

Chaptering with Entrypoints

Michael Tam <mtam@vcdimager.org>

(c) 17 January, 2002

Introduction

Beyond the creation of menus [simple](#) or [otherwise](#), one of the most functional aspects to be authored onto a VCD is that of chapters. And with "**chapters**", I refer to the chapters like on a DVD. That is, one chapter seamlessly flows onto the next with NO pause or skip or blemish of any kind.

Now, the chapters that can be on a VCD are not quite like that of a DVD, but the results can be still quite pleasing. To author this feature, the system works by using one large MPEG file as a [<sequence-item>](#) with **entrypoints**. What the entrypoints allow is to begin playback of a [<sequence-item>](#) from somewhere other than the beginning of the track.

There will not be a CUE/BIN image of an example of chapters. There is already an example of this in the PAL/NTSC Demo VCD. This can be downloaded here: <http://www.vcdimager.org/discs.phtml>

About MPEG Sequence Headers

Now, for **entrypoints** to be able to be set (using the [<entry>](#) tag under the element of [<sequence-item>](#)), the MPEG source file must either have **regular MPEG sequence headers**, or these headers at the specific frames you want for chapters. You can consider the sequence headers as sort of "information way points". They give information about the timing of the MPEG, the playback specifications of the MPEG (e.g., framesize), etc. Without these, random entry (and thus chapters) are not possible. Although the only MPEG sequence headers required are those at the entrypoints, there is no harm in putting regular ones in (i.e., an MPEG sequence header before each GOP). There is negligible change in file size. Furthermore, some stand-alone players (e.g., Philips DVD player) require regular MPEG sequence headers for the FFW and REW functions to operate during playback.

Now, all this would have no significance if all MPEG-1 files are naturally encoded with MPEG sequence headers. Unfortunately, this is not the case. Methods to overcome this deficiency will be discussed when using the two best and most popular MPEG-1 encoders for VCD.

The Panasonic MPEG Encoder

This is an excellent (in terms of video and audio quality) MPEG-1 encoder at VCD bitrates. However, it does have several problems. Firstly, it does not insert regular MPEG sequence headers. Secondly, there is an error with the way the video and audio is multiplexed (can cause A/V synchronisation problems on some stand-alone players). Both issues will have to

be addressed before an MPEG encoded with this encoder can be used in a "chaptered" VCD).

As it doesn't add regular MPEG sequence headers, these will have to be inserted using another tool afterwards. Firstly, we need to demultiplex the MPEG into the video and audio components. This can be accomplished with a number of freeware tools, including TMPGEnc, bbTools and so forth.

The tool used to add MPEG sequence headers to the elemental video stream is known as "MPEG Sequence Maker v0.2". It can be downloaded from VCDHelp.com here: <http://www.vcdhelp.com/MPEGSequenceMaker02.zip>

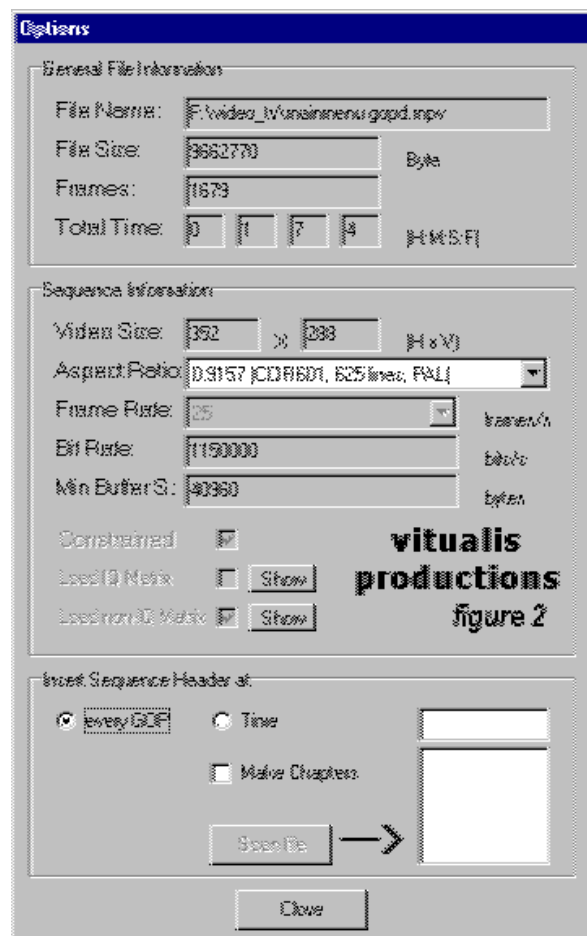
You will need a program like WinZip to decompress it.

