffer & )
```

assign a Buffer to another

7.2.3.10 operator==()

```
bool mpeg7cdva::Buffer::operator== (
    const Buffer & other ) const
```

compare if two Buffer(s) are equal (i.e. if they have the same size and contain the same data)

7.2.3.11 read()

```
void mpeg7cdva::Buffer::read (
    const char * fname )
```

read Buffer from a file

7.2.3.12 resize()

```
bool mpeg7cdva::Buffer::resize (
    size_t newsize )
```

change buffer size; content is lost if newsize if less than the current size

7.2.3.13 `sdata()` [1/2]

```
char* mpeg7cdva::Buffer::sdata ( )
```

access to [Buffer](#)'s data as signed char (writable)

7.2.3.14 `sdata()` [2/2]

```
const char* mpeg7cdva::Buffer::sdata ( ) const
```

access to [Buffer](#)'s data as signed char (read only)

7.2.3.15 `size()`

```
size_t mpeg7cdva::Buffer::size ( ) const
```

return the current size of the [Buffer](#)

7.2.3.16 `swap()`

```
void mpeg7cdva::Buffer::swap (
    Buffer & x )
```

swap the content of two [Buffer](#)(s)

7.2.3.17 `write()`

```
void mpeg7cdva::Buffer::write (
    const char * fname ) const
```

write [Buffer](#) to file

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

- [Buffer.h](#)

7.3 mpeg7cdva::CdvaException Class Reference

Class defining a specific exception for CDVA.

```
#include <CdvaException.h>
```

Inheritance diagram for mpeg7cdva::CdvaException:

Collaboration diagram for mpeg7cdva::CdvaException:

Public Member Functions

- [CdvaException](#) (std::string str)
Create a new CDVA exception.
- virtual [~CdvaException](#) () throw ()
- const char * [what](#) () const throw ()
Get the exception message.

7.3.1 Detailed Description

Class defining a specific exception for CDVA.

7.3.2 Constructor & Destructor Documentation

7.3.2.1 CdvaException()

```
mpeg7cdva::CdvaException::CdvaException (
    std::string str ) [inline]
```

Create a new CDVA exception.

Parameters

<i>str</i>	the exception message string.
------------	-------------------------------

7.3.2.2 ~CdvaException()

```
virtual mpeg7cdva::CdvaException::~~CdvaException ( ) throw () [inline], [virtual]
```

7.3.3 Member Function Documentation

7.3.3.1 what()

```
const char* mpeg7cdva::CdvaException::what ( ) const throw ( ) [inline]
```

Get the exception message.

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

- [CdvaException.h](#)

7.4 mpeg7cdva::CdvalImpl Class Reference

A CDVA implementation based on multiple CDVS descriptors.

```
#include <CdvaImpl.h>
```

Public Member Functions

- [CdvalImpl](#) ()
- virtual [~CdvalImpl](#) ()
- virtual void [init](#) ([OPERATION](#) op, bool verbose, size_t n_videos, int querybitrate, int refbitrate=0, bool calcdescsizes=false, bool rwCompressed=false, bool [optMatch](#)=false, double set_drop_th=-1, double set_encode_th=-1)

initialization method - called once before processing videos.
- void [extract](#) (const std::string &descname, const std::string &videopathname, int bitrate, [ExtractData](#) &outdata) const

Video processing method - called once for each video in the list.
- virtual double [match](#) ([MatchData](#) &matchResults, const std::string &qdescname, const std::string &rdescname, int qbitrate, int rbitrate)

Video matching method - called once for each pair of videos in the list.
- virtual void [makeindex](#) (const std::string &cdva_descriptor, const std::string &relativepathname)

Video indexing method - builds a DB of reference video descriptors.
- virtual void [retrieve](#) (std::vector< [MatchData](#) > &retrievalResults, const std::string &qdescname, int qbitrate)

Video retrieval method - returns a list of reference videos matching the query video.
- void [commitDB](#) ()

Save database information.
- virtual void [close](#) ()

de-initialization method - called once at the end of processing.

Static Public Member Functions

- static bool [checkBitrate](#) (int bitrate)

check if the given bitrate is one of the standard values defined in the CDVA evaluation framework.
- static const char * [getDescriptorExt](#) (int bitrate)

get the file extension corresponding to the given bitrate.

Protected Member Functions

- virtual void [parse](#) (const std::string &descFile, [ShotDescriptorList](#) &shotList)
- virtual double [match](#) ([MatchData](#) &matchResults, const [ShotDescriptorList](#) &qDescList, const [ShotDescriptorList](#) &rDescList)
- virtual double [match_med1](#) ([MatchData](#) &matchResults, const [ShotDescriptorList](#) &qDescList, const [ShotDescriptorList](#) &rDescList)
matching using medoid - variant 1 check medoids against min threshold, then perform full match
- virtual double [match_med2](#) ([MatchData](#) &matchResults, const [ShotDescriptorList](#) &qDescList, const [ShotDescriptorList](#) &rDescList)
matching using medoid - variant 2 check medoids against min threshold, then iteratively match frames
- virtual int [encodeShot](#) ([SegmentDescriptor](#) &shot, unsigned long endTime, mpeg7cdvs::CdvsDescriptor &medoid, std::ofstream &fout, [ExtractData](#) &outdata) const
- virtual void [getDiffSignature](#) (mpeg7cdvs::SCFVSignature &medoid, mpeg7cdvs::SCFVSignature &other, mpeg7cdvs::SCFVSignature &diffSig, unsigned int &bufferSize, unsigned char *buffer, unsigned int *globalstatsBin) const
- virtual void [localDescCodingAbs](#) ([SegmentDescriptor](#) &shot, std::vector< int > &framesToCode, unsigned int &bufferSize, unsigned char *buffer, int medoidIdx, unsigned int *localstats) const
lossy local descriptor coding, applying ABAC to absolute descriptors
- void [generateDTM](#) ([CompressedSegmentDescriptor](#) &shot, std::vector< int > &framesToCode, int medoidIdx, std::vector< [RefFeature](#) > refFeatureList, std::vector< int > ldOffsets, std::vector< int > localFrameOrder, int totalNrFeatures) const
generate frame - descriptor map
- virtual void [encodeCoordinates](#) ([CompressedSegmentDescriptor](#) &shot, std::vector< int > &framesToCode, int medoidIdx, std::vector< [RefFeature](#) > refFeatureList, std::vector< int > ldOffsets) const
encode coordinates of local descriptors of as one block

Static Protected Member Functions

- static bool [byDescendingScore](#) (const [MatchData](#) &m1, const [MatchData](#) &m2)
- static int [nBitsSet](#) (unsigned char byte)
- static int [nBitsSet](#) (unsigned int word)
- static std::string [getExt](#) (const std::string &imageName)

Protected Attributes

- mpeg7cdvs::CdvsConfiguration * [cdvsconfig](#)
- mpeg7cdvs::CdvsClient * [cdvsclient](#)
- mpeg7cdvs::CdvsServer * [cdvsserver](#)
- bool [verboseMode](#)
verbose mode indicator
- [OPERATION](#) [current_op](#)
the current operation
- int [cdvsMode](#)
the CDVS mode that will be used to encode keyframe descriptors
- int [skip_before](#)
number of video frames to skip before decoding one
- int [skip_after](#)
number of video frames to skip after decoding one
- double [drop_frame_th](#)
drop frame threshold
- double [shot_cut_th](#)

- shot cut threshold*
- double `shot_ver_th`
 - shot verification threshold*
- double `encode_th`
 - threshold for encoding other frames than median*
- size_t `max_retrieved`
 - maximum number of retrieved images*
- bool `calc_desc_sizes`
 - calculate size of descriptor components*
- int `forceSampleMs`
 - enforce a sample every x milliseconds*
- int `minShotLen`
 - minimum shot length*
- int `minLocalDiff`
 - min local difference for lossy local descriptor coding (if diff is larger, descriptor is encoded, otherwise replaced by reference)*
- bool `readWriteCompressed`
 - read and write compressed descriptors*
- bool `optMatch`
 - approx. matching of medoid-based descriptors*
- int `optMatch_mode`
 - mode for optimised matching (1 or 2)*
- double `optMatch_tau`
 - factor to define threshold to continue matching (opt matching modes 1 and 2)*
- double `optMatch_b`
 - divider for fraction of key frames to check to decide for match (opt matching mode 2)*

7.4.1 Detailed Description

A CDVA implementation based on multiple CDVS descriptors.

7.4.2 Constructor & Destructor Documentation

7.4.2.1 CdvaImpl()

```
mpeg7cdva::CdvaImpl::CdvaImpl ( )
```

7.4.2.2 ~CdvaImpl()

```
virtual mpeg7cdva::CdvaImpl::~CdvaImpl ( ) [virtual]
```

7.4.3 Member Function Documentation

7.4.3.1 byDescendingScore()

```
static bool mpeg7cdva::CdvaImpl::byDescendingScore (
    const MatchData & m1,
    const MatchData & m2 ) [static], [protected]
```

7.4.3.2 checkBitrate()

```
static bool mpeg7cdva::CdvaImpl::checkBitrate (
    int bitrate ) [static]
```

check if the given bitrate is one of the standard values defined in the CDVA evaluation framework.

Parameters

<i>bitrate</i>	the bitrate in Kilo-byte per second (KB/s)
----------------	--

Returns

true if valid

7.4.3.3 close()

```
virtual void mpeg7cdva::CdvaImpl::close ( ) [virtual]
```

de-initialization method - called once at the end of processing.

7.4.3.4 commitDB()

```
void mpeg7cdva::CdvaImpl::commitDB ( )
```

Save database information.

7.4.3.5 encodeCoordinates()

```
virtual void mpeg7cdva::CdvaImpl::encodeCoordinates (
    CompressedSegmentDescriptor & shot,
    std::vector< int > & framesToCode,
    int medoidIdx,
    std::vector< RefFeature > refFeatureList,
    std::vector< int > ldOffsets ) const [protected], [virtual]
```

encode coordinates of local descriptors of as one block

7.4.3.6 encodeShot()

```
virtual int mpeg7cdva::CdvaImpl::encodeShot (
    SegmentDescriptor & shot,
    unsigned long endTime,
    mpeg7cdvs::CdvsDescriptor & medoid,
    std::ofstream & fout,
    ExtractData & outdata ) const [protected], [virtual]
```

7.4.3.7 extract()

```
void mpeg7cdva::CdvaImpl::extract (
    const std::string & descrname,
    const std::string & videopathname,
    int bitrate,
    ExtractData & outdata ) const
```

Video processing method - called once for each video in the list.

Parameters

<i>descrname</i>	output descriptor pathname
<i>videopathname</i>	input video stream pathname
<i>bitrate</i>	encoding bitrate (one of 0,16,64,256)
<i>outdata</i>	the container for output data

7.4.3.8 generateDTM()

```
void mpeg7cdva::CdvaImpl::generateDTM (
    CompressedSegmentDescriptor & shot,
    std::vector< int > & framesToCode,
    int medoidIdx,
    std::vector< RefFeature > refFeatureList,
```

```
std::vector< int > ldOffsets,
std::vector< int > localFrameOrder,
int totalNrFeatures ) const [protected]
```

generate frame - descriptor map

7.4.3.9 getDescriptorExt()

```
static const char* mpeg7cdva::CdvaImpl::getDescriptorExt (
    int bitrate ) [static]
```

get the file extension corresponding to the given bitrate.

Parameters

<i>bitrate</i>	the bitrate in Kilo-byte per second (KB/s)
----------------	--

Returns

the file extension

7.4.3.10 getDiffSignature()

```
virtual void mpeg7cdva::CdvaImpl::getDiffSignature (
    mpeg7cdvs::SCFVSignature & medoid,
    mpeg7cdvs::SCFVSignature & other,
    mpeg7cdvs::SCFVSignature & diffSig,
    unsigned int & bufferSize,
    unsigned char * buffer,
    unsigned int * globaldstatsBin ) const [protected], [virtual]
```

7.4.3.11 getExt()

```
static std::string mpeg7cdva::CdvaImpl::getExt (
    const std::string & imageName ) [static], [protected]
```

7.4.3.12 init()

```
virtual void mpeg7cdva::CdvaImpl::init (
    OPERATION op,
    bool verbose,
    size_t n_videos,
    int querybitrate,
    int refbitrate = 0,
    bool calcdescsizes = false,
    bool rwCompressed = false,
    bool optMatch = false,
    double set_drop_th = -1,
    double set_encode_th = -1 ) [virtual]
```

initialization method - called once before processing videos.

Parameters

<i>op</i>	one of EXTRACT, MATCH, RETRIEVE
<i>verbose</i>	when set, more information is provided
<i>n_videos</i>	the number of videos to be processed
<i>querybitrate</i>	the query encoding bitrate (one of 0,16,64,256)
<i>refbitrate</i>	the reference encoding bitrate (one of 0,16,64,256)
<i>calcdescsizes</i>	if true, the size of individual elements of the CDVS descriptor is reported in the output log files
<i>rwCompressed</i>	if true, compressed descriptor representations are used for serialising and parsing
<i>optMatch</i>	if true, use approximate matching of medoid-based descriptors
<i>set_drop_th</i>	value to override default drop frame threshold
<i>set_encode_th</i>	value to override default encode frame threshold

7.4.3.13 localDescCodingAbs()

```
virtual void mpeg7cdva::CdvaImpl::localDescCodingAbs (
    SegmentDescriptor & shot,
    std::vector< int > & framesToCode,
    unsigned int & bufferSize,
    unsigned char * buffer,
    int medoidIdx,
    unsigned int * localdstats ) const [protected], [virtual]
```

lossy local descriptor coding, applying ABAC to absolute descriptors

Parameters

<i>localdstats</i>	for gathering encoding statistics only
--------------------	--

7.4.3.14 makeindex()

```
virtual void mpeg7cdva::CdvaImpl::makeindex (
    const std::string & cdva_descriptor,
    const std::string & relativepathname ) [virtual]
```

Video indexing method - builds a DB of reference video descriptors.

Parameters

<i>cdva_descriptor</i>	the descriptor to add to the DB
<i>relativepathname</i>	the relative pathname of the video file to be used as unique identifier

7.4.3.15 match() [1/2]

```
virtual double mpeg7cdva::CdvaImpl::match (
    MatchData & matchResults,
    const ShotDescriptorList & qDescList,
    const ShotDescriptorList & rDescList ) [protected], [virtual]
```

7.4.3.16 match() [2/2]

```
virtual double mpeg7cdva::CdvaImpl::match (
    MatchData & matchResults,
    const std::string & qdescrname,
    const std::string & rdescrname,
    int qbitrate,
    int rbitrate ) [virtual]
```

Video matching method - called once for each pair of videos in the list.

Parameters

<i>matchResults</i>	container for the results of matching
<i>qdescrname</i>	input query descriptor name
<i>rdescrname</i>	input reference descriptor name
<i>qbitrate</i>	query bitrate (one of 16,64,256)
<i>rbitrate</i>	reference bitrate (one of 16,64,256)

Returns

the matching score (normalized in the [0..1] range)

7.4.3.17 match_med1()

```
virtual double mpeg7cdva::CdvaImpl::match_med1 (
    MatchData & matchResults,
    const ShotDescriptorList & qDescList,
    const ShotDescriptorList & rDescList ) [protected], [virtual]
```

matching using medoid - variant 1 check medoids against min threshold, then perform full match

7.4.3.18 match_med2()

```
virtual double mpeg7cdva::CdvaImpl::match_med2 (
    MatchData & matchResults,
    const ShotDescriptorList & qDescList,
    const ShotDescriptorList & rDescList ) [protected], [virtual]
```

matching using medoid - variant 2 check medoids against min threshold, then iteratively match frames

7.4.3.19 nBitsSet() [1/2]

```
static int mpeg7cdva::CdvaImpl::nBitsSet (
    unsigned char byte ) [inline], [static], [protected]
```

7.4.3.20 nBitsSet() [2/2]

```
static int mpeg7cdva::CdvaImpl::nBitsSet (
    unsigned int word ) [inline], [static], [protected]
```

7.4.3.21 parse()

```
virtual void mpeg7cdva::CdvaImpl::parse (
    const std::string & descFile,
    ShotDescriptorList & shotList ) [protected], [virtual]
```

7.4.3.22 retrieve()

