

# CDVA Evaluation Framework

## 2.0

Generated by Doxygen 1.8.9.1

Mon Nov 2 2015 18:03:15



# Contents

<b>1</b>	<b>Documentation</b>	<b>1</b>
<b>2</b>	<b>Namespace Index</b>	<b>3</b>
2.1	Namespace List . . . . .	3
<b>3</b>	<b>Hierarchical Index</b>	<b>5</b>
3.1	Class Hierarchy . . . . .	5
<b>4</b>	<b>Data Structure Index</b>	<b>7</b>
4.1	Data Structures . . . . .	7
<b>5</b>	<b>File Index</b>	<b>9</b>
5.1	File List . . . . .	9
<b>6</b>	<b>Namespace Documentation</b>	<b>11</b>
6.1	mpeg7cdva Namespace Reference . . . . .	11
6.1.1	Detailed Description . . . . .	11
6.1.2	Enumeration Type Documentation . . . . .	12
6.1.2.1	LogFormat . . . . .	12
<b>7</b>	<b>Data Structure Documentation</b>	<b>13</b>
7.1	mpeg7cdva::Buffer Class Reference . . . . .	13
7.1.1	Detailed Description . . . . .	14
7.1.2	Constructor & Destructor Documentation . . . . .	14
7.1.2.1	Buffer . . . . .	14
7.1.2.2	~Buffer . . . . .	14
7.1.2.3	Buffer . . . . .	14
7.1.2.4	Buffer . . . . .	14
7.1.2.5	Buffer . . . . .	14
7.1.3	Member Function Documentation . . . . .	14
7.1.3.1	assign . . . . .	14
7.1.3.2	clear . . . . .	15
7.1.3.3	compare . . . . .	15
7.1.3.4	data . . . . .	15

7.1.3.5	data	15
7.1.3.6	empty	15
7.1.3.7	equals	15
7.1.3.8	fill	15
7.1.3.9	operator=	15
7.1.3.10	operator==	15
7.1.3.11	read	15
7.1.3.12	resize	15
7.1.3.13	sdata	16
7.1.3.14	sdata	16
7.1.3.15	size	16
7.1.3.16	swap	16
7.1.3.17	write	16
7.2	mpeg7cdva::CdvaException Class Reference	16
7.2.1	Detailed Description	17
7.2.2	Constructor & Destructor Documentation	17
7.2.2.1	CdvaException	17
7.2.2.2	~CdvaException	17
7.2.3	Member Function Documentation	17
7.2.3.1	what	17
7.3	mpeg7cdva::CdvaImpl Class Reference	18
7.3.1	Detailed Description	18
7.3.2	Constructor & Destructor Documentation	18
7.3.2.1	CdvaImpl	18
7.3.2.2	~CdvaImpl	18
7.3.3	Member Function Documentation	18
7.3.3.1	checkBitrate	18
7.3.3.2	close	19
7.3.3.3	extract	19
7.3.3.4	getDescriptorExt	19
7.3.3.5	init	19
7.3.3.6	makeindex	19
7.3.3.7	match	20
7.3.3.8	retrieve	20
7.4	mpeg7cdva::DbRecord Class Reference	20
7.4.1	Detailed Description	21
7.4.2	Field Documentation	21
7.4.2.1	descriptor	21
7.4.2.2	descriptorId	21
7.5	mpeg7cdva::FileManager Class Reference	21

7.5.1	Detailed Description	22
7.5.2	Constructor & Destructor Documentation	22
7.5.2.1	FileManager	22
7.5.2.2	~FileManager	23
7.5.3	Member Function Documentation	23
7.5.3.1	countNames	23
7.5.3.2	getDatasetName	23
7.5.3.3	getDatasetPath	23
7.5.3.4	getDatasetPathName	23
7.5.3.5	getDatasetSize	23
7.5.3.6	getQueryName	23
7.5.3.7	getReferenceName	24
7.5.3.8	getWorkspaceDir	24
7.5.3.9	replaceExt	24
7.5.3.10	replacePath	24
7.5.3.11	setWorkspaceDir	25
7.6	mpeg7cdva::LogManager Class Reference	25
7.6.1	Detailed Description	26
7.6.2	Constructor & Destructor Documentation	26
7.6.2.1	LogManager	26
7.6.2.2	~LogManager	26
7.6.3	Member Function Documentation	26
7.6.3.1	close	26
7.6.3.2	init	26
7.6.3.3	printExtractData	26
7.6.3.4	printExtractHeader	26
7.6.3.5	printMatchData	26
7.6.3.6	printMatchHeader	27
7.6.3.7	printRetrievalData	27
7.6.3.8	printRetrievalHeader	27
7.6.3.9	setDescriptorLength	27
7.6.3.10	setNumFrames	27
7.6.3.11	setVideoDuration	27
7.7	mpeg7cdva::MatchData Class Reference	27
7.7.1	Detailed Description	28
7.7.2	Constructor & Destructor Documentation	28
7.7.2.1	MatchData	28
7.7.2.2	~MatchData	28
7.7.3	Member Function Documentation	28
7.7.3.1	getFirstMatchingTime	28