Load up your elemental video stream (figure 1) and set and output file name.

Click on "Options" and figure 2 will pop up. All the settings should be left as they are. Make sure that the bottom setting of "Insert Sequence Header at" on set to "every GOP".

Click on "Start"

The new video and audio streams will need to be remultiplexed together. Both bbMPEG and TMPGEnc can do this too. With TMPGEnc, be sure to have the setting on "MPEG-1 Video-CD" rather than "Auto".

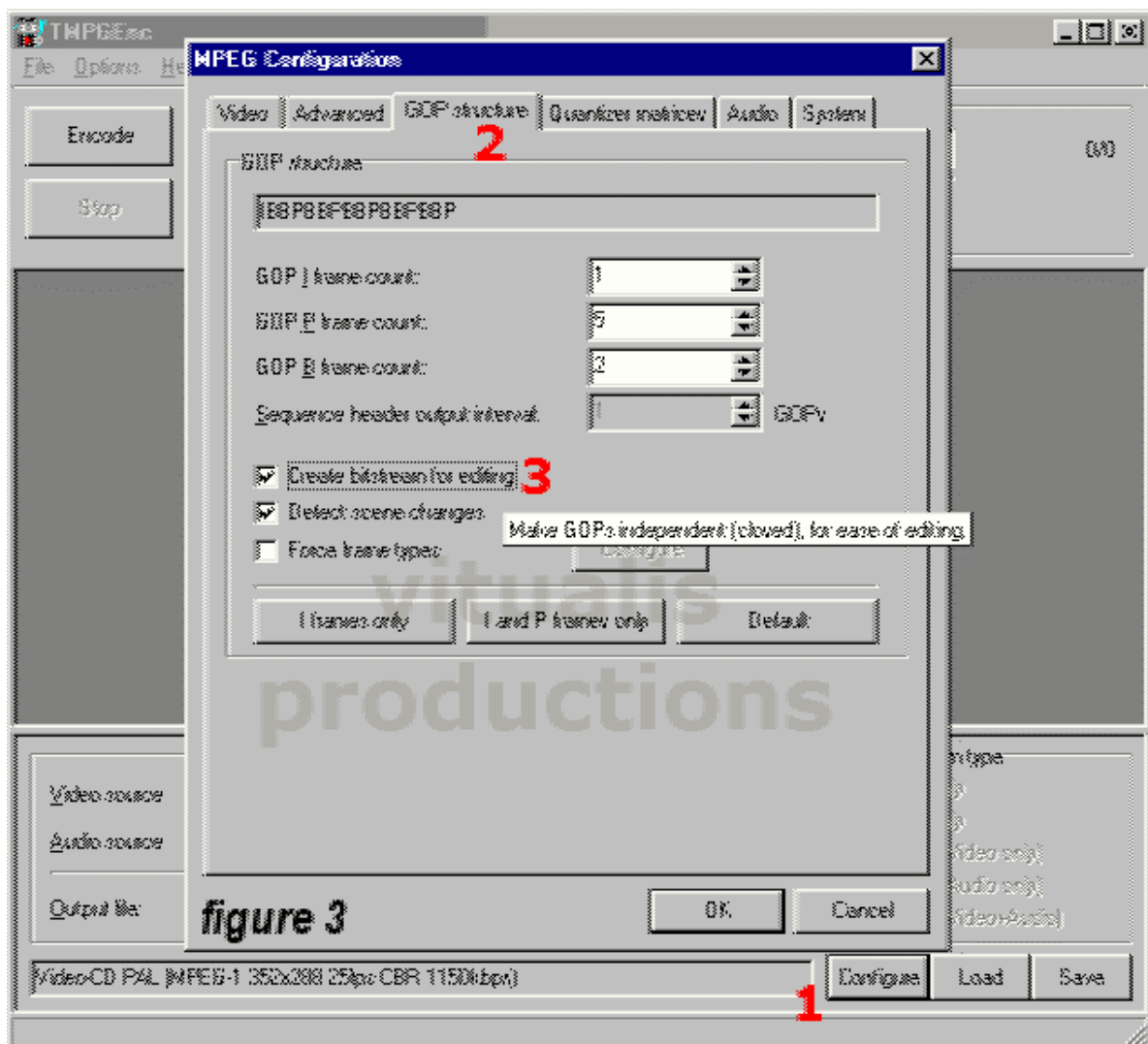


TMPGEnc (Tsunami MPEG Encoder)

Another excellent MPEG encoder all round. This one does insert regular MPEG sequence headers, but there is a setting should probably be changed from default in the template:

1. If encoding a VCD, make sure that either the PAL VCD or NTSC VCD template is loaded. Then click on **"Configure"**
2. Click on **GOP structure**
3. Now, click and tick the option **"Create bitstream for editing"**

This will ensure that there are regular closed GOPs and thus regular MPEG sequence headers (as you can see in figure 3, TMPGEnc automatically puts in 1 sequence header per GOP).