```
virtual void mpeg7cdva::CdvaImpl::retrieve (
    std::vector< MatchData > & retrievalResults,
    const std::string & qdescrname,
    int qbitrate ) [virtual]
```

Video retrieval method - returns a list of reference videos matching the query video.

Parameters

<i>retrievalResults</i>	the output vector containing an ordered list of matching reference videos
<i>qdescrname</i>	the video query descriptor
<i>qbitrate</i>	query bitrate (one of 16,64,256)

7.4.4 Field Documentation

7.4.4.1 calc_desc_sizes

```
bool mpeg7cdva::CdvaImpl::calc_desc_sizes [protected]
```

calculate size of descriptor components

7.4.4.2 cdvsclient

```
mpeg7cdvs::CdvsClient* mpeg7cdva::CdvaImpl::cdvsclient [protected]
```

7.4.4.3 cdvsconfig

```
mpeg7cdvs::CdvsConfiguration* mpeg7cdva::CdvaImpl::cdvsconfig [protected]
```

7.4.4.4 cdvsMode

```
int mpeg7cdva::CdvaImpl::cdvsMode [protected]
```

the CDVS mode that will be used to encode keyframe descriptors

7.4.4.5 cdvsserver

```
mpeg7cdvs::CdvsServer* mpeg7cdva::CdvaImpl::cdvsserver [protected]
```

7.4.4.6 current_op

`OPERATION` `mpeg7cdva::CdvaImpl::current_op` [protected]

the current operation

7.4.4.7 drop_frame_th

`double` `mpeg7cdva::CdvaImpl::drop_frame_th` [protected]

drop frame threshold

7.4.4.8 encode_th

`double` `mpeg7cdva::CdvaImpl::encode_th` [protected]

threshold for encoding other frames than median

7.4.4.9 forceSampleMs

`int` `mpeg7cdva::CdvaImpl::forceSampleMs` [protected]

enforce a sample every x milliseconds

7.4.4.10 max_retrieved

`size_t` `mpeg7cdva::CdvaImpl::max_retrieved` [protected]

maximum number of retrieved images

7.4.4.11 minLocalDiff

`int` `mpeg7cdva::CdvaImpl::minLocalDiff` [protected]

min local difference for lossy local descriptor coding (if diff is larger, descriptor is encoded, otherwise replaced by reference)

7.4.4.12 minShotLen

`int mpeg7cdva::CdvaImpl::minShotLen [protected]`

minimum shot length

7.4.4.13 optMatch

`bool mpeg7cdva::CdvaImpl::optMatch [protected]`

approx. matching of medoid-based descriptors

7.4.4.14 optMatch_b

`double mpeg7cdva::CdvaImpl::optMatch_b [protected]`

divider for fraction of key frames to check to decide for match (opt matching mode 2)

7.4.4.15 optMatch_mode

`int mpeg7cdva::CdvaImpl::optMatch_mode [protected]`

mode for optimised matching (1 or 2)

7.4.4.16 optMatch_tau

`double mpeg7cdva::CdvaImpl::optMatch_tau [protected]`

factor to define threshold to continue matching (opt matching modes 1 and 2)

7.4.4.17 readWriteCompressed

`bool mpeg7cdva::CdvaImpl::readWriteCompressed [protected]`

read and write compressed descriptors

7.4.4.18 shot_cut_th

```
double mpeg7cdva::CdvaImpl::shot_cut_th [protected]
```

shot cut threshold

7.4.4.19 shot_ver_th

```
double mpeg7cdva::CdvaImpl::shot_ver_th [protected]
```

shot verification threshold

7.4.4.20 skip_after

```
int mpeg7cdva::CdvaImpl::skip_after [protected]
```

number of video frames to skip after decoding one

7.4.4.21 skip_before

```
int mpeg7cdva::CdvaImpl::skip_before [protected]
```

number of video frames to skip before decoding one

7.4.4.22 verboseMode

```
bool mpeg7cdva::CdvaImpl::verboseMode [protected]
```

verbose mode indicator

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

- [CdvaImpl.h](#)

7.5 mpeg7cdva::CoordinateCoding::CircularSumContext Struct Reference

Basic structure for Cssc.

```
#include <CoordinateCoding.h>
```

Data Fields

- long [vCount](#) [[SUM_HIST_COUNT_SIZE](#)]
Histogram Count Arithmetic Coding model initialization data.
- long [vInitialMap](#) [2]
Circular scan map Arithmetic Coding model initialization data (initial)
- long [vMap](#) [[MAXIMUM_SUM_CONTEXT](#)+1][2]
Circular scan map Arithmetic Coding model initialization data (following)

7.5.1 Detailed Description

Basic structure for Cssc.

7.5.2 Field Documentation

7.5.2.1 vCount

```
long mpeg7cdva::CoordinateCoding::CircularSumContext::vCount [SUM\_HIST\_COUNT\_SIZE]
```

Histogram Count Arithmetic Coding model initialization data.

7.5.2.2 vInitialMap

```
long mpeg7cdva::CoordinateCoding::CircularSumContext::vInitialMap [2]
```

Circular scan map Arithmetic Coding model initialization data (initial)

7.5.2.3 vMap

```
long mpeg7cdva::CoordinateCoding::CircularSumContext::vMap [MAXIMUM\_SUM\_CONTEXT+1] [2]
```

Circular scan map Arithmetic Coding model initialization data (following)

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

- [CoordinateCoding.h](#)

7.6 mpeg7cdva::CompressedFeatureList Class Reference

```
#include <CompressedFeatureList.h>
```

Inheritance diagram for mpeg7cdva::CompressedFeatureList:

Collaboration diagram for mpeg7cdva::CompressedFeatureList:

Public Member Functions

- [CompressedFeatureList](#) ()
- [CompressedFeatureList](#) (const [CompressedFeatureList](#) &a, bool relevantOnly=false)
- [CompressedFeatureList](#) (const mpeg7cdvs::FeatureList &a, bool relevantOnly)
- [CompressedFeatureList](#) & operator= ([CompressedFeatureList](#) a)
- [CompressedFeatureList](#) & operator= (mpeg7cdvs::CompressedFeatureList a)
- void [toCDVScfl](#) (mpeg7cdvs::CompressedFeatureList &cfl)
- void [allocate](#) (int nFeatures, int descLen)
- virtual int [matchDescriptors_oneWay](#) (mpeg7cdvs::PointPairs &pairs, const [CompressedFeatureList](#) &other←
FeatureList, float ratioThreshold, [DescriptorTimeMap](#) &thisDtm, [DescriptorTimeMap](#) &otherDtm, [CoordList](#)
&thisFl, [CoordList](#) &otherFl, int thisIdx, int otherIdx, int **distanceMatrix) const
- virtual int [matchDescriptors_twoWay](#) (mpeg7cdvs::PointPairs &pairs, const [CompressedFeatureList](#) &other←
FeatureList, float ratioThreshold, [DescriptorTimeMap](#) &thisDtm, [DescriptorTimeMap](#) &otherDtm, [CoordList](#)
&thisFl, [CoordList](#) &otherFl, int thisIdx, int otherIdx, int **distanceMatrix) const
- void [getFeatureList](#) (mpeg7cdvs::FeatureList &fl)

Data Fields

- std::vector< int > [relevances](#)

7.6.1 Constructor & Destructor Documentation

7.6.1.1 CompressedFeatureList() [1/3]

```
mpeg7cdva::CompressedFeatureList::CompressedFeatureList ( )
```

7.6.1.2 CompressedFeatureList() [2/3]

```
mpeg7cdva::CompressedFeatureList::CompressedFeatureList (
    const CompressedFeatureList & a,
    bool relevantOnly = false )
```

7.6.1.3 CompressedFeatureList() [3/3]

```
mpeg7cdva::CompressedFeatureList::CompressedFeatureList (
    const mpeg7cdvs::FeatureList & a,
    bool relevantOnly )
```

7.6.2 Member Function Documentation

7.6.2.1 allocate()

```
void mpeg7cdva::CompressedFeatureList::allocate (
    int nFeatures,
    int descLen )
```

7.6.2.2 getFeatureList()

```
void mpeg7cdva::CompressedFeatureList::getFeatureList (
    mpeg7cdvs::FeatureList & fl )
```

7.6.2.3 matchDescriptors_oneWay()

```
virtual int mpeg7cdva::CompressedFeatureList::matchDescriptors_oneWay (
    mpeg7cdvs::PointPairs & pairs,
    const CompressedFeatureList & otherFeatureList,
    float ratioThreshold,
    DescriptorTimeMap & thisDtm,
    DescriptorTimeMap & otherDtm,
    CoordList & thisFl,
    CoordList & otherFl,
    int thisIdx,
    int otherIdx,
    int ** distanceMatrix ) const [virtual]
```

7.6.2.4 matchDescriptors_twoWay()

```
virtual int mpeg7cdva::CompressedFeatureList::matchDescriptors_twoWay (
    mpeg7cdvs::PointPairs & pairs,
    const CompressedFeatureList & otherFeatureList,
    float ratioThreshold,
    DescriptorTimeMap & thisDtm,
    DescriptorTimeMap & otherDtm,
    CoordList & thisFl,
    CoordList & otherFl,
    int thisIdx,
    int otherIdx,
    int ** distanceMatrix ) const [virtual]
```

7.6.2.5 operator=() [1/2]

```
CompressedFeatureList& mpeg7cdva::CompressedFeatureList::operator= (
    CompressedFeatureList a )
```

7.6.2.6 operator=() [2/2]

```
CompressedFeatureList& mpeg7cdva::CompressedFeatureList::operator= (
    mpeg7cdvs::CompressedFeatureList a )
```

7.6.2.7 toCDVScfl()

```
void mpeg7cdva::CompressedFeatureList::toCDVScfl (
    mpeg7cdvs::CompressedFeatureList & cfl )
```

7.6.3 Field Documentation**7.6.3.1 relevances**

```
std::vector<int> mpeg7cdva::CompressedFeatureList::relevances
```

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

- [CompressedFeatureList.h](#)

7.7 mpeg7cdva::CompressedSegmentDescriptor Class Reference

Extension of segment descriptor container to also hold compressed data for a segment.

```
#include <CdvaImpl.h>
```

Inheritance diagram for mpeg7cdva::CompressedSegmentDescriptor:

Collaboration diagram for mpeg7cdva::CompressedSegmentDescriptor:

Public Member Functions

- [CompressedSegmentDescriptor](#) (const mpeg7cdvs::Parameters ¶ms)
- virtual [~CompressedSegmentDescriptor](#) ()
- virtual size_t [write](#) (std::ofstream &fout) const
write (appending) this segment to a file
- virtual size_t [read](#) (std::ifstream &fin, mpeg7cdvs::CdvsServer *cdvsServer)
read (from the current position) this segment from a file
- virtual unsigned char * [allocateLocalBuffer](#) (unsigned long sz)
- virtual unsigned char * [allocateGlobalBuffer](#) (unsigned long sz)
- virtual unsigned char * [allocateHistoBuffer](#) (unsigned long sz)
- virtual unsigned char * [getLocalBuffer](#) ()
- virtual unsigned char * [getGlobalBuffer](#) ()
- virtual unsigned char * [getHistoBuffer](#) ()
- virtual unsigned long [getLocalBufSz](#) ()
- virtual unsigned long [getGlobalBufSz](#) ()
- virtual unsigned long [getHistoBufSz](#) ()
- virtual void [setHistoBufSz](#) (unsigned long sz)
- virtual void [setLocalUncompressedBufSz](#) (unsigned long sz)
- virtual void [setGlobalUncompressedBufSz](#) (unsigned long sz)
- virtual [DescriptorTimeMap](#) & [getDtm](#) ()
- virtual void [clear](#) ()
clear the shot container

Protected Member Functions

- void [decode](#) (char *globalMedBuf, unsigned int globalMedBufSz, std::vector< unsigned int > &nLocalDesc, unsigned char cdvsDescriptorMode, unsigned short origImgX, unsigned short origImgY, unsigned short histoMapSzX, unsigned short histoMapSzY, bool hasBitSel, bool hasVar, bool hasRelB)
- void [reconstructGlobalDifferences](#) (char *uncompressedData, unsigned int uncompressedSz, int nrGDesc, bool hasVar, bool globalHasBitSelection, std::vector< unsigned int > &nLocalDesc)
- mpeg7cdvs::FeatureList [readFeatureListFromBinaryAbs](#) (unsigned char *buffer, unsigned int bufferSz, bool readRelevance, unsigned int nrElemGr)
- int [numGlobFctPresent](#) (const mpeg7cdvs::CdvsDescriptor &desc) const
- int [nBitsSet](#) (unsigned char byte) const

Protected Attributes

- unsigned char * [globalBuf](#)
buffer for differential global descriptors
- unsigned char * [localBuf](#)
buffer for local descriptors
- unsigned char * [histoBuf](#)
buffer for histogram descriptors
- unsigned long [globalBufSz](#)
size of encoded global buffer
- unsigned long [localBufSz](#)
size of encoded local buffer
- unsigned long [histoBufSz](#)
size of encoded histogram buffer
- unsigned long [globalUncompressedBufSz](#)

- size of unencoded global buffer*
- unsigned long [localUncompressedBufSz](#)
size of unencoded local buffer
- [DescriptorTimeMap dtm](#)
map for frame - descriptor association
- const mpeg7cdvs::Parameters & [params](#)
parameters needed to invoke histogram decoding

Static Protected Attributes

- static const int [SHOT_HEADER_SIZE](#) = 35

Additional Inherited Members

7.7.1 Detailed Description

Extension of segment descriptor container to also hold compressed data for a segment.

7.7.2 Constructor & Destructor Documentation

7.7.2.1 CompressedSegmentDescriptor()

```
mpeg7cdva::CompressedSegmentDescriptor::CompressedSegmentDescriptor (
    const mpeg7cdvs::Parameters & params )
```

7.7.2.2 ~CompressedSegmentDescriptor()

```
virtual mpeg7cdva::CompressedSegmentDescriptor::~~CompressedSegmentDescriptor ( ) [virtual]
```

7.7.3 Member Function Documentation

7.7.3.1 allocateGlobalBuffer()

```
virtual unsigned char* mpeg7cdva::CompressedSegmentDescriptor::allocateGlobalBuffer (
    unsigned long sz ) [virtual]
```

7.7.3.2 allocateHistoBuffer()

```
virtual unsigned char* mpeg7cdva::CompressedSegmentDescriptor::allocateHistoBuffer (
    unsigned long sz ) [virtual]
```

7.7.3.3 allocateLocalBuffer()

```
virtual unsigned char* mpeg7cdva::CompressedSegmentDescriptor::allocateLocalBuffer (
    unsigned long sz ) [virtual]
```

7.7.3.4 clear()

```
virtual void mpeg7cdva::CompressedSegmentDescriptor::clear ( ) [virtual]
```

clear the shot container

Reimplemented from [mpeg7cdva::SegmentDescriptor](#).

7.7.3.5 decode()

```
void mpeg7cdva::CompressedSegmentDescriptor::decode (
    char * globalMedBuf,
    unsigned int globalMedBufSz,
    std::vector< unsigned int > & nLocalDesc,
    unsigned char cdvsDescriptorMode,
    unsigned short origImgX,
    unsigned short origImgY,
    unsigned short histoMapSzX,
    unsigned short histoMapSzY,
    bool hasBitSel,
    bool hasVar,
    bool hasRelB ) [protected]
```

Parameters

<i>nLocalDesc</i>	decode compressed descriptor structure
-------------------	--

7.7.3.6 getDtm()

```
virtual DescriptorTimeMap& mpeg7cdva::CompressedSegmentDescriptor::getDtm ( ) [inline], [virtual]
```

References [mpeg7cdva::SegmentDescriptor::clear\(\)](#).

Here is the call graph for this function:

7.7.3.7 getGlobalBuffer()

```
virtual unsigned char* mpeg7cdva::CompressedSegmentDescriptor::getGlobalBuffer ( ) [inline],  
[virtual]
```

7.7.3.8 getGlobalBufSz()

```
virtual unsigned long mpeg7cdva::CompressedSegmentDescriptor::getGlobalBufSz ( ) [inline],  
[virtual]
```

7.7.3.9 getHistoBuffer()

```
virtual unsigned char* mpeg7cdva::CompressedSegmentDescriptor::getHistoBuffer ( ) [inline],  
[virtual]
```

7.7.3.10 getHistoBufSz()

```
virtual unsigned long mpeg7cdva::CompressedSegmentDescriptor::getHistoBufSz ( ) [inline],  
[virtual]
```

7.7.3.11 getLocalBuffer()

```
virtual unsigned char* mpeg7cdva::CompressedSegmentDescriptor::getLocalBuffer ( ) [inline],  
[virtual]
```

7.7.3.12 getLocalBufSz()