7.7.3.2	<a href="#">getLastMatchingTime</a>	28
7.7.3.3	<a href="#">getReferenceId</a>	28
7.7.3.4	<a href="#">getScore</a>	29
7.7.3.5	<a href="#">setMatchingScore</a>	29
7.7.3.6	<a href="#">setMatchingTime</a>	29
7.7.3.7	<a href="#">setReferenceID</a>	29
<b>8</b>	<b>File Documentation</b>	<b>31</b>
8.1	<a href="#">Buffer.h File Reference</a>	31
8.2	<a href="#">cdva.h File Reference</a>	32
8.3	<a href="#">CdvaException.h File Reference</a>	33
8.4	<a href="#">CdvaImpl.h File Reference</a>	34
8.5	<a href="#">FileManager.h File Reference</a>	34
8.6	<a href="#">LogManager.h File Reference</a>	35
<b>Index</b>		<b>37</b>

# Chapter 1

## Documentation

This is the documentation of the C++ classes implementing the MPEG CDVA Evaluation Framework. The software implements the recommendations contained in the following document:

- N15338: "Evaluation Framework for Compact Descriptors for Video Analysis - Search and Retrieval", July 2015, Warsaw, Poland

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 that are visible when the CD↔  
VA Library headers (in particular CdvaInterface.h) are included . . . . . 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
mpeg7cdva::DbRecord . . . . .	20
exception	
mpeg7cdva::CdvaException . . . . .	16
mpeg7cdva::FileManager . . . . .	21
mpeg7cdva::LogManager . . . . .	25
mpeg7cdva::MatchData . . . . .	27



## Chapter 4

# Data Structure Index

### 4.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">mpeg7cdva::Buffer</a>	A container class for a byte array, intended to replace all malloc() and new() instructions in the main code . . . . .	13
<a href="#">mpeg7cdva::CdvaException</a>	Class defining a specific exception for CDVA . . . . .	16
<a href="#">mpeg7cdva::CdvaImpl</a>	A CDVA implementation template class . . . . .	18
<a href="#">mpeg7cdva::DbRecord</a>	Container class for DB records . . . . .	20
<a href="#">mpeg7cdva::FileManager</a>	Helper class to manage lists of file names . . . . .	21
<a href="#">mpeg7cdva::LogManager</a>	Helper class to produce log files in various formats (csv, text, XML, etc.) . . . . .	25
<a href="#">mpeg7cdva::MatchData</a>	A class containing the results of a matching or retrieval operation . . . . .	27



## Chapter 5

# File Index

### 5.1 File List

Here is a list of all files with brief descriptions:

<a href="#">Buffer.h</a>	31
<a href="#">cdva.h</a>	32
<a href="#">CdvaException.h</a>	33
<a href="#">Cdvalmpl.h</a>	34
<a href="#">FileManager.h</a>	34
<a href="#">LogManager.h</a>	35





## Chapter 6

# Namespace Documentation

### 6.1 mpeg7cdva Namespace Reference

Namespace used to encapsulate all MPEG-7 CDVA declarations that are visible when the CDVA Library headers (in particular `CdvaInterface.h`) are included.

#### 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 [CdvaImpl](#)  
*A CDVA implementation template class.*
- class [DbRecord](#)  
*a container class for DB records*
- 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.*

#### Enumerations

- 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 that are visible when the CDVA Library headers (in particular `CdvaInterface.h`) are included.

