

# Voyager CDLink (VCD) Control Language Reference

*For Voyager CDLink version 1.0b1*

*subject to continual revision; last modified 8 August 1995*

## **General:**

The Voyager CDLink software is designed to play Voyager CDLink control files (VCDs). A VCD is a text file containing commands like play, pause, delay, or loop, followed often by track or time figure(s). The vocabulary and grammar of VCDs comprises the VCD control language, developed by Voyager for CDLink. This document describes the VCD control language, and is annotated with HTML links to examples. A CDLink Web page is simply one that contains links to VCDs.

## **Working with VCDs**

You can open and study the VCDs linked to this document in any text editor, edit them to suit your purposes, and create your own CDLink application within minutes, be it a Web page, a folder full of custom mixes for your figure-skating routines, or part of a more elaborate media project. While a third-party [authoring tool for Mac OS](#) is available, and Voyager is currently developing CDLog, a cross-platform VCD authoring tool, no special tools are needed to get started, and many may prefer to make VCDs "by hand" in any case, especially until the concepts become clear.

Sound intimidating? Web pages for the first 25 CDs enhanced with CDLink were prepared in just a few days by a handful of liberal-arts graduates using only text editors (BBEdit, SimpleText) and the AppleCD Audio Player desk accessory. NotePad and Media Player are common Windows equivalents of these Mac OS tools.

A VCD should be named with the extension ".vcd". This identifies the file properly to Windows computers and to Web servers. In the Mac OS, VCDs must have the creator attribute "VGCD" in order to be double-click testable (otherwise, you can just drag-and-drop a plain text file onto CDLink Kernel). Several excellent shareware utilities are available for setting Mac OS file attributes. BBEEdit can even save files directly with the VGCD (CDLink Kernel) creator attribute, allowing you to test VCDs while their windows remain open for editing. Because BBEEdit is also a powerful HTML formatting tool, we recommend it highly. Easiest of all is to use an existing VCD (downloaded from this page) as a template--drag a VCD onto any text editor and edit away, while double-clicking the file to test as you go! You can use a word processor to make and edit VCDs, but if you do, remember to save the files as plain (ascii) text--not the default format.

## **Serving VCDs on the Web**

Web servers must be updated to map VCDs' new MIME type of application/x-cdlink to the .vcd extension before they can serve CDLink files properly. Do this by having your server's system administrator edit the mime.types file in the CONF subdirectory of the browser daemon directory. Add the following line to the file:

```
application/x-cdlink vcd
```

The server may need to be restarted to register the change.

## **The VCD Control Language**

This section uses the following notation:

CAPITALIZED WORDS - A command or literal

Data in <>'s - Required parameters

Data in {}'s - Optional parameters

tmsf - Track, Minute, Second, Frame (1/75th-second)

The "example usage" texts linked below represent the complete and exact contents of VCDs. Clicking on a link will send your browser a "live" VCD containing the linked text.

**PLAY {t,m,s,f}{,t,m,s,f}**

Example Usage:

[\[COMMANDS\]](#)

[PLAY](#) -- plays the CD from its current position, clearing any previously set stop markers

[\[COMMANDS\]](#)

[PLAY 03,01,00,00](#) -- plays from track 3, minute 1, second and frame 0 to end

[\[COMMANDS\]](#)

[PLAY 03,01,00,00,03,03,20,00](#) -- plays a 2-minute, 20-second passage beginning at track 3, minute 1, second and frame 0

**Command Description:**

- 1) Issuing PLAY with no parameters starts the CD playing until the end of the CD is reached. Any previous stop markers are cleared.
- 2) Issuing PLAY with just a start tmsf searches the CD to the supplied position and begins playing the CD. The CD stops when the end of the CD is reached. Any previous stop markers are cleared.
- 3) Issuing PLAY with both start tmsf and end tmsf values searches the CD to the supplied start position, sets a stop marker at the supplied end position, and begins playing the CD.
- 4) There is no way to issue a PLAY command with just an end tmsf figure.