```
virtual unsigned long mpeg7cdva::CompressedSegmentDescriptor::getLocalBufSz ( ) [inline],  
[virtual]
```

7.7.3.13 nBitsSet()

```
int mpeg7cdva::CompressedSegmentDescriptor::nBitsSet (  
    unsigned char byte ) const [inline], [protected]
```

References `mpeg7cdva::SegmentDescriptor::read()`, and `mpeg7cdva::SegmentDescriptor::write()`.

Here is the call graph for this function:

7.7.3.14 numGlobFctPresent()

```
int mpeg7cdva::CompressedSegmentDescriptor::numGlobFctPresent (
    const mpeg7cdvs::CdvsDescriptor & desc ) const [inline], [protected]
```

7.7.3.15 read()

```
virtual size_t mpeg7cdva::CompressedSegmentDescriptor::read (
    std::ifstream & fin,
    mpeg7cdvs::CdvsServer * cdvsserver ) [virtual]
```

read (from the current position) this segment from a file

Reimplemented from [mpeg7cdva::SegmentDescriptor](#).

7.7.3.16 readFeatureListFromBinaryAbs()

```
mpeg7cdvs::FeatureList mpeg7cdva::CompressedSegmentDescriptor::readFeatureListFromBinaryAbs (
    unsigned char * buffer,
    unsigned int bufferSize,
    bool readRelevance,
    unsigned int nrElemGr ) [protected]
```

7.7.3.17 reconstructGlobalDifferences()

```
void mpeg7cdva::CompressedSegmentDescriptor::reconstructGlobalDifferences (
    char * uncompressedData,
    unsigned int uncompressedSz,
    int nrGDesc,
    bool hasVar,
    bool globalHasBitSelection,
    std::vector< unsigned int > & nLocalDesc ) [protected]
```

7.7.3.18 setGlobalUncompressedBufSz()

```
virtual void mpeg7cdva::CompressedSegmentDescriptor::setGlobalUncompressedBufSz (
    unsigned long sz ) [inline], [virtual]
```

7.7.3.19 setHistoBufSz()

```
virtual void mpeg7cdva::CompressedSegmentDescriptor::setHistoBufSz (
    unsigned long sz ) [inline], [virtual]
```

7.7.3.20 setLocalUncompressedBufSz()

```
virtual void mpeg7cdva::CompressedSegmentDescriptor::setLocalUncompressedBufSz (
    unsigned long sz ) [inline], [virtual]
```

7.7.3.21 write()

```
virtual size_t mpeg7cdva::CompressedSegmentDescriptor::write (
    std::ostream & fout ) const [virtual]
```

write (appending) this segment to a file

Reimplemented from [mpeg7cdva::SegmentDescriptor](#).

7.7.4 Field Documentation

7.7.4.1 dtm

[DescriptorTimeMap](#) mpeg7cdva::CompressedSegmentDescriptor::dtm [protected]

map for frame - descriptor association

7.7.4.2 globalBuf

unsigned char* mpeg7cdva::CompressedSegmentDescriptor::globalBuf [protected]

buffer for differential global descriptors

7.7.4.3 globalBufSz

unsigned long mpeg7cdva::CompressedSegmentDescriptor::globalBufSz [protected]

size of encoded global buffer

7.7.4.4 globalUncompressedBufSz

unsigned long mpeg7cdva::CompressedSegmentDescriptor::globalUncompressedBufSz [protected]

size of unencoded global buffer

7.7.4.5 histoBuf

unsigned char* mpeg7cdva::CompressedSegmentDescriptor::histoBuf [protected]

buffer for histogram descriptors

7.7.4.6 histoBufSz

unsigned long mpeg7cdva::CompressedSegmentDescriptor::histoBufSz [protected]

size of encoded histogram buffer

7.7.4.7 localBuf

unsigned char* mpeg7cdva::CompressedSegmentDescriptor::localBuf [protected]

buffer for local descriptors

7.7.4.8 localBufSz

unsigned long mpeg7cdva::CompressedSegmentDescriptor::localBufSz [protected]

size of encoded local buffer

7.7.4.9 localUncompressedBufSz

unsigned long mpeg7cdva::CompressedSegmentDescriptor::localUncompressedBufSz [protected]

size of unencoded local buffer

7.7.4.10 params

```
const mpeg7cdvs::Parameters& mpeg7cdva::CompressedSegmentDescriptor::params [protected]
```

parameters needed to invoke histogram decoding

7.7.4.11 SHOT_HEADER_SIZE

```
const int mpeg7cdva::CompressedSegmentDescriptor::SHOT_HEADER_SIZE = 35 [static], [protected]
```

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

- [Cdvalmpl.h](#)

7.8 mpeg7cdva::CoordinateCoding Class Reference

```
#include <CoordinateCoding.h>
```

Data Structures

- struct [CircularSumContext](#)
Basic structure for Cssc.

Public Member Functions

- [CoordinateCoding](#) (const Parameters ¶m)
Constructor using the given parameters to set the [CsscCoordinateCoding](#) behaviour.
- virtual [~CoordinateCoding](#) ()
- void [toBinary](#) (BitOutputStream &writer, bool writeSize=true)
Convert the stored information into a binary stream.
- void [fromBinary](#) (BitInputStream &reader)
Convert a binary stream into the stored information.
- void [fromBinary](#) (BitInputStream &reader, unsigned int hMapSizeX, unsigned int hMapSizeY)
Convert a binary stream into the stored information.
- int [compare](#) (const [CoordinateCoding](#) &other)
Compare this instance with another one.
- void [exportVars](#) (unsigned int &histogramCountSize, unsigned int &histogramMapSizeX, unsigned int &histogramMapSizeY) const
Export the value of the histogram count and size.
- void [generateHistogramMap](#) (FeatureList &featurelist, int numPoints)
Generation of new matrix representation based on circular scanning.
- void [generateHistogramMap](#) (std::vector< [RefFeature](#) > &refFeaturelist, FeatureList &featurelist, int offset, int frameldx, [DescriptorTimeMap](#) &dtm, int numPoints, int xSize, int ySize)
Generation of new matrix representation based on circular scanning.
- void [generateFeatureList](#) (FeatureList &descriptors)

Reconstruction of the original histogram map starting from circular scanning representation.

- void [StartTrainingMode](#) ()
- void [EndTrainingMode](#) ()
- int [AddImageSample](#) (FeatureList &featurelist)
- int [writeSeparateContext](#) (char *filename)
- int [readSeparateContext](#) (char *filename)
- void [setVerbose](#) ()
- void [difference](#) (CoordinateCoding &other)

Set object to be the difference of the current histogram and the one passed as argument same image size is assumed call after generateHistogramMap2.

- void [generateHistogramMap2](#) (FeatureList &featurelist, int numPoints)
- void [setIsOK](#) (bool ok)

Static Public Member Functions

- static int [readSeparateContext](#) (char *filename, [CircularSumContext](#) &cCsc)

Static Public Attributes

- static const int [SUM_HIST_COUNT_SIZE](#) = 64
Histogram count context lenght.
- static const int [CONTEXT_RANGE](#) = 5
Sum-based context range.
- static const int [MAXIMUM_SUM_CONTEXT](#) = (2*[CONTEXT_RANGE](#)*[CONTEXT_RANGE](#) + [CONTEXT_RANGE](#))
Maximum value of sum context.

7.8.1 Constructor & Destructor Documentation

7.8.1.1 CoordinateCoding()

```
mpeg7cdva::CoordinateCoding::CoordinateCoding (
    const Parameters & param )
```

Constructor using the given parameters to set the [CsscCoordinateCoding](#) behaviour.

Parameters

<i>param</i>	the set of parameters to initialize this object.
--------------	--

7.8.1.2 ~CoordinateCoding()

```
virtual mpeg7cdva::CoordinateCoding::~~CoordinateCoding ( ) [virtual]
```

7.8.2 Member Function Documentation

7.8.2.1 AddImageSample()

```
int mpeg7cdva::CoordinateCoding::AddImageSample (
    FeatureList & featurelist )
```

7.8.2.2 compare()

```
int mpeg7cdva::CoordinateCoding::compare (
    const CoordinateCoding & other )
```

Compare this instance with another one.

Parameters

<i>other</i>	the other instance.
--------------	---------------------

Returns

0 if equal, or the number of different values if different.

7.8.2.3 difference()

```
void mpeg7cdva::CoordinateCoding::difference (
    CoordinateCoding & other )
```

Set object to be the difference of the current histogram and the one passed as argument same image size is assumed call after generateHistogramMap2.

7.8.2.4 EndTrainingMode()

```
void mpeg7cdva::CoordinateCoding::EndTrainingMode ( )
```

7.8.2.5 exportVars()

```
void mpeg7cdva::CoordinateCoding::exportVars (
    unsigned int & histogramCountSize,
    unsigned int & histogramMapSizeX,
    unsigned int & histogramMapSizeY ) const
```

Export the value of the histogram count and size.

Parameters

<i>histogramCountSize</i>	size of histogram count
<i>histogramMapSizeX</i>	size of histogram map (X)
<i>histogramMapSizeY</i>	size of histogram map (Y)

7.8.2.6 fromBinary() [1/2]

```
void mpeg7cdva::CoordinateCoding::fromBinary (
    BitInputStream & reader )
```

Convert a binary stream into the stored information.

Parameters

<i>reader</i>	the bitstream reader object.
---------------	------------------------------

7.8.2.7 fromBinary() [2/2]

```
void mpeg7cdva::CoordinateCoding::fromBinary (
    BitInputStream & reader,
    unsigned int hMapSizeX,
    unsigned int hMapSizeY )
```

Convert a binary stream into the stored information.

In this case the buffer does not contain histo X and Y size, but they are specified as parameters.

Parameters

<i>reader</i>	the bitstream reader object.

7.8.2.8 generateFeatureList()

```
void mpeg7cdva::CoordinateCoding::generateFeatureList (
    FeatureList & descriptors )
```

Reconstruction of the original histogram map starting from circular scanning representation.

Parameters

<i>descriptors</i>	reconstructed list of keypoints and descriptors
--------------------	---

7.8.2.9 generateHistogramMap() [1/2]

```
void mpeg7cdva::CoordinateCoding::generateHistogramMap (
    FeatureList & featurelist,
    int numPoints )
```

Generation of new matrix representation based on circular scanning.

Parameters

<i>featurelist</i>	list of keypoints.
<i>numPoints</i>	the number of features to encode (only the first numPoint features in featurelist will be encoded)

7.8.2.10 generateHistogramMap() [2/2]

```
void mpeg7cdva::CoordinateCoding::generateHistogramMap (
    std::vector< RefFeature > & refFeaturelist,
    FeatureList & featurelist,
    int offset,
    int frameIdx,
    DescriptorTimeMap & dtm,
    int numPoints,
    int xSize,
    int ySize )
```

Generation of new matrix representation based on circular scanning.

keeps order specified by ref feature list (using populated entries only)

Parameters

<i>featurelist</i>	list of keypoints.
<i>numPoints</i>	the number of features to encode (only the first numPoint features in featurelist will be encoded)

7.8.2.11 generateHistogramMap2()

```
void mpeg7cdva::CoordinateCoding::generateHistogramMap2 (
    FeatureList & featurelist,
    int numPoints )
```

7.8.2.12 readSeparateContext() [1/2]

```
static int mpeg7cdva::CoordinateCoding::readSeparateContext (
    char * filename,
    CircularSumContext & cCsc ) [static]
```

7.8.2.13 readSeparateContext() [2/2]

```
int mpeg7cdva::CoordinateCoding::readSeparateContext (
    char * filename )
```

7.8.2.14 setIsOK()

```
void mpeg7cdva::CoordinateCoding::setIsOK (
    bool ok ) [inline]
```

7.8.2.15 setVerbose()

```
void mpeg7cdva::CoordinateCoding::setVerbose ( ) [inline]
```

7.8.2.16 StartTrainingMode()

```
void mpeg7cdva::CoordinateCoding::StartTrainingMode ( )
```

7.8.2.17 toBinary()

```
void mpeg7cdva::CoordinateCoding::toBinary (
    BitOutputStream & writer,
    bool writeSize = true )
```

Convert the stored information into a binary stream.

Parameters

<i>writer</i>	the bitstream writer object.
---------------	------------------------------

7.8.2.18 writeSeparateContext()

```
int mpeg7cdva::CoordinateCoding::writeSeparateContext (
    char * filename )
```

7.8.3 Field Documentation

7.8.3.1 CONTEXT_RANGE

```
const int mpeg7cdva::CoordinateCoding::CONTEXT_RANGE = 5 [static]
```

Sum-based context range.

7.8.3.2 MAXIMUM_SUM_CONTEXT

```
const int mpeg7cdva::CoordinateCoding::MAXIMUM_SUM_CONTEXT = (2*CONTEXT_RANGE*CONTEXT_RANGE +
CONTEXT_RANGE) [static]
```

Maximum value of sum context.

7.8.3.3 SUM_HIST_COUNT_SIZE

```
const int mpeg7cdva::CoordinateCoding::SUM_HIST_COUNT_SIZE = 64 [static]
```

Histogram count context lenght.

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

- [CoordinateCoding.h](#)

7.9 mpeg7cdva::CoordList Class Reference

```
#include <CompressedFeatureList.h>
```

Inheritance diagram for mpeg7cdva::CoordList:

Public Member Functions

- [CoordList](#) ()
- [CoordList](#) (mpeg7cdvs::FeatureList &fl)
- virtual unsigned int [getX](#) (int i)=0
- virtual unsigned int [getY](#) (int i)=0

7.9.1 Constructor & Destructor Documentation

7.9.1.1 CoordList() [1/2]

```
mpeg7cdva::CoordList::CoordList ( ) [inline]
```

References [getX\(\)](#), and [getY\(\)](#).

Here is the call graph for this function:

7.9.1.2 CoordList() [2/2]

```
mpeg7cdva::CoordList::CoordList (
    mpeg7cdvs::FeatureList & fl )
```

7.9.2 Member Function Documentation

7.9.2.1 getX()

```
virtual unsigned int mpeg7cdva::CoordList::getX (
    int i ) [pure virtual]
```

Implemented in [mpeg7cdva::LeanCoordList](#), and [mpeg7cdva::WrappedCoordList](#).

Referenced by [CoordList\(\)](#).

7.9.2.2 getY()

```
virtual unsigned int mpeg7cdva::CoordList::getY (
    int i ) [pure virtual]
```

Implemented in [mpeg7cdva::LeanCoordList](#), and [mpeg7cdva::WrappedCoordList](#).

Referenced by [CoordList\(\)](#).

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

- [CompressedFeatureList.h](#)

7.10 CsscCoordinateCoding Class Reference

Class that converts the coordinates of all descriptors of an image into a bitstream, and vice versa.

