

MPEG CDVA Experimentation Model (CXM)

1.0

Generated by Doxygen 1.8.11

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	mpeg7cdva Namespace Reference	11
6.1.1	Detailed Description	12
6.1.2	Typedef Documentation	12
6.1.2.1	ShotDescriptorList	12
6.1.3	Enumeration Type Documentation	12
6.1.3.1	LogFormat	12
6.1.3.2	OPERATION	12

7 Data Structure Documentation	13
7.1 <code>mpeg7cdva::Buffer</code> Class Reference	13
7.1.1 Detailed Description	14
7.1.2 Constructor & Destructor Documentation	14
7.1.2.1 <code>Buffer()</code>	14
7.1.2.2 <code>~Buffer()</code>	14
7.1.2.3 <code>Buffer(size_t size)</code>	14
7.1.2.4 <code>Buffer(unsigned char *data, size_t size)</code>	14
7.1.2.5 <code>Buffer(const Buffer &)</code>	15
7.1.3 Member Function Documentation	15
7.1.3.1 <code>assign(const unsigned char *data, size_t size)</code>	15
7.1.3.2 <code>clear()</code>	15
7.1.3.3 <code>compare(const Buffer &other) const</code>	15
7.1.3.4 <code>data()</code>	15
7.1.3.5 <code>data() const</code>	15
7.1.3.6 <code>empty() const</code>	15
7.1.3.7 <code>equals(Buffer &buffer)</code>	15
7.1.3.8 <code>fill(unsigned char value=0)</code>	16
7.1.3.9 <code>operator=(const Buffer &)</code>	16
7.1.3.10 <code>operator==(const Buffer &other) const</code>	16
7.1.3.11 <code>read(const char *fname)</code>	16
7.1.3.12 <code>resize(size_t newsize)</code>	16
7.1.3.13 <code>sdata()</code>	16
7.1.3.14 <code>sdata() const</code>	16
7.1.3.15 <code>size() const</code>	16
7.1.3.16 <code>swap(Buffer &x)</code>	16
7.1.3.17 <code>write(const char *fname) const</code>	16
7.2 <code>mpeg7cdva::CdvaException</code> Class Reference	17
7.2.1 Detailed Description	17
7.2.2 Constructor & Destructor Documentation	17

7.2.2.1	CdvaException(std::string str)	17
7.2.2.2	~CdvaException()	18
7.2.3	Member Function Documentation	18
7.2.3.1	what() const	18
7.3	mpeg7cdva::CdvaImpl Class Reference	18
7.3.1	Detailed Description	19
7.3.2	Constructor & Destructor Documentation	20
7.3.2.1	CdvaImpl()	20
7.3.2.2	~CdvaImpl()	20
7.3.3	Member Function Documentation	20
7.3.3.1	byDescendingScore(const MatchData &m1, const MatchData &m2)	20
7.3.3.2	checkBitrate(int bitrate)	20
7.3.3.3	close()	20
7.3.3.4	commitDB()	20
7.3.3.5	extract(const std::string &descname, const std::string &videopathname, int bitrate, ExtractData &outdata) const	20
7.3.3.6	getDescriptorExt(int bitrate)	21
7.3.3.7	getExt(const std::string &imageName)	21
7.3.3.8	init(OPERATION op, bool verbose, size_t n_videos, int querybitrate, int refbitrate=0, bool calcdescsizes=false)	21
7.3.3.9	makeindex(const std::string &cdva_descriptor, const std::string &relativepathname)	21
7.3.3.10	match(MatchData &matchResults, const ShotDescriptorList &qDescList, const ShotDescriptorList &rDescList)	21
7.3.3.11	match(MatchData &matchResults, const std::string &qdescname, const std::string &rdescname, int qbitrate, int rbitrate)	21
7.3.3.12	parse(const std::string &descFile, ShotDescriptorList &shotList)	22
7.3.3.13	retrieve(std::vector< MatchData > &retrievalResults, const std::string &qdescname, int qbitrate)	22
7.3.4	Field Documentation	22
7.3.4.1	calc_desc_sizes	22
7.3.4.2	cdvsclient	22
7.3.4.3	cdvsconfig	22
7.3.4.4	cdvsMode	22

7.3.4.5	cdvsserver	22
7.3.4.6	current_op	22
7.3.4.7	drop_frame_th	23
7.3.4.8	max_retrieved	23
7.3.4.9	shot_cut_th	23
7.3.4.10	skip_after	23
7.3.4.11	skip_before	23
7.3.4.12	verboseMode	23
7.4	mpeg7cdva::ExtractData Class Reference	23
7.4.1	Detailed Description	24
7.4.2	Constructor & Destructor Documentation	24
7.4.2.1	ExtractData()	24
7.4.3	Member Function Documentation	24
7.4.3.1	setDescriptorLength(double length)	24
7.4.3.2	setNumFrames(double nframes)	24
7.4.3.3	setNumShots(double nshots)	24
7.4.3.4	setVideoDuration(double time)	25
7.4.4	Field Documentation	25
7.4.4.1	clip_duration	25
7.4.4.2	coordinate_bit_count	25
7.4.4.3	descriptorlength	25
7.4.4.4	global_bit_count	25
7.4.4.5	header_bit_count	25
7.4.4.6	local_bit_count	25
7.4.4.7	n_keyframes	25
7.4.4.8	numframes	25
7.4.4.9	numshots	25
7.5	mpeg7cdva::FileManager Class Reference	25
7.5.1	Detailed Description	26
7.5.2	Constructor & Destructor Documentation	26