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

## 6.1.2 Enumeration Type Documentation

### 6.1.2.1 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

## 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](#) ()
- 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)*

### 7.1.1 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 <a href="#">Buffer</a>
--------------	----------------------------------

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&amp; mpeg7cdva::Buffer::operator= ( const Buffer &amp; )

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:

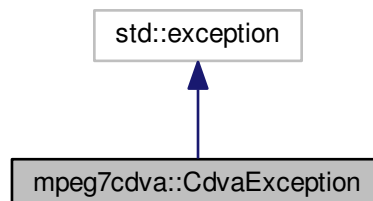
- [Buffer.h](#)

## 7.2 mpeg7cdva::CdvaException Class Reference

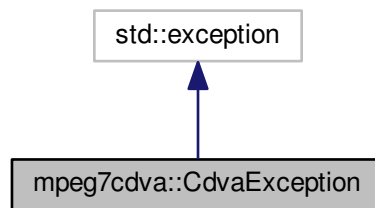
Class defining a specific exception for CDVA.

```
#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::Cdvalmpl Class Reference

A CDVA implementation template class.

```
#include <Cdvalmpl.h>
```

### Public Member Functions

- [Cdvalmpl](#) ()
- virtual [~Cdvalmpl](#) ()
- void [init](#) (bool verbose, size\_t n\_videos, int querybitrate, int refbitrate=0)  
*initialization method - called once before processing videos.*
- void [extract](#) (const std::string &descname, const std::string &videopathname, int bitrate, [LogManager](#) &logmanager)  
*Video processing method - called once for each video in the list.*
- 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.*
- void [makeindex](#) (const std::string &cdva\_descriptor, const std::string &relativepathname)  
*Video indexing method - builds a DB of reference video descriptors.*
- void [retrieve](#) (std::vector< [MatchData](#) > &retrievalResults, const std::string &qdescname, int qbitrate, [LogManager](#) &logmanager)  
*Video retrieval method - returns a list of reference videos matching the query video.*
- 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.*

#### 7.3.1 Detailed Description

A CDVA implementation template class.

#### 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::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

## 7.3.3.2 void mpeg7cdva::Cdvalmpl::close ( )

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

7.3.3.3 void mpeg7cdva::Cdvalmpl::extract ( const std::string & *descrname*, const std::string & *videopathname*, int *bitrate*, [LogManager](#) & *logmanager* )

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>logmanager</i>	the <a href="#">LogManager</a> instance that saves data into log files

7.3.3.4 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

7.3.3.5 void mpeg7cdva::Cdvalmpl::init ( bool *verbose*, size\_t *n\_videos*, int *querybitrate*, int *refbitrate* = 0 )

initialization method - called once before processing videos.

## Parameters

<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)

7.3.3.6 void mpeg7cdva::Cdvalmpl::makeindex ( const std::string & *cdva\_descriptor*, const std::string & *relativepathname* )

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

## Parameters

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

7.3.3.7 `double mpeg7cdva::CdvalImpl::match ( MatchData & matchResults, const std::string & qdescrname, const std::string & rdescrname, int qbitrate, int rbitrate )`

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)

7.3.3.8 `void mpeg7cdva::CdvalImpl::retrieve ( std::vector< MatchData > & retrievalResults, const std::string & qdescrname, int qbitrate, LogManager & logmanager )`

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)
<i>logmanager</i>	the <a href="#">LogManager</a> instance that saves data into log files

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

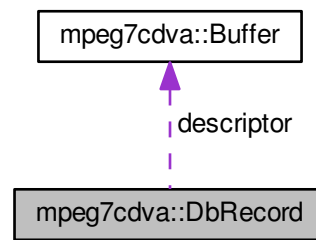
- [CdvalImpl.h](#)

## 7.4 mpeg7cdva::DbRecord Class Reference

a container class for DB records

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

Collaboration diagram for mpeg7cdva::DbRecord:



## Data Fields

- `std::string` [descriptorId](#)  
the string that identifies the video (e.g. pathname)
- [Buffer](#) [descriptor](#)  
the buffer containing the video descriptor

### 7.4.1 Detailed Description

a container class for DB records

### 7.4.2 Field Documentation

#### 7.4.2.1 Buffer mpeg7cdva::DbRecord::descriptor

the buffer containing the video descriptor

#### 7.4.2.2 std::string mpeg7cdva::DbRecord::descriptorId

the string that identifies the video (e.g. pathname)

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

- [CdvalImpl.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>annotationpath- name</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>absolute↔ Pathname</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>absolute↔ Pathname</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 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 [setVideoDuration](#) (double time)  
*set the video duration in seconds.*
- void [setNumFrames](#) (double nframes)  
*set the number of frames of the video clip.*
- void [setDescriptorLength](#) (double length)  
*set the actual descriptor length (in bytes).*
- void [printExtractHeader](#) ()  
*method used by the Evaluation Framework to produce log files; do not change.*
- void [printExtractData](#) (int index, const std::string &videoname)  
*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 )`

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.

7.6.3.9 void mpeg7cdva::LogManager::setDescriptorLength ( double *length* )

set the actual descriptor length (in bytes).

Parameters

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

7.6.3.10 void mpeg7cdva::LogManager::setNumFrames ( double *nframes* )

set the number of frames of the video clip.

Parameters

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

7.6.3.11 void mpeg7cdva::LogManager::setVideoDuration ( double *time* )

set the video duration in seconds.

Parameters

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

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](#)



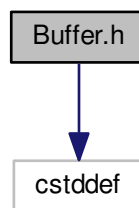
## Chapter 8

# File Documentation

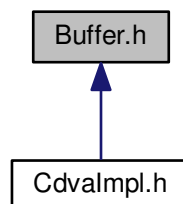
### 8.1 Buffer.h File Reference