```
#include <CoordinateCoding.h>
```

7.10.1 Detailed Description

Class that converts the coordinates of all descriptors of an image into a bitstream, and vice versa.

Date

2012

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

- [CoordinateCoding.h](#)

7.11 mpeg7cdva::DescriptorTimeMap Class Reference

```
#include <DescriptorTimeMap.h>
```

Public Member Functions

- [DescriptorTimeMap](#) ()
- [DescriptorTimeMap](#) ([DescriptorTimeMap](#) &other)
- [DescriptorTimeMap](#) (int nDescr, int nAbsDescr, int nFrames)
- [DescriptorTimeMap](#) (int nDescr, int nAbsDescr, int nFrames, char *buf)
construct from parsed buffer contents of buffer will be copied into buffer under ownership of class
- virtual [~DescriptorTimeMap](#) ()
- [DescriptorTimeMap](#) & [operator=](#) (const [DescriptorTimeMap](#) &other)
- void [init](#) (int nDescr, int nAbsDescr, int nFrames)
- void [add](#) (int descIdx, int frameIdx)
- bool [get](#) (int descIdx, int frameIdx)
- int [getCount](#) (int descIdx, int frameIdx)
- int [getFeatureStartIndex](#) (int descIdx, int frameIdx)
- int [getBitSize](#) () const
- char * [getBuffer](#) () const
- int [getNDescr](#) () const
- int [getNAbsDescr](#) () const
- int [getNFrames](#) () const
- std::vector< unsigned short > & [getDescIndex](#) (int frameIdx)
- void [addDescIndex](#) (std::vector< unsigned short > &di, int frame)
- void [writeDescIndex](#) (unsigned char *buffer, bool compressed, int &outIdx) const
- void [parseDescIndex](#) (unsigned char *buffer, std::vector< unsigned int > &nLocalDesc, bool compressed)
- void [clear](#) ()

Data Fields

- `std::vector< int >` [ldOffsets](#)

Protected Attributes

- `char *` [m_buffer](#)
- `int` [m_nDescr](#)
- `int` [m_nAbsDescr](#)
- `int` [m_nFrames](#)
- `int` [m_nbits](#)
- `std::vector< std::vector< unsigned short > >` [descIndex](#)

7.11.1 Constructor & Destructor Documentation

7.11.1.1 DescriptorTimeMap() [1/4]

```
mpeg7cdva::DescriptorTimeMap::DescriptorTimeMap ( )
```

7.11.1.2 DescriptorTimeMap() [2/4]

```
mpeg7cdva::DescriptorTimeMap::DescriptorTimeMap (
    DescriptorTimeMap & other )
```

7.11.1.3 DescriptorTimeMap() [3/4]

```
mpeg7cdva::DescriptorTimeMap::DescriptorTimeMap (
    int nDescr,
    int nAbsDescr,
    int nFrames )
```

7.11.1.4 DescriptorTimeMap() [4/4]

```
mpeg7cdva::DescriptorTimeMap::DescriptorTimeMap (
    int nDescr,
    int nAbsDescr,
    int nFrames,
    char * buf )
```

construct from parsed buffer contents of buffer will be copied into buffer under ownership of class

7.11.1.5 ~DescriptorTimeMap()

```
virtual mpeg7cdva::DescriptorTimeMap::~~DescriptorTimeMap ( ) [virtual]
```

7.11.2 Member Function Documentation

7.11.2.1 add()

```
void mpeg7cdva::DescriptorTimeMap::add (
    int descIdx,
    int frameIdx ) [inline]
```

7.11.2.2 addDescIndex()

```
void mpeg7cdva::DescriptorTimeMap::addDescIndex (
    std::vector< unsigned short > & di,
    int frame )
```

Referenced by getDescIndex().

7.11.2.3 clear()

```
void mpeg7cdva::DescriptorTimeMap::clear ( )
```

Referenced by getDescIndex().

7.11.2.4 get()

```
bool mpeg7cdva::DescriptorTimeMap::get (
    int descIdx,
    int frameIdx ) [inline]
```

References getCount().

Here is the call graph for this function:

7.11.2.5 getBitSize()

```
int mpeg7cdva::DescriptorTimeMap::getBitSize ( ) const [inline]
```

References `m_nbits`.

7.11.2.6 getBuffer()

```
char* mpeg7cdva::DescriptorTimeMap::getBuffer ( ) const [inline]
```

References `m_buffer`.

7.11.2.7 getCount()

```
int mpeg7cdva::DescriptorTimeMap::getCount (
    int descIdx,
    int frameIdx ) [inline]
```

Referenced by `get()`, and `getFeatureStartIndex()`.

7.11.2.8 getDescIndex()

```
std::vector<unsigned short>& mpeg7cdva::DescriptorTimeMap::getDescIndex (
    int frameIdx ) [inline]
```

References `addDescIndex()`, `clear()`, `parseDescIndex()`, and `writeDescIndex()`.

Here is the call graph for this function:

7.11.2.9 getFeatureStartIndex()

```
int mpeg7cdva::DescriptorTimeMap::getFeatureStartIndex (
    int descIdx,
    int frameIdx ) [inline]
```

References `getCount()`.

Here is the call graph for this function:

7.11.2.10 getNAbsDescr()

```
int mpeg7cdva::DescriptorTimeMap::getNAbsDescr ( ) const [inline]
```

References `m_nAbsDescr`.

7.11.2.11 getNDescr()

```
int mpeg7cdva::DescriptorTimeMap::getNDescr ( ) const [inline]
```

References `m_nDescr`.

7.11.2.12 getNFrames()

```
int mpeg7cdva::DescriptorTimeMap::getNFrames ( ) const [inline]
```

References `m_nFrames`.

7.11.2.13 init()

```
void mpeg7cdva::DescriptorTimeMap::init (
    int nDescr,
    int nAbsDescr,
    int nFrames )
```

7.11.2.14 operator=()

```
DescriptorTimeMap& mpeg7cdva::DescriptorTimeMap::operator= (
    const DescriptorTimeMap & other )
```

7.11.2.15 parseDescIndex()

```
void mpeg7cdva::DescriptorTimeMap::parseDescIndex (
    unsigned char * buffer,
    std::vector< unsigned int > & nLocalDesc,
    bool compressed )
```

Referenced by `getDescIndex()`.

7.11.2.16 writeDescIndex()

```
void mpeg7cdva::DescriptorTimeMap::writeDescIndex (
    unsigned char * buffer,
    bool compressed,
    int & outIdx ) const
```

Referenced by `getDescIndex()`.

7.11.3 Field Documentation

7.11.3.1 descIndex

`std::vector< std::vector<unsigned short> > mpeg7cdva::DescriptorTimeMap::descIndex` [protected]

7.11.3.2 ldOffsets

`std::vector<int> mpeg7cdva::DescriptorTimeMap::ldOffsets`

7.11.3.3 m_buffer

`char* mpeg7cdva::DescriptorTimeMap::m_buffer` [protected]

Referenced by `getBuffer()`.

7.11.3.4 m_nAbsDescr

`int mpeg7cdva::DescriptorTimeMap::m_nAbsDescr` [protected]

Referenced by `getNAbsDescr()`.

7.11.3.5 m_nbits

`int mpeg7cdva::DescriptorTimeMap::m_nbits` [protected]

Referenced by `getBitSize()`.

7.11.3.6 m_nDescr

`int mpeg7cdva::DescriptorTimeMap::m_nDescr` [protected]

Referenced by `getNDescr()`.

7.11.3.7 m_nFrames

```
int mpeg7cdva::DescriptorTimeMap::m_nFrames [protected]
```

Referenced by getNFrames().

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

- [DescriptorTimeMap.h](#)

7.12 evx::entropy_coder Class Reference

```
#include <cabac.h>
```

Public Member Functions

- [entropy_coder](#) ()
- [entropy_coder](#) (uint32 input_model)
- void [clear](#) ()
- [evx_status](#) [encode](#) (bitstream *source, bitstream *dest, bool auto_finish=true)
- [evx_status](#) [decode](#) (uint32 symbol_count, bitstream *source, bitstream *dest, bool auto_start=true)
- [evx_status](#) [start_decode](#) (bitstream *source)
- [evx_status](#) [finish_encode](#) (bitstream *dest)

7.12.1 Constructor & Destructor Documentation

7.12.1.1 entropy_coder() [1/2]

```
evx::entropy_coder::entropy_coder ( )
```

7.12.1.2 entropy_coder() [2/2]

```
evx::entropy_coder::entropy_coder (
    uint32 input_model ) [explicit]
```

7.12.2 Member Function Documentation

7.12.2.1 clear()

```
void evx::entropy_coder::clear ( )
```

7.12.2.2 decode()

```
evx_status evx::entropy_coder::decode (
    uint32 symbol_count,
    bitstream * source,
    bitstream * dest,
    bool auto_start = true )
```

7.12.2.3 encode()

```
evx_status evx::entropy_coder::encode (
    bitstream * source,
    bitstream * dest,
    bool auto_finish = true )
```

7.12.2.4 finish_encode()

```
evx_status evx::entropy_coder::finish_encode (
    bitstream * dest )
```

7.12.2.5 start_decode()

```
evx_status evx::entropy_coder::start_decode (
    bitstream * source )
```

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

- [cabac.h](#)

7.13 mpeg7cdva::ExtractData Class Reference

A class containing the results of an extraction operation.

```
#include <cdva.h>
```

Public Member Functions

- [ExtractData](#) ()
- void [setVideoDuration](#) (double time)
set the video duration in seconds.
- void [setNumFrames](#) (double nframes)
set the number of frames of the video clip.
- void [setNumShots](#) (double nshots)
set the number of shots of the video clip.
- void [setDescriptorLength](#) (double length)
set the actual descriptor length (in bytes).

Data Fields

- double [descriptorlength](#)
- double [clip_duration](#)
- double [numframes](#)
- double [numshots](#)
- int [header_bit_count](#)
- int [local_bit_count](#)
- int [global_bit_count](#)
- int [coordinate_bit_count](#)
- int [n_keyframes](#)

7.13.1 Detailed Description

A class containing the results of an extraction operation.

7.13.2 Constructor & Destructor Documentation

7.13.2.1 ExtractData()

```
mpeg7cdva::ExtractData::ExtractData ( ) [inline]
```

7.13.3 Member Function Documentation

7.13.3.1 setDescriptorLength()

```
void mpeg7cdva::ExtractData::setDescriptorLength (
    double length ) [inline]
```

set the actual descriptor length (in bytes).

Parameters

<i>length</i>	the size in bytes of the encoded descriptor.
---------------	--

7.13.3.2 setNumFrames()

```
void mpeg7cdva::ExtractData::setNumFrames (
    double nframes ) [inline]
```

set the number of frames of the video clip.

Parameters

<i>nframes</i>	number of frames of the video clip.
----------------	-------------------------------------

7.13.3.3 setNumShots()

```
void mpeg7cdva::ExtractData::setNumShots (
    double nshots ) [inline]
```

set the number of shots of the video clip.

Parameters

<i>nshots</i>	number of shots of the video clip.
---------------	------------------------------------

7.13.3.4 setVideoDuration()

```
void mpeg7cdva::ExtractData::setVideoDuration (
    double time ) [inline]
```

set the video duration in seconds.

Parameters

<i>time</i>	the time in seconds
-------------	---------------------

7.13.4 Field Documentation

7.13.4.1 clip_duration

```
double mpeg7cdva::ExtractData::clip_duration
```

7.13.4.2 coordinate_bit_count

```
int mpeg7cdva::ExtractData::coordinate_bit_count
```

7.13.4.3 descriptorlength

```
double mpeg7cdva::ExtractData::descriptorlength
```

7.13.4.4 global_bit_count

```
int mpeg7cdva::ExtractData::global_bit_count
```

7.13.4.5 header_bit_count

```
int mpeg7cdva::ExtractData::header_bit_count
```

7.13.4.6 local_bit_count

```
int mpeg7cdva::ExtractData::local_bit_count
```

7.13.4.7 n_keyframes

```
int mpeg7cdva::ExtractData::n_keyframes
```

7.13.4.8 numframes

```
double mpeg7cdva::ExtractData::numframes
```

7.13.4.9 numshots

```
double mpeg7cdva::ExtractData::numshots
```

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

- [cdva.h](#)

7.14 mpeg7cdva::FileManager Class Reference

Helper class to manage lists of file names.

```
#include <FileManager.h>
```

Public Member Functions

- [FileManager](#) (const char *annotationpathname, int level=0)
Read the list of images from the given annotation file.
- virtual [~FileManager](#) ()
- std::string [getDatasetPath](#) () const
Get the dataset base directory.
- std::string [getDatasetName](#) () const
Get the dataset name.
- std::string [getDatasetPathName](#) () const
Get the dataset full pathname.
- void [setWorkspaceDir](#) (const char *workdir)
Set the workspace directory.
- std::string [getWorkspaceDir](#) () const
Get the workspace directory.
- size_t [getDatasetSize](#) () const
Get the dataset size.
- std::string [replaceExt](#) (const std::string &imageName, const char *ext) const
Convert a pathname into a pathname with the given extension.
- std::string [getQueryName](#) (size_t i, bool absolutePathname=true) const
Get the first image name found at the i-th position in the annotation file.
- std::string [getReferenceName](#) (size_t i, bool absolutePathname=true) const
Get the second image name found at the i-th position in the annotation file.
- int [countNames](#) (size_t i) const
Count how many pathnames are contained in the i-th line.

Static Public Member Functions

- static std::string [replacePath](#) (const std::string &imageName, const char *newpath)
Convert a pathname into a pathname with the given new path.

7.14.1 Detailed Description

Helper class to manage lists of file names.

Author

Massimo Balestri

Date

2012

7.14.2 Constructor & Destructor Documentation

7.14.2.1 FileManager()

```
mpeg7cdva::FileManager::FileManager (
    const char * annotationpathname,
    int level = 0 )
```

Read the list of images from the given annotation file.

Parameters

<i>annotationpathname</i>	the pathname of the annotation text file containing the list of images.
<i>level</i>	the recursion level (to avoid infinite loops); must be zero when called the first time.

7.14.2.2 ~FileManager()

```
virtual mpeg7cdva::FileManager::~~FileManager ( ) [virtual]
```

7.14.3 Member Function Documentation

7.14.3.1 countNames()

```
int mpeg7cdva::FileManager::countNames (
    size_t i ) const
```

Count how many pathnames are contained in the *i*-th line.

Parameters

<i>i</i>	the index of the image in the annotation file.
----------	--

Returns

the number of pathnames found.

7.14.3.2 getDatasetName()

```
std::string mpeg7cdva::FileManager::getDatasetName ( ) const
```

Get the dataset name.

Returns

the dataset name

7.14.3.3 getDatasetPath()

```
std::string mpeg7cdva::FileManager::getDatasetPath ( ) const
```

Get the dataset base directory.

Returns

the dataset path

7.14.3.4 getDatasetPathName()

```
std::string mpeg7cdva::FileManager::getDatasetPathName ( ) const
```

Get the dataset full pathname.

Returns

the dataset pathname

7.14.3.5 `getDatasetSize()`

```
size_t mpeg7cdva::FileManager::getDatasetSize ( ) const
```

Get the dataset size.

Returns

the number of lines read from the filename.

7.14.3.6 `getQueryName()`

```
std::string mpeg7cdva::FileManager::getQueryName (
    size_t i,
    bool absolutePathname = true ) const
```

Get the first image name found at the i-th position in the annotation file.

The image name is provided as an absolute pathname.

Parameters

<i>i</i>	the index of the image in the annotation file.
<i>absolutePathname</i>	if true, the absolute pathname of the file is returned; otherwise, the relative pathname is returned.

Returns

the relative or absolute pathname of the image.

7.14.3.7 `getReferenceName()`

```
std::string mpeg7cdva::FileManager::getReferenceName (
    size_t i,
    bool absolutePathname = true ) const
```

Get the second image name found at the i-th position in the annotation file.

The image name is provided as an absolute pathname.

Parameters

<i>i</i>	the index of the image in the annotation file.
<i>absolutePathname</i>	if true, the absolute pathname of the file is returned; otherwise, the relative pathname is returned.

Returns

the relative or absolute pathname of the image.

7.14.3.8 getWorkspaceDir()

```
std::string mpeg7cdva::FileManager::getWorkspaceDir ( ) const
```

Get the workspace directory.

This is the directory where output files will be stored.

Returns

the workspace directory

7.14.3.9 replaceExt()

```
std::string mpeg7cdva::FileManager::replaceExt (
    const std::string & imageName,
    const char * ext ) const
```

Convert a pathname into a pathname with the given extension.

Parameters

<i>imageName</i>	the original image/video name;
<i>ext</i>	new extension;

Returns

the modified pathname.

7.14.3.10 replacePath()

```
static std::string mpeg7cdva::FileManager::replacePath (
    const std::string & imageName,
    const char * newpath ) [static]
```

Convert a pathname into a pathname with the given new path.

Parameters

<i>imageName</i>	the original image name;
<i>newpath</i>	the new path;

Returns

the modified pathname.

7.14.3.11 `setWorkspaceDir()`

```
void mpeg7cdva::FileManager::setWorkspaceDir (
    const char * workdir )
```

Set the workspace directory.

This is the directory where output files will be stored.

Parameters

<i>workdir</i>	the workspace directory
----------------	-------------------------

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

- [FileManager.h](#)

7.15 `mpeg7cdva::LeanCoordList` Class Reference

```
#include <CompressedFeatureList.h>
```

Inheritance diagram for `mpeg7cdva::LeanCoordList`:

Collaboration diagram for `mpeg7cdva::LeanCoordList`:

Public Member Functions

- [LeanCoordList](#) ()
- [LeanCoordList](#) (int size)
- [LeanCoordList](#) (mpeg7cdvs::FeatureList &fl)
- [LeanCoordList](#) (const [LeanCoordList](#) &other)
- [LeanCoordList](#) & `operator=` (const [LeanCoordList](#) &other)
- virtual `~LeanCoordList` ()
- virtual unsigned int [getX](#) (int i)
- virtual unsigned int [getY](#) (int i)
- virtual void [setX](#) (int i, unsigned int x)
- virtual void [setY](#) (int i, unsigned int y)
- virtual int [getLength](#) ()
- virtual mpeg7cdvs::FeatureList & [getFeatureList](#) ()
- virtual void [getFeatureList](#) (mpeg7cdvs::FeatureList &fl)

Protected Attributes

- unsigned int * [xCoord](#)
- unsigned int * [yCoord](#)
- int [nFeatures](#)

7.15.1 Constructor & Destructor Documentation

7.15.1.1 LeanCoordList() [1/4]