7.5.2.1	FileManager(const char *annotationpathname, int level=0)	26
7.5.2.2	~FileManager()	27
7.5.3	Member Function Documentation	27
7.5.3.1	countNames(size_t i) const	27
7.5.3.2	getDatasetName() const	27
7.5.3.3	getDatasetPath() const	27
7.5.3.4	getDatasetPathName() const	27
7.5.3.5	getDatasetSize() const	28
7.5.3.6	getQueryName(size_t i, bool absolutePathname=true) const	28
7.5.3.7	getReferenceName(size_t i, bool absolutePathname=true) const	28
7.5.3.8	getWorkspaceDir() const	28
7.5.3.9	replaceExt(const std::string &imageName, const char *ext) const	29
7.5.3.10	replacePath(const std::string &imageName, const char *newpath)	29
7.5.3.11	setWorkspaceDir(const char *workdir)	29
7.6	mpeg7cdva::LogManager Class Reference	29
7.6.1	Detailed Description	30
7.6.2	Constructor & Destructor Documentation	30
7.6.2.1	LogManager()	30
7.6.2.2	~LogManager()	30
7.6.3	Member Function Documentation	31
7.6.3.1	close()	31
7.6.3.2	init(int formats, const std::string &datasetpath, const std::string &datasetname, size_t n_videos, int mode, int refmode=0)	31
7.6.3.3	printExtractData(int index, const std::string &videoname, const ExtractData &data)	31
7.6.3.4	printExtractHeader()	31
7.6.3.5	printMatchData(int index, const std::string &queryvideoname, const std::string &refvideoname, const MatchData &matchData)	31
7.6.3.6	printMatchHeader()	31
7.6.3.7	printRetrievalData(int index, const std::string &queryvideoname, const std::vector< MatchData > &retrievalResults)	31
7.6.3.8	printRetrievalHeader()	31
7.7	mpeg7cdva::MatchData Class Reference	31

7.7.1	Detailed Description	32
7.7.2	Constructor & Destructor Documentation	32
7.7.2.1	MatchData()	32
7.7.2.2	~MatchData()	32
7.7.3	Member Function Documentation	32
7.7.3.1	getFirstMatchingTime() const	32
7.7.3.2	getLastMatchingTime() const	32
7.7.3.3	getReferenceId() const	33
7.7.3.4	getScore() const	33
7.7.3.5	setMatchingScore(double myscore)	33
7.7.3.6	setMatchingTime(double time_s)	33
7.7.3.7	setReferenceID(const std::string reference)	33
7.8	mpeg7cdva::ShotDescriptor Class Reference	34
7.8.1	Detailed Description	34
7.8.2	Member Function Documentation	34
7.8.2.1	clear()	34
7.8.2.2	empty() const	35
7.8.2.3	getEndTimeMs() const	35
7.8.2.4	getParity() const	35
7.8.2.5	getStartTimeMs() const	35
7.8.2.6	read(std::ifstream &fin, mpeg7cdvs::CdvsServer *cdvsserver)	35
7.8.2.7	setEndTimeMs(unsigned long position_ms)	35
7.8.2.8	setParity(unsigned int value)	35
7.8.2.9	setStartTimeMs(unsigned long position_ms)	35
7.8.2.10	write(std::ofstream &fout) const	35
7.8.3	Field Documentation	35
7.8.3.1	keyframes	35
8	File Documentation	37
8.1	Buffer.h File Reference	37
8.2	cdva.h File Reference	38
8.3	CdvaException.h File Reference	39
8.4	Cdvalmpl.h File Reference	40
8.5	FileManager.h File Reference	41
8.6	LogManager.h File Reference	42
	Index	45

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:

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

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

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

mpeg7cdva::Buffer	13
mpeg7cdva::CdvaImpl	18
exception	
mpeg7cdva::CdvaException	17
mpeg7cdva::ExtractData	23
mpeg7cdva::FileManager	25
mpeg7cdva::LogManager	29
mpeg7cdva::MatchData	31
mpeg7cdva::ShotDescriptor	34