```
#include <cstdint>
```

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

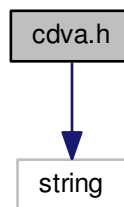
- [mpeg7cdva](#)

*Namespace used to encapsulate all MPEG-7 CDVA declarations that are visible when the CDVA Library headers (in particular `CdvaInterface.h`) are included.*

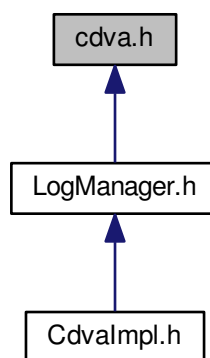
## 8.2 cdva.h File Reference

```
#include <string>
```

Include dependency graph for `cdva.h`:



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



## Data Structures

- class [mpeg7cdva::MatchData](#)

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

## Namespaces

- [mpeg7cdva](#)

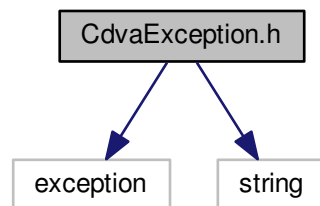
*Namespace used to encapsulate all MPEG-7 CDVA declarations that are visible when the CDVA Library headers (in particular CdvaInterface.h) are included.*

## 8.3 CdvaException.h File Reference

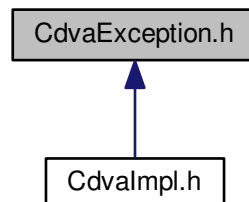
```
#include <exception>
```

```
#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.*

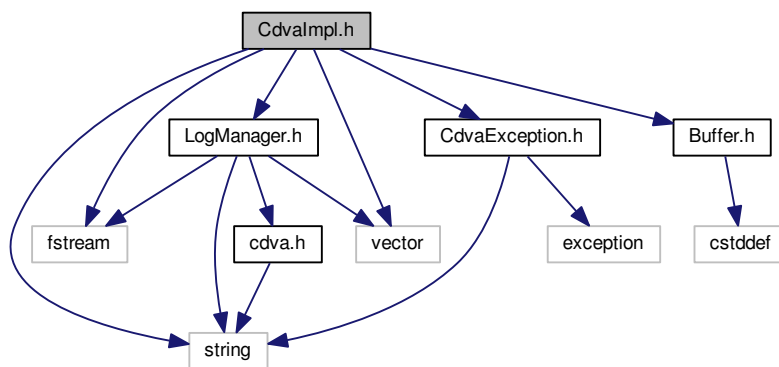
## Namespaces

- [mpeg7cdva](#)

*Namespace used to encapsulate all MPEG-7 CDVA declarations that are visible when the CDVA Library headers (in particular CdvaInterface.h) are included.*

## 8.4 CdvalImpl.h File Reference

```
#include <fstream>
#include <string>
#include <vector>
#include "CdvaException.h"
#include "LogManager.h"
#include "Buffer.h"
Include dependency graph for CdvalImpl.h:
```



## Data Structures

- class [mpeg7cdva::DbRecord](#)  
a container class for DB records
- class [mpeg7cdva::CdvalImpl](#)  
A CDVA implementation template class.

## Namespaces

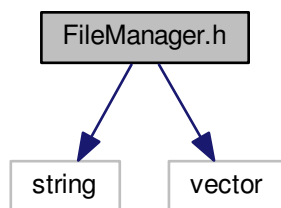
- [mpeg7cdva](#)  
Namespace used to encapsulate all MPEG-7 CDVA declarations that are visible when the CDVA Library headers (in particular `CdvalInterface.h`) are included.

