

**rxcd**

<b>COLLABORATORS</b>
----------------------

	<i>TITLE :</i> rxcd		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		July 31, 2024	

<b>REVISION HISTORY</b>
-------------------------

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>rxcD</b>	<b>1</b>
1.1	RxCD 11.0	1
1.2	RxCD Introduction	1
1.3	RxCD Author	3
1.4	RxCD Warning, Requirements, Installation and Distribution	3
1.5	RxCD Terms	4
1.6	RxCD Functions	4
1.7	RxCD Functions - CDCreate	5
1.8	RxCD Functions - CDDispose	6
1.9	RxCD Functions - CDAddChangeInt	6
1.10	RxCD Functions - CDRemChangeInt	7
1.11	RxCD Functions - CDAddFrameInt	7
1.12	RxCD Functions - CDRemFrameInt	7
1.13	RxCD Functions - CDEject	8
1.14	RxCD Functions - CDPause	8
1.15	RxCD Functions - CDStartStop	9
1.16	RxCD Functions - CDPlay	9
1.17	RxCD Functions - CDPlay10	10
1.18	RxCD Functions - CDPlay12	10
1.19	RxCD Functions - CDPlayMSF	11
1.20	RxCD Functions - CDPlayTrackIndex	11
1.21	RxCD Functions - CDPlayTrackRelative10	12
1.22	RxCD Functions - CDPlayTrackRelative12	13
1.23	RxCD Functions - CDCddb4String	13
1.24	RxCD Functions - CDInfo	13
1.25	RxCD Functions - CDModeSense	15
1.26	RxCD Functions - CDStatus	16
1.27	RxCD Functions - CDToc	17
1.28	RxCD Functions - CDRequestSense	18
1.29	RxCD Functions - CDTestUnit	18

---

---

1.30 RxCD Functions - CDVolume . . . . .	19
1.31 RxCD Functions - GetRxCDString . . . . .	19
1.32 RxCD Functions - Help . . . . .	20
1.33 RxCD FAQ . . . . .	21
1.34 RxCD Sense . . . . .	21

---

# Chapter 1

## rxcd

### 1.1 RxCD 11.0

RxCD 11.0 © 2001 Alfonso Ranieri

1. Introduction
2. Author
3. WRID
4. Terms
5. Functions
6. FAQ
7. Sense

### 1.2 RxCD Introduction

#### Introduction

This library contains low level functions to manipulate audio cd. Note that your cd soft or hard device might refuse to perform some operation.

A cd device is referred by a name, e.g. CD0 CD2, or by an unique cd descriptor id returned by `CDCreate()`.

You may use `CDCreate()` for speed reasons if in your macro you use a lot of RxCD functions calls.

You have to use `CDCreate()` anytime you want to add an interrupt.

So the first argument of any `CDXXX()` function will always be a cd definition:

- o id - the id returned by `CDCreate()`
- o cd - a cd name or an id
- o cdName - a cd name, not an id

The second argument of most `CDXXX()` functions is an ARexx stem name.

---

It is used to store error informations; quite all functions return an integer:

- o 0  
Success
- o >0  
Failure

On failure, if you have supplied a valid second argument, there will be set the fields:

- o DevErr  
The `exec.library` device error
  - o Status  
The SCSI2 Status
  - o CC  
The Check Condition flag of Status. This is very important.
  - o ASCIISense  
The sense data in ascii format
  - o Valid
  - o ErrorCode  
Always 0x 70x 71x
  - o SegmentNumber
  - o FileMark
  - o EOM
  - o ILI
  - o SenseKey  
The type of the error
  - o Error  
The SenseKey in human form
  - o Information
  - o ASL
  - o CSI
  - o ASC  
With ASCQ you can know what happen
  - o ASCQ
  - o Reasone  
The ASC/ASCQ in human form
-

- o FRUC
- o SKSV
- o SKS

See sense format.

Audio cd programming is not that easy. Different hardware devices and exec.library software devices may react in different mode.

Info about SCSI2 standard may be found at  
<http://www.uni-mainz.de/~tacke/scsi/SCSI2-index.html>

## 1.3 RxCD Author

Author

I am Alfonso Ranieri

My e-mail address is [alforan@tin.it](mailto:alforan@tin.it)

My home page is at <http://web.tiscalinet.it/amiga/>

## 1.4 RxCD Warning, Requirements, Installation and Distribution

Warning, Requirements, Installation and Distribution

Warning

THIS SOFTWARE AND INFORMATION ARE PROVIDED AS IS.  
ALL USE IS AT YOUR OWN RISK, AND NO LIABILITY OR  
RESPONSIBILITY IS ASSUMED. NO WARRANTY IS MADE,

Requirements

The library needs:

- o AmigaOS, version 3 or higher

Installation

- o Run the installer script
- o The library dispatch offset is -30

Distribution

RxCD is Freeware

You are free to distribute it as long as the original archive is kept intact. Commercial use or its inclusion in other software package is prohibited without prior written consent from the Author.

---

## 1.5 RxCD Terms

### Terms

The main terms used in this guide are:

- o id  
The device descriptor as returned by CDCreate()
- o cd  
An AmigaDOS device name or an id
- o cdName  
An AmigaDOS device name
- o track  
A sub-division of the CD-ROM media. A disc has from 1 to 99 tracks. The data within a track are always of the same type. A track can be either CD-ROM or CD-Audio. A disc can start at any track number.
- o index  
An index is a subdivision of a CD-ROM track. A track can have from 1 to 99 index numbers. Index numbers within a track are sequential starting with 1.
- o address  
Physical address of a frame on a cd
- o frame  
1/75 of a second. A physical sector on CD-ROM media. Also the F field unit of a MSF CD-ROM address
- o MSF  
Minutes/Seconds/Frames represents an address on a cd  
 $addr = m * 75 * 60 + S * 75 + F$
- o stem or stemName  
A valid ARexx variable name e.g. var var.0 var.name
- o types of arguments:
 

D	any data	--	
N	numeric	/N	ARexx integral number
S	symbol	/S	ARexx valid symbol
V	stemName	/V	As S but with length<32

## 1.6 RxCD Functions

### Functions

CD objects



```
    CDCreate
    CDDispose

Interrupts
    CDAddChangeInt
    CDRemChangeInt
    CDAddFrameInt
    CDRemFrameInt

Loading
    CDEject
    CDPause
    CDStartStop

Play
    CDPlay
    CDPlay10
    CDPlay12
    CDPlayMSF
    CDPlayTrackIndex
    CDPlaytrackRelative10
    CDPlaytrackRelative12

Info
    CDInfo
    CDModeSense
    CDStatus
    CDToc
    CDRequestSense

Various
    CDCddb4String
    CDTestUnit
    CDVolume

Help
    GetRxCDString
    Help
```

## 1.7 RxCD Functions - CDCreate

CDCreate - creates a cd object

### Synopsis

```
id = CDCreate(cdName)
<cdName>
```

### Function

Creates an internal cd object and returns its descriptor.