If this setting isn't ticked, TMPGEnc can sometimes encode the MPEG in such a way that the GOP isn't closed for an extended interval (> 2 seconds). During that interval, no MPEG sequence header will be present and thus there may be an issue with entrypoint (and thus chapter) placement.

Authoring the chapters

Both the theory and logic behind the authoring and the actual authoring will be discussed. If you wish to jump ahead to the authoring, [click here](#).

Logic and Theory behind the structure

Now this is a diagram (figure 4) of an idealised perfect chaptering system:



figure 4

Pressing "previous" and "next" go to the previous and next chapter points while playback is seamless throughout. This is achievable on a DVD but not quite on a VCD. However, this is our goal.

Why not just use a simple multitrack?

Good question. Those who read the article on the simple multitrack may recall this diagram from that page:

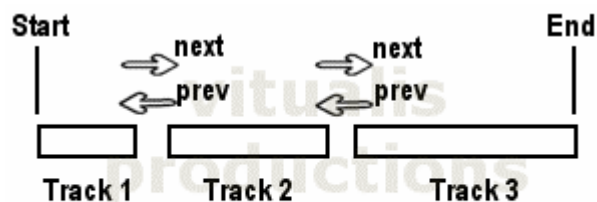


figure 1

Now this would be a good way except for one very important point... **You cannot get seamless playback between tracks.** It is as simple as that. From track 1 playing to track 2 in that diagram, there will almost always be a pause, a blip, at least a noticeable blemish of some kind. Some DVD players may be able to play seamless multitracks if authored a [particular way](#) but definitely not the majority.