## 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.*

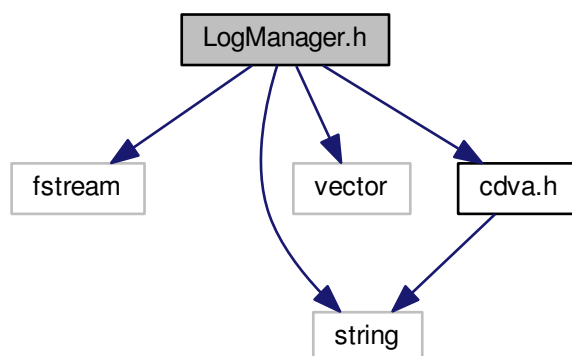
## Namespaces

- `mpeg7cdva`  
*Namespace used to encapsulate all MPEG-7 CDVA declarations that are visible when the CDVA Library headers (in particular `CdvaInterface.h`) are included.*

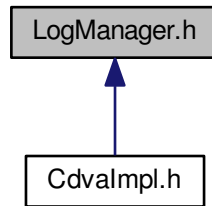
## 8.6 LogManager.h File Reference

```
#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 that are visible when the CDVA Library headers (in particular [CdvaInterface.h](#)) are included.*

## 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, [17](#)
- ~CdvalImpl
  - mpeg7cdva::CdvalImpl, [18](#)
- ~FileManager
  - mpeg7cdva::FileManager, [23](#)
- ~LogManager
  - mpeg7cdva::LogManager, [26](#)
- ~MatchData
  - mpeg7cdva::MatchData, [28](#)
- assign
  - mpeg7cdva::Buffer, [14](#)
- Buffer
  - mpeg7cdva::Buffer, [14](#)
- Buffer.h, [31](#)
- cdva.h, [32](#)
- CdvaException
  - mpeg7cdva::CdvaException, [17](#)
- CdvaException.h, [33](#)
- CdvalImpl
  - mpeg7cdva::CdvalImpl, [18](#)
- CdvalImpl.h, [34](#)
- checkBitrate
  - mpeg7cdva::CdvalImpl, [18](#)
- clear
  - mpeg7cdva::Buffer, [14](#)
- close
  - mpeg7cdva::CdvalImpl, [19](#)
  - mpeg7cdva::LogManager, [26](#)
- compare
  - mpeg7cdva::Buffer, [15](#)
- countNames
  - mpeg7cdva::FileManager, [23](#)
- data
  - mpeg7cdva::Buffer, [15](#)
- descriptor
  - mpeg7cdva::DbRecord, [21](#)
- descriptorId
  - mpeg7cdva::DbRecord, [21](#)
- empty
  - mpeg7cdva::Buffer, [15](#)
- equals
  - mpeg7cdva::Buffer, [15](#)
- extract
  - mpeg7cdva::CdvalImpl, [19](#)
- FORMAT\_CSV
  - mpeg7cdva, [12](#)
- FORMAT\_HTML
  - mpeg7cdva, [12](#)
- FORMAT\_NONE
  - mpeg7cdva, [12](#)
- FORMAT\_TEXT
  - mpeg7cdva, [12](#)
- FileManager
  - mpeg7cdva::FileManager, [22](#)
- FileManager.h, [34](#)
- fill
  - mpeg7cdva::Buffer, [15](#)
- getDatasetName
  - mpeg7cdva::FileManager, [23](#)
- getDatasetPath
  - mpeg7cdva::FileManager, [23](#)
- getDatasetPathName
  - mpeg7cdva::FileManager, [23](#)
- getDatasetSize
  - mpeg7cdva::FileManager, [23](#)
- getDescriptorExt
  - mpeg7cdva::CdvalImpl, [19](#)
- getFirstMatchingTime
  - mpeg7cdva::MatchData, [28](#)
- getLastMatchingTime
  - mpeg7cdva::MatchData, [28](#)
- getQueryName
  - mpeg7cdva::FileManager, [23](#)
- getReferenceld
  - mpeg7cdva::MatchData, [28](#)
- getReferenceName
  - mpeg7cdva::FileManager, [24](#)
- getScore
  - mpeg7cdva::MatchData, [28](#)
- getWorkspaceDir
  - mpeg7cdva::FileManager, [24](#)
- init
  - mpeg7cdva::CdvalImpl, [19](#)
  - mpeg7cdva::LogManager, [26](#)
- LogFormat
  - mpeg7cdva, [12](#)
- LogManager
  - mpeg7cdva::LogManager, [26](#)
- LogManager.h, [35](#)

- makeindex
  - mpeg7cdva::CdvaImpl, 19
- match
  - mpeg7cdva::CdvaImpl, 20
- MatchData
  - mpeg7cdva::MatchData, 28