```
mpeg7cdva::LeanCoordList::LeanCoordList ( )
```

7.15.1.2 LeanCoordList() [2/4]

```
mpeg7cdva::LeanCoordList::LeanCoordList (
    int size )
```

7.15.1.3 LeanCoordList() [3/4]

```
mpeg7cdva::LeanCoordList::LeanCoordList (
    mpeg7cdvs::FeatureList & fl )
```

7.15.1.4 LeanCoordList() [4/4]

```
mpeg7cdva::LeanCoordList::LeanCoordList (
    const LeanCoordList & other )
```

7.15.1.5 ~LeanCoordList()

```
virtual mpeg7cdva::LeanCoordList::~~LeanCoordList ( ) [virtual]
```

7.15.2 Member Function Documentation

7.15.2.1 `getFeatureList()` [1/2]

```
virtual mpeg7cdvs::FeatureList& mpeg7cdva::LeanCoordList::getFeatureList ( ) [virtual]
```

7.15.2.2 `getFeatureList()` [2/2]

```
virtual void mpeg7cdva::LeanCoordList::getFeatureList (
    mpeg7cdvs::FeatureList & fl ) [virtual]
```

7.15.2.3 `getLength()`

```
virtual int mpeg7cdva::LeanCoordList::getLength ( ) [inline], [virtual]
```

7.15.2.4 `getX()`

```
virtual unsigned int mpeg7cdva::LeanCoordList::getX (
    int i ) [inline], [virtual]
```

Implements [mpeg7cdva::CoordList](#).

7.15.2.5 `getY()`

```
virtual unsigned int mpeg7cdva::LeanCoordList::getY (
    int i ) [inline], [virtual]
```

Implements [mpeg7cdva::CoordList](#).

7.15.2.6 `operator=()`

```
LeanCoordList& mpeg7cdva::LeanCoordList::operator= (
    const LeanCoordList & other )
```

7.15.2.7 setX()

```
virtual void mpeg7cdva::LeanCoordList::setX (  
    int i,  
    unsigned int x )    [inline], [virtual]
```

7.15.2.8 setY()

```
virtual void mpeg7cdva::LeanCoordList::setY (  
    int i,  
    unsigned int y )    [inline], [virtual]
```

7.15.3 Field Documentation

7.15.3.1 nFeatures

```
int mpeg7cdva::LeanCoordList::nFeatures    [protected]
```

7.15.3.2 xCoord

```
unsigned int* mpeg7cdva::LeanCoordList::xCoord    [protected]
```

7.15.3.3 yCoord

```
unsigned int* mpeg7cdva::LeanCoordList::yCoord    [protected]
```

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

- [CompressedFeatureList.h](#)

7.16 mpeg7cdva::LogManager Class Reference

Helper class to produce log files in various formats (csv, text, XML, etc.)

```
#include <LogManager.h>
```

Public Member Functions

- [LogManager](#) ()
constructor
- virtual [~LogManager](#) ()
destructor
- void [init](#) (int formats, const std::string &datasetpath, const std::string &datasetname, size_t n_videos, int mode, int refmode=0)
initialization method used by the Evaluation Framework; do not change.
- void [printExtractHeader](#) ()
method used by the Evaluation Framework to produce log files; do not change.
- void [printExtractData](#) (int index, const std::string &videoname, const [ExtractData](#) &data)
method used by the Evaluation Framework to produce log files; do not change.
- void [printMatchHeader](#) ()
method used by the Evaluation Framework to produce log files; do not change.
- void [printMatchData](#) (int index, const std::string &queryvideoname, const std::string &refvideoname, const [MatchData](#) &matchData)
method used by the Evaluation Framework to produce log files; do not change.
- void [printRetrievalHeader](#) ()
method used by the Evaluation Framework to produce log files; do not change.
- void [printRetrievalData](#) (int index, const std::string &queryvideoname, const std::vector< [MatchData](#) > &retrievalResults)
method used by the Evaluation Framework to produce log files; do not change.
- void [close](#) ()

7.16.1 Detailed Description

Helper class to produce log files in various formats (csv, text, XML, etc.)

Author

Massimo Balestri

Date

2015

7.16.2 Constructor & Destructor Documentation

7.16.2.1 LogManager()

```
mpeg7cdva::LogManager::LogManager ( )
```

constructor

7.16.2.2 ~LogManager()

```
virtual mpeg7cdva::LogManager::~~LogManager ( ) [virtual]
```

destructor

7.16.3 Member Function Documentation

7.16.3.1 close()

```
void mpeg7cdva::LogManager::close ( )
```

7.16.3.2 init()

```
void mpeg7cdva::LogManager::init (
    int formats,
    const std::string & datasetpath,
    const std::string & datasetname,
    size_t n_videos,
    int mode,
    int refmode = 0 )
```

initialization method used by the Evaluation Framework; do not change.

7.16.3.3 printExtractData()

```
void mpeg7cdva::LogManager::printExtractData (
    int index,
    const std::string & videoname,
    const ExtractData & data )
```

method used by the Evaluation Framework to produce log files; do not change.

7.16.3.4 printExtractHeader()

```
void mpeg7cdva::LogManager::printExtractHeader ( )
```

method used by the Evaluation Framework to produce log files; do not change.

7.16.3.5 printMatchData()

```
void mpeg7cdva::LogManager::printMatchData (
    int index,
    const std::string & queryvideoname,
    const std::string & refvideoname,
    const MatchData & matchData )
```

method used by the Evaluation Framework to produce log files; do not change.

7.16.3.6 printMatchHeader()

```
void mpeg7cdva::LogManager::printMatchHeader ( )
```

method used by the Evaluation Framework to produce log files; do not change.

7.16.3.7 printRetrievalData()

```
void mpeg7cdva::LogManager::printRetrievalData (
    int index,
    const std::string & queryvideoname,
    const std::vector< MatchData > & retrievalResults )
```

method used by the Evaluation Framework to produce log files; do not change.

7.16.3.8 printRetrievalHeader()

```
void mpeg7cdva::LogManager::printRetrievalHeader ( )
```

method used by the Evaluation Framework to produce log files; do not change.

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

- [LogManager.h](#)

7.17 mpeg7cdva::MatchData Class Reference

A class containing the results of a matching or retrieval operation.

```
#include <cdva.h>
```

Public Member Functions

- [MatchData](#) ()
- virtual [~MatchData](#) ()
- void [setMatchingScore](#) (double myscore)
set the score of matching the query image with the reference image.
- void [setMatchingTime](#) (double time_s)
set the time of each frame matching (only the first and the last will be saved).
- void [setReferenceID](#) (const std::string reference)
set the string that identifies the matching reference video clip.
- double [getScore](#) () const
Get the matching score.
- double [getFirstMatchingTime](#) () const
get the time in seconds indicating the fist matching frame of the query clip.
- double [getLastMatchingTime](#) () const
get the time in seconds indicating the last matching frame of the query clip.
- std::string [getReferenceId](#) () const
get the string that identifies the matching reference video clip.

7.17.1 Detailed Description

A class containing the results of a matching or retrieval operation.

7.17.2 Constructor & Destructor Documentation

7.17.2.1 MatchData()

```
mpeg7cdva::MatchData::MatchData ( ) [inline]
```

7.17.2.2 ~MatchData()

```
virtual mpeg7cdva::MatchData::~~MatchData ( ) [inline], [virtual]
```

7.17.3 Member Function Documentation

7.17.3.1 `getFirstMatchingTime()`

```
double mpeg7cdva::MatchData::getFirstMatchingTime ( ) const [inline]
```

get the time in seconds indicating the first matching frame of the query clip.

Returns

the time in seconds from the start of the video clip

7.17.3.2 `getLastMatchingTime()`

```
double mpeg7cdva::MatchData::getLastMatchingTime ( ) const [inline]
```

get the time in seconds indicating the last matching frame of the query clip.

Returns

the time in seconds from the start of the video clip

7.17.3.3 `getReferenceId()`

```
std::string mpeg7cdva::MatchData::getReferenceId ( ) const [inline]
```

get the string that identifies the matching reference video clip.

Returns

the video clip relative pathname

7.17.3.4 `getScore()`

```
double mpeg7cdva::MatchData::getScore ( ) const [inline]
```

Get the matching score.

Returns

the score

7.17.3.5 `setMatchingScore()`

```
void mpeg7cdva::MatchData::setMatchingScore (
    double myscore ) [inline]
```

set the score of matching the query image with the reference image.

Parameters

<i>myscore</i>	the overall matching score
----------------	----------------------------

7.17.3.6 setMatchingTime()

```
void mpeg7cdva::MatchData::setMatchingTime (
    double time_s ) [inline]
```

set the time of each frame matching (only the first and the last will be saved).

Parameters

<i>time_s</i>	the time in seconds from the start of the query video sequence
---------------	--

7.17.3.7 setReferenceID()

```
void mpeg7cdva::MatchData::setReferenceID (
    const std::string reference ) [inline]
```

set the string that identifies the matching reference video clip.

Parameters

<i>reference</i>	the identifier (usually the relative pathname) of the matching reference video.
------------------	---

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

- [cdva.h](#)

7.18 mpeg7cdva::RefFeature Class Reference

```
#include <RefFeature.h>
```

Inheritance diagram for mpeg7cdva::RefFeature:

Collaboration diagram for mpeg7cdva::RefFeature:

Public Member Functions

- [RefFeature](#) (mpeg7cdvs::Feature &other)
- void [compress](#) ()

Data Fields

- int [refFrame](#)
- int [refFeatureIndex](#)
- int [diff](#)
- int [descrDiff](#) [descrLength]
- char [compressed](#) [descrLength >> 2]
- bool [refOnly](#)
- std::vector< unsigned int > [timesPresent](#)

7.18.1 Constructor & Destructor Documentation

7.18.1.1 RefFeature()

```
mpeg7cdva::RefFeature::RefFeature (
    mpeg7cdvs::Feature & other ) [inline]
```

References [compress\(\)](#).

Here is the call graph for this function:

7.18.2 Member Function Documentation

7.18.2.1 compress()

```
void mpeg7cdva::RefFeature::compress ( ) [inline]
```

Referenced by [RefFeature\(\)](#).

7.18.3 Field Documentation

7.18.3.1 compressed

```
char mpeg7cdva::RefFeature::compressed[descrLength >> 2]
```

7.18.3.2 descrDiff

```
int mpeg7cdva::RefFeature::descrDiff[descrLength]
```

7.18.3.3 diff

```
int mpeg7cdva::RefFeature::diff
```

7.18.3.4 refFeatureIndex

```
int mpeg7cdva::RefFeature::refFeatureIndex
```

7.18.3.5 refFrame

```
int mpeg7cdva::RefFeature::refFrame
```

7.18.3.6 refOnly

```
bool mpeg7cdva::RefFeature::refOnly
```

7.18.3.7 timesPresent

```
std::vector<unsigned int> mpeg7cdva::RefFeature::timesPresent
```

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

- [RefFeature.h](#)

7.19 mpeg7cdva::SegmentDescriptor Class Reference

A container for CdvsDescriptor instances belonging to the same video segment.

```
#include <CdvaImpl.h>
```

Inheritance diagram for mpeg7cdva::SegmentDescriptor:

Public Member Functions

- [SegmentDescriptor](#) ()
- virtual [~SegmentDescriptor](#) ()
- void [setParity](#) (unsigned int value)
Set the shot parity.
- unsigned int [getParity](#) () const
Get the shot parity.
- void [setStartTimeMs](#) (unsigned long position_ms)
Set the start time of the shot in milliseconds.
- void [setEndTimeMs](#) (unsigned long position_ms)
Set the start time of the frame in milliseconds.
- unsigned long [getStartTimeMs](#) () const
Get the shot start time in milliseconds.
- unsigned long [getEndTimeMs](#) () const
Get the frame start time in milliseconds.
- virtual size_t [write](#) (std::ofstream &fout) const
write (appending) this segment to a file
- virtual size_t [read](#) (std::ifstream &fin, mpeg7cdvs::CdvsServer *cdvsServer)
read (from the current position) this segment from a file
- bool [empty](#) () const
returns true if the segment is empty
- virtual void [clear](#) ()
clear the shot container

Static Public Member Functions

- static size_t [addSegmentToDB](#) (std::ifstream &fin, mpeg7cdvs::CdvsServer *cdvsServer, const std::string &relativepathname, bool optMatch)
read (from the current position) this segment from a file and store it into the CdvsServer DB

Data Fields

- std::vector< mpeg7cdvs::CdvsDescriptor > [keyframes](#)
keyframes belonging to this shot

Protected Member Functions

- size_t [getSize](#) () const

Protected Attributes

- unsigned long [startTime](#)
- unsigned long [endTime](#)
- unsigned int [parity](#)

Static Protected Attributes

- static const int [SHOT_HEADER_SIZE](#) = 10

7.19.1 Detailed Description

A container for CdvsDescriptor instances belonging to the same video segment.

7.19.2 Constructor & Destructor Documentation

7.19.2.1 SegmentDescriptor()

```
mpeg7cdva::SegmentDescriptor::SegmentDescriptor ( )
```

7.19.2.2 ~SegmentDescriptor()

```
virtual mpeg7cdva::SegmentDescriptor::~~SegmentDescriptor ( ) [virtual]
```

7.19.3 Member Function Documentation

7.19.3.1 addSegmentToDB()

```
static size_t mpeg7cdva::SegmentDescriptor::addSegmentToDB (
    std::ifstream & fin,
    mpeg7cdvs::CdvsServer * cdvsserver,
    const std::string & relativepathname,
    bool optMatch ) [static]
```

read (from the current position) this segment from a file and store it into the CdvsServer DB

7.19.3.2 clear()

```
virtual void mpeg7cdva::SegmentDescriptor::clear ( ) [virtual]
```

clear the shot container

Reimplemented in [mpeg7cdva::CompressedSegmentDescriptor](#).

Referenced by [mpeg7cdva::CompressedSegmentDescriptor::getDtm\(\)](#).

7.19.3.3 empty()

```
bool mpeg7cdva::SegmentDescriptor::empty ( ) const
```

returns true if the segment is empty

7.19.3.4 getEndTimeMs()

```
unsigned long mpeg7cdva::SegmentDescriptor::getEndTimeMs ( ) const
```

Get the frame start time in milliseconds.

7.19.3.5 getParity()

```
unsigned int mpeg7cdva::SegmentDescriptor::getParity ( ) const
```

Get the shot parity.

7.19.3.6 getSize()

```
size_t mpeg7cdva::SegmentDescriptor::getSize ( ) const [protected]
```

7.19.3.7 getStartTimeMs()

```
unsigned long mpeg7cdva::SegmentDescriptor::getStartTimeMs ( ) const
```

Get the shot start time in milliseconds.

7.19.3.8 read()

```
virtual size_t mpeg7cdva::SegmentDescriptor::read (
    std::ifstream & fin,
    mpeg7cdvs::CdvsServer * cdvsServer ) [virtual]
```

read (from the current position) this segment from a file

Reimplemented in [mpeg7cdva::CompressedSegmentDescriptor](#).

Referenced by [mpeg7cdva::CompressedSegmentDescriptor::nBitsSet\(\)](#).

7.19.3.9 setEndTimeMs()

```
void mpeg7cdva::SegmentDescriptor::setEndTimeMs (
    unsigned long position_ms )
```

Set the start time of the frame in milliseconds.

7.19.3.10 setParity()

```
void mpeg7cdva::SegmentDescriptor::setParity (
    unsigned int value )
```

Set the shot parity.

7.19.3.11 setStartTimeMs()

```
void mpeg7cdva::SegmentDescriptor::setStartTimeMs (
    unsigned long position_ms )
```

Set the start time of the shot in milliseconds.

7.19.3.12 write()

```
virtual size_t mpeg7cdva::SegmentDescriptor::write (
    std::ostream & fout ) const [virtual]
```

write (appending) this segment to a file

Reimplemented in [mpeg7cdva::CompressedSegmentDescriptor](#).

Referenced by `mpeg7cdva::CompressedSegmentDescriptor::nBitsSet()`.

7.19.4 Field Documentation

7.19.4.1 endTime

```
unsigned long mpeg7cdva::SegmentDescriptor::endTime [protected]
```

7.19.4.2 keyframes