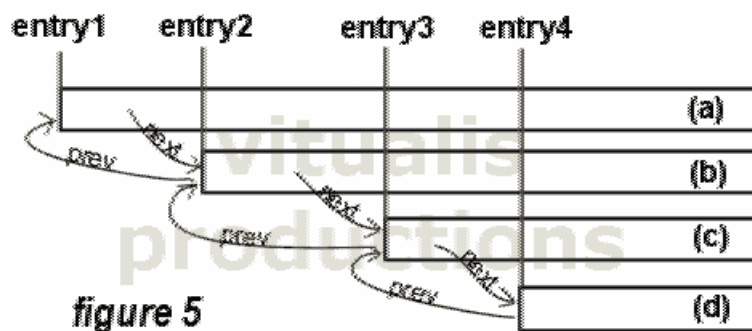
So how can it be authored?

As the title reads, the secret is all in using entrypoints... There will be some degree of assumed knowledge in the following. I suggest that you also read [this](#) before hand:

Now say I have an MPEG file called **track1.mpg** that I want to have chapters that start at 0s, 60s, 120s and 180s. Then, I would author the XML as such:

```
<sequence-items>
  <sequence-item src="track1.mpg" id="seq-track1">
    <entry id="entry1">0</entry>
    <entry id="entry2">60</entry>
    <entry id="entry3">120</entry>
    <entry id="entry4">180</entry>
  </sequence-item>
</sequence-items>
```

Now, if **entry1-4** is used as the **<play-item>** in either **<selection>** or **<playlist>**, then the subsequent **<selection>** or **<playlist>** will start at 0, 60, 120 and 180s of the actual track1.mpg file. Now, if I link up these **<selection>** or **<playlist>** in a **"multi-track" (method 3)** then I would get the following:

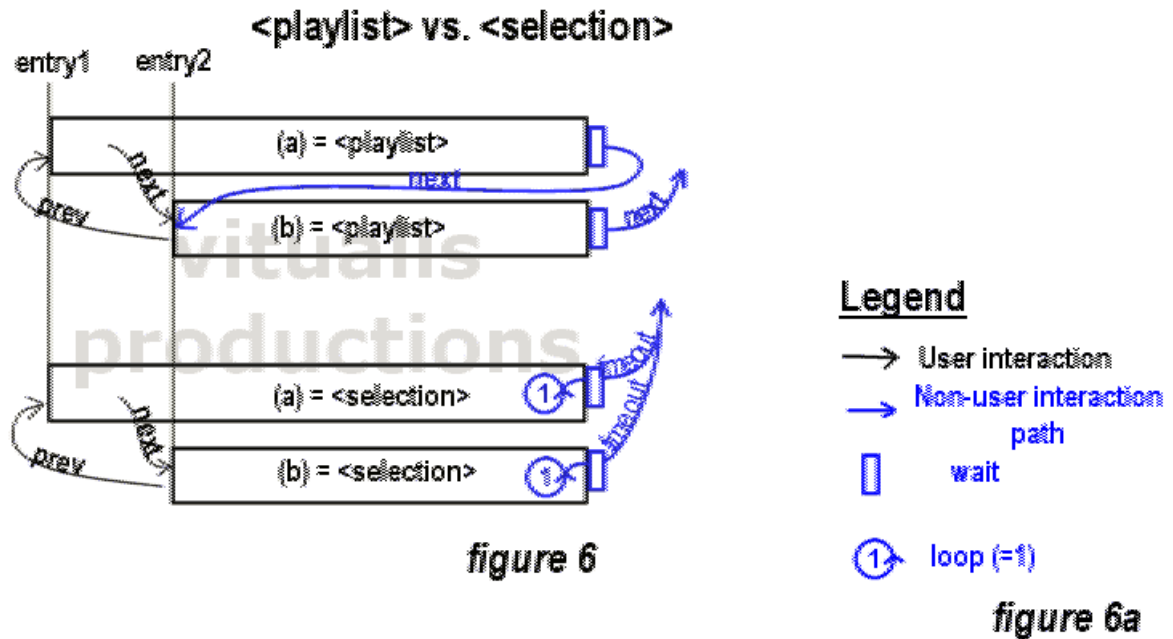


(a)-(d): each of these are either a **<selection>** or **<playlist>**

- (a)** has the **<play-item>** of **"entry1"**
- (b)** has the **<play-item>** of **"entry2"**
- (c)** has the **<play-item>** of **"entry3"**
- (d)** has the **<play-item>** of **"entry4"**

As can be seen, each **<selection>** or **<playlist>** that uses entry1-4 as the **<play-item>** will play the entire remainder of "track1.mpg" seamlessly. That is, (a) will play from entry1 (0s) all the way to the end, (b) will play from entry2 (60s) all the way to the end, and so forth.