Chapter 4

Data Structure Index

4.1 Data Structures

Here are the data structures with brief descriptions:

mpeg7cdva::Buffer	A container class for a byte array, intended to replace all malloc() and new() instructions in the main code	13
mpeg7cdva::CdvaException	Class defining a specific exception for CDVA	17
mpeg7cdva::CdvaImpl	A CDVA implementation based on multiple CDVS descriptors	18
mpeg7cdva::ExtractData	A class containing the results of an extraction operation	23
mpeg7cdva::FileManager	Helper class to manage lists of file names	25
mpeg7cdva::LogManager	Helper class to produce log files in various formats (csv, text, XML, etc.)	29
mpeg7cdva::MatchData	A class containing the results of a matching or retrieval operation	31
mpeg7cdva::ShotDescriptor	A container for CdvsDescriptors belonging to the same shot	34

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

Buffer.h	37
cdva.h	38
CdvaException.h	39
CdvaImpl.h	40
FileManager.h	41
LogManager.h	42

Chapter 6

Namespace Documentation

6.1 mpeg7cdva Namespace Reference

Namespace used to encapsulate all MPEG-7 CDVA declarations.

Data Structures

- class [Buffer](#)
A container class for a byte array, intended to replace all malloc() and new() instructions in the main code.
- class [CdvaException](#)
Class defining a specific exception for CDVA.
- class [Cdvalmpl](#)
A CDVA implementation based on multiple CDVS descriptors.
- class [ExtractData](#)
A class containing the results of an extraction operation.
- class [FileManager](#)
Helper class to manage lists of file names.
- class [LogManager](#)
Helper class to produce log files in various formats (csv, text, XML, etc.)
- class [MatchData](#)
A class containing the results of a matching or retrieval operation.
- class [ShotDescriptor](#)
A container for CdvsDescriptors belonging to the same shot.

Typedefs

- typedef std::vector< [ShotDescriptor](#) > [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.

6.1.1 Detailed Description

Namespace used to encapsulate all MPEG-7 CDVA declarations.

6.1.2 Typedef Documentation

6.1.2.1 `typedef std::vector<ShotDescriptor> mpeg7cdva::ShotDescriptorList`

vector of shots descriptors of a video

6.1.3 Enumeration Type Documentation

6.1.3.1 `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

6.1.3.2 `enum mpeg7cdva::OPERATION`

Enumerator

UNKNOWN

EXTRACT

MATCH

RETRIEVE

Chapter 7

Data Structure Documentation

7.1 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](#) ()
create a buffer of the given size
- virtual [~Buffer](#) ()
- [Buffer](#) (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 newsz)
change buffer size; content is lost if newsz 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)

7.1.1 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

7.1.2 Constructor & Destructor Documentation

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

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

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

create a buffer of the given size

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

copy the given array into this `Buffer`

7.1.2.5 mpeg7cdva::Buffer::Buffer (const Buffer &)

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

7.1.3 Member Function Documentation

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

assign the given data to [Buffer](#)

7.1.3.2 void mpeg7cdva::Buffer::clear ()

clear the [Buffer](#)

7.1.3.3 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.

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

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

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

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

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

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

7.1.3.7 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)

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

fill a `Buffer` with the given value

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

assign a `Buffer` to another

7.1.3.10 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)

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

read `Buffer` from a file

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

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

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

access to `Buffer`'s data as signed char (writable)

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

access to `Buffer`'s data as signed char (read only)

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

return the current size of the `Buffer`

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

swap the content of two `Buffer`(s)

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

write `Buffer` to file

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

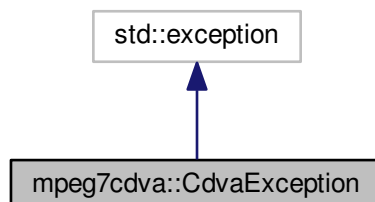
- [Buffer.h](#)

