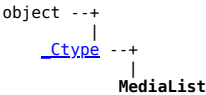


Class MediaList

[source code](#)



Create a new `MediaList` instance.

Usage: MediaList(list_of_MRLs)

See `vlc.Instance.media_list` new documentation for details.

Instance Methods		[hide private]
	<code>get_instance(self)</code>	source code
	<code>add_media(self, mrl)</code> Add media instance to media list.	source code
	<code>release(self)</code> Release media list created with new().	source code
	<code>retain(self)</code> Retain reference to a media list.	source code
	<code>set_media(self, p_md)</code> Associate media instance with this media list instance.	source code
	<code>media(self)</code> Get media instance from this media list instance.	source code
	<code>insert_media(self, p_md, i_pos)</code> Insert media instance in media list on a position The lock should be held upon entering this function.	source code
	<code>remove_index(self, i_pos)</code> Remove media instance from media list on a position The lock should be held upon entering this function.	source code
	<code>count(self)</code> Get count on media list items The lock should be held upon entering this function.	source code
	<code>__len__(self)</code>	source code
	<code>item_at_index(self, i_pos)</code> List media instance in media list at a position The lock should be held upon entering this function.	source code
	<code>__getitem__(self, i)</code>	source code
	<code>__iter__(self)</code>	source code
	<code>index_of_item(self, p_md)</code> Find index position of List media instance in media list.	source code
	<code>is_readonly(self)</code> This indicates if this media list is read-only from a user point of view.	source code
	<code>lock(self)</code> Get lock on media list items.	source code
	<code>unlock(self)</code> Release lock on media list items The lock should be held upon entering this function.	source code
	<code>event_manager(self)</code> Decorator.	source code
Inherited from object: <code>__delattr__</code> , <code>__format__</code> , <code>__getattr__</code> , <code>__hash__</code> , <code>__init__</code> , <code>__reduce__</code> , <code>__reduce_ex__</code> , <code>__repr__</code> , <code>__setattr__</code> , <code>__sizeof__</code> , <code>__str__</code> , <code>__subclasshook__</code>		

Static Methods		[hide private]
a new object with type S, a subtype of T	<u><code>new</code></u> (cls, *args)	source code
Inherited from <u><code>Ctype</code></u> : from_param		

Properties

Inherited from object: `__class__`

Method Details
<p><code>__new__(cls, *args)</code> source code</p> <p><i>Static Method</i></p> <p>Returns: a new object with type S, a subtype of T</p> <p>Overrides: <code>object.__new__</code> <i>(inherited documentation)</i></p>

```
add_media(self, mrl)
```

Add media instance to media list.

The **lock** should be held upon entering this function

Parameters:

- `mrl` - a media instance or a MRL.

Returns:

0 on success, -1 if the media list is read-only.

set_media(self, p_md)[source code](#)

Associate media instance with this media list instance. If another media instance was present it will be released. The [lock](#) should NOT be held upon entering this function.

Parameters:

- `p_md` - media instance to add.

media(self)[source code](#)

Get media instance from this media list instance. This action will increase the refcount on the media instance. The [lock](#) should NOT be held upon entering this function.

Returns:

media instance.

insert_media(self, p_md, i_pos)[source code](#)

Insert media instance in media list on a position The [lock](#) should be held upon entering this function.

Parameters:

- `p_md` - a media instance.
- `i_pos` - position in array where to insert.

Returns:

0 on success, -1 if the media list is read-only.

remove_index(self, i_pos)[source code](#)

Remove media instance from media list on a position The [lock](#) should be held upon entering this function.

Parameters:

- `i_pos` - position in array where to insert.

Returns:

0 on success, -1 if the list is read-only or the item was not found.

count(self)[source code](#)

Get count on media list items The [lock](#) should be held upon entering this function.

Returns:

number of items in media list.

item_at_index(self, i_pos)[source code](#)

List media instance in media list at a position The [lock](#) should be held upon entering this function.

Parameters:

- `i_pos` - position in array where to insert.

Returns:

media instance at position `i_pos`, or None if not found. In case of success, `media_retain()` is called to increase the refcount on the media.

index_of_item(self, p_md)[source code](#)

Find index position of List media instance in media list. Warning: the function will return the first matched position. The [lock](#) should be held upon entering this function.

Parameters:

- `p_md` - media instance.

Returns:

position of media instance or -1 if media not found.

is_readonly(self)[source code](#)

This indicates if this media list is read-only from a user point of view.

Returns:

1 on readonly, 0 on readwrite \libvlc_ret

`event_manager(self)`

Decorator. Caches a parameterless method's return value each time it is called.

If called later with the same arguments, the cached value is returned (not reevaluated). Adapted from <https://wiki.python.org/moin/PythonDecoratorLibrary>

Decorators:

- `@memoize_parameterless`

[Home](#) [Trees](#) [Indices](#) [Help](#)

Generated by Epydoc 3.0.1 on Sat Jul 25 13:13:08 2020

<http://epydoc.sourceforge.net>