So should I use <selection> or <playlist>?



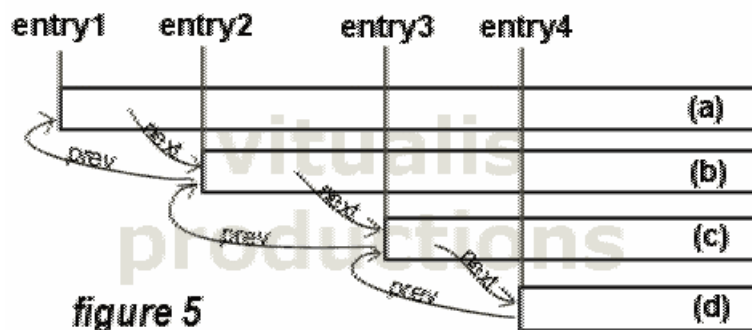
The answer to this question is undeniably **<selection>**.

Look at figure 5 and 6 and consider what would happen if you decided to watch the movie all the way through. That is, the <play-item> in (a) has finished playing. In a **<playlist>** the VCD will automatically go to what is defined in <next>... and that is (b)!!! That means, after (a) has finished playing, it will go back to (b), the **second chapter** and so on. This is quite a ridiculous outcome and completely unsatisfactory.

Now, with a **<selection>**, when the <play-item> in (a) has finished playing, **it does not automatically go to <next>**. This is a fundamental difference between <playlist> and <selection>. Rather, it automatically goes to **<timeout>**. Thus, if (a)-(d) are <selection>, then it can be authored that the VCD will do something else when the end has been reached (e.g., go to a "next disc" clip or go to the main menu or an <endlist>). This is much more acceptable.

In what way are VCD chapters not like the perfect idealised (DVD-like/figure 4) chapters?

If **figure 5** is considered again, the answer is apparent:



Consider the playback of **(a)** (remembering that (a) is either a <playlist> or <selection> with the <play-item> or "entry1").

Now, if "next" is pressed BEFORE reaching the time point of "entry2" then all is well. However, what happens if "next" is pressed while the playback is between "entry2" and "entry3"? It will go to <next> and that links to (b)! That is, in terms of function, we pressed "next" and went to a point in the video we have already watched!

Indeed, pressing "next" any time while in (a) will go to (b) and thus beginning playback at "entry2".

Unfortunately, this particular aspect of VCD chaptering is its limitation. There is no **real** way to get around this. One just has to accept that the next and prev buttons on the remote are not quite context specific...

