

cdplay_library

COLLABORATORS

	<i>TITLE :</i> cdplay_library		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		February 24, 2025	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	cdplay_library	1
1.1	cdplay_library -- Inhalt	1
1.2	cdplay.library/CDClose()	1
1.3	cdplay.library/CDCloseDrive()	2
1.4	cdplay.library/CDInquiry()	2
1.5	cdplay.library/CDJump()	3
1.6	cdplay.library/CDLockDrive()	4
1.7	cdplay.library/CDNext()	5
1.8	cdplay.library/CDOpen()	5
1.9	cdplay.library/CDOpenDrive()	6
1.10	cdplay.library/CDPause()	7
1.11	cdplay.library/CDPlay()	8
1.12	cdplay.library/CDPrev()	8
1.13	cdplay.library/CDStop()	9
1.14	cdplay.library/CDUpdate()	9

Chapter 1

cdplay_library

1.1 cdplay_library -- Inhalt

cdplay.library

CDClose()	CDCloseDrive()	CDInquiry()	CDJump()	CDLockDrive()
CDNext()	CDOpen()	CDOpenDrive()	CDPause()	CDPlay()
CDPrev()	CDStop()	CDUpdate()		

1.2 cdplay.library/CDClose()

NAME

CDClose -- schließt die Schublade des CDRom Laufwerkes

SYNOPSIS

```
result = CDClose (request)
D0          A1
```

```
ULONG CDClose (struct CDRequest *);
```

FUNCTION

Schließt die Schublade des CDRom Laufwerkes.

INPUTS

```
request          --          "struct CDRequest *" des entsprechenden
                   Drives
```

RESULT

```
result          --          liefert im Fehlerfall einen von 0 verschiedenen
                   Wert, der dem Errorcode entspricht.
```

NOTES

Wird nicht von allen Laufwerken unterstützt

BUGS

keine bekannt.

SEE ALSO
cdplay.library/CDOpen()

1.3 cdplay.library/CDCloseDrive()

NAME

CDCloseDrive -- SCSI Device schließen und Speicher freigeben

SYNOPSIS

```
result = CDCloseDrive (request)
D0                A1
```

```
struct CDRequest *CDCloseDrive (struct CDRequest *);
```

FUNCTION

Schließt ein mit CDOpenDrive() geöffnet Device und gibt den benutzten Speicher wieder frei.

INPUTS

request -- CDRequest des zu schließenden Devices, diese Struktur wird von CDOpenDrive() geliefert.

RESULT

result -- Ergebniss des Kommandos, in der Regel muß alles klappen und es wird TRUE geliefert.

EXAMPLE

siehe cdplay.library/CDOpenDrive()

BUGS

keine bekannt

SEE ALSO

cdplay.library/CDOpenDrive(), cdplay.library/CDUpdate(),
cdplay.library/CDInquiry()

1.4 cdplay.library/CDInquiry()

NAME

CDInquiry -- liest spezifische Informationen aus dem Device aus

SYNOPSIS

```
result = CDInquiry (inquiry, request)
D0                A0                A1
```

```
ULONG CDInquiry (struct CDInquiry *, struct CDRequest *);
```

FUNCTION

Liest Informationen aus dem SCSI-Device aus. Dazu gehören unter anderen Hersteller und Produkt, Features des Devices, Typ (CDRom,

Festplatte, ...). Normalerweise wird diese Funktion nur einmal pro Device aufgerufen. CDOpenDrive() ruft die Funktion beim Öffnen auf, so das der normale Programmierer keine Verwendung haben dürfte.

INPUTS

```
inquiry      --          "struct CDInquiry *", Struktur die mit den
                    erhätlichen Informationen des Devices ge-
                    füllt wird.
request      --          "struct CDRequest *", wird von CDOpenDrive()
                    geliefert
```

RESULT

```
result      --          Im Fehlerfall ein von 0 verschiedener Wert
```

EXAMPLE

```
struct CDRequest *mycd;
struct CDInquiry *myinq;

if (mycd = CDOpenDrive ("scsi.device", 2))
{
    if (!CDInquiry (myinq, mycd))
    {
        /* erfolgreich */
    }
    else
    {
        /* fehlgeschlagen ;(( */
    }

    CDCloseDrive (mycd);
}
```

NOTES

Wird normalerweise nie gebraucht.

BUGS

keine bekannt

SEE ALSO

cdplay.library/CDOpenDrive(), cdplay.library/CDUpdate()

1.5 cdplay.library/CDJump()

NAME

CDJump -- springt innerhalb eines Tracks vor/zurück

SYNOPSIS

```
result = CDJump (offset, request)
D0          D0          A1
```

```
ULONG CDJump (LONG, struct CDRequest *);
```

FUNCTION

Springt innerhalb eines Tracks zu einer neuen Position. Dabei sowohl vorwärts als auch rückwärts gesprungen werden. Der Offset gibt dabei die Richtung an (Vorzeichen) und die Sprungweite. Ein Wert von 75 entspricht dabei einer Sekunde.