```
std::vector<mpeg7cdvs::CdvsDescriptor> mpeg7cdva::SegmentDescriptor::keyframes
```

keyframes belonging to this shot

7.19.4.3 parity

```
unsigned int mpeg7cdva::SegmentDescriptor::parity [protected]
```

7.19.4.4 SHOT_HEADER_SIZE

```
const int mpeg7cdva::SegmentDescriptor::SHOT_HEADER_SIZE = 10 [static], [protected]
```

7.19.4.5 startTime

```
unsigned long mpeg7cdva::SegmentDescriptor::startTime [protected]
```

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

- [Cdvalmpl.h](#)

7.20 mpeg7cdva::WrappedCoordList Class Reference

```
#include <CompressedFeatureList.h>
```

Inheritance diagram for mpeg7cdva::WrappedCoordList:

Collaboration diagram for mpeg7cdva::WrappedCoordList:

Public Member Functions

- [WrappedCoordList](#) (mpeg7cdvs::FeatureList &fl)
- virtual unsigned int [getX](#) (int i)
- virtual unsigned int [getY](#) (int i)

Protected Member Functions

- virtual unsigned int [clampIdx](#) (unsigned int i)

Protected Attributes

- mpeg7cdvs::FeatureList & [mfl](#)

7.20.1 Constructor & Destructor Documentation

7.20.1.1 WrappedCoordList()

```
mpeg7cdva::WrappedCoordList::WrappedCoordList (  
    mpeg7cdvs::FeatureList & fl ) [inline]
```

7.20.2 Member Function Documentation

7.20.2.1 clampIdx()

```
virtual unsigned int mpeg7cdva::WrappedCoordList::clampIdx (  
    unsigned int i ) [inline], [protected], [virtual]
```

7.20.2.2 getX()

```
virtual unsigned int mpeg7cdva::WrappedCoordList::getX (  
    int i ) [inline], [virtual]
```

Implements [mpeg7cdva::CoordList](#).

7.20.2.3 getY()

```
virtual unsigned int mpeg7cdva::WrappedCoordList::getY (  
    int i ) [inline], [virtual]
```

Implements [mpeg7cdva::CoordList](#).

7.20.3 Field Documentation

7.20.3.1 mfl

```
mpeg7cdvs::FeatureList& mpeg7cdva::WrappedCoordList::mfl [protected]
```

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

- [CompressedFeatureList.h](#)

Chapter 8

File Documentation

8.1 base.h File Reference

```
#include "unistd.h"
#include "sys/types.h"
#include "ctype.h"
#include "stdio.h"
#include "stdlib.h"
#include "stdarg.h"
Include dependency graph for base.h:
```

8.2 bitstream.h File Reference

```
#include "base.h"
Include dependency graph for bitstream.h: This graph shows which files directly or indirectly include this file:
```

Data Structures

- class [evx::bitstream](#)

Namespaces

- [evx](#)

Macros

- #define [EVX_READ_BIT](#)(source, bit) (((source) >> (bit)) & 0x1)
- #define [EVX_WRITE_BIT](#)(dest, bit, value)

8.2.1 Macro Definition Documentation

8.2.1.1 EVX_READ_BIT

```
#define EVX_READ_BIT(
    source,
    bit ) (((source) >> (bit)) & 0x1)
```

8.2.1.2 EVX_WRITE_BIT

```
#define EVX_WRITE_BIT(
    dest,
    bit,
    value )
```

Value:

```
(dest) = (((dest) & ~(0x1 << (bit))) | \
          (((value) & 0x1) << (bit)))
```

8.3 Buffer.h File Reference

```
#include <cstddef>
```

Include dependency graph for Buffer.h: This graph shows which files directly or indirectly include this file:

Data Structures

- class [mpeg7cdva::Buffer](#)

A container class for a byte array, intended to replace all malloc() and new() instructions in the main code.

Namespaces

- [mpeg7cdva](#)

Namespace used to encapsulate all MPEG-7 CDVA declarations.

8.4 cabac.h File Reference

```
#include "bitstream.h"
```

Include dependency graph for cabac.h:

Data Structures

- class [evx::entropy_coder](#)

Namespaces

- [evx](#)

8.5 cdva.h File Reference

```
#include <string>
```

Include dependency graph for cdva.h: This graph shows which files directly or indirectly include this file:

Data Structures

- class [mpeg7cdva::ExtractData](#)
A class containing the results of an extraction operation.
- class [mpeg7cdva::MatchData](#)
A class containing the results of a matching or retrieval operation.

Namespaces

- [mpeg7cdva](#)
Namespace used to encapsulate all MPEG-7 CDVA declarations.

8.6 CdvaException.h File Reference

```
#include <exception>
```

```
#include <string>
```

Include dependency graph for CdvaException.h: This graph shows which files directly or indirectly include this file:

Data Structures

- class [mpeg7cdva::CdvaException](#)
Class defining a specific exception for CDVA.

Namespaces

- [mpeg7cdva](#)
Namespace used to encapsulate all MPEG-7 CDVA declarations.

8.7 CdvalImpl.h File Reference

```
#include <string>
#include <vector>
#include <map>
#include "cdva.h"
#include "CdvaException.h"
#include "LogManager.h"
#include "Buffer.h"
#include "CdvsInterface.h"
#include "SCFVIndex.h"
#include "DescriptorTimeMap.h"
#include "CompressedFeatureList.h"
Include dependency graph for CdvalImpl.h:
```

Data Structures

- class [mpeg7cdva::SegmentDescriptor](#)
A container for CdvsDescriptor instances belonging to the same video segment.
- class [mpeg7cdva::CompressedSegmentDescriptor](#)
Extension of segment descriptor container to also hold compressed data for a segment.
- class [mpeg7cdva::CdvalImpl](#)
A CDVA implementation based on multiple CDVS descriptors.

Namespaces

- [mpeg7cdva](#)
Namespace used to encapsulate all MPEG-7 CDVA declarations.

Typedefs

- typedef std::vector< SegmentDescriptor > [mpeg7cdva::ShotDescriptorList](#)
vector of shots descriptors of a video

Enumerations

- enum [mpeg7cdva::OPERATION](#) { [mpeg7cdva::UNKNOWN](#), [mpeg7cdva::EXTRACT](#), [mpeg7cdva::MATCH](#), [mpeg7cdva::RETRIEVE](#) }

8.8 CompressedFeatureList.h File Reference

```
#include "FeatureList.h"
#include "DescriptorTimeMap.h"
#include "CdvaException.h"
#include <iostream>
```

Include dependency graph for CompressedFeatureList.h: This graph shows which files directly or indirectly include this file:

Data Structures

- class [mpeg7cdva::CoordList](#)
- class [mpeg7cdva::WrappedCoordList](#)
- class [mpeg7cdva::LeanCoordList](#)
- class [mpeg7cdva::CompressedFeatureList](#)

Namespaces

- [mpeg7cdva](#)
Namespace used to encapsulate all MPEG-7 CDVA declarations.

8.9 CoordinateCoding.h File Reference

```
#include "BitOutputStream.h"
#include "BitInputStream.h"
#include "FeatureList.h"
#include "Parameters.h"
#include "RefFeature.h"
#include "DescriptorTimeMap.h"
Include dependency graph for CoordinateCoding.h:
```

Data Structures

- class [mpeg7cdva::CoordinateCoding](#)
- struct [mpeg7cdva::CoordinateCoding::CircularSumContext](#)
Basic structure for Cssc.

Namespaces

- [mpeg7cdva](#)
Namespace used to encapsulate all MPEG-7 CDVA declarations.

8.10 DescriptorTimeMap.h File Reference

```
#include <math.h>
#include <assert.h>
#include <vector>
Include dependency graph for DescriptorTimeMap.h: This graph shows which files directly or indirectly include this file:
```

Data Structures

- class [mpeg7cdva::DescriptorTimeMap](#)

Namespaces

- [mpeg7cdva](#)

Namespace used to encapsulate all MPEG-7 CDVA declarations.

8.11 FileManager.h File Reference

```
#include <string>
```

```
#include <vector>
```

Include dependency graph for FileManager.h:

Data Structures

- class [mpeg7cdva::FileManager](#)

Helper class to manage lists of file names.

Namespaces

- [mpeg7cdva](#)

Namespace used to encapsulate all MPEG-7 CDVA declarations.

8.12 LogManager.h File Reference

```
#include <fstream>
```

```
#include <string>
```

```
#include <vector>
```

```
#include "cdva.h"
```

Include dependency graph for LogManager.h: This graph shows which files directly or indirectly include this file:

Data Structures

- class [mpeg7cdva::LogManager](#)

Helper class to produce log files in various formats (csv, text, XML, etc.)

Namespaces

- [mpeg7cdva](#)

Namespace used to encapsulate all MPEG-7 CDVA declarations.

Enumerations

- enum [mpeg7cdva::LogFormat](#) { [mpeg7cdva::FORMAT_NONE](#) = 0, [mpeg7cdva::FORMAT_CSV](#) = 1, [mpeg7cdva::FORMAT_TEXT](#) = 2, [mpeg7cdva::FORMAT_HTML](#) = 4 }

Format of output logs.

8.13 math.h File Reference

```
#include "base.h"
```

Include dependency graph for math.h: This graph shows which files directly or indirectly include this file:

Namespaces

- [evx](#)

Macros

- `#define EVX_KB ((uint32) 1024)`
- `#define EVX_MB (EVX_KB * EVX_KB)`
- `#define EVX_GB (EVX_MB * EVX_KB)`
- `#define EVX_MAX_INT64 (0x7FFFFFFFFFFFFFFF)`
- `#define EVX_MAX_INT32 (0x7FFFFFFF)`
- `#define EVX_MAX_INT16 (0x7FFF)`
- `#define EVX_MAX_INT8 (0x7F)`
- `#define EVX_MAX_UINT64 (0xFFFFFFFFFFFFFFFF)`
- `#define EVX_MAX_UINT32 (0xFFFFFFFF)`
- `#define EVX_MAX_UINT16 (0xFFFF)`
- `#define EVX_MAX_UINT8 (0xFF)`
- `#define EVX_MIN_INT64 (-EVX_MAX_INT64 - 1)`
- `#define EVX_MIN_INT32 (-EVX_MAX_INT32 - 1)`
- `#define EVX_MIN_INT16 (-EVX_MAX_INT16 - 1)`
- `#define EVX_MIN_INT8 (-EVX_MAX_INT8 - 1)`
- `#define evx_min2(a, b) ((a) < (b) ? (a) : (b))`
- `#define evx_max2(a, b) ((a) > (b) ? (a) : (b))`
- `#define evx_min3(a, b, c) ((c) < (a) ? ((c) < (b) ? (c) : (b)) : (a) < (b) ? (a) : (b))`
- `#define evx_max3(a, b, c) ((c) > (a) ? ((c) > (b) ? (c) : (b)) : (a) > (b) ? (a) : (b))`

Functions

- `uint8 evx::log2 (uint8 value)`
- `uint8 evx::log2 (uint16 value)`
- `uint8 evx::log2 (uint32 value)`
- `int8 evx::abs (int8 value)`
- `int16 evx::abs (int16 value)`
- `int32 evx::abs (int32 value)`
- `int16 evx::clip_range (int16 value, int16 min, int16 max)`
- `uint32 evx::greater_multiple (uint32 value, uint32 multiple)`
- `uint32 evx::align (uint32 value, uint32 alignment)`

Variables

- `const uint8 evx::log2_byte_lut []`

8.13.1 Macro Definition Documentation

8.13.1.1 EVX_GB

```
#define EVX_GB (EVX_MB * EVX_KB)
```

8.13.1.2 EVX_KB

```
#define EVX_KB ((uint32) 1024)
```

8.13.1.3 evx_max2

```
#define evx_max2(  
    a,  
    b ) ((a) > (b) ? (a) : (b))
```

8.13.1.4 evx_max3

```
#define evx_max3(  
    a,  
    b,  
    c ) ((c) > (a) ? ((c) > (b) ? (c) : (b)) : (a) > (b) ? (a) : (b))
```

8.13.1.5 EVX_MAX_INT16

```
#define EVX_MAX_INT16 (0x7FFF)
```

8.13.1.6 EVX_MAX_INT32

```
#define EVX_MAX_INT32 (0x7FFFFFFF)
```

8.13.1.7 EVX_MAX_INT64

```
#define EVX_MAX_INT64 (0xFFFFFFFFFFFFFFFF)
```

8.13.1.8 EVX_MAX_INT8

```
#define EVX_MAX_INT8 (0x7F)
```

8.13.1.9 EVX_MAX_UINT16

```
#define EVX_MAX_UINT16 (0xFFFF)
```

8.13.1.10 EVX_MAX_UINT32

```
#define EVX_MAX_UINT32 (0xFFFFFFFF)
```

8.13.1.11 EVX_MAX_UINT64

```
#define EVX_MAX_UINT64 (0xFFFFFFFFFFFFFFFF)
```

8.13.1.12 EVX_MAX_UINT8

```
#define EVX_MAX_UINT8 (0xFF)
```

8.13.1.13 EVX_MB

```
#define EVX_MB (EVX_KB * EVX_KB)
```

8.13.1.14 evx_min2

```
#define evx_min2(  
    a,  
    b ) ((a) < (b) ? (a) : (b))
```

8.13.1.15 evx_min3

```
#define evx_min3(
    a,
    b,
    c ) ((c) < (a) ? ((c) < (b) ? (c) : (b)) : (a) < (b) ? (a) : (b))
```

8.13.1.16 EVX_MIN_INT16

```
#define EVX_MIN_INT16 (-EVX_MAX_INT16 - 1)
```

8.13.1.17 EVX_MIN_INT32

```
#define EVX_MIN_INT32 (-EVX_MAX_INT32 - 1)
```

8.13.1.18 EVX_MIN_INT64

```
#define EVX_MIN_INT64 (-EVX_MAX_INT64 - 1)
```

8.13.1.19 EVX_MIN_INT8

```
#define EVX_MIN_INT8 (-EVX_MAX_INT8 - 1)
```

8.14 memory.h File Reference

```
#include "base.h"
```

Include dependency graph for memory.h:

Namespaces

- [evx](#)

Functions

- uint32 [evx::aligned_bit_copy](#) (uint8 *dest, uint32 dest_bit_offset, uint8 *source, uint32 source_bit_offset, uint32 copy_bit_count)
- uint32 [evx::unaligned_bit_copy](#) (uint8 *dest, uint32 dest_offset, uint8 *source, uint32 source_offset, uint32 copy_bit_count)

8.15 RefFeature.h File Reference

```
#include <stdio.h>
#include <string.h>
#include "Feature.h"
```

Include dependency graph for RefFeature.h: This graph shows which files directly or indirectly include this file:

Data Structures

- class [mpeg7cdva::RefFeature](#)

Namespaces

- [mpeg7cdva](#)
Namespace used to encapsulate all MPEG-7 CDVA declarations.

8.16 version.h File Reference

Macros

- #define [EVX_VERSION_MAJOR](#) 1
- #define [EVX_VERSION_MINOR](#) 29
- #define [EVX_VERSION_CHANGELIST](#) 30
- #define [EVX_VERSION_WORD](#)(major, minor) (((major) & 0xFF) << 8) | ((minor) & 0xFF)
- #define [EVX_MAJOR_VERSION](#)(x) (((x) >> 8) & 0xFF)
- #define [EVX_MINOR_VERSION](#)(x) ((x) & 0xFF)

8.16.1 Macro Definition Documentation

8.16.1.1 EVX_MAJOR_VERSION

```
#define EVX_MAJOR_VERSION(  
    x ) ((x) >> 8) & 0xFF)
```

8.16.1.2 EVX_MINOR_VERSION