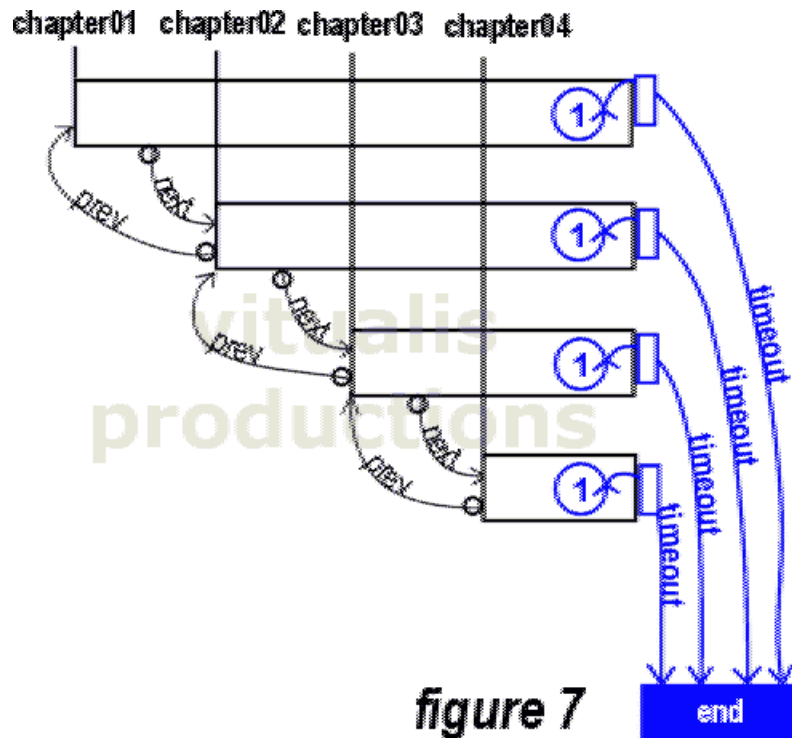
However, we can augment the underlying chaptering system by creating "**scene select**" menus to help cover to some degree this particular limitation.

Authoring the XML example

There will be some degree of assumed knowledge in the following. I suggest reading [this](#) before hand, concentrating on how `<entry>` works under [<sequence-item>](#) and how [<selection>](#) works.

Example 1

In this example just the raw chapting system is created.



The source files:

- track1.mpg (our main video clip -- entrypoints at 0s, 60s, 120s and 180s)
- CD-i application ([click here](#) for more information)

In chapters are created as `<selection>`. The reasoning behind this is further up this page [here](#):

- The loop value is set to "1" (i.e., the `<play-item>` plays once)
- The wait value is set to "1" (i.e., wait for 1 second before `<timeout>`)
- The chapters timeout to an `<endlist>`

The individual `<selection>` 's forming the chapters are chained together in a way similar to a [simple multitrack \(method 3\)](#)

```
<?xml version="1.0"?>
<!DOCTYPE videocd PUBLIC "-//GNU//DTD VideoCD//EN"
"http://www.gnu.org/software/vcdimager/videocd.dtd">
```

```
<videocd xmlns="http://www.gnu.org/software/vcdimager/1.0/" class="vcd"
version="2.0">
```

```
<info>
  <album-id>CHAPTERTEST</album-id>
  <volume-count>1</volume-count>
  <volume-number>1</volume-number>
  <restriction>0</restriction>
</info>

<pvd>
  <volume-id>CHAPTERTEST</volume-id>
  <system-id>CD-RTOS CD-BRIDGE</system-id>
  <application-id>CDI/CDI_VCD.APP;1</application-id>
  <preparer-id/>
  <publisher-id>MICHAEL TAM - VITUALIS PRODUCTIONS</publisher-id>
</pvd>

<filesystem>
  <folder>
    <name>CDI</name>
    <file src="cdi_imag.rtf" format="mixed">
      <name>CDI_IMAG.RTF</name>
    </file>
    <file src="cdi_text.fnt">
      <name>CDI_TEXT.FNT</name>
    </file>
    <file src="cdi_vcd.app">
      <name>CDI_VCD.APP</name>
    </file>
  </folder>
</filesystem>

<sequence-items>
  <sequence-item src="track1.mpg" id="seq-track1">
    <entry id="chapter01">0</entry>
    <entry id="chapter02">60</entry>
    <entry id="chapter03">120</entry>
    <entry id="chapter04">180</entry>
  </sequence-item>
</sequence-items>

<pbcc>

  <selection id="select-chapter01">
    <next ref="select-chapter02"/>
    <timeout ref="end"/>
    <wait>1</wait>
    <loop jump-timing="immediate">1</loop>
    <play-item ref="chapter01"/>
  </selection>

  <selection id="select-chapter02">
    <prev ref="select-chapter01"/>
    <next ref="select-chapter03"/>
    <timeout ref="end"/>
    <wait>1</wait>
    <loop jump-timing="immediate">1</loop>
    <play-item ref="chapter02"/>
  </selection>

  <selection id="select-chapter03">
    <prev ref="select-chapter02"/>
    <next ref="select-chapter04"/>
    <timeout ref="end"/>
```

```
<wait>1</wait>
<loop jump-timing="immediate">1</loop>
<play-item ref="chapter03"/>
</selection>

<selection id="select-chapter04">
  <prev ref="select-chapter03"/>
  <timeout ref="end"/>
  <wait>1</wait>
  <loop jump-timing="immediate">1</loop>
  <play-item ref="chapter04"/>
</selection>

<endlist id="end" rejected="true"/>

</pbc>
</videocd>
```