**PLAYTRACK <track>{,track}**

Example Usage:

[\[COMMANDS\]](#)

[PLAYTRACK 02](#) -- plays from the beginning of track 2 to the end of the CD.

[\[COMMANDS\]](#)

[PLAYTRACK 02,03](#) -- plays from the beginning of track 2 to the end of track 3

[\[COMMANDS\]](#)

[PLAYTRACK 03,03](#) -- plays track 3 only

**Command Description:**

- 1) Issuing PLAYTRACK with just a start track searches the CD to the supplied position and begins playing the CD. The CD stops when the end of the CD is reached.
- 2) Issuing PLAYTRACK with both start track and end track times searches the CD to the supplied start position, sets a stop marker at the supplied end position and begins playing the CD.
- 3) There is no way to issue a PLAYTRACK command with just an end track

**LOOP {t,m,s,f}{,t,m,s,f}**

Example Usage:

[\[COMMANDS\]](#)

[LOOP 03,02,30,40,03,02,39,21](#)

**Command Description:**

- 1) Issuing LOOP with no parameters clears any loop state of CDLink.
- 2) Issuing LOOP with both start tmsf and end tmsf parameters searches the CD to the start tmsf and plays the CD to the end tmsf. Then CDLink loops the CD back to the beginning.
- 3) Issuing LOOP with only one tmsf is the same as command (2) except that CDLink uses the current time on the CD as the beginning of the loop.

**STOP**

Example Usage:

[\[COMMANDS\]](#)  
[STOP](#)

**Command Description:**

Stops the CD, clears the current stop marker (if any) and sets the current time to the beginning of the CD.

**PAUSE**

Example Usage:

[\[COMMANDS\]](#)  
[PAUSE](#)

**Command Description:**

Pauses the CD. No other action is taken.

**PLAYPAUSE**

Example Usage:

[\[COMMANDS\]](#)  
[PLAYPAUSE](#)

**Command Description:**

PLAYPAUSE toggles between play and pause states. PLAYPAUSE takes no parameters. Unlike using PAUSE and PLAY independently, PLAYPAUSE does not clear any previously set stop marker.

**WAIT**

Example Usage:

[\[COMMANDS\]](#)  
[PLAY 01,00,00,00,01,02,00,00](#)  
[WAIT -- Wait until the CD reaches track 1, 2 minutes, 0 seconds, 0 frames](#)  
[LOOP 01,00,00,00,01,00,10,00 -- ten second loops](#)  
[DELAY 1800 -- delay 30 seconds](#)  
[WAIT -- Wait until at the end of the loop](#)  
[STOP -- and stop](#)

**Command Description:**

WAIT takes no parameters. It waits until the last stop marker set is reached. Generally, WAIT causes CDLink to finish executing one command line before starting another.

**DELAY <ticks>**

Example Usage:

[\[COMMANDS\]](#).

[PLAYPAUSE](#)

[DELAY 240 -- Delay 4 seconds](#)

[PLAYPAUSE](#)

**Command Description:**

DELAY takes one argument, a value in ticks, or sixtieths of a second. DELAY causes CDLink to delay n ticks before executing the next command in the VCD. Note that this is different than WAIT, which waits until the CD finishes executing the previous line. NOTE ALSO THAT THE "DELAY" ARGUMENT MAY CHANGE TO FRAMES (75ths of a second) IN A SUBSEQUENT RELEASE OF THE CDLINK SOFTWARE.

**EJECT**

Example Usage:

[\[COMMANDS\]](#).

[EJECT](#)

**Command Description:**

EJECT stops the CD (if any) and ejects the CD tray or caddy from the CD-ROM drive.

**COMMENTS**

Example Usage:

[-- This is a comment](#)

[\[COMMANDS\]](#).

[PLAY 03,02,20,30,03,02,24,35 --this is where the bells come in](#)

[WAIT -- let those bells finish](#)

[PLAY 04,00,20,71 -- and here come the whistles](#)

**Description:**

Not commands, comments are parsed out of the command file. A comment begins with the '--' character sequence and continues to the end of the line.