7.2 mpeg7cdva::CdvaException Class Reference

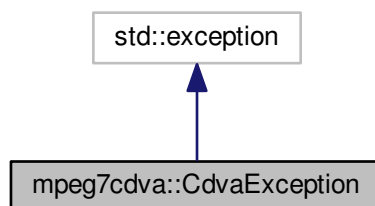
Class defining a specific exception for CDVA.

```
#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.

7.2.1 Detailed Description

Class defining a specific exception for CDVA.

7.2.2 Constructor & Destructor Documentation

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

Create a new CDVA exception.

Parameters

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

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

7.2.3 Member Function Documentation

7.2.3.1 `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](#)

7.3 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)`
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)

Static Protected Member Functions

- static bool [byDescendingScore](#) (const [MatchData](#) &m1, const [MatchData](#) &m2)
- 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
- size_t [max_retrieved](#)
maximum number of retrieved images
- bool [calc_desc_sizes](#)
calculate size of descriptor components

7.3.1 Detailed Description

A CDVA implementation based on multiple CDVS descriptors.

7.3.2 Constructor & Destructor Documentation

7.3.2.1 `mpeg7cdva::Cdvalmpl::Cdvalmpl ()`

7.3.2.2 `virtual mpeg7cdva::Cdvalmpl::~~Cdvalmpl ()` `[virtual]`

7.3.3 Member Function Documentation

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

7.3.3.2 `static bool mpeg7cdva::Cdvalmpl::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

7.3.3.3 `virtual void mpeg7cdva::Cdvalmpl::close ()` `[virtual]`

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

7.3.3.4 `void mpeg7cdva::Cdvalmpl::commitDB ()`

Save database information.

7.3.3.5 `void mpeg7cdva::Cdvalmpl::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

7.3.3.6 static const char* mpeg7cdva::Cdvalmpl::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

7.3.3.7 static std::string mpeg7cdva::Cdvalmpl::getExt (const std::string & *imageName*) [static], [protected]

7.3.3.8 virtual void mpeg7cdva::Cdvalmpl::init (OPERATION *op*, bool *verbose*, size_t *n_videos*, int *querybitrate*, int *refbitrate* = 0, bool *calcdescsizes* = false) [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

7.3.3.9 virtual void mpeg7cdva::Cdvalmpl::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

7.3.3.10 virtual double mpeg7cdva::Cdvalmpl::match (MatchData & *matchResults*, const ShotDescriptorList & *qDescList*, const ShotDescriptorList & *rDescList*) [protected], [virtual]

7.3.3.11 virtual double mpeg7cdva::Cdvalmpl::match (MatchData & *matchResults*, const std::string & *qdescname*, const std::string & *rdescname*, 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>qdescname</i>	input query descriptor name
<i>rdescname</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)

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

7.3.3.13 `virtual void mpeg7cdva::Cdvalmpl::retrieve (std::vector< MatchData > & retrievalResults, const std::string & qdescname, 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>qdescname</i>	the video query descriptor
<i>qbitrate</i>	query bitrate (one of 16,64,256)

7.3.4 Field Documentation

7.3.4.1 `bool mpeg7cdva::Cdvalmpl::calc_desc_sizes` [protected]

calculate size of descriptor components

7.3.4.2 `mpeg7cdvs::CdvsClient* mpeg7cdva::Cdvalmpl::cdvsclient` [protected]

7.3.4.3 `mpeg7cdvs::CdvsConfiguration* mpeg7cdva::Cdvalmpl::cdvsconfig` [protected]

7.3.4.4 `int mpeg7cdva::Cdvalmpl::cdvsMode` [protected]

the CDVS mode that will be used to encode keyframe descriptors

7.3.4.5 `mpeg7cdvs::CdvsServer* mpeg7cdva::Cdvalmpl::cdvsserver` [protected]

7.3.4.6 `OPERATION mpeg7cdva::Cdvalmpl::current_op` [protected]

the current operation

7.3.4.7 `double mpeg7cdva::Cdvalmpl::drop_frame_th` [protected]

drop frame threshold

7.3.4.8 `size_t mpeg7cdva::Cdvalmpl::max_retrieved` [protected]

maximum number of retrieved images

7.3.4.9 `double mpeg7cdva::Cdvalmpl::shot_cut_th` [protected]

shot cut threshold

7.3.4.10 `int mpeg7cdva::Cdvalmpl::skip_after` [protected]

number of video frames to skip after decoding one

7.3.4.11 `int mpeg7cdva::Cdvalmpl::skip_before` [protected]

number of video frames to skip before decoding one

7.3.4.12 `bool mpeg7cdva::Cdvalmpl::verboseMode` [protected]

verbose mode indicator

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

- [Cdvalmpl.h](#)

7.4 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](#)

7.4.1 Detailed Description

A class containing the results of an extraction operation.

7.4.2 Constructor & Destructor Documentation

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

7.4.3 Member Function Documentation

7.4.3.1 `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.
---------------	--

7.4.3.2 `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.
----------------	-------------------------------------

7.4.3.3 `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.
---------------	------------------------------------

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

set the video duration in seconds.

Parameters

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

7.4.4 Field Documentation

7.4.4.1 double mpeg7cdva::ExtractData::clip_duration

7.4.4.2 int mpeg7cdva::ExtractData::coordinate_bit_count

7.4.4.3 double mpeg7cdva::ExtractData::descriptorlength

7.4.4.4 int mpeg7cdva::ExtractData::global_bit_count

7.4.4.5 int mpeg7cdva::ExtractData::header_bit_count

7.4.4.6 int mpeg7cdva::ExtractData::local_bit_count

7.4.4.7 int mpeg7cdva::ExtractData::n_keyframes

7.4.4.8 double mpeg7cdva::ExtractData::numframes

7.4.4.9 double mpeg7cdva::ExtractData::numshots

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

- [cdva.h](#)

7.5 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.