### Inputs

cdName - audio cd device name

### Result

id - an integer:

---

0 - failure, a RxCD error code in RC  
>0 - success, a valid cd descriptor

See

CDDispose()

## 1.8 RxCD Functions - CDDispose

CDDispose - disposes a cd object

Synopsis

```
call CDDispose(id)
<id/N>
```

Function

Disposes an internal cd object created with CDCreate().

Inputs

id - audio cd device descriptor

Result

Always returns 0.

See

CDCreate()

## 1.9 RxCD Functions - CDAddChangeInt

CDAddChangeInt - adds a change-disk interrupt

Synopsis

```
sig = CDAddChangeInt(id)
<id/N>
```

Function

Adds a change interrupt so that anytime 'id' is open/closed, your macro is signaled via 'sig'.

Warning: your cd exec.library device may refuse to add the interrupt.

Inputs

id - audio cd device descriptor

Result

res - 0 on failure with in RC the device Error,  
the signal to wait otherwise.

See

CDRemChangeInt()

## 1.10 RxCD Functions - CDRemChangeInt

CDRemChangeInt - removes a change-disk interrupt

### Synopsis

```
call = CDRemChangeInt(id)
<id/N>
```

### Function

Removes a change-disk interrupt added by CDAddChangeInt().

### Inputs

id - a device descriptor

### Result

res - always 0

### See

CDAddChangeInt()

## 1.11 RxCD Functions - CDAddFrameInt

CDAddFrameInt - adds a frame interrupt

### Synopsis

```
sig = CDAddFrameInt(id)
<id/N>
```

### Function

Adds a frame interrupt so that anytime 'id' is playing and the current frame changes your macro is signaled via 'sig'.

Warning: your cd exec.library device will refuse to add the interrupt for sure :-)  
It means mainly that to control the playing you have to poll like all cd players do.

### Inputs

id - audio cd device descriptor

### Result

res - 0 on failure with in RC the device Error,  
the signal to wait otherwise.

### See

CDRemFrameInt()

## 1.12 RxCD Functions - CDRemFrameInt

---

CDRemFrameInt - removes a change-disk interrupt

#### Synopsis

```
call = CDRemFrameInt(id)
<id/N>
```

#### Function

Removes a frame interrupt added by CDAddFrameInt().

#### Inputs

id - a device descriptor

#### Result

res - always 0

#### See

CDAddFrameInt()

## 1.13 RxCD Functions - CDEject

CDEject - ejects a cd

#### Synopsis

```
res = CDEject(cd,status)
<cd>, [status/N]
```

#### Function

Ejects or injects a cd.

Warning: your cd exec.library device may refuse to eject the cd. You should use CDStartStop() to perform this operation.

#### Inputs

cd - audio cd device name or descriptor  
status - 0 inject, 1 eject

#### Result

res - 0 on success a RxCD error code otherwise with in RC the device Error

#### See

CDPause() CDStartStop()

## 1.14 RxCD Functions - CDPause

CDPause - pause/continue a cd

#### Synopsis

```
res = CDPause(cd,debug,status)
<cd>, <debug/V>, [status/N]
```

#### Function

Pauses or continues a cd.

#### Inputs

cd - audio cd device name or descriptor  
debug - an ARexx stem name  
status - 0 pause, 1 continue

#### Result

res - 0 on success a RxCD error code otherwise

#### See

CDEject() CDStartStop()

## 1.15 RxCD Functions - CDStartStop

CDStartStop - starts/stops a cd

#### Synopsis

```
res = CDStartStop(cd,debug,l,s)
<cd>,<debug/V>,[l/N],[s/N]
```

#### Function

Ejects or injects a cd.

#### Inputs

cd - audio cd device name or descriptor  
debug - an ARexx stem name  
l s action  
-----  
0 0 stop  
0 1 ready  
1 1 load  
1 0 eject

#### Result

res - 0 on success a RxCD error code otherwise

#### See

CDEject() CDPause()

## 1.16 RxCD Functions - CDPlay

CDPlay - plays a cd by a track pair

#### Synopsis

```
res = CDPlay(cd,debug,fromTrack,toTrack)
<cd>,<debug/V>,[fromTrack/N],[toTrack/N]
```

#### Function

Plays a cd starting from 'fromTrack' track to 'toTrack' track

---

Default for 'fromTrack' is 1.

Default for 'toTrack' is 99 (MAX standard number of tracks).

#### Inputs

cd - audio cd device name or descriptor  
debug - an ARexx stem name  
fromTrack - the track to start at  
toTrack - the track to stop at

#### Result

res - 0 on success a RxCD error code otherwise

#### See

CDPlay10() CDPlay12() CDPlayMSF()  
CDPlayTrackIndex() CDPlaytrackRelative10() CDPlaytrackRelative12()

## 1.17 RxCD Functions - CDPlay10

CDPlay10 - plays a cd by address/len

#### Synopsis

res = CDPlay10(cd, debug, absAddr, shortLen)  
<cd>, <debug/V>, [absAddr/N], [shortLen/N]

#### Function

Plays a cd starting at long 'absAddr' address for short 'shortLen' frames.

#### Inputs

cd - audio cd device name or descriptor  
debug - an ARexx stem name  
absAddr - the address to start at  
shortLen - the short frames to play

#### Result

res - 0 on success a RxCD error code otherwise

#### See

CDPlay() CDPlay12() CDPlayMSF()  
CDPlayTrackIndex() CDPlaytrackRelative10() CDPlaytrackRelative12()

## 1.18 RxCD Functions - CDPlay12

CDPlay12 - plays a cd by address/len

#### Synopsis

res = CDPlay12(cd, debug, absAddr, longLen)  
<cd>, <debug/V>, [absAddr/N], [longLen/N]

#### Function

---

Plays a cd starting at long 'absAddr' address for long 'longLen' frames.

#### Inputs

cd - audio cd device name or descriptor  
 debug - an ARExx stem name  
 absAddr - the address to start at  
 longLen - the long frames to play

#### Result

res - 0 on success a RxCD error code otherwise

#### See

CDPlay() CDPlay10() CDPlayMSF()  
 CDPlayTrackIndex() CDPlaytrackRelative10() CDPlaytrackRelative12()

## 1.19 RxCD Functions - CDPlayMSF

CDPlayMSF - plays a cd by MSF pair

#### Synopsis

```
res = CDPlayMSF(cd, debug, startMin, startSec, startFrame, endMin, endSec, endFrame)
<cd>, <debug/V>, [startMin/N], [startSec/N], [startFrame/N], [endMin/N], [endSec/N] ←
    ], [endFrame/N]
```

#### Function

Plays a cd from 'startMin'-'startSec'-'startFrame' to 'endMin'-'endSec'-'endFrame'.

It means that the function plays the cd from the addr  
 startMin+60\*75+startSec\*75+startFrame to the addr  
 endMin+60\*75+endSec\*75+endFrame .