```
#define EVX_MINOR_VERSION(  
    x ) ((x) & 0xFF)
```

8.16.1.3 EVX_VERSION_CHANGELIST

```
#define EVX_VERSION_CHANGELIST 30
```

8.16.1.4 EVX_VERSION_MAJOR

```
#define EVX_VERSION_MAJOR 1
```

8.16.1.5 EVX_VERSION_MINOR

```
#define EVX_VERSION_MINOR 29
```

8.16.1.6 EVX_VERSION_WORD

```
#define EVX_VERSION_WORD(  
    major,  
    minor ) (((major) & 0xFF) << 8) | ((minor) & 0xFF))
```

Index

- ~Buffer
 - mpeg7cdva::Buffer, 25
- ~CdvaException
 - mpeg7cdva::CdvaException, 29
- ~CdvalImpl
 - mpeg7cdva::CdvalImpl, 32
- ~CompressedSegmentDescriptor
 - mpeg7cdva::CompressedSegmentDescriptor, 48
- ~CoordinateCoding
 - mpeg7cdva::CoordinateCoding, 55
- ~DescriptorTimeMap
 - mpeg7cdva::DescriptorTimeMap, 63
- ~FileManager
 - mpeg7cdva::FileManager, 74
- ~LeanCoordList
 - mpeg7cdva::LeanCoordList, 79
- ~LogManager
 - mpeg7cdva::LogManager, 82
- ~MatchData
 - mpeg7cdva::MatchData, 85
- ~SegmentDescriptor
 - mpeg7cdva::SegmentDescriptor, 91
- ~bitstream
 - evx::bitstream, 20
- abs
 - evx, 13, 14
- add
 - mpeg7cdva::DescriptorTimeMap, 64
- addDescIndex
 - mpeg7cdva::DescriptorTimeMap, 64
- AddImageSample
 - mpeg7cdva::CoordinateCoding, 56
- addSegmentToDB
 - mpeg7cdva::SegmentDescriptor, 91
- align
 - evx, 14
- aligned_bit_copy
 - evx, 14
- allocate
 - mpeg7cdva::CompressedFeatureList, 45
- allocateGlobalBuffer
 - mpeg7cdva::CompressedSegmentDescriptor, 48
- allocateHistoBuffer
 - mpeg7cdva::CompressedSegmentDescriptor, 48
- allocateLocalBuffer
 - mpeg7cdva::CompressedSegmentDescriptor, 49
- assign
 - evx::bitstream, 20
 - mpeg7cdva::Buffer, 25
- base.h, 97
- bitstream
 - evx::bitstream, 19, 20
- bitstream.h, 97
 - EVX_READ_BIT, 97
 - EVX_WRITE_BIT, 98
- Buffer
 - mpeg7cdva::Buffer, 24, 25
- Buffer.h, 98
- byDescendingScore
 - mpeg7cdva::CdvalImpl, 33
- CONTEXT_RANGE
 - mpeg7cdva::CoordinateCoding, 60
- cabac.h, 98
- calc_desc_sizes
 - mpeg7cdva::CdvalImpl, 39
- cdva.h, 99
- CdvaException
 - mpeg7cdva::CdvaException, 29
- CdvaException.h, 99
- CdvalImpl
 - mpeg7cdva::CdvalImpl, 32
- CdvalImpl.h, 100
- cdvsMode
 - mpeg7cdva::CdvalImpl, 39
- cdvsclient
 - mpeg7cdva::CdvalImpl, 39
- cdvsconfig
 - mpeg7cdva::CdvalImpl, 39
- cdvsserver
 - mpeg7cdva::CdvalImpl, 39
- checkBitrate
 - mpeg7cdva::CdvalImpl, 33
- clampIdx
 - mpeg7cdva::WrappedCoordList, 95
- clear
 - evx::bitstream, 20
 - evx::entropy_coder, 68
 - mpeg7cdva::Buffer, 25
 - mpeg7cdva::CompressedSegmentDescriptor, 49
 - mpeg7cdva::DescriptorTimeMap, 64
 - mpeg7cdva::SegmentDescriptor, 91
- clip_duration
 - mpeg7cdva::ExtractData, 71
- clip_range
 - evx, 14
- close
 - mpeg7cdva::CdvalImpl, 33
 - mpeg7cdva::LogManager, 83

- commitDB
 - mpeg7cdva::CdvalImpl, 33
- compare
 - mpeg7cdva::Buffer, 26
 - mpeg7cdva::CoordinateCoding, 56
- compress
 - mpeg7cdva::RefFeature, 88
- compressed
 - mpeg7cdva::RefFeature, 88
- CompressedFeatureList
 - mpeg7cdva::CompressedFeatureList, 44
- CompressedFeatureList.h, 100
- CompressedSegmentDescriptor
 - mpeg7cdva::CompressedSegmentDescriptor, 48
- CoordList
 - mpeg7cdva::CoordList, 61
- coordinate_bit_count
 - mpeg7cdva::ExtractData, 72
- CoordinateCoding
 - mpeg7cdva::CoordinateCoding, 55
- CoordinateCoding.h, 101
- countNames
 - mpeg7cdva::FileManager, 74
- CsscCoordinateCoding, 62
- current_op
 - mpeg7cdva::CdvalImpl, 39
- data
 - mpeg7cdva::Buffer, 26
- decode
 - evx::entropy_coder, 69
 - mpeg7cdva::CompressedSegmentDescriptor, 49
- descIndex
 - mpeg7cdva::DescriptorTimeMap, 67
- descrDiff
 - mpeg7cdva::RefFeature, 88
- DescriptorTimeMap
 - mpeg7cdva::DescriptorTimeMap, 63
- DescriptorTimeMap.h, 101
- descriptorlength
 - mpeg7cdva::ExtractData, 72
- diff
 - mpeg7cdva::RefFeature, 89
- difference
 - mpeg7cdva::CoordinateCoding, 56
- drop_frame_th
 - mpeg7cdva::CdvalImpl, 40
- dtm
 - mpeg7cdva::CompressedSegmentDescriptor, 52
- EVX_GB
 - math.h, 103
- EVX_KB
 - math.h, 104
- EVX_MAJOR_VERSION
 - version.h, 107
- EVX_MAX_INT16
 - math.h, 104
- EVX_MAX_INT32
 - math.h, 104
- EVX_MAX_INT64
 - math.h, 104
- EVX_MAX_INT8
 - math.h, 104
- EVX_MAX_UINT16
 - math.h, 105
- EVX_MAX_UINT32
 - math.h, 105
- EVX_MAX_UINT64
 - math.h, 105
- EVX_MAX_UINT8
 - math.h, 105
- EVX_MIN_INT16
 - math.h, 106
- EVX_MIN_INT32
 - math.h, 106
- EVX_MIN_INT64
 - math.h, 106
- EVX_MIN_INT8
 - math.h, 106
- EVX_MINOR_VERSION
 - version.h, 107
- EVX_MB
 - math.h, 105
- EVX_READ_BIT
 - bitstream.h, 97
- EVX_VERSION_CHANGELIST
 - version.h, 107
- EVX_VERSION_MAJOR
 - version.h, 108
- EVX_VERSION_MINOR
 - version.h, 108
- EVX_VERSION_WORD
 - version.h, 108
- EVX_WRITE_BIT
 - bitstream.h, 98
- empty
 - evx::bitstream, 21
 - mpeg7cdva::Buffer, 26
 - mpeg7cdva::SegmentDescriptor, 91
- encode
 - evx::entropy_coder, 69
- encode_th
 - mpeg7cdva::CdvalImpl, 40
- encodeCoordinates
 - mpeg7cdva::CdvalImpl, 33
- encodeShot
 - mpeg7cdva::CdvalImpl, 34
- endTime
 - mpeg7cdva::SegmentDescriptor, 93
- EndTrainingMode
 - mpeg7cdva::CoordinateCoding, 56
- entropy_coder
 - evx::entropy_coder, 68
- equals
 - mpeg7cdva::Buffer, 26
- evx, 11

- abs, [13](#), [14](#)
- align, [14](#)
- aligned_bit_copy, [14](#)
- clip_range, [14](#)
- evx_status, [12](#)
- float32, [12](#)
- float64, [12](#)
- greater_multiple, [14](#)
- int16, [12](#)
- int32, [12](#)
- int64, [12](#)
- int8, [12](#)
- log2, [14](#), [15](#)
- log2_byte_lut, [16](#)
- post_error_i, [15](#)
- uint16, [13](#)
- uint32, [13](#)
- uint64, [13](#)
- uint8, [13](#)
- unaligned_bit_copy, [15](#)
- wchar, [13](#)
- evx::bitstream, [19](#)
 - ~bitstream, [20](#)
 - assign, [20](#)
 - bitstream, [19](#), [20](#)
 - clear, [20](#)
 - empty, [21](#)
 - is_empty, [21](#)
 - is_full, [21](#)
 - query_byte_occupancy, [21](#)
 - query_capacity, [21](#)
 - query_data, [21](#)
 - query_occupancy, [21](#)
 - read_bit, [21](#)
 - read_bits, [22](#)
 - read_byte, [22](#)
 - read_bytes, [22](#)
 - resize_capacity, [22](#)
 - seek, [22](#)
 - write_bit, [22](#)
 - write_bits, [22](#)
 - write_byte, [23](#)
 - write_bytes, [23](#)
- evx::entropy_coder, [68](#)
 - clear, [68](#)
 - decode, [69](#)
 - encode, [69](#)
 - entropy_coder, [68](#)
 - finish_encode, [69](#)
 - start_decode, [69](#)
- evx_max2
 - math.h, [104](#)
- evx_max3
 - math.h, [104](#)
- evx_min2
 - math.h, [105](#)
- evx_min3
 - math.h, [105](#)
- evx_status
 - evx, [12](#)
- exportVars
 - mpeg7cdva::CoordinateCoding, [56](#)
- extract
 - mpeg7cdva::CdvaImpl, [34](#)
- ExtractData
 - mpeg7cdva::ExtractData, [70](#)
- FileManager
 - mpeg7cdva::FileManager, [74](#)
- FileManager.h, [102](#)
- fill
 - mpeg7cdva::Buffer, [27](#)
- finish_encode
 - evx::entropy_coder, [69](#)
- float32
 - evx, [12](#)
- float64
 - evx, [12](#)
- forceSampleMs
 - mpeg7cdva::CdvaImpl, [40](#)
- fromBinary
 - mpeg7cdva::CoordinateCoding, [57](#)
- generateDTM
 - mpeg7cdva::CdvaImpl, [34](#)
- generateFeatureList
 - mpeg7cdva::CoordinateCoding, [57](#)
- generateHistogramMap
 - mpeg7cdva::CoordinateCoding, [58](#)
- generateHistogramMap2
 - mpeg7cdva::CoordinateCoding, [58](#)
- get
 - mpeg7cdva::DescriptorTimeMap, [64](#)
- getBitSize
 - mpeg7cdva::DescriptorTimeMap, [64](#)
- getBuffer
 - mpeg7cdva::DescriptorTimeMap, [65](#)
- getCount
 - mpeg7cdva::DescriptorTimeMap, [65](#)
- getDatasetName
 - mpeg7cdva::FileManager, [75](#)
- getDatasetPath
 - mpeg7cdva::FileManager, [75](#)
- getDatasetPathName
 - mpeg7cdva::FileManager, [75](#)
- getDatasetSize
 - mpeg7cdva::FileManager, [75](#)
- getDescIndex
 - mpeg7cdva::DescriptorTimeMap, [65](#)
- getDescriptorExt
 - mpeg7cdva::CdvaImpl, [35](#)
- getDiffSignature
 - mpeg7cdva::CdvaImpl, [35](#)
- getDtm
 - mpeg7cdva::CompressedSegmentDescriptor, [49](#)
- getEndTimeMs
 - mpeg7cdva::SegmentDescriptor, [92](#)

- getExt
 - mpeg7cdva::CdvalImpl, 35
- getFeatureList
 - mpeg7cdva::CompressedFeatureList, 45
 - mpeg7cdva::LeanCoordList, 79, 80
- getFeatureStartIndex
 - mpeg7cdva::DescriptorTimeMap, 65
- getFirstMatchingTime
 - mpeg7cdva::MatchData, 85
- getGlobalBufSz
 - mpeg7cdva::CompressedSegmentDescriptor, 50
- getGlobalBuffer
 - mpeg7cdva::CompressedSegmentDescriptor, 49
- getHistoBufSz
 - mpeg7cdva::CompressedSegmentDescriptor, 50
- getHistoBuffer
 - mpeg7cdva::CompressedSegmentDescriptor, 50
- getLastMatchingTime
 - mpeg7cdva::MatchData, 86
- getLength
 - mpeg7cdva::LeanCoordList, 80
- getLocalBufSz
 - mpeg7cdva::CompressedSegmentDescriptor, 50
- getLocalBuffer
 - mpeg7cdva::CompressedSegmentDescriptor, 50
- getNAbsDescr
 - mpeg7cdva::DescriptorTimeMap, 65
- getNDescr
 - mpeg7cdva::DescriptorTimeMap, 65
- getNFrames
 - mpeg7cdva::DescriptorTimeMap, 66
- getParity
 - mpeg7cdva::SegmentDescriptor, 92
- getQueryName
 - mpeg7cdva::FileManager, 76
- getReferenceId
 - mpeg7cdva::MatchData, 86
- getReferenceName
 - mpeg7cdva::FileManager, 76
- getScore
 - mpeg7cdva::MatchData, 86
- getSize
 - mpeg7cdva::SegmentDescriptor, 92
- getStartTimeMs
 - mpeg7cdva::SegmentDescriptor, 92
- getWorkspaceDir
 - mpeg7cdva::FileManager, 77
- getX
 - mpeg7cdva::CoordList, 61
 - mpeg7cdva::LeanCoordList, 80
 - mpeg7cdva::WrappedCoordList, 95
- getY
 - mpeg7cdva::CoordList, 61
 - mpeg7cdva::LeanCoordList, 80
 - mpeg7cdva::WrappedCoordList, 95
- global_bit_count
 - mpeg7cdva::ExtractData, 72
- globalBuf
 - mpeg7cdva::CompressedSegmentDescriptor, 52
- globalBufSz
 - mpeg7cdva::CompressedSegmentDescriptor, 52
- globalUncompressedBufSz
 - mpeg7cdva::CompressedSegmentDescriptor, 52
- greater_multiple
 - evx, 14
- header_bit_count
 - mpeg7cdva::ExtractData, 72
- histoBuf
 - mpeg7cdva::CompressedSegmentDescriptor, 53
- histoBufSz
 - mpeg7cdva::CompressedSegmentDescriptor, 53
- init
 - mpeg7cdva::CdvalImpl, 35
 - mpeg7cdva::DescriptorTimeMap, 66
 - mpeg7cdva::LogManager, 83
- int16
 - evx, 12
- int32
 - evx, 12
- int64
 - evx, 12
- int8
 - evx, 12
- is_empty
 - evx::bitstream, 21
- is_full
 - evx::bitstream, 21
- keyframes
 - mpeg7cdva::SegmentDescriptor, 93
- ldOffsets
 - mpeg7cdva::DescriptorTimeMap, 67
- LeanCoordList
 - mpeg7cdva::LeanCoordList, 79
- local_bit_count
 - mpeg7cdva::ExtractData, 72
- localBuf
 - mpeg7cdva::CompressedSegmentDescriptor, 53
- localBufSz
 - mpeg7cdva::CompressedSegmentDescriptor, 53
- localDescCodingAbs
 - mpeg7cdva::CdvalImpl, 36
- localUncompressedBufSz
 - mpeg7cdva::CompressedSegmentDescriptor, 53
- log2
 - evx, 14, 15
- log2_byte_lut
 - evx, 16
- LogFormat
 - mpeg7cdva, 17
- LogManager
 - mpeg7cdva::LogManager, 82
- LogManager.h, 102
- m_buffer

- mpeg7cdva::DescriptorTimeMap, 67
- m_nAbsDesc
 - mpeg7cdva::DescriptorTimeMap, 67
- m_nDesc
 - mpeg7cdva::DescriptorTimeMap, 67
- m_nFrames
 - mpeg7cdva::DescriptorTimeMap, 67
- m_nbits
 - mpeg7cdva::DescriptorTimeMap, 67
- MAXIMUM_SUM_CONTEXT
 - mpeg7cdva::CoordinateCoding, 60
- makeindex
 - mpeg7cdva::CdvalImpl, 36
- match
 - mpeg7cdva::CdvalImpl, 37
- match_med1
 - mpeg7cdva::CdvalImpl, 37
- match_med2
 - mpeg7cdva::CdvalImpl, 38
- MatchData
 - mpeg7cdva::MatchData, 85
- matchDescriptors_oneWay
 - mpeg7cdva::CompressedFeatureList, 45
- matchDescriptors_twoWay
 - mpeg7cdva::CompressedFeatureList, 45
- math.h, 103
 - EVX_GB, 103
 - EVX_KB, 104
 - EVX_MAX_INT16, 104
 - EVX_MAX_INT32, 104
 - EVX_MAX_INT64, 104
 - EVX_MAX_INT8, 104
