NAME

media_getattr, media_setattr – get and set media attributes

SYNOPSIS

cc [*flag* . . .] *file*. . . –**lvolmgt** [*library*. . . .]

#include <volmgt.h>

char *****media_getattr(char** **vol_path***, char** **attr***);**

int media_setattr(char **vol_path***, char** **attr***, char** **value***);**

MT-LEVEL

MT-Safe

DESCRIPTION

media_setattr( ) and media_getattr( ) respectively set and get attribute-value pairs (called properties) on a per-volume basis.

Volume Management supports system properties and user properties. System properties are ones that Volume Management predefines. Some of these system properties are writable, but only by the user that owns the volume being specified, and some system properties are read only:

Attribute	Writable	Value	Description
s-access	RO	"seq", "rand"	sequential or random access
s-density	RO	"low", "medium", "high"	media density
s-parts	RO	comma separated list of slice numbers	list of partitions on this volume
s-location	RO	*pathname*	Volume Management pathname to media
s-mejectable	RO	"true", "false"	whether or not media is manually ejectable
s-rmoneject	R/W	"true", "false"	should media access points be removed from database upon ejection
s-enxio	R/W	"true", "false"	if set return **ENXIO** when media access attempted

Properties can also be defined by the user. In this case the value can be any string the user wishes.

RETURN VALUES

Upon successful completion **media_getattr( )** returns a pointer to the value corresponding to the specified attribute. A null pointer is returned if the specified volume doesn't exist, if the specified attribute for that volume doesn't exist, if the specified attribute is boolean and its value is false, or if **malloc**(3C) fails to allocate space for the return value.

media_setattr( ) returns **1** upon success, and **0** upon failure.

ERRORS

Both **media_getattr( )** and **media_setattr( )** can fail returning a null pointer if an **open**(2) of the specified *vol_path* fails, if an **fstat**(2) of that pathname fails, or if that pathname is not a block or character special device.

media_getattr( ) can also fail if the specified attribute was not found, and **media_setattr( )** can also fail if the caller doesn't have permission to set the attribute, either because it's is a system attribute, or because the caller doesn't own the specified volume.

Additionally, either routine can fail returning the following error values:

ENXIO The Volume Management daemon, **vold**, is not running

EINTR The routine was interrupted by the user before finishing

EXAMPLES

The following example checks to see if the volume called *fred* that Volume Management is managing can be ejected via software, or if it can only be manually ejected:

```
if (media_getattr("/vol/rdsk/fred", "s-mejectable") != NULL) {
      (void) printf("\"fred\" must be manually ejected\n");
} else {
      (void) printf("software can eject \"fred\"\n");
}
```

This example shows setting the *s-enxio* property for the floppy volume currently in the first floppy drive:

```
int     res;

if ((res = media_setattr("/vol/dev/aliases/floppy0", "s-enxio",
   "true")) == 0) {
      (void) printf("can't set s-enxio flag for floppy0\n");
}
```

SEE ALSO

cc(1B), **vold**(1M), **lstat**(2), **open**(2), **readlink**(2), **stat**(2), **free**(3C), **malloc**(3C), **media_findname**(3X), **volmgt_check**(3X), **volmgt_inuse**(3X), **volmgt_root**(3X), **volmgt_running**(3X), **volmgt_symname**(3X)

NOTES

Upon success **media_getattr( )** returns a pointer to a string which has been allocated, and should be freed when no longer in use (see **free**(3C)).