INPUTS

offset -- Sprungweite von der aktuellen Position, das Vorzeichen bestimmt die Richtung
request -- "struct CDRequest *" des entsprechenden Drives

RESULT

result -- liefert im Fehlerfall einen von 0 verschiedenen Wert, der dem Errorcode entspricht.

EXAMPLE

```
/* .... */
CDJump (-(10 * 75), mycd);            /* 10 Sekunden zurück springen */
/* .... */
```

BUGS

keine bekannt.

1.6 cdplay.library/CDLockDrive()

NAME

CDLockDrive -- ermöglicht das Öffnen des CD-Rom's zu sperren

SYNOPSIS

```
result = CDLockDrive (mode, request);
D0                           D0    A1

ULONG CDLockDrive (ULONG, struct CDRequest *);
```

FUNCTION

Ermöglicht es, den Auswurf von CD-Rom's zu vermeiden. Mit der Funktion kann die Klappe des Laufwerks verriegelt werden bzw. eine solche Verriegelung wieder aufgehoben werden.

INPUTS

mode -- Einen von 0 verschiedenen Wert zum Verriegeln oder 0, um das Laufwerk freizugeben.
request -- "struct CDRequest *" des entsprechenden Drives

RESULT

result -- liefert im Fehlerfall einen von 0 verschiedenen Wert, der dem Errorcode entspricht.

EXAMPLE

```
struct CDRequest *mycd;

if (mycd = CDOpenDrive ("scsi.device", 2))
```

```
{
    /* Laufwerk verriegeln */
    CDLockDrive (TRUE, mycd);

    CDCloseDrive (mycd);
}
```

NOTES

Nach einem Reset ist die Verriegelung bei einigen Laufwerken aufgehoben, bei anderen nicht. Ausprobieren!

BUGS

keine bekannt.

1.7 cdplay.library/CDNext()

NAME

CDNext -- wählt den nächsten Track an

SYNOPSIS

```
result = CDNext (request)
D0          A1

ULONG CDNext (struct CDRequest *);
```

FUNCTION

Springt zum nächsten Track. Wenn es keinen nächsten Track gibt, hat der Aufruf keine Wirkung.

INPUTS

request -- "struct CDRequest *" des entsprechenden Drives

RESULT

result -- liefert im Fehlerfall einen von 0 verschiedenen Wert, der dem Errorcode entspricht.

BUGS

keine bekannt.

SEE ALSO

cdplay.library/CDPrev()

1.8 cdplay.library/CDOpen()

NAME

CDOpen -- öffnet die Schublade des CDRom Laufwerkes

SYNOPSIS

```
result = CDOpen (request)
D0          A1
```

```

        ULONG CDOpen (struct CDRequest *);

FUNCTION
    Öffnet die Schublade des CDROM Laufwerkes.

INPUTS
    request      --      "struct CDRequest *" des entsprechenden Drives

RESULT
    result       --      liefert im Fehlerfall einen von 0 verschiedenen
                        Wert, der dem Errorcode entspricht.

NOTES
    Wird nicht von allen Laufwerken unterstützt

BUGS
    keine bekannt.

SEE ALSO
    cdplay.library/CDClose()

```

1.9 cdplay.library/CDOpenDrive()

```

NAME
    CDOpenDrive -- SCSI Device als Audio-Player initialisieren

SYNOPSIS
    request = CDOpenDrive (device, unit)
    D0      ~      A0      D0

    struct CDRequest *CDOpenDrive (STRPTR, ULONG);

FUNCTION
    Öffnet das SCSI-Device mit der angegebenen Unit und initialisiert
    die CDRequest Struktur mit den nötigen Werten. Diese Funktion kann
    erfolgreich auf alle SCSI-Devices angewendet werden, es muß selbst-
    ständig gecheckt werden, ob das Device ein CD-ROM ist. Das dürfte
    aber kein Problem darstellen, in jeder Programmiersprache sollte
    dieser Test in einer Zeile durchführbar sein. Die Funktion liefert
    im Fehlerfall NULL zurück.

INPUTS
    device      --      Name des SCSI-Devices, z.B. "scsi.device"
    unit        --      Unit, an der das CD-Rom ansprechbar ist, z.B. 2

RESULT
    request     --      "struct CDRequest *", ein Zeiger auf eine
                        CDRequest-Struktur oder NULL im Fehlerfall.

EXAMPLE
    struct CDRequest *mycd;

```

```

if (mycd = CDOpenDrive ("scsi.device", 2))
{
    /* etwas mit dem Laufwerk anstellen */

    CDCloseDrive (mycd);
}

```

NOTES

Diese Funktion kann natürlich von beliebigen Programmen benutzt werden und hat nicht unmittelbar mit Audio-CD's zu tun. CDOpenDrive() und CDCloseDrive() können dazu verwendet werden, Informationen über SCSI-Devices zu bekommen.