7.5.1 Detailed Description

Helper class to manage lists of file names.

Author

Massimo Balestri

Date

2012

7.5.2 Constructor & Destructor Documentation

7.5.2.1 `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.

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

7.5.3 Member Function Documentation

7.5.3.1 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.

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

Get the dataset name.

Returns

the dataset name

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

Get the dataset base directory.

Returns

the dataset path

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

Get the dataset full pathname.

Returns

the dataset pathname

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

Get the dataset size.

Returns

the number of lines read from the filename.

7.5.3.6 `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.

7.5.3.7 `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.

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

Get the workspace directory.

This is the directory where output files will be stored.

Returns

the workspace directory

7.5.3.9 `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.

7.5.3.10 `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.

7.5.3.11 `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](#)

7.6 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](#) ()

7.6.1 Detailed Description

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

Author

Massimo Balestri

Date

2015

7.6.2 Constructor & Destructor Documentation

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

constructor

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

destructor

7.6.3 Member Function Documentation

7.6.3.1 void mpeg7cdva::LogManager::close ()

7.6.3.2 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.

7.6.3.3 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.

7.6.3.4 void mpeg7cdva::LogManager::printExtractHeader ()

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

7.6.3.5 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.

7.6.3.6 void mpeg7cdva::LogManager::printMatchHeader ()

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

7.6.3.7 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.

7.6.3.8 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](#)

7.7 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.

7.7.1 Detailed Description

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

7.7.2 Constructor & Destructor Documentation

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

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

7.7.3 Member Function Documentation

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

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

Returns

the time in seconds from the start of the video clip

7.7.3.2 `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

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

get the string that identifies the matching reference video clip.

Returns

the video clip relative pathname

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

Get the matching score.

Returns

the score

7.7.3.5 `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
----------------	----------------------------

7.7.3.6 `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
---------------	--

7.7.3.7 `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](#)

7.8 mpeg7cdva::ShotDescriptor Class Reference

A container for CdvsDescriptors belonging to the same shot.

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

Public Member Functions

- 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.
- size_t [write](#) (std::ofstream &fout) const
write (appending) this descriptor to a file
- size_t [read](#) (std::ifstream &fin, mpeg7cdvs::CdvsServer *cdvsserver)
read (from the current position) this descriptor from a file
- bool [empty](#) () const
returns true if the shot is empty
- void [clear](#) ()
clear the shot container

Data Fields

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

7.8.1 Detailed Description

A container for CdvsDescriptors belonging to the same shot.

7.8.2 Member Function Documentation

7.8.2.1 void mpeg7cdva::ShotDescriptor::clear ()

clear the shot container

7.8.2.2 `bool mpeg7cdva::ShotDescriptor::empty () const`

returns true if the shot is empty

7.8.2.3 `unsigned long mpeg7cdva::ShotDescriptor::getEndTimeMs () const`

Get the frame start time in milliseconds.

7.8.2.4 `unsigned int mpeg7cdva::ShotDescriptor::getParity () const`

Get the shot parity.

7.8.2.5 `unsigned long mpeg7cdva::ShotDescriptor::getStartTimeMs () const`

Get the shot start time in milliseconds.

7.8.2.6 `size_t mpeg7cdva::ShotDescriptor::read (std::ifstream & fin, mpeg7cdvs::CdvsServer * cdvsserver)`

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

7.8.2.7 `void mpeg7cdva::ShotDescriptor::setEndTimeMs (unsigned long position_ms)`

Set the start time of the frame in milliseconds.

7.8.2.8 `void mpeg7cdva::ShotDescriptor::setParity (unsigned int value)`

Set the shot parity.

7.8.2.9 `void mpeg7cdva::ShotDescriptor::setStartTimeMs (unsigned long position_ms)`

Set the start time of the shot in milliseconds.

7.8.2.10 `size_t mpeg7cdva::ShotDescriptor::write (std::ofstream & fout) const`

write (appending) this descriptor to a file

7.8.3 Field Documentation

7.8.3.1 `std::vector<mpeg7cdvs::CdvsDescriptor> mpeg7cdva::ShotDescriptor::keyframes`

keyframes belonging to this shot

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

- [CdvalImpl.h](#)

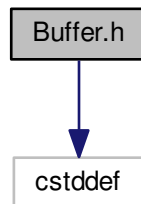
Chapter 8

File Documentation

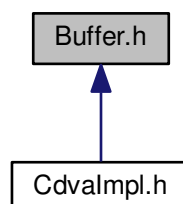
8.1 Buffer.h File Reference