#### Inputs

cd - audio cd device name or descriptor  
 debug - an ARExx stem name  
 startMin - the minute to start at  
 startSec - the second to start at  
 startFrame - the frame to start at  
 endMin - the minute to stop at  
 endSec - the second to stop at  
 endFrame - the frame to stop at

#### Result

res - 0 on success a RxCD error code otherwise

#### See

CDPlay() CDPlay10() CDPlay12()  
 CDPlayTrackIndex() CDPlaytrackRelative10() CDPlaytrackRelative12()

## 1.20 RxCD Functions - CDPlayTrackIndex

CDPlayTrackIndex - plays a cd by track/index pair

#### Synopsis

```
res = CDPlayTrackIndex(cd,debug,startTrack,startIndex,endTrack,endIndex)
<cd>,<debug/V>,[startTrack/N],[startIndex/N],[endTrack/N],[endIndex/N]
```

#### Function

Plays a cd starting at track 'startTrack' index 'startIndex' to track 'endTrack' index 'endIndex'.

#### Inputs

cd - audio cd device name or descriptor  
debug - an ARexx stem name  
startTrack - the track to start at  
startIndex - the index (inside 'startTrack') to start at  
endTrack - the track to stop at  
endIndex - the index (inside 'endTrack') to stop at

#### Result

res - 0 on success a RxCD error code otherwise

#### See

CDPlay() CDPlay10() CDPlay12()  
CDPlayMSF() CDPlaytrackRelative10() CDPlaytrackRelative12()

## 1.21 RxCD Functions - CDPlayTrackRelative10

CDPlayTrackRelative10 - plays a cd by track/addr/len

#### Synopsis

```
res = CDPlayTrackRelative10(cd,debug,relAddr,track,shortLen)
<cd>,<debug/V>,[relAddr/N],[track/N],[shortLen/N]
```

#### Function

Plays a cd starting at address 'relAddr' relative to the track 'track' for short 'shortLen' frames.

#### Inputs

cd - audio cd device name or descriptor  
debug - an ARexx stem name  
relAddr - the address, relative to 'track', to start at  
track - the track 'relAddr' is relative to  
shortLen - the short frames to play

#### Result

res - 0 on success a RxCD error code otherwise

#### See

CDPlay() CDPlay10() CDPlay12()  
CDPlayMSF() CDPlayTrackIndex() CDPlaytrackRelative12()

---



## 1.22 RxCD Functions - CDPlayTrackRelative12

CDPlayTrackRelative12 - plays a cd by track/addr/len

### Synopsis

```
res = CDPlayTrackRelative11(cd, debug, relAddr, track, longLen)
<cd>, <debug/V>, [relAddr/N], [track/N], [len/N]
```

### Function

Plays a cd starting at address 'relAddr' relative to the track 'track' for long 'longLen' frames.

### Inputs

cd - audio cd device name or descriptor  
debug - an ARexx stem name  
relAddr - the address to start at, relative to 'track'  
track - the track 'relAddr' is relative to  
longLen - the long frames to play

### Result

res - 0 on success a RxCD error code otherwise

### See

CDPlay() CDPlay10() CDPlay12()  
CDPlayMSF() CDPlayTrackIndex() CDPlaytrackRelative10()

## 1.23 RxCD Functions - CDCDDB4String

CDCDDB4String - returns CDDB protocol-4 query string

### Synopsis

```
res = CDCDDB4String(cd)
<cd>, <debug/V>, <var/S>
```

### Function

Retrieves CDDB protocol-4 query string

### Inputs

cd - audio cd device name or descriptor  
debug - an ARexx stem name  
var - an ARexx var name where to store result

### Result

res - 0 on success a RxCD error code otherwise

### See

CDToc()

## 1.24 RxCD Functions - CDInfo

CDInfo - retrieves CD driver informations

---

## Synopsis

```
res = CDInfo(cd, debug, stem)
<cd>, <debug/V>, <stem/V>
```

## Function

Retrieves CD driver informations, writing in 'stem' the fields:

- o PeripheralQualifier
- o PeripheralType
- o Removable
- o Modifier
- o Iso
- o Ecma
- o ANSI
- o AENC
- o TrmIOP
- o Rdf
- o ALength
- o RelAddr
- o Wbus32
- o Wbus16
- o Sync
- o Linked
- o CmdQueue
- o SoftReset
- o Vendor
- o Product
- o Revision

### PeripheralQualifier:

- o 000b The specified peripheral device type is currently connected to this logical unit. If the target cannot determine whether or not a physical device is currently connected, it shall also use this peripheral qualifier when returning the INQUIRY data. This peripheral qualifier does not mean that the device is ready for access by the initiator.
- o 001b The target is capable of supporting the specified peripheral device type on this logical unit; however, the physical device is not currently connected to this logical unit.
- o 010b Reserved
- o 011b The target is not capable of supporting a physical device on this logical unit. For this peripheral qualifier the peripheral type shall be set to 1Fh to provide compatibility with versions of SCSI. All other peripheral device type are reserved for this peripheral qualifier.
- o 1XXb Vendor-specific

### PeripheralType:

- o 00h Direct-access device (e.g. magnetic disk)
- o 01h Sequential-access device (e.g. magnetic tape)
- o 02h Printer device
- o 03h Processor device
- o 04h Write-once device (e.g. some optical disks)
- o 05h CD-ROM device
- o 06h Scanner device
- o 07h Optical memory device (e.g. some optical disks)
- o 08h Medium changer device (e.g. jukeboxes)
- o 09h Communications device

- o 0Ah - 0Bh Defined by ASC IT8 (Graphic arts pre-press devices)
- o 0Ch - 1Eh Reserved
- o 1Fh Unknown or no device type

Of course, we are interested to 05h device type (humm maybe to 08h ?).

#### Inputs

cd - audio cd device name or descriptor  
debug - an ARexx stem name  
stem - where to write the fields

#### Result

res - 0 on success a RxCD error code otherwise

#### See

CDModeSense()

## 1.25 RxCD Functions - CDModeSense

CDModeSense - retrieves current media parameters

#### Synopsis

```
res = CDModeSense(cd, debug, stem)
<cd>, <debug/V>, <stem/V>
```

#### Function

Retrieves current media parameters, writing in stem the fields:

- o DataLength
- o MediumType
- o Empty
- o Opened
- o Specific
- o BDlength
- o PS
- o Immed
- o Sotc
- o APRVal
- o LBA
- o PBS
- o Out0
- o Vol0
- o Out1
- o Vol1
- o Out2
- o Vol2
- o Out3
- o Vol3

#### MediumType codes:

- o 00h Default (only one type supported)
  - o 01h 120 mm CD-ROM data only
  - o 02h 120 mm CD-DA audio only
  - o 03h 120 mm CD-ROM data and audio combined
  - o 04h Reserved
-

- o 05h            80 mm CD-ROM data only
- o 06h            80 mm CD-DA audio only
- o 07h            80 mm CD-ROM data and audio combined
- o 08h - 7Fh Reserved
- o 80h - FFh Vendor-specific

#### Inputs

cd     - audio cd device name or descriptor  
debug - an ARexx stem name  
stem   - where to write the fields

#### Result

res - 0 on success a RxCD error code otherwise

#### See

CDInfo()

## 1.26 RxCD Functions - CDStatus

CDStatus - retrieves the cd status

#### Synopsis

```
res = CDStatus(cd, debug, stem)
<cd>, <debug/V>, <stem/V>
```

#### Function

Retrieves the cd status, writing in stem the fields:

- o AudioStatus/N
- o Play
- o Pause
- o Complete
- o DLength
- o Adr
- o PreEmphasis
- o CopyPerm
- o Audio
- o 4Channels
- o Track
- o Index
- o AbsAddr
- o RelAddr

#### AudioStatus codes:

- o 00h - Audio status byte not supported or not valid
- o 11h - Audio play operation in progress
- o 12h - Audio play operation paused
- o 13h - Audio play operation successfully completed
- o 14h - Audio play operation stopped due to error
- o 15h - No current audio status to return

#### Inputs

cd     - audio cd device name or descriptor  
debug - an ARexx stem name  
stem   - where to write the fields

---

#### Result

res - 0 on success a RxCD error code otherwise

## 1.27 RxCD Functions - CDToc

CDToc - reads the table of contents

#### Synopsis

```
res = CDToc(cd, debug, stem)
<cd>, <debug/V>, <stem/V>
```

#### Function

Reads the table of contents, writing in stem the fields:

- o FirstTrack  
First track index. Note that this may not be 1.
- o LastTrack  
Last track index. Valid tracks are FirstTrack,...,LastTrack
- o StartAddr  
Where the first track starts
- o EndAddr  
Where the last track ends
- o Frame  
Size of the cd in frames
- o Min  
Size of the cd in minutes and seconds
- o Sec
- o ID  
The CDDDB proto4 ID of the disk
- o Num  
The number of the tracks

And int stem.i , i=0,...,NUM-1

- o Track  
Track number. Note that this may be not i+1
  - o Startaddr  
Address of the track
  - o Frames  
Size of the track in frames
  - o StartMin  
Where the track starts in MSF
  - o StartSec
-

- o Startframe
- o Min  
size of the track in MSF
- o Sec
- o Frame
- o Audio  
This track is an audio track
- o PreEmphasis
- o CopyPerm
- o 4Channels

#### Inputs

cd - audio cd device name or descriptor  
debug - an ARexx stem name  
stem - where to write the fields

#### Result

res - 0 on success a RxCD error code otherwise

#### See

CDCDDString()

## 1.28 RxCD Functions - CDRequestSense

CDRequestSense - returns sense data

#### Synopsis

```
res = CDRequestSense(cd, debug)
<cd>, <debug/V>
```

#### Function

Asks to the device for the sense data.

#### Inputs

id - a device descriptor

#### Result

sense - the sense data  
debug - an ARexx stem name

## 1.29 RxCD Functions - CDTestUnit

CDTestUnit - tests unit ready

---

#### Synopsis

```
res = CDTestUnit(cd, debug)
<cd>, <debug/V>
```

#### Function

Tests if the cd unit is ready.  
You should do something like:

```
cd = CDCreate("cd0")
call CDTestUnit(cd)
sense = CDSense(cd)
/* check the sense data here */
```

#### Inputs

cd - audio cd device name or descriptor  
debug - an ARExx stem name

#### Result

res - an ARExx boolean

## 1.30 RxCD Functions - CDVolume

CDVolume - changes audio volumes

#### Synopsis

```
res = CDVolume(cd, debug, leftVol, rightVol)
<cd>, <debug/V>, [leftVol/N], [rightVol/N]
```

#### Function

Changes audio volume. Note that not all cd hardware devices support the software volume control.

#### Inputs

cd - audio cd device name or descriptor  
debug - an ARExx stem name  
leftVol - left volume (0,...255)  
rightVol - right volume (0,...255)

#### Result

res - 0 on success a RxCD error code otherwise

## 1.31 RxCD Functions - GetRxCDString

GetRxCDString - returns a RxCD error string

#### Synopsis

```
string = GetRxCDString(code)
<code/N>
```

#### Function

Returns the error associated with code.

---

The error string defined are:

Code	Short	String
-----		
51	[NO MEM]	too few memory
52	[ARG MISS]	required argument missed
53	[BAD NUMBER]	bad number
54	[BAD VALUE]	bad value
55	[NO DEVICE]	device not found The device doesn't exist
56	[NO SIGNAL]	no signal available Exec signals are over
57	[CANT OPEN DEVICE]	can't open device The device cannot be opened
58	[DEVICE ERROR]	device error The device returned an error
59	[NO CD]	device is not a CD-ROM The device is not a CD-ROM
60	[BADTOC]	invalid TOC format Can't read this TOC

#### Inputs

code - the code to obtain the string associated with

#### Returns

string - the string associated with code

## 1.32 RxCD Functions - Help

Help - returns RxCD functions help strings

#### Synopsis

```
help=help(funName)
<funName>
```

#### Function

Returns the arguments mask string of rxcd.library  
function 'funName'.

#### Inputs

funName - a RxCD function name

#### Result

help - the help string associated with 'funName'