BUGS

keine bekannt, allerdings sollte die Funktion wenn möglich MemoryPools nutzen statt mehrere kleine Blöcke zu allokkieren, das wird in einer der nächsten Versionen geändert.

SEE ALSO

cdplay.library/CDCloseDrive(), cdplay.library/CDUpdate(),
cdplay.library/CDInquiry()

1.10 cdplay.library/CDPause()

NAME

CDPause -- stoppt/startet die Wiedergabe

SYNOPSIS

```

result = CDPause (pause, request)
D0                D0                A1

ULONG CDPause (ULONG, struct CDRequest *);

```

FUNCTION

Mit dieser Funktion kann die Wiedergabe für eine gewünschte Zeit angehalten werden, um dann an der selben Stelle wieder fortzufahren.

INPUTS

```

pause            --          Ein von 0 verschiedener Wert stoppt die Wieder-
                        gabe, 0 selber startet sie, sofern das Lauf-
                        werk vorher aktiv war
request          --          "struct CDRequest *" des entsprechenden Drives

```

RESULT

```

result          --          liefert im Fehlerfall einen von 0 verschiedenen
                        Wert, der dem Errorcode entspricht.

```

BUGS

keine bekannt.

SEE ALSO

cdplay.library/CDStop(), cdplay.library/CDPlay()

1.11 cdplay.library/CDPlay()

NAME

CDPlay -- beginnt die Audiowiedergabe eines Tracks

SYNOPSIS

```
result = CDPlay (start, request)
D0          D0          A1
```

```
ULONG CDPlay (ULONG, struct CDRequest *);
```

FUNCTION

Beginnt die Wiedergabe eines Audiotracks. Der Starttrack selber wird als Parameter übergeben. Die Ausgabe wird erst beendet, wenn die ganze CD abgespielt worden ist, eine Funktion wie Pause oder Stop gewählt wurde oder die Funktion mit einem neuen Starttrack aufgerufen wird.

INPUTS

```
start          --          Nummer des Starttracks
request        --          "struct CDRequest *" des entsprechenden Drives
```

RESULT

```
result         --          liefert im Fehlerfall einen von 0 verschiedenen
                          Wert, der dem Errorcode entspricht.
```

BUGS

keine bekannt.

SEE ALSO

cdplay.library/CDStop(), cdplay.library/CDPause()

1.12 cdplay.library/CDPrev()

NAME

CDPrev -- wählt den vorherigen Track an

SYNOPSIS

```
result = CDPrev (request)
D0          A1
```

```
ULONG CDPrev (struct CDRequest *);
```

FUNCTION

Springt zum vorherigen Track. Wenn es keinen vorhergehenden Track gibt, hat der Aufruf keine Wirkung.

INPUTS

```
request        --          "struct CDRequest *" des entsprechenden Drives
```

RESULT

```
result         --          liefert im Fehlerfall einen von 0 verschiedenen
```

Wert, der dem Errorcode entspricht.

BUGS

keine bekannt.

SEE ALSO

cdplay.library/CDNext()

1.13 cdplay.library/CDStop()

NAME

CDStop -- stoppt die Wiedergabe einer Audio-CD

SYNOPSIS

```
result = CDStop (request)
D0          A1
```

```
ULONG CDStop (struct CDRequest *);
```

FUNCTION

Stoppt die Wiedergabe einer Audio-CD, die zuvor mit CDPlay() gestartet wurde.

INPUTS

request -- "struct CDRequest *" des entsprechenden Drives

RESULT

result -- liefert im Fehlerfall einen von 0 verschiedenen Wert, der dem Errorcode entspricht.

BUGS

keine bekannt.

SEE ALSO

cdplay.library/CDPause(), cdplay.library/CDPlay()

1.14 cdplay.library/CDUpdate()

NAME

CDUpdate -- aktualisiert alle Daten des CDRom's die im laufenden Betrieb veränderlich sind (z.B. CD-Wechsel)

SYNOPSIS

```
result = CDUpdate (request)
D0          A1
```

```
ULONG CDUpdate (struct CDRequest *);
```

FUNCTION

Alle Daten über das SCSI-Device, die sich im laufenden Betrieb

ändern können (also NICHT Herstellername etc.) werden mit dieser Funktion neu ermittelt und im CDRequest abgelegt. Dazu zählen Informationen über die CD, aber auch die Position (Zeit) auf der CD während des Audioplays.

INPUTS

request -- "struct CDRequest *", wird von CDOpenDrive()
 geliefert

RESULT

result -- Fehlercode, der dem Rückgabewert der DoIO()
 Funktion entspricht. Im Fehlerfall ist dies
 ein von 0 verschiedener Wert.

NOTES

Einige Funktionen rufen CDUpdate() von selber auf, ein Aufruf ist dann i.d.R. nicht mehr nötig. Zu diesen Funktionen gehören cdplay.library/CDJump(), cdplay.library/CDNext() und cdplay.library/CDPrev()

BUGS

keine bekannt

SEE ALSO

cdplay.library/CDJump(), cdplay.library/CDNext(),
cdplay.library/CDPrev()
