

master Script Syntax

This chapter provides a description of how master interprets script files to generate disc images.

The master script uses the following syntax to build disc images:

```
[options [:] option...]  
album_definition  
{volume_definition  
file_definition...  
[psd_definition]  
directory_definition}...
```

The master script uses the following syntax to build realtime files:

```
[options [:] {option}...]  
{record_definition}...
```

The master script elements are defined on the following pages.



The syntax limited to vcdmaster is the -vcd and -vcd2 options and the following definitions:

```
volume_options  
psd_list  
karinfo_data  
avseq_dat file
```

All other syntax is allowed for both master and vcdmaster.

```

option  :=  -a=char_file
          |  -c
          |  -d
          |  -e
          |  -g
          |  -l[=leadout]
          |  -m[=map_file]
          |  -nt
          |  -o[=PMM_file]
          |  -p
          |  -q

          |
          |  -r
          |  -s
          |  -u
          |  -vcd
          |  -vcd2
          |  -vn=volnum
          |  -xa
          |  -z

char_file := pathlist
leadout  := decimal_number
map_file := pathlist
PMM_file := pathlist
toc_file := pathlist
volnum   := integer

album_definition := define album album_name [label]
                  | red album
album_name       := string
label            := publisher string preparer string
                  | publisher string
                  | preparer string

```

```

volume_definition := volume volume_name in pathlist [volume_options]
    volume_name := string
    volume_options := [disc_restrict {0|1|2|3}] [strt_nxt_dsc_l1d1]
                        [strt_nxt_dsc_trk2]

file_definition := green file ident from pathlist
    | red file ident from [swapped] source
                        [with pre_emphasis]
    | yellow file ident [rounded] from source
    | application file ident [rounded] from pathlist

    | biblio file ident [rounded] from pathlist
    | copyright file ident [rounded] from pathlist
    | abstract file ident [rounded] from pathlist
    | application_use from pathlist
    | message from [swapped] source
    | seg_play_item file ident from segment_type source
                        autopause {at {lsn | expr} [every {lsn | expr}}...
    | avseq_dat file ident from mpeg pathlist
                        [scan_list file ident [spaced seconds]] [verbose]]
                        {[chapter_marks [first_source_tc SMPTE]
                        {"string" {source_tc | mpeg_tc} SMPTE}...}
                        | [ep_list from pathlist]}
    | karinfo_data from pathlist
    | green file ident from record_definition
                        [no plm_maps]

segment_type := wb_still
    | wb_still_seq
    | wb_a
    | wb_v
    | wb_av

SMPTE := hours:minutes:seconds:frames
    | hours, minutes, seconds, frames

```

```

record_definition := block [rtl[, rtl...]] stream_definition...
                  | record [rtl[, rtl...]] stream_definition...

stream_definition := mpeg_str
                  | rt_audio_str
                  | audio_str
                  | video_str
                  | data_str
                  | event_str

mpeg_str := mpeg_type mpeg [rtl[, rtl...]] [at rsn] in channel
           mpeg_channel from mpeg_source[, mpeg_source...]

mpeg_type := non_real_time
           | real_time
           | non_realtime still
           | real_time still

mpeg_channel := 0..15

mpeg_source := [rtl:...] mpeg_asset [{at | by} rsn]

mpeg_asset := pathlist

rt_audio_str := real_time audio [rtl[, rtl...]] [at rsn]
              in channel audio_channel
              from audio_source[, audio_source...]

audio_str := [non_real_time] audio [packed] [rtl[, rtl...]] [at rsn]
            in channel audio_channel
            from audio_source[, audio_source...]

audio_channel := 0..15

audio_source := [rtl:...] audio_asset [mask pattern] [offset offset
              length length] [{at | by} rsn]
              | [at rsn] silence sn[type[emphasis]] [{at | by} rsn]

audio_asset := pathlist [audio_type[emphasis]]
              | pathlist>chunkpath

```

```

audio_type := mono_a
              | mono_b
              | mono_c
              | stereo_a
              | stereo_b
              | stereo_c

pattern    := 16bit_number
offset     := number_of_bytes
length     := number_of_bytes


sn         := number_of_sectors
              | hours:minutes:seconds


video_str  := [{non_real_time | form1 non_realtime}]
               video [packed] [rtl[, rtl...]] [at rsn]
               in channel data_channel
               from video_source[, video_source...]

video_source := [rtl:...]video_asset [mask pattern] [{at | by} rsn]
video_asset  := pathlist[video_mask]
              | pathlist chunkpath
video_mask   := video_type [resolution] [scan_lines]
video_type   := clut4
              | clut7
              | clut8
              | rl3
              | rl7
              | dyuv
              | rgb555l
              | rgb555u
              | rgb555
              | qhy
              | specific mask

```

```

    mask := decimal_number
           | hex_number
resolution := normal
           | high
           | double
scan_lines := even
           | odd

data_str := data [packed] [rtl[, rtl...]] [at rsn]
           in channel data_channel
           from data_source[, data_source...]
data_channel := 0..31
data_source := [rtl:...] data_asset [mask pattern] [offset offset
           length length] [{at | by} rsn]
data_asset := pathlist
           | pathlist>chunkpath

event_str := {triggers | eors} at event[, event...]
event := [ident.first | ident.last | rsn] [every sn]

psd_definition := psd...
psd := sel_list ident play_item ident
      [repeat]
      [default]
      [next {ident [hot_spot] | off}]
      [previous ident [hot_spot]]
      [return ident [hotspot]]
      {selection number ident [hot_spot]}...

```

```

|   play_list ident
    [next {ident [hot_spot] | off}]
    [previous ident [hot_spot]]
    [return ident [hotspot]]
    [playing_time {all | 1/15second_units}]
    [after_play_wait {lsn | forever | none}]
    [auto_pause_wait {lsn | forever | none}]
    {play_item ident}...

```

```

repeat := repeat forever [finish]
| repeat number [finish] then
    [wait {lsn | forever | none}]
    {random_sel | play_item ident}

```

```

default := default off
| default ident [hotspot]
| default multi_default [hotspot] [no_numeric_sel]

```

```

hotspot := x1, y1, x2, y2

```

```

x1 := 0..255

```

```

x2 := 0..255

```

```

y1 := 0..255

```

```

y2 := 0..255

```

```

directory_def := {[file_entry...] [dir_name {file_entry...}]}

```

```

file_entry := file_name [attribute...] from ident

```

```

file_name := string

```

```

dir_name := string

```

attribute := *owner group.user*
 | *protection permissions*
 | *hidden*
 | *pad*
 | *extra number*
 | *intervleave number:number*

source := *pathlist*
 | *pathlist>chunkpath*
pathlist := “[string/...]string”

chunkpath := [*>wrapper...*][*>chunk*]
wrapper := {*FORM* | *LIST* | *CAT*} #*chunk*
chunk := *IFFID*[#*index*]
index := *decimal_number*
hours := *decimal_number*
minutes := *decimal_number*
sectors := *decimal_number*
frames := *decimal_number*
1/15second_units := *decimal_number*

rtl := *ident*
 | *ident(expr)*
expr := *hex_number*
 | *decimal_number*
 | (*expr*)
 | *expr + expr*
 | *expr - expr*
 | *expr / expr*
 | *expr * expr*