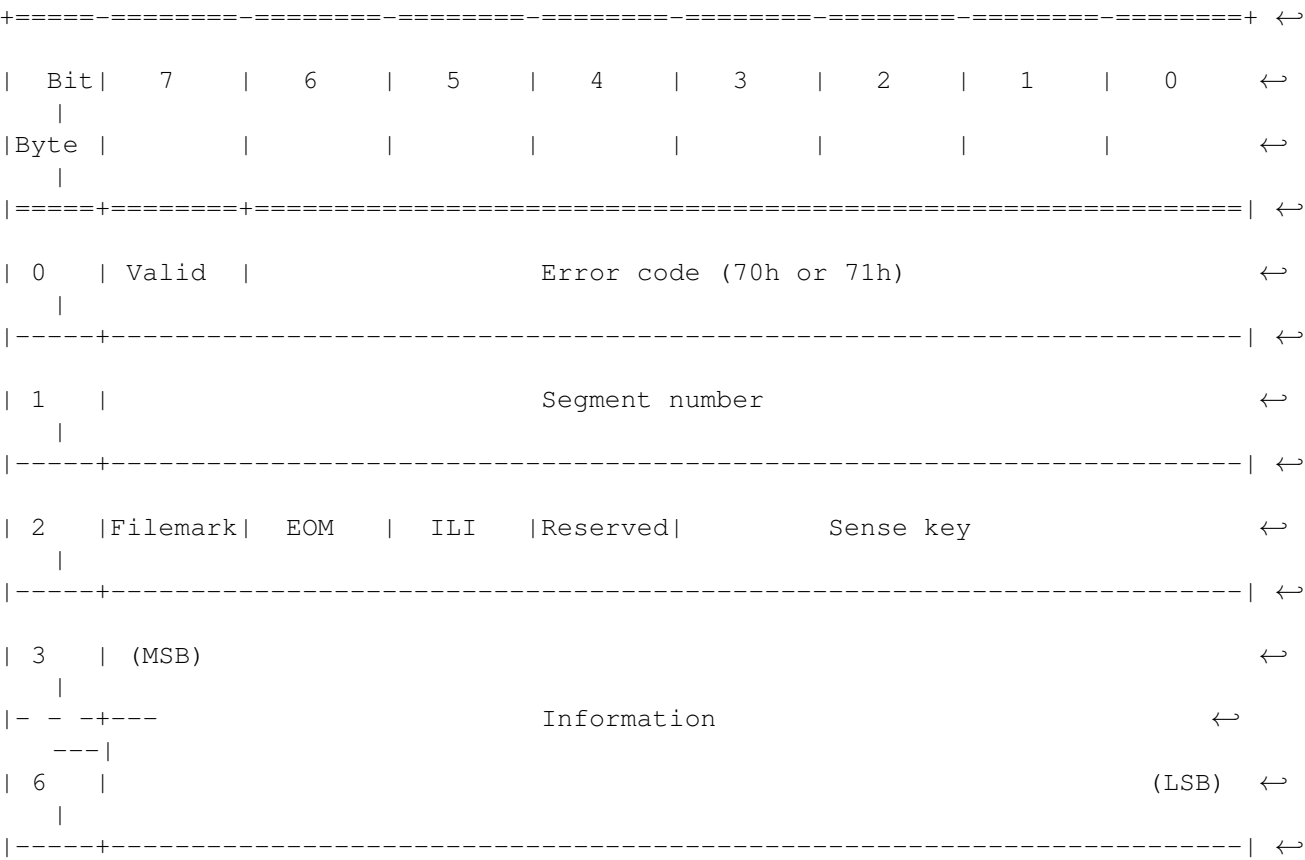
### 1.33 RxCD FAQ

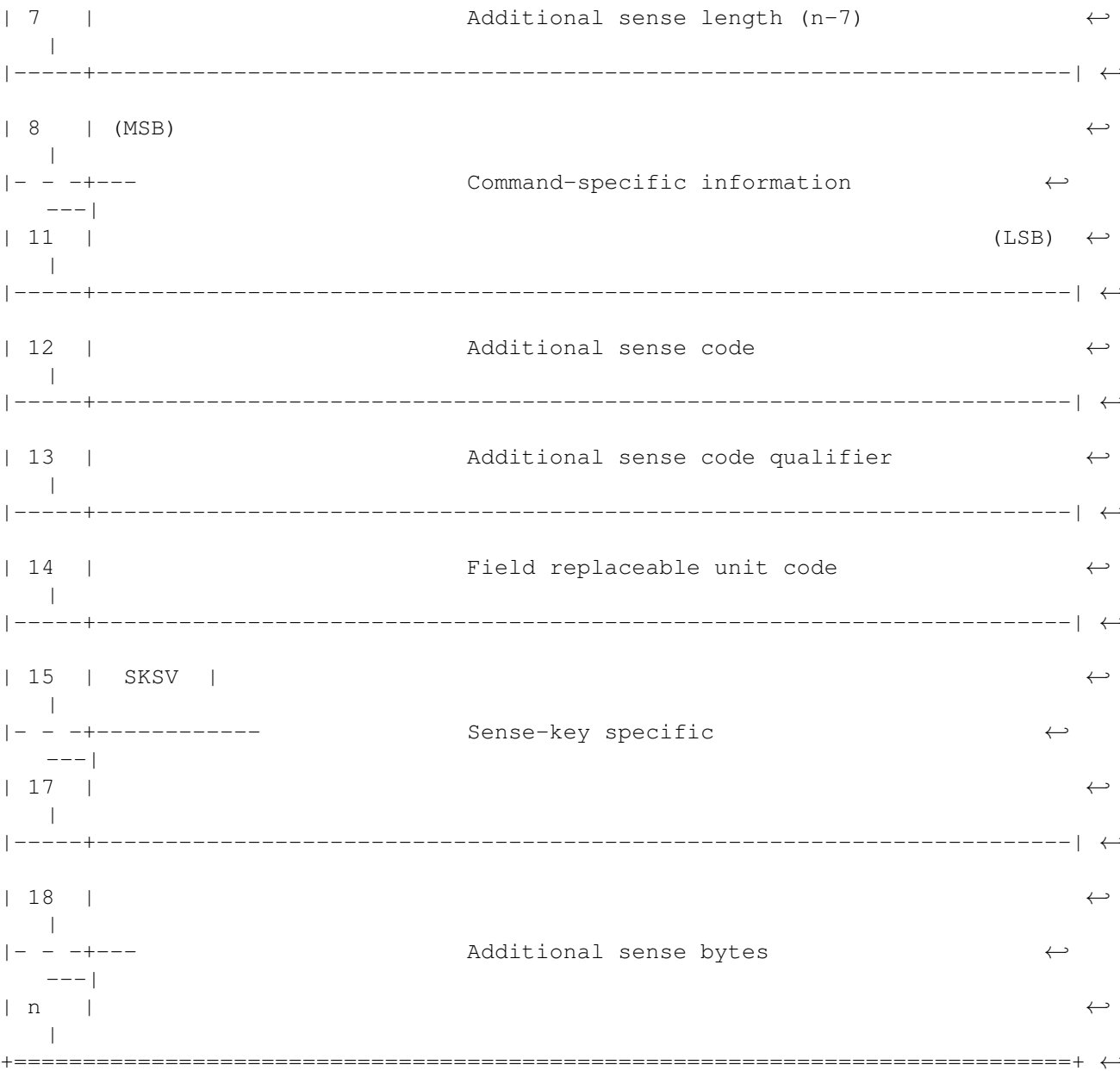
FAQ

1. How to check for a cd rom device:  
if CDInfo(cd,"db","info")==0 then iscd=(info.PeripheralType==5)  
else iscd=0
2. How to check for a anytype cd inserted:  
if CDMoDeSense(cd,"db","MS")==0 then iscdin=(and(a.MediumType,112)==0)  
else iscdin=0
3. How to check for an audio cd inserted:  
if CDMoDeSense(cd,"db","MS")==0 then isaudiocdin=(and(a.MediumType,2)~=0)  
else isaudiocdin=0
4. How to check for a data cd inserted:  
if CDMoDeSense(cd,"db","MS")==0 then isdatacdin=(and(a.MediumType,1)~=0)  
else isdatacdin=0

### 1.34 RxCD Sense

Sense





Sense key

+=====+			↔
Sense	Description		↔
key			↔
-----+-----			↔
0h	NO SENSE. Indicates that there is no specific sense key		↔
	information to be reported for the designated logical unit. This		↔
	would be the case for a successful command or a command that		↔

		received CHECK CONDITION or COMMAND TERMINATED status because one	↔
		of the filemark, EOM, or ILI bits is set to one.	↔
		-----	↔
	1h	RECOVERED ERROR. Indicates that the last command completed	↔
		successfully with some recovery action performed by the target.	↔
		Details may be determinable by examining the additional sense	↔
		bytes and the information field. When multiple recovered errors	↔
		occur during one command, the choice of which error to report	↔
		(first, last, most severe, etc.) is device specific.	↔
		-----	↔
	2h	NOT READY. Indicates that the logical unit addressed cannot be	↔
		accessed. Operator intervention may be required to correct this	↔
		condition.	↔
		-----	↔
	3h	MEDIUM ERROR. Indicates that the command terminated with a non-	↔
		recovered error condition that was probably caused by a flaw in	↔
		the medium or an error in the recorded data. This sense key may	↔
		also be returned if the target is unable to distinguish between a	↔
		flaw in the medium and a specific hardware failure (sense key 4h)	↔
		-----	↔
	4h	HARDWARE ERROR. Indicates that the target detected a non-	↔
		recoverable hardware failure (for example, controller failure,	↔
		device failure, parity error, etc.) while performing the command	↔
		or during a self test.	↔
		-----	↔
	5h	ILLEGAL REQUEST. Indicates that there was an illegal parameter in	↔
		the command descriptor block or in the additional parameters	↔
		supplied as data for some commands (FORMAT UNIT, SEARCH DATA,	↔
		-----	↔