Example 2

In this example a scene select menu has been added. That is, from this menu, chapters can be chosen. Furthermore, by pressing "return" on the the remote control while in the video clip, the VCD goes back to scene select menu.

The source files:

- track1.mpg (our main video clip -- entrypoints at 0s, 60s, 120s and 180s)
- ss.mpg (high resolution still image of the "scene select" menu)
- CD-i application ([click here](#) for more information)

In chapters are created as <selection>. The reasoning behind this is further up this page [here](#).

- The loop value is set to "1" (i.e., the <play-item> plays once)
- The wait value is set to "1" (i.e., wait for 1 second before <timeout>)
- The chapters timeout back to the "scene select" menu
- Pressing "return" while in the chapters also goes back to the "scene select" menu

The "scene select" menu:

- loops once and then pauses indefinitely (still image)

pressing # 1-4 goes to chapters 1-4

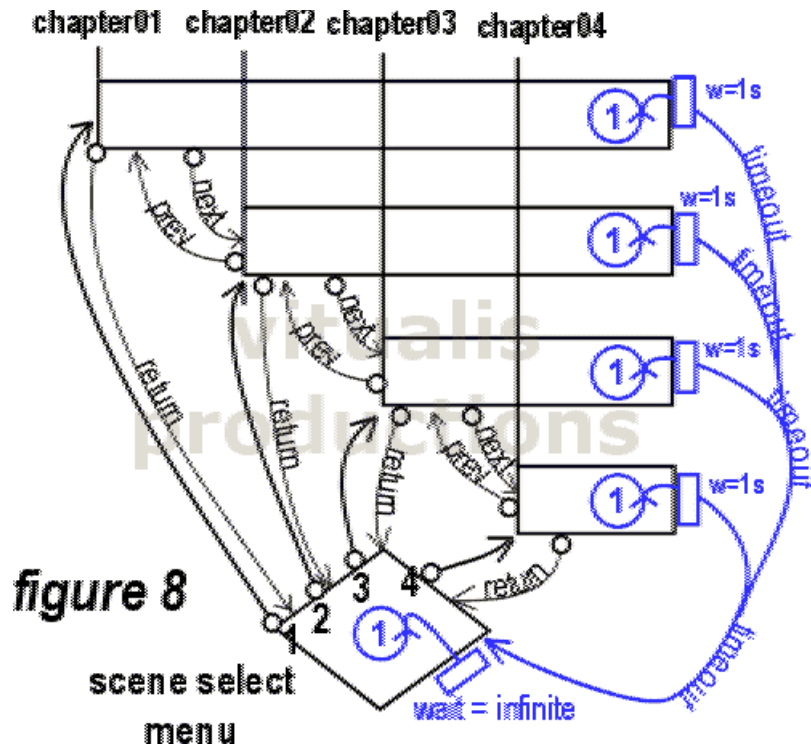


figure 8

```

<?xml version="1.0"?>
<!DOCTYPE videocd PUBLIC "-//GNU//DTD VideoCD//EN"
"http://www.gnu.org/software/vcdimager/videocd.dtd">

<videocd xmlns="http://www.gnu.org/software/vcdimager/1.0/" class="vcd"
version="2.0">

  <info>
    <album-id>CHAPTERTEST</album-id>
    <volume-count>1</volume-count>
    <volume-number>1</volume-number>
    <restriction>0</restriction>
  </info>

  <pvd>
    <volume-id>CHAPTERTEST</volume-id>
    <system-id>CD-RTOS CD-BRIDGE</system-id>
    <application-id>CDI/CDI_VCD.APP;1</application-id>
    <preparer-id/>
    <publisher-id>MICHAEL TAM - VITUALIS PRODUCTIONS</publisher-id>
  </pvd>

  <filesystem>
    <folder>
      <name>CDI</name>
      <file src="cdi_imag.rtf" format="mixed">
        <name>CDI_IMAG.RTF</name>
      </file>
      <file src="cdi_text.fnt">
        <name>CDI_TEXT.FNT</name>
      </file>
      <file src="cdi_vcd.app">
        <name>CDI_VCD.APP</name>
      </file>
    </folder>
  </filesystem>

```

```
<segment-items>
  <segment-item src="ss.mpg" id="seg-ss"/>
</segment-items>

<sequence-items>
  <sequence-item src="track1.mpg" id="seq-track1">
    <entry id="chapter01">0</entry>
    <entry id="chapter02">60</entry>