- mpeg7cdva, 11
  - FORMAT\_CSV, 12
  - FORMAT\_HTML, 12
  - FORMAT\_NONE, 12
  - FORMAT\_TEXT, 12
  - LogFormat, 12
- mpeg7cdva::Buffer, 13
  - ~Buffer, 14
  - assign, 14
  - Buffer, 14
  - clear, 14
  - compare, 15
  - data, 15
  - empty, 15
  - equals, 15
  - fill, 15
  - operator=, 15
  - operator==, 15
  - read, 15
  - resize, 15
  - sdata, 15, 16
  - size, 16
  - swap, 16
  - write, 16
- mpeg7cdva::CdvaException, 16
  - ~CdvaException, 17
  - CdvaException, 17
  - what, 17
- mpeg7cdva::CdvaImpl, 18
  - ~CdvaImpl, 18
  - CdvaImpl, 18
  - checkBitrate, 18
  - close, 19
  - extract, 19
  - getDescriptorExt, 19
  - init, 19
  - makeindex, 19
  - match, 20
  - retrieve, 20
- mpeg7cdva::DbRecord, 20
  - descriptor, 21
  - descriptorId, 21
- mpeg7cdva::FileManager, 21
  - ~FileManager, 23
  - countNames, 23
  - FileManager, 22
  - getDatasetName, 23
  - getDatasetPath, 23
  - getDatasetPathName, 23
  - getDatasetSize, 23
  - getQueryName, 23
  - getReferenceName, 24
  - getWorkspaceDir, 24
  - replaceExt, 24
  - replacePath, 24
  - setWorkspaceDir, 25
- mpeg7cdva::LogManager, 25
  - ~LogManager, 26
  - close, 26
  - init, 26
  - LogManager, 26
  - printExtractData, 26
  - printExtractHeader, 26
  - printMatchData, 26
  - printMatchHeader, 26
  - printRetrievalData, 27
  - printRetrievalHeader, 27
  - setDescriptorLength, 27
  - setNumFrames, 27
  - setVideoDuration, 27
- mpeg7cdva::MatchData, 27
  - ~MatchData, 28
  - getFirstMatchingTime, 28
  - getLastMatchingTime, 28
  - getReferenceId, 28
  - getScore, 28
  - MatchData, 28
  - setMatchingScore, 29
  - setMatchingTime, 29
  - setReferenceID, 29
- operator=
  - mpeg7cdva::Buffer, 15
- operator==
  - mpeg7cdva::Buffer, 15
- printExtractData
  - mpeg7cdva::LogManager, 26
- printExtractHeader
  - mpeg7cdva::LogManager, 26
- printMatchData
  - mpeg7cdva::LogManager, 26
- printMatchHeader
  - mpeg7cdva::LogManager, 26
- printRetrievalData
  - mpeg7cdva::LogManager, 27
- printRetrievalHeader
  - mpeg7cdva::LogManager, 27
- read
  - mpeg7cdva::Buffer, 15
- replaceExt
  - mpeg7cdva::FileManager, 24
- replacePath
  - mpeg7cdva::FileManager, 24
- resize
  - mpeg7cdva::Buffer, 15
- retrieve
  - mpeg7cdva::CdvaImpl, 20
- sdata

- mpeg7cdva::Buffer, [15](#), [16](#)
- setDescriptionLength
  - mpeg7cdva::LogManager, [27](#)
- setMatchingScore
  - mpeg7cdva::MatchData, [29](#)
- setMatchingTime
  - mpeg7cdva::MatchData, [29](#)
- setNumFrames
  - mpeg7cdva::LogManager, [27](#)
- setReferenceID
  - mpeg7cdva::MatchData, [29](#)
- setVideoDuration
  - mpeg7cdva::LogManager, [27](#)
- setWorkspaceDir
  - mpeg7cdva::FileManager, [25](#)
- size
  - mpeg7cdva::Buffer, [16](#)
- swap
  - mpeg7cdva::Buffer, [16](#)
- what
  - mpeg7cdva::CdvaException, [17](#)
- write
  - mpeg7cdva::Buffer, [16](#)