		etc.). If the target detects an invalid parameter in the command	↔
		descriptor block, then it shall terminate the command without	↔
		altering the medium. If the target detects an invalid parameter	↔
		in the additional parameters supplied as data, then the target may	↔
		have already altered the medium. This sense key may also indicate	↔
		that an invalid IDENTIFY message was received (6.6.7).	↔
		-----+-----	↔
	6h	UNIT ATTENTION. Indicates that the removable medium may have been	↔
		changed or the target has been reset.	↔
		-----+-----	↔
	7h	DATA PROTECT. Indicates that a command that reads or writes the	↔
		medium was attempted on a block that is protected from this	↔
		operation. The read or write operation is not performed.	↔
		-----+-----	↔
	8h	BLANK CHECK. Indicates that a write-once device or a sequential-	↔
		access device encountered blank medium or format-defined end-of-	↔
		data indication while reading or a write-once device encountered a	↔
		non-blank medium while writing.	↔
		-----+-----	↔
	9h	VENDOR-SPECIFIC. This sense key is available for reporting vendor	↔
		specific conditions.	↔
		-----+-----	↔
	Ah	COPY ABORTED. Indicates a COPY, COMPARE, or COPY AND VERIFY	↔
		command was aborted due to an error condition on the source	↔
		device, the destination device, or both. (See 8.2.3.2 for	↔
		additional information about this sense key.)	↔
		-----+-----	↔
	Bh	ABORTED COMMAND. Indicates that the target aborted the command.	↔

			The initiator may be able to recover by trying the command again.	↵
			-----+-----	↵
	Ch		EQUAL. Indicates a SEARCH DATA command has satisfied an equal	↵
			comparison.	↵
			-----+-----	↵
	Dh		VOLUME OVERFLOW. Indicates that a buffered peripheral device has	↵
			reached the end-of-partition and data may remain in the buffer	↵
			that has not been written to the medium. A RECOVER BUFFERED DATA	↵
			command(s) may be issued to read the unwritten data from the	↵
			buffer.	↵
			-----+-----	↵
	Eh		MISCOMPARE. Indicates that the source data did not match the data	↵
			read from the medium.	↵
			-----+-----	↵
	Fh		RESERVED.	↵
			=====+	↵

#### ASC and ASCQ assignments

			=====+	↵
			D - DIRECT ACCESS DEVICE	↵
			.T - SEQUENTIAL ACCESS DEVICE	↵
			. L - PRINTER DEVICE	↵
			. P - PROCESSOR DEVICE	↵
			. .W - WRITE ONCE READ MULTIPLE DEVICE	↵
			. . R - READ ONLY (CD-ROM) DEVICE	↵
			. . S - SCANNER DEVICE	↵
			. . .O - OPTICAL MEMORY DEVICE	↵
			. . . M - MEDIA CHANGER DEVICE	↵
			. . . C - COMMUNICATION DEVICE	↵

				. . . .	←
	ASC	ASCQ	DTLPWRSOMC	DESCRIPTION	←
	---	---		-----	←
	00	00	DTLPWRSOMC	NO ADDITIONAL SENSE INFORMATION	←
	00	01	T	FILEMARK DETECTED	←
	00	02	T S	END-OF-PARTITION/MEDIUM DETECTED	←
	00	03	T	SETMARK DETECTED	←
	00	04	T S	BEGINNING-OF-PARTITION/MEDIUM DETECTED	←
	00	05	T S	END-OF-DATA DETECTED	←
	00	06	DTLPWRSOMC	I/O PROCESS TERMINATED	←
	00	11	R	AUDIO PLAY OPERATION IN PROGRESS	←
	00	12	R	AUDIO PLAY OPERATION PAUSED	←
	00	13	R	AUDIO PLAY OPERATION SUCCESSFULLY COMPLETED	←
	00	14	R	AUDIO PLAY OPERATION STOPPED DUE TO ERROR	←
	00	15	R	NO CURRENT AUDIO STATUS TO RETURN	←
	01	00	DW O	NO INDEX/SECTOR SIGNAL	←
	02	00	DWR OM	NO SEEK COMPLETE	←
	03	00	DTL W SO	PERIPHERAL DEVICE WRITE FAULT	←
	03	01	T	NO WRITE CURRENT	←
	03	02	T	EXCESSIVE WRITE ERRORS	←
	04	00	DTLPWRSOMC	LOGICAL UNIT NOT READY, CAUSE NOT REPORTABLE	←
	04	01	DTLPWRSOMC	LOGICAL UNIT IS IN PROCESS OF BECOMING READY	←
	04	02	DTLPWRSOMC	LOGICAL UNIT NOT READY, INITIALIZING command REQUIRED	←
	04	03	DTLPWRSOMC	LOGICAL UNIT NOT READY, MANUAL INTERVENTION REQUIRED	←
	04	04	DTL O	LOGICAL UNIT NOT READY, FORMAT IN PROGRESS	←
	05	00	DTL WRSOMC	LOGICAL UNIT DOES NOT RESPOND TO SELECTION	←
	06	00	DWR OM NO	REFERENCE POSITION FOUND	←
	07	00	DTL WRSOM	MULTIPLE PERIPHERAL DEVICES SELECTED	←

	08	00	DTL	WRSOMC	LOGICAL UNIT COMMUNICATION FAILURE	↵
	08	01	DTL	WRSOMC	LOGICAL UNIT COMMUNICATION TIME-OUT	↵
	08	02	DTL	WRSOMC	LOGICAL UNIT COMMUNICATION PARITY ERROR	↵
	09	00	DT	WR O	TRACK FOLLOWING ERROR	↵
	09	01		WR O	TRACKING SERVO FAILURE	↵
	09	02		WR O	FOCUS SERVO FAILURE	↵
	09	03		WR O	SPINDLE SERVO FAILURE	↵
	0A	00	DTLPWRSOMC		ERROR LOG OVERFLOW	↵
	0B	00				↵
	0C	00	T	S	WRITE ERROR	↵
	0C	01	D	W O	WRITE ERROR RECOVERED WITH AUTO REALLOCATION	↵
	0C	02	D	W O	WRITE ERROR - AUTO REALLOCATION FAILED	↵
	0D	00				↵
	0E	00				↵
	0F	00				↵
	10	00	D	W O	ID CRC OR ECC ERROR	↵
	11	00	DT	WRSO	UNRECOVERED READ ERROR	↵
	11	01	DT	W SO	READ RETRIES EXHAUSTED	↵
	11	02	DT	W SO	ERROR TOO LONG TO CORRECT	↵
	11	03	DT	W SO	MULTIPLE READ ERRORS	↵
	11	04	D	W O	UNRECOVERED READ ERROR - AUTO REALLOCATE FAILED	↵
	11	05		WR O	L-EC UNCORRECTABLE ERROR	↵
	11	06		WR O	CIRC UNRECOVERED ERROR	↵
	11	07		W O	DATA RESYNCHRONIZATION ERROR	↵
	11	08	T		INCOMPLETE BLOCK READ	↵
	11	09	T		NO GAP FOUND	↵
	11	0A	DT	O	MISCORRECTED ERROR	↵
	11	0B	D	W O	UNRECOVERED READ ERROR - RECOMMEND REASSIGNMENT	↵