 - EVX_MAX_UINT16, 105
 - EVX_MAX_UINT32, 105
 - EVX_MAX_UINT64, 105
 - EVX_MAX_UINT8, 105
 - EVX_MIN_INT16, 106
 - EVX_MIN_INT32, 106
 - EVX_MIN_INT64, 106
 - EVX_MIN_INT8, 106
 - EVX_MB, 105
 - evx_max2, 104
 - evx_max3, 104
 - evx_min2, 105
 - evx_min3, 105
- max_retrieved
 - mpeg7cdva::CdvalImpl, 40
- memory.h, 106
- mfl
 - mpeg7cdva::WrappedCoordList, 95
- minLocalDiff
 - mpeg7cdva::CdvalImpl, 40
- minShotLen
 - mpeg7cdva::CdvalImpl, 40
- mpeg7cdva, 16
 - LogFormat, 17
 - OPERATION, 18
 - ShotDescriptorList, 17
- mpeg7cdva::Buffer, 23
 - ~Buffer, 25
 - assign, 25
 - Buffer, 24, 25
 - clear, 25
 - compare, 26
 - data, 26
 - empty, 26
 - equals, 26
 - fill, 27
 - operator=, 27
 - operator==, 27
 - read, 27
 - resize, 27
 - sdata, 27, 28
 - size, 28
 - swap, 28
 - write, 28
- mpeg7cdva::CdvaException, 29
 - ~CdvaException, 29
 - CdvaException, 29
 - what, 29
- mpeg7cdva::CdvalImpl, 30
 - ~CdvalImpl, 32
 - byDescendingScore, 33
 - calc_desc_sizes, 39
 - CdvalImpl, 32
 - cdvsMode, 39
 - cdvsclient, 39
 - cdvsconfig, 39
 - cdvsserver, 39
 - checkBitrate, 33
 - close, 33
 - commitDB, 33
 - current_op, 39
 - drop_frame_th, 40
 - encode_th, 40
 - encodeCoordinates, 33
 - encodeShot, 34
 - extract, 34
 - forceSampleMs, 40
 - generateDTM, 34
 - getDescriptorExt, 35
 - getDiffSignature, 35
 - getExt, 35
 - init, 35
 - localDescCodingAbs, 36
 - makeindex, 36
 - match, 37
 - match_med1, 37
 - match_med2, 38
 - max_retrieved, 40
 - minLocalDiff, 40
 - minShotLen, 40
 - nBitsSet, 38
 - optMatch, 41
 - optMatch_b, 41
 - optMatch_mode, 41

- optMatch_tau, 41
- parse, 38
- readWriteCompressed, 41
- retrieve, 38
- shot_cut_th, 41
- shot_ver_th, 42
- skip_after, 42
- skip_before, 42
- verboseMode, 42
- mpeg7cdva::CompressedFeatureList, 44
 - allocate, 45
 - CompressedFeatureList, 44
 - getFeatureList, 45
 - matchDescriptors_oneWay, 45
 - matchDescriptors_twoWay, 45
 - operator=, 45, 46
 - relevances, 46
 - toCDVScfl, 46
- mpeg7cdva::CompressedSegmentDescriptor, 46
 - ~CompressedSegmentDescriptor, 48
 - allocateGlobalBuffer, 48
 - allocateHistoBuffer, 48
 - allocateLocalBuffer, 49
 - clear, 49
 - CompressedSegmentDescriptor, 48
 - decode, 49
 - dtm, 52
 - getDtm, 49
 - getGlobalBufSz, 50
 - getGlobalBuffer, 49
 - getHistoBufSz, 50
 - getHistoBuffer, 50
 - getLocalBufSz, 50
 - getLocalBuffer, 50
 - globalBuf, 52
 - globalBufSz, 52
 - globalUncompressedBufSz, 52
 - histoBuf, 53
 - histoBufSz, 53
 - localBuf, 53
 - localBufSz, 53
 - localUncompressedBufSz, 53
 - nBitsSet, 50
 - numGlobFctPresent, 50
 - params, 53
 - read, 51
 - readFeatureListFromBinaryAbs, 51
 - reconstructGlobalDifferences, 51
 - SHOT_HEADER_SIZE, 54
 - setGlobalUncompressedBufSz, 51
 - setHistoBufSz, 51
 - setLocalUncompressedBufSz, 52
 - write, 52
- mpeg7cdva::CoordList, 60
 - CoordList, 61
 - getX, 61
 - getY, 61
- mpeg7cdva::CoordinateCoding, 54
 - ~CoordinateCoding, 55
 - AddImageSample, 56
 - CONTEXT_RANGE, 60
 - compare, 56
 - CoordinateCoding, 55
 - difference, 56
 - EndTrainingMode, 56
 - exportVars, 56
 - fromBinary, 57
 - generateFeatureList, 57
 - generateHistogramMap, 58
 - generateHistogramMap2, 58
 - MAXIMUM_SUM_CONTEXT, 60
 - readSeparateContext, 59
 - SUM_HIST_COUNT_SIZE, 60
 - setIsOK, 59
 - setVerbose, 59
 - StartTrainingMode, 59
 - toBinary, 59
 - writeSeparateContext, 60
- mpeg7cdva::CoordinateCoding::CircularSumContext, 42
 - vCount, 43
 - vInitialMap, 43
 - vMap, 43
- mpeg7cdva::DescriptorTimeMap, 62
 - ~DescriptorTimeMap, 63
 - add, 64
 - addDescIndex, 64
 - clear, 64
 - descIndex, 67
 - DescriptorTimeMap, 63
 - get, 64
 - getBitSize, 64
 - getBuffer, 65
 - getCount, 65
 - getDescIndex, 65
 - getFeatureStartIndex, 65
 - getNAbsDescr, 65
 - getNDescr, 65
 - getNFrames, 66
 - init, 66
 - ldOffsets, 67
 - m_buffer, 67
 - m_nAbsDescr, 67
 - m_nDescr, 67
 - m_nFrames, 67
 - m_nbits, 67
 - operator=, 66
 - parseDescIndex, 66
 - writeDescIndex, 66
- mpeg7cdva::ExtractData, 69
 - clip_duration, 71
 - coordinate_bit_count, 72
 - descriptorlength, 72
 - ExtractData, 70
 - global_bit_count, 72
 - header_bit_count, 72

- local_bit_count, 72
- n_keyframes, 72
- numframes, 72
- numshots, 72
- setDescriptorLength, 70
- setNumFrames, 71
- setNumShots, 71
- setVideoDuration, 71
- mpeg7cdva::FileManager, 73
 - ~FileManager, 74
 - countNames, 74
 - FileManager, 74
 - getDatasetName, 75
 - getDatasetPath, 75
 - getDatasetPathName, 75
 - getDatasetSize, 75
 - getQueryName, 76
 - getReferenceName, 76
 - getWorkspaceDir, 77
 - replaceExt, 77
 - replacePath, 77
 - setWorkspaceDir, 78
- mpeg7cdva::LeanCoordList, 78
 - ~LeanCoordList, 79
 - getFeatureList, 79, 80
 - getLength, 80
 - getX, 80
 - getY, 80
 - LeanCoordList, 79
 - nFeatures, 81
 - operator=, 80
 - setX, 80
 - setY, 81
 - xCoord, 81
 - yCoord, 81
- mpeg7cdva::LogManager, 81
 - ~LogManager, 82
 - close, 83
 - init, 83
 - LogManager, 82
 - printExtractData, 83
 - printExtractHeader, 83
 - printMatchData, 83
 - printMatchHeader, 84
 - printRetrievalData, 84
 - printRetrievalHeader, 84
- mpeg7cdva::MatchData, 84
 - ~MatchData, 85
 - getFirstMatchingTime, 85
 - getLastMatchingTime, 86
 - getReferenceId, 86
 - getScore, 86
 - MatchData, 85
 - setMatchingScore, 86
 - setMatchingTime, 87
 - setReferenceID, 87
- mpeg7cdva::RefFeature, 87
 - compress, 88
 - compressed, 88
 - descrDiff, 88
 - diff, 89
 - RefFeature, 88
 - refFeatureIndex, 89
 - refFrame, 89
 - refOnly, 89
 - timesPresent, 89
- mpeg7cdva::SegmentDescriptor, 89
 - ~SegmentDescriptor, 91
 - addSegmentToDB, 91
 - clear, 91
 - empty, 91
 - endTime, 93
 - getEndTimeMs, 92
 - getParity, 92
 - getSize, 92
 - getStartTimeMs, 92
 - keyframes, 93
 - parity, 94
 - read, 92
 - SHOT_HEADER_SIZE, 94
 - SegmentDescriptor, 91
 - setEndTimeMs, 92
 - setParity, 93
 - setStartTimeMs, 93
 - startTime, 94
 - write, 93
- mpeg7cdva::WrappedCoordList, 94
 - clampIdx, 95
 - getX, 95
 - getY, 95
 - mfl, 95
 - WrappedCoordList, 95
- n_keyframes
 - mpeg7cdva::ExtractData, 72
- nBitsSet
 - mpeg7cdva::CdvalImpl, 38
 - mpeg7cdva::CompressedSegmentDescriptor, 50
- nFeatures
 - mpeg7cdva::LeanCoordList, 81
- numGlobFctPresent
 - mpeg7cdva::CompressedSegmentDescriptor, 50
- numframes
 - mpeg7cdva::ExtractData, 72
- numshots
 - mpeg7cdva::ExtractData, 72
- OPERATION
 - mpeg7cdva, 18
- operator=
 - mpeg7cdva::Buffer, 27
 - mpeg7cdva::CompressedFeatureList, 45, 46
 - mpeg7cdva::DescriptorTimeMap, 66
 - mpeg7cdva::LeanCoordList, 80
- operator==
 - mpeg7cdva::Buffer, 27
- optMatch

- mpeg7cdva::CdvalImpl, 41
- optMatch_b
 - mpeg7cdva::CdvalImpl, 41
- optMatch_mode
 - mpeg7cdva::CdvalImpl, 41
- optMatch_tau
 - mpeg7cdva::CdvalImpl, 41
- params
 - mpeg7cdva::CompressedSegmentDescriptor, 53
- parity
 - mpeg7cdva::SegmentDescriptor, 94
- parse
 - mpeg7cdva::CdvalImpl, 38
- parseDescIndex
 - mpeg7cdva::DescriptorTimeMap, 66
- post_error_i
 - evx, 15
- printExtractData
 - mpeg7cdva::LogManager, 83
- printExtractHeader
 - mpeg7cdva::LogManager, 83
- printMatchData
 - mpeg7cdva::LogManager, 83
- printMatchHeader
 - mpeg7cdva::LogManager, 84
- printRetrievalData
 - mpeg7cdva::LogManager, 84
- printRetrievalHeader
 - mpeg7cdva::LogManager, 84
- query_byte_occupancy
 - evx::bitstream, 21
- query_capacity
 - evx::bitstream, 21
- query_data
 - evx::bitstream, 21
- query_occupancy
 - evx::bitstream, 21
- read
 - mpeg7cdva::Buffer, 27
 - mpeg7cdva::CompressedSegmentDescriptor, 51
 - mpeg7cdva::SegmentDescriptor, 92
- read_bit
 - evx::bitstream, 21
- read_bits
 - evx::bitstream, 22
- read_byte
 - evx::bitstream, 22
- read_bytes
 - evx::bitstream, 22
- readFeatureListFromBinaryAbs
 - mpeg7cdva::CompressedSegmentDescriptor, 51
- readSeparateContext
 - mpeg7cdva::CoordinateCoding, 59
- readWriteCompressed
 - mpeg7cdva::CdvalImpl, 41
- reconstructGlobalDifferences
 - mpeg7cdva::CompressedSegmentDescriptor, 51
- RefFeature
 - mpeg7cdva::RefFeature, 88
- RefFeature.h, 107
- refFeatureIndex
 - mpeg7cdva::RefFeature, 89
- refFrame
 - mpeg7cdva::RefFeature, 89
- refOnly
 - mpeg7cdva::RefFeature, 89
- relevances
 - mpeg7cdva::CompressedFeatureList, 46
- replaceExt
 - mpeg7cdva::FileManager, 77
- replacePath
 - mpeg7cdva::FileManager, 77
- resize
 - mpeg7cdva::Buffer, 27
- resize_capacity
 - evx::bitstream, 22
- retrieve
 - mpeg7cdva::CdvalImpl, 38
- SHOT_HEADER_SIZE
 - mpeg7cdva::CompressedSegmentDescriptor, 54
 - mpeg7cdva::SegmentDescriptor, 94
- SUM_HIST_COUNT_SIZE
 - mpeg7cdva::CoordinateCoding, 60
- sdata
 - mpeg7cdva::Buffer, 27, 28
- seek
 - evx::bitstream, 22
- SegmentDescriptor
 - mpeg7cdva::SegmentDescriptor, 91
- setDescriptorLength
 - mpeg7cdva::ExtractData, 70
- setEndTimeMs
 - mpeg7cdva::SegmentDescriptor, 92
- setGlobalUncompressedBufSz
 - mpeg7cdva::CompressedSegmentDescriptor, 51
- setHistoBufSz
 - mpeg7cdva::CompressedSegmentDescriptor, 51
- setIsOK
 - mpeg7cdva::CoordinateCoding, 59
- setLocalUncompressedBufSz
 - mpeg7cdva::CompressedSegmentDescriptor, 52
- setMatchingScore
 - mpeg7cdva::MatchData, 86
- setMatchingTime
 - mpeg7cdva::MatchData, 87
- setNumFrames
 - mpeg7cdva::ExtractData, 71
- setNumShots
 - mpeg7cdva::ExtractData, 71
- setParity
 - mpeg7cdva::SegmentDescriptor, 93
- setReferenceID
 - mpeg7cdva::MatchData, 87
- setStartTimeMs

- mpeg7cdva::SegmentDescriptor, 93
- setVerbose
 - mpeg7cdva::CoordinateCoding, 59
- setVideoDuration
 - mpeg7cdva::ExtractData, 71
- setWorkspaceDir
 - mpeg7cdva::FileManager, 78
- setX
 - mpeg7cdva::LeanCoordList, 80
- setY
 - mpeg7cdva::LeanCoordList, 81
- shot_cut_th
 - mpeg7cdva::CdvaImpl, 41
- shot_ver_th
 - mpeg7cdva::CdvaImpl, 42
- ShotDescriptorList
 - mpeg7cdva, 17
- size
 - mpeg7cdva::Buffer, 28
- skip_after
 - mpeg7cdva::CdvaImpl, 42
- skip_before
 - mpeg7cdva::CdvaImpl, 42
- start_decode
 - evx::entropy_coder, 69
- startTime
 - mpeg7cdva::SegmentDescriptor, 94
- StartTrainingMode
 - mpeg7cdva::CoordinateCoding, 59
- swap
 - mpeg7cdva::Buffer, 28
- timesPresent
 - mpeg7cdva::RefFeature, 89
- toBinary
 - mpeg7cdva::CoordinateCoding, 59
- toCDVScfl
 - mpeg7cdva::CompressedFeatureList, 46
- uint16
 - evx, 13
- uint32
 - evx, 13
- uint64
 - evx, 13
- uint8
 - evx, 13
- unaligned_bit_copy
 - evx, 15
- vCount
 - mpeg7cdva::CoordinateCoding::CircularSum↔ Context, 43
- vInitialMap
 - mpeg7cdva::CoordinateCoding::CircularSum↔ Context, 43
- vMap
 - mpeg7cdva::CoordinateCoding::CircularSum↔ Context, 43
- verboseMode
 - mpeg7cdva::CdvaImpl, 42
- version.h, 107
 - EVX_MAJOR_VERSION, 107
 - EVX_MINOR_VERSION, 107
 - EVX_VERSION_CHANGELIST, 107
 - EVX_VERSION_MAJOR, 108
 - EVX_VERSION_MINOR, 108
 - EVX_VERSION_WORD, 108
- wchar
 - evx, 13
- what
 - mpeg7cdva::CdvaException, 29
- WrappedCoordList
 - mpeg7cdva::WrappedCoordList, 95
- write
 - mpeg7cdva::Buffer, 28
 - mpeg7cdva::CompressedSegmentDescriptor, 52
 - mpeg7cdva::SegmentDescriptor, 93
- write_bit
 - evx::bitstream, 22
- write_bits
 - evx::bitstream, 22
- write_byte
 - evx::bitstream, 23
- write_bytes
 - evx::bitstream, 23
- writeDescIndex
 - mpeg7cdva::DescriptorTimeMap, 66
- writeSeparateContext
 - mpeg7cdva::CoordinateCoding, 60
- xCoord
 - mpeg7cdva::LeanCoordList, 81
- yCoord
 - mpeg7cdva::LeanCoordList, 81