```
#include <stddef>
```

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

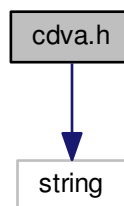
- `mpeg7cdva`

Namespace used to encapsulate all MPEG-7 CDVA declarations.

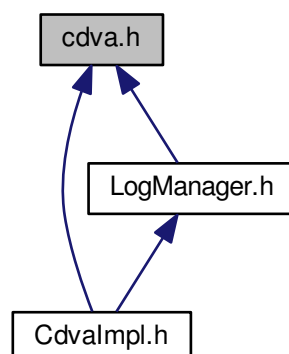
8.2 cdva.h File Reference

```
#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.

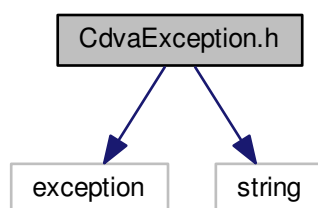
Namespaces

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

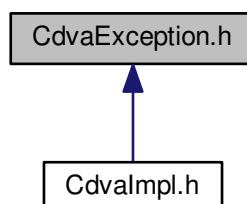
8.3 CdvaException.h File Reference

```
#include <exception>
#include <string>
```

Include dependency graph for CdvaException.h:



This graph shows which files directly or indirectly include this file:



Data Structures

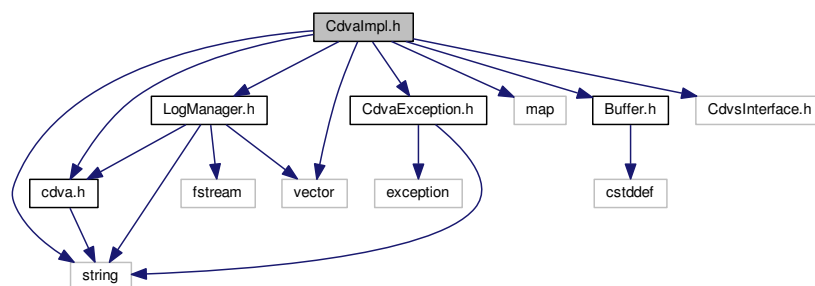
- class [mpeg7cdva::CdvaException](#)
Class defining a specific exception for CDVA.

Namespaces

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

8.4 Cdvalmpl.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 dependency graph for Cdvalmpl.h:
```



Data Structures

- class [mpeg7cdva::ShotDescriptor](#)
A container for `CdvsDescriptors` belonging to the same shot.
- class [mpeg7cdva::Cdvalmpl](#)
A CDVA implementation based on multiple CDVS descriptors.

Namespaces

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

Typedefs

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

Enumerations

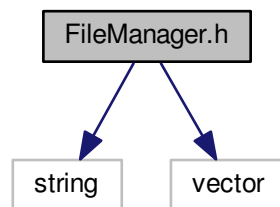
- enum [mpeg7cdva::OPERATION](#) { [mpeg7cdva::UNKNOWN](#), [mpeg7cdva::EXTRACT](#), [mpeg7cdva::MATCH](#), [mpeg7cdva::RETRIEVE](#) }

8.5 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.

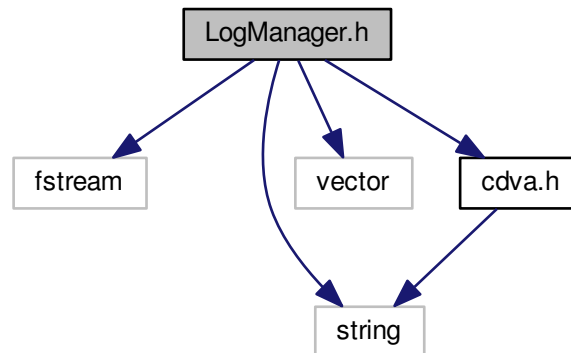
Namespaces

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

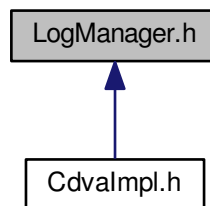
8.6 LogManager.h File Reference

```
#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.

Index

- ~Buffer
 - mpeg7cdva::Buffer, [14](#)
- ~CdvaException
 - mpeg7cdva::CdvaException, [18](#)
- ~CdvalImpl
 - mpeg7cdva::CdvalImpl, [20](#)
- ~FileManager
 - mpeg7cdva::FileManager, [27](#)
- ~LogManager
 - mpeg7cdva::LogManager, [30](#)
- ~MatchData
 - mpeg7cdva::MatchData, [32](#)
- assign
 - mpeg7cdva::Buffer, [15](#)
- Buffer
 - mpeg7cdva::Buffer, [14](#)
- Buffer.h, [37](#)
- byDescendingScore
 - mpeg7cdva::CdvalImpl, [20](#)
- calc_desc_sizes
 - mpeg7cdva::CdvalImpl, [22](#)
- cdva.h, [38](#)
- CdvaException
 - mpeg7cdva::CdvaException, [17](#)
- CdvaException.h, [39](#)
- CdvalImpl
 - mpeg7cdva::CdvalImpl, [20](#)
- CdvalImpl.h, [40](#)
- cdvsMode
 - mpeg7cdva::CdvalImpl, [22](#)
- cdvsclient
 - mpeg7cdva::CdvalImpl, [22](#)
- cdvsconfig
 - mpeg7cdva::CdvalImpl, [22](#)
- cdvsserver
 - mpeg7cdva::CdvalImpl, [22](#)
- checkBitrate
 - mpeg7cdva::CdvalImpl, [20](#)
- clear
 - mpeg7cdva::Buffer, [15](#)
 - mpeg7cdva::ShotDescriptor, [34](#)
- clip_duration
 - mpeg7cdva::ExtractData, [25](#)
- close
 - mpeg7cdva::CdvalImpl, [20](#)
 - mpeg7cdva::LogManager, [31](#)
- commitDB
 - mpeg7cdva::CdvalImpl, [20](#)
- compare
 - mpeg7cdva::Buffer, [15](#)
- coordinate_bit_count
 - mpeg7cdva::ExtractData, [25](#)
- countNames
 - mpeg7cdva::FileManager, [27](#)
- current_op
 - mpeg7cdva::CdvalImpl, [22](#)
- data
 - mpeg7cdva::Buffer, [15](#)
- descriptorlength
 - mpeg7cdva::ExtractData, [25](#)
- drop_frame_th
 - mpeg7cdva::CdvalImpl, [22](#)
- EXTRACT
 - mpeg7cdva, [12](#)
- empty
 - mpeg7cdva::Buffer, [15](#)
 - mpeg7cdva::ShotDescriptor, [34](#)
- equals
 - mpeg7cdva::Buffer, [15](#)
- extract
 - mpeg7cdva::CdvalImpl, [20](#)
- ExtractData
 - mpeg7cdva::ExtractData, [24](#)
- FORMAT_CSV
 - mpeg7cdva, [12](#)
- FORMAT_HTML
 - mpeg7cdva, [12](#)
- FORMAT_NONE
 - mpeg7cdva, [12](#)
- FORMAT_TEXT
 - mpeg7cdva, [12](#)
- FileManager
 - mpeg7cdva::FileManager, [26](#)
- FileManager.h, [41](#)
- fill
 - mpeg7cdva::Buffer, [15](#)
- getDatasetName
 - mpeg7cdva::FileManager, [27](#)
- getDatasetPath
 - mpeg7cdva::FileManager, [27](#)
- getDatasetPathName
 - mpeg7cdva::FileManager, [27](#)
- getDatasetSize