	11	0C	D	W	O	UNRECOVERED READ ERROR - RECOMMEND REWRITE THE DATA	↵
	12	00	D	W	O	ADDRESS MARK NOT FOUND FOR ID FIELD	↵
	13	00	D	W	O	ADDRESS MARK NOT FOUND FOR DATA FIELD	↵
	14	00	DTL	WRSO		RECORDED ENTITY NOT FOUND	↵
	14	01	DT	WR	O	RECORD NOT FOUND	↵
	14	02	T			FILEMARK OR SETMARK NOT FOUND	↵
	14	03	T			END-OF-DATA NOT FOUND	↵
	14	04	T			BLOCK SEQUENCE ERROR	↵
	15	00	DTL	WRSOM		RANDOM POSITIONING ERROR	↵
	15	01	DTL	WRSOM		MECHANICAL POSITIONING ERROR	↵
	15	02	DT	WR	O	POSITIONING ERROR DETECTED BY READ OF MEDIUM	↵
	16	00	DW		O	DATA SYNCHRONIZATION MARK ERROR	↵
	17	00	DT	WRSO		RECOVERED DATA WITH NO ERROR CORRECTION APPLIED	↵
	17	01	DT	WRSO		RECOVERED DATA WITH RETRIES	↵
	17	02	DT	WR	O	RECOVERED DATA WITH POSITIVE HEAD OFFSET	↵
	17	03	DT	WR	O	RECOVERED DATA WITH NEGATIVE HEAD OFFSET	↵
	17	04		WR	O	RECOVERED DATA WITH RETRIES AND/OR CIRC APPLIED	↵
	17	05	D	WR	O	RECOVERED DATA USING PREVIOUS SECTOR ID	↵
	17	06	D	W	O	RECOVERED DATA WITHOUT ECC - DATA AUTO-REALLOCATED	↵
	17	07	D	W	O	RECOVERED DATA WITHOUT ECC - RECOMMEND REASSIGNMENT	↵
	17	08	D	W	O	RECOVERED DATA WITHOUT ECC - RECOMMEND REWRITE	↵
	18	00	DT	WR	O	RECOVERED DATA WITH ERROR CORRECTION APPLIED	↵
	18	01	D	WR	O	RECOVERED DATA WITH ERROR CORRECTION & RETRIES	↵
	18	02	D	WR	O	RECOVERED DATA - DATA AUTO-REALLOCATED	↵
	18	03		R		RECOVERED DATA WITH CIRC	↵
	18	04		R		RECOVERED DATA WITH LEC	↵
	18	05	D	WR	O	RECOVERED DATA - RECOMMEND REASSIGNMENT	↵
	18	06	D	WR	O	RECOVERED DATA - RECOMMEND REWRITE	↵



	19	00	D	O	DEFECT LIST ERROR	↵
	19	01	D	O	DEFECT LIST NOT AVAILABLE	↵
	19	02	D	O	DEFECT LIST ERROR IN PRIMARY LIST	↵
	19	03	D	O	DEFECT LIST ERROR IN GROWN LIST	↵
	1A	00	DTLPWRSOMC		PARAMETER LIST LENGTH ERROR	↵
	1B	00	DTLPWRSOMC		SYNCHRONOUS DATA TRANSFER ERROR	↵
	1C	00	D	O	DEFECT LIST NOT FOUND	↵
	1C	01	D	O	PRIMARY DEFECT LIST NOT FOUND	↵
	1C	02	D	O	GROWN DEFECT LIST NOT FOUND	↵
	1D	00	D	W O	MISCOMPARE DURING VERIFY OPERATION	↵
	1E	00	D	W O	RECOVERED ID WITH ECC	↵
	1F	00				↵
	20	00	DTLPWRSOMC		INVALID command OPERATION CODE	↵
	21	00	DT	WR OM	LOGICAL BLOCK ADDRESS OUT OF RANGE	↵
	21	01		M	INVALID ELEMENT ADDRESS	↵
	22	00	D		ILLEGAL FUNCTION (SHOULD USE 20 00, 24 00, OR 26 00)	↵
	23	00				↵
	24	00	DTLPWRSOMC		INVALID FIELD IN CDB	↵
	25	00	DTLPWRSOMC		LOGICAL UNIT NOT SUPPORTED	↵
	26	00	DTLPWRSOMC		INVALID FIELD IN PARAMETER LIST	↵
	26	01	DTLPWRSOMC		PARAMETER NOT SUPPORTED	↵
	26	02	DTLPWRSOMC		PARAMETER VALUE INVALID	↵
	26	03	DTLPWRSOMC		THRESHOLD PARAMETERS NOT SUPPORTED	↵
	27	00	DT	W O	WRITE PROTECTED	↵
	28	00	DTLPWRSOMC		NOT READY TO READY TRANSITION (MEDIUM MAY HAVE CHANGED)	↵
	28	01		M	IMPORT OR EXPORT ELEMENT ACCESSED	↵
	29	00	DTLPWRSOMC		POWER ON, RESET, OR BUS DEVICE RESET OCCURRED	↵
	2A	00	DTL WRSOMC		PARAMETERS CHANGED	↵

	2A	01	DTL WRSOMC	MODE PARAMETERS CHANGED	↵
	2A	02	DTL WRSOMC	LOG PARAMETERS CHANGED	↵
	2B	00	DTLPWRSO C	COPY CANNOT EXECUTE SINCE HOST CANNOT DISCONNECT	↵
	2C	00	DTLPWRSOMC	command SEQUENCE ERROR	↵
	2C	01	S	TOO MANY WINDOWS SPECIFIED	↵
	2C	02	S	INVALID COMBINATION OF WINDOWS SPECIFIED	↵
	2D	00	T	OVERWRITE ERROR ON UPDATE IN PLACE	↵
	2E	00			↵
	2F	00	DTLPWRSOMC	commands CLEARED BY ANOTHER INITIATOR	↵
	30	00	DT WR OM	INCOMPATIBLE MEDIUM INSTALLED	↵
	30	01	DT WR O	CANNOT READ MEDIUM - UNKNOWN FORMAT	↵
	30	02	DT WR O	CANNOT READ MEDIUM - INCOMPATIBLE FORMAT	↵
	30	03	DT	CLEANING CARTRIDGE INSTALLED	↵
	31	00	DT W O	MEDIUM FORMAT CORRUPTED	↵
	31	01	D L O	FORMAT command FAILED	↵
	32	00	D W O	NO DEFECT SPARE LOCATION AVAILABLE	↵
	32	01	D W O	DEFECT LIST UPDATE FAILURE	↵
	33	00	T	TAPE LENGTH ERROR	↵
	34	00			↵
	35	00			↵
	36	00	L	RIBBON, INK, OR TONER FAILURE	↵
	37	00	DTL WRSOMC	ROUNDED PARAMETER	↵
	38	00			↵
	39	00	DTL WRSOMC	SAVING PARAMETERS NOT SUPPORTED	↵
	3A	00	DTL WRSOM	MEDIUM NOT PRESENT	↵
	3B	00	TL	SEQUENTIAL POSITIONING ERROR	↵
	3B	01	T	TAPE POSITION ERROR AT BEGINNING-OF-MEDIUM	↵
	3B	02	T	TAPE POSITION ERROR AT END-OF-MEDIUM	↵