    <entry id="chapter03">120</entry>
    <entry id="chapter04">180</entry>
  </sequence-item>
</sequence-items>

<pbcs>

  <selection id="select-ss">
    <bsn>1</bsn>
    <wait>-1</wait>
    <loop jump-timing="immediate">1</loop>
    <play-item ref="seg-ss"/>
    <select ref="select-chapter01"/>
    <select ref="select-chapter02"/>
    <select ref="select-chapter03"/>
    <select ref="select-chapter04"/>
  </selection>

  <selection id="select-chapter01">
    <next ref="select-chapter02"/>
    <return ref="select-ss"/>
    <timeout ref="select-ss"/>
    <wait>1</wait>
    <loop jump-timing="immediate">1</loop>
    <play-item ref="chapter01"/>
  </selection>

  <selection id="select-chapter02">
    <prev ref="select-chapter01"/>
    <next ref="select-chapter03"/>
    <return ref="select-ss"/>
    <timeout ref="select-ss"/>
    <wait>1</wait>
    <loop jump-timing="immediate">1</loop>
    <play-item ref="chapter02"/>
  </selection>

  <selection id="select-chapter03">
    <prev ref="select-chapter02"/>
    <next ref="select-chapter04"/>
    <return ref="select-ss"/>
    <timeout ref="select-ss"/>
    <wait>1</wait>
    <loop jump-timing="immediate">1</loop>
    <play-item ref="chapter03"/>
  </selection>

  <selection id="select-chapter04">
    <prev ref="select-chapter03"/>
    <return ref="select-ss"/>
    <timeout ref="select-ss"/>
    <wait>1</wait>
    <loop jump-timing="immediate">1</loop>
    <play-item ref="chapter04"/>
  </selection>
```

```
</selection>  
  
</pbc>  
</videocd>
```

Conclusion

Once this process of chaptering is understood and mastered, it will seem rather obvious. This is because it is not a difficult method, once there is comprehension of how a VCD works. Although it may seem somewhat arbitrary at first, you'd realise in time that this is simply the logical conclusion of what the various interactive components on a VCD allows.

Obviously, this method of chaptering shouldn't be confined simply to itself. I encourage the reader to combine the information in this article with the methods in the other guides, especially the [multi-menu VCD](#). In doing so, very complex VCD structures can be authored indeed.

Good luck.

Michael Tam <mtam@vcdimager.org>

(c) 17 January, 2002