- mpeg7cdva::FileManager, 27
- getDescriptorExt
 - mpeg7cdva::CdvalImpl, 20
- getEndTimeMs
 - mpeg7cdva::ShotDescriptor, 35
- getExt
 - mpeg7cdva::CdvalImpl, 21
- getFirstMatchingTime
 - mpeg7cdva::MatchData, 32
- getLastMatchingTime
 - mpeg7cdva::MatchData, 32
- getParity
 - mpeg7cdva::ShotDescriptor, 35
- getQueryName
 - mpeg7cdva::FileManager, 28
- getReferenceld
 - mpeg7cdva::MatchData, 32
- getReferenceName
 - mpeg7cdva::FileManager, 28
- getScore
 - mpeg7cdva::MatchData, 33
- getStartTimeMs
 - mpeg7cdva::ShotDescriptor, 35
- getWorkspaceDir
 - mpeg7cdva::FileManager, 28
- global_bit_count
 - mpeg7cdva::ExtractData, 25
- header_bit_count
 - mpeg7cdva::ExtractData, 25
- init
 - mpeg7cdva::CdvalImpl, 21
 - mpeg7cdva::LogManager, 31
- keyframes
 - mpeg7cdva::ShotDescriptor, 35
- local_bit_count
 - mpeg7cdva::ExtractData, 25
- LogFormat
 - mpeg7cdva, 12
- LogManager
 - mpeg7cdva::LogManager, 30
- LogManager.h, 42
- MATCH
 - mpeg7cdva, 12
- makeindex
 - mpeg7cdva::CdvalImpl, 21
- match
 - mpeg7cdva::CdvalImpl, 21
- MatchData
 - mpeg7cdva::MatchData, 32
- max_retrieved
 - mpeg7cdva::CdvalImpl, 23
- mpeg7cdva, 11
 - EXTRACT, 12
 - FORMAT_CSV, 12
 - FORMAT_HTML, 12
 - FORMAT_NONE, 12
 - FORMAT_TEXT, 12
 - LogFormat, 12
 - MATCH, 12
 - OPERATION, 12
 - RETRIEVE, 12
 - ShotDescriptorList, 12
 - UNKNOWN, 12
- mpeg7cdva::Buffer, 13
 - ~Buffer, 14
 - assign, 15
 - Buffer, 14
 - clear, 15
 - compare, 15
 - data, 15
 - empty, 15
 - equals, 15
 - fill, 15
 - operator=, 16
 - operator==, 16
 - read, 16
 - resize, 16
 - sdata, 16
 - size, 16
 - swap, 16
 - write, 16
- mpeg7cdva::CdvaException, 17
 - ~CdvaException, 18
 - CdvaException, 17
 - what, 18
- mpeg7cdva::CdvalImpl, 18
 - ~CdvalImpl, 20
 - byDescendingScore, 20
 - calc_desc_sizes, 22
 - CdvalImpl, 20
 - cdvsMode, 22
 - cdvsclient, 22
 - cdvsconfig, 22
 - cdvsserver, 22
 - checkBitrate, 20
 - close, 20
 - commitDB, 20
 - current_op, 22
 - drop_frame_th, 22
 - extract, 20
 - getDescriptorExt, 20
 - getExt, 21
 - init, 21
 - makeindex, 21
 - match, 21
 - max_retrieved, 23
 - parse, 22
 - retrieve, 22
 - shot_cut_th, 23
 - skip_after, 23
 - skip_before, 23
 - verboseMode, 23