	3B	03	L	TAPE OR ELECTRONIC VERTICAL FORMS UNIT NOT READY	↵
	3B	04	L	SLEW FAILURE	↵
	3B	05	L	PAPER JAM	↵
	3B	06	L	FAILED TO SENSE TOP-OF-FORM	↵
	3B	07	L	FAILED TO SENSE BOTTOM-OF-FORM	↵
	3B	08	T	REPOSITION ERROR	↵
	3B	09	S	READ PAST END OF MEDIUM	↵
	3B	0A	S	READ PAST BEGINNING OF MEDIUM	↵
	3B	0B	S	POSITION PAST END OF MEDIUM	↵
	3B	0C	S	POSITION PAST BEGINNING OF MEDIUM	↵
	3B	0D	M	MEDIUM DESTINATION ELEMENT FULL	↵
	3B	0E	M	MEDIUM SOURCE ELEMENT EMPTY	↵
	3C	00			↵
	3D	00	DTLPWRSOMC	INVALID BITS IN IDENTIFY MESSAGE	↵
	3E	00	DTLPWRSOMC	LOGICAL UNIT HAS NOT SELF-CONFIGURED YET	↵
	3F	00	DTLPWRSOMC	TARGET OPERATING CONDITIONS HAVE CHANGED	↵
	3F	01	DTLPWRSOMC	MICROCODE HAS BEEN CHANGED	↵
	3F	02	DTLPWRSOMC	CHANGED OPERATING DEFINITION	↵
	3F	03	DTLPWRSOMC	INQUIRY DATA HAS CHANGED	↵
	40	00	D	RAM FAILURE (SHOULD USE 40 NN)	↵
	40	NN	DTLPWRSOMC	DIAGNOSTIC FAILURE ON COMPONENT NN (80H-FFH)	↵
	41	00	D	DATA PATH FAILURE (SHOULD USE 40 NN)	↵
	42	00	D	POWER-ON OR SELF-TEST FAILURE (SHOULD USE 40 NN)	↵
	43	00	DTLPWRSOMC	MESSAGE ERROR	↵
	44	00	DTLPWRSOMC	INTERNAL TARGET FAILURE	↵
	45	00	DTLPWRSOMC	SELECT OR RESELECT FAILURE	↵
	46	00	DTLPWRSOMC	UNSUCCESSFUL SOFT RESET	↵
	47	00	DTLPWRSOMC	SCSI PARITY ERROR	↵

	48	00	DTLPWRSOMC	INITIATOR DETECTED ERROR MESSAGE RECEIVED	↵
	49	00	DTLPWRSOMC	INVALID MESSAGE ERROR	↵
	4A	00	DTLPWRSOMC	command PHASE ERROR	↵
	4B	00	DTLPWRSOMC	DATA PHASE ERROR	↵
	4C	00	DTLPWRSOMC	LOGICAL UNIT FAILED SELF-CONFIGURATION	↵
	4D	00			↵
	4E	00	DTLPWRSOMC	OVERLAPPED COMMANDS ATTEMPTED	↵
	4F	00			↵
	50	00	T	WRITE APPEND ERROR	↵
	50	01	T	WRITE APPEND POSITION ERROR	↵
	50	02	T	POSITION ERROR RELATED TO TIMING	↵
	51	00	T O	ERASE FAILURE	↵
	52	00	T	CARTRIDGE FAULT	↵
	53	00	DTL WRSOM	MEDIA LOAD OR EJECT FAILED	↵
	53	01	T	UNLOAD TAPE FAILURE	↵
	53	02	DT WR OM	MEDIUM REMOVAL PREVENTED	↵
	54	00	P	SCSI TO HOST SYSTEM INTERFACE FAILURE	↵
	55	00	P	SYSTEM RESOURCE FAILURE	↵
	56	00			↵
	57	00	R	UNABLE TO RECOVER TABLE-OF-CONTENTS	↵
	58	00	O	GENERATION DOES NOT EXIST	↵
	59	00	O	UPDATED BLOCK READ	↵
	5A	00	DTLPWRSOM	OPERATOR REQUEST OR STATE CHANGE INPUT (UNSPECIFIED)	↵
	5A	01	DT WR OM	OPERATOR MEDIUM REMOVAL REQUEST	↵
	5A	02	DT W O	OPERATOR SELECTED WRITE PROTECT	↵
	5A	03	DT W O	OPERATOR SELECTED WRITE PERMIT	↵
	5B	00	DTLPWRSOM	LOG EXCEPTION	↵
	5B	01	DTLPWRSOM	THRESHOLD CONDITION MET	↵

	5B	02	DTLPWRSOM	LOG COUNTER AT MAXIMUM	↵
	5B	03	DTLPWRSOM	LOG LIST CODES EXHAUSTED	↵
	5C	00	D O	RPL STATUS CHANGE	↵
	5C	01	D O	SPINDLES SYNCHRONIZED	↵
	5C	02	D O	SPINDLES NOT SYNCHRONIZED	↵
	5D	00			↵
	5E	00			↵
	5F	00			↵
	60	00	S	LAMP FAILURE	↵
	61	00	S	VIDEO ACQUISITION ERROR	↵
	61	01	S	UNABLE TO ACQUIRE VIDEO	↵
	61	02	S	OUT OF FOCUS	↵
	62	00	S	SCAN HEAD POSITIONING ERROR	↵
	63	00	R	END OF USER AREA ENCOUNTERED ON THIS TRACK	↵
	64	00	R	ILLEGAL MODE FOR THIS TRACK	↵
	65	00			↵
	66	00			↵
	67	00			↵
	68	00			↵
	69	00			↵
	6A	00			↵
	6B	00			↵
	6C	00			↵
	6D	00			↵
	6E	00			↵
	6F	00			↵
	70	00			↵
	71	00			↵

	72	00		↵
	73	00		↵
	74	00		↵
	75	00		↵
	76	00		↵
	77	00		↵
	78	00		↵
	79	00		↵
	7A	00		↵
	7B	00		↵
	7C	00		↵
	7D	00		↵
	7E	00		↵
	7F	00		↵
				↵
	80	xxh \		↵
	THROUGH >	Vendor-specific.		↵
	FF	xxh /		↵
				↵
	xxh 80 \			↵
	THROUGH >	Vendor-specific QUALIFICATION OF STANDARD ASC.		↵
	xxh FF /			↵
		ALL CODES NOT SHOWN OR BLANK ARE RESERVED.		↵
+=====+ ↵				