- mpeg7cdva::ExtractData, 23
 - clip_duration, 25
 - coordinate_bit_count, 25
 - descriptorlength, 25
 - ExtractData, 24
 - global_bit_count, 25
 - header_bit_count, 25
 - local_bit_count, 25
 - n_keyframes, 25
 - numframes, 25
 - numshots, 25
 - setDescriptionLength, 24
 - setNumFrames, 24
 - setNumShots, 24
 - setVideoDuration, 25
- mpeg7cdva::FileManager, 25
 - ~FileManager, 27
 - countNames, 27
 - FileManager, 26
 - getDatasetName, 27
 - getDatasetPath, 27
 - getDatasetPathName, 27
 - getDatasetSize, 27
 - getQueryName, 28
 - getReferenceName, 28
 - getWorkspaceDir, 28
 - replaceExt, 28
 - replacePath, 29
 - setWorkspaceDir, 29
- mpeg7cdva::LogManager, 29
 - ~LogManager, 30
 - close, 31
 - init, 31
 - LogManager, 30
 - printExtractData, 31
 - printExtractHeader, 31
 - printMatchData, 31
 - printMatchHeader, 31
 - printRetrievalData, 31
 - printRetrievalHeader, 31
- mpeg7cdva::MatchData, 31
 - ~MatchData, 32
 - getFirstMatchingTime, 32
 - getLastMatchingTime, 32
 - getReferenceId, 32
 - getScore, 33
 - MatchData, 32
 - setMatchingScore, 33
 - setMatchingTime, 33
 - setReferenceID, 33
- mpeg7cdva::ShotDescriptor, 34
 - clear, 34
 - empty, 34
 - getEndTimeMs, 35
 - getParity, 35
 - getStartTimeMs, 35
 - keyframes, 35
 - read, 35
 - setEndTimeMs, 35
 - setParity, 35
 - setStartTimeMs, 35
 - write, 35
- n_keyframes
 - mpeg7cdva::ExtractData, 25
- numframes
 - mpeg7cdva::ExtractData, 25
- numshots
 - mpeg7cdva::ExtractData, 25
- OPERATION
 - mpeg7cdva, 12
- operator=
 - mpeg7cdva::Buffer, 16
- operator==
 - mpeg7cdva::Buffer, 16
- parse
 - mpeg7cdva::CdvaImpl, 22
- printExtractData
 - mpeg7cdva::LogManager, 31
- printExtractHeader
 - mpeg7cdva::LogManager, 31
- printMatchData
 - mpeg7cdva::LogManager, 31
- printMatchHeader
 - mpeg7cdva::LogManager, 31
- printRetrievalData
 - mpeg7cdva::LogManager, 31
- printRetrievalHeader
 - mpeg7cdva::LogManager, 31
- RETRIEVE
 - mpeg7cdva, 12
- read
 - mpeg7cdva::Buffer, 16
 - mpeg7cdva::ShotDescriptor, 35
- replaceExt
 - mpeg7cdva::FileManager, 28
- replacePath
 - mpeg7cdva::FileManager, 29
- resize
 - mpeg7cdva::Buffer, 16
- retrieve
 - mpeg7cdva::CdvaImpl, 22
- sdata
 - mpeg7cdva::Buffer, 16
- setDescriptionLength
 - mpeg7cdva::ExtractData, 24
- setEndTimeMs
 - mpeg7cdva::ShotDescriptor, 35
- setMatchingScore
 - mpeg7cdva::MatchData, 33
- setMatchingTime
 - mpeg7cdva::MatchData, 33
- setNumFrames

- mpeg7cdva::ExtractData, [24](#)
- setNumShots
 - mpeg7cdva::ExtractData, [24](#)
- setParity
 - mpeg7cdva::ShotDescriptor, [35](#)
- setReferenceID
 - mpeg7cdva::MatchData, [33](#)
- setStartTimeMs
 - mpeg7cdva::ShotDescriptor, [35](#)
- setVideoDuration
 - mpeg7cdva::ExtractData, [25](#)
- setWorkspaceDir
 - mpeg7cdva::FileManager, [29](#)
- shot_cut_th
 - mpeg7cdva::CdvaImpl, [23](#)
- ShotDescriptorList
 - mpeg7cdva, [12](#)
- size
 - mpeg7cdva::Buffer, [16](#)
- skip_after
 - mpeg7cdva::CdvaImpl, [23](#)
- skip_before
 - mpeg7cdva::CdvaImpl, [23](#)
- swap
 - mpeg7cdva::Buffer, [16](#)
- UNKNOWN
 - mpeg7cdva, [12](#)
- verboseMode
 - mpeg7cdva::CdvaImpl, [23](#)
- what
 - mpeg7cdva::CdvaException, [18](#)
- write
 - mpeg7cdva::Buffer, [16](#)
 - mpeg7cdva::ShotDescriptor, [35](#)