Making your

VDOLive Content

available on the

World Wide Web

Documentation Version 2.3

December 26, 1996

VDOnet Corporation

4009 Miranda Avenue
Suite 250
Palo Alto, CA   94304

Tel: (415) 846-7700
Fax: (415) 846-7900

http://www.vdo.net

Trademarks

VDOLive, Lite VDO, VDOLive Trial Version, VDOLive Video Server, VDOLive Presentation Server, VDOLive Tools, VDO Capture, VDO Clip and VDOWave are trademarks of VDOnet Corporation Ltd.

Other product and company names appearing in VDOnet's products and materials are used for identification purposes only and may be trademarks or registered trademarks of their respective owners. Registered and unregistered trademarks used in any VDOnet products and materials are the exclusive property of their respective owners.

Table of Contents

Overview

This manual explains how to connect your users to your content; it links the information provided in the VDOLive Server and the VDOLive Tools manuals.

Your audience views your content--video clips or broadcasts--from the World Wide Web. They reach your site by using a browser and activate the video by clicking on text or graphic links provided on your site.

The complete procedure for linking your audience with your video content includes these steps:

1. Setting up your VDOLive Server.
2. Creating files of compressed video using VDOLive encoding either from the VDOLive Tools, or using another editing tool together with VDOLive compression. These files are in AVI file format.
3. Creating a VDO file for each clip or broadcast.
4. Creating an HTML link to each VDO file.
5. Configuring the MIME type of the VDO file types/s used depending on the type of video you are serving. This is done on your HTTP (Web) Server.

- The exception to the procedure above, and to the information in the sections following, is when you are serving audiences who are prevented from receiving video by a **firewall**. For more information about reaching this audience, please see the section beginning on page 21. Users who are either behind a packet filtering firewall, or a proxy firewall, may not require special treatment.
- Administrators of a **packet filtering firewall** can open a port for incoming and outgoing VDOLive data. Refer the system or firewall administrator to the VDOLive Server manual, or to VDOnet Corp. for information and guidelines.
- Some **proxy firewalls** support VDOLive video streaming; if this type of firewall is in use, your VDOLive video content can reach these users.

The following diagram illustrates the process by which users access your content (when not prevented by any firewall). The example uses a video clip, but is equally valid for a VDOLive broadcast.

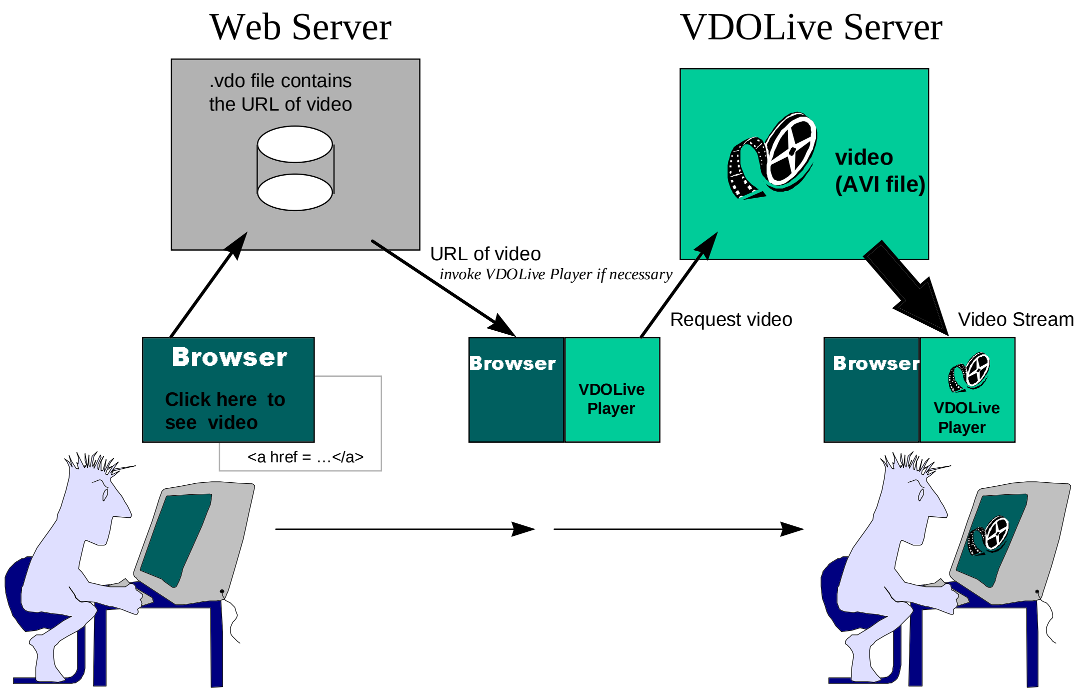

Figure 1 - How users access and view your video

About VDO Files

Characteristics of the VDO File

A VDO file provides a transparent link for your users from your web site, which is on your HTTP Server, to the VDOLive Server actually serving your video clips or broadcasts. (See also Figure 1 on page 8)

The VDO file:

- is a text file.
- contains the URL of your video content: either a VDOLive clip (compressed file) or a VDOLive broadcast.
- uses the suffix ***.vdo***
- is stored on an HTTP (Web) server. This can be the same HTTP Server on which your web page is stored, or a remote HTTP Server.

Creating VDO Files

Create VDO files using any text editor, such as Notepad.

➢ To create a VDO file:

1. Open a text editor.
2. Enter the URL of the clip or broadcast.
3. Save the file with the suffix ***.vdo*** and store it on an HTTP server.

Syntax of the URL in VDO files

A **URL** (**U**niform **R**esource **L**ocator) is a standard way of identifying the precise location of information on a computer network. On the World Wide Web, each graphic or text hyperlink displayed on a web page has HTML code behind it; this code includes a URL specifying the destination or "target" of the link.

Each VDO file indicates the location of a specific AVI file (containing VDOLive compressed data) or broadcast channel on a VDOLive Server. This locator information is what enables the user, who is browsing a web site, to reach the video, which is coming from a VDOLive Server. This location information is organized and presented in the VDO file as follows:

```
protocol://Server address[:Server TCP port]/Path
```

where:

- **Protocol**: the protocol for VDOLive video is vdo://
  The colon and the forward slashes must be included. Backslashes (\ ) do not work.

- **Server address**: identification of the VDOLive Server where the AVI file or broadcast is located by the Server's host name **or** IP address. (use only one of the two possible addresses).
    - Advantage of using host name to identify server address: minimizes time until video reaches end user.
    - Advantage of using IP address to identify server address: easier to maintain VDO files, since you can change the IP of the host computer where the AVI files are stored and still use the same host name.
    - Server address (of either type) must be followed by a forward slash (/) not a backslash (\ ). If the location includes the server port number (see below), the slash should follow the port number.
- **Server TCP Port**: the number of the TCP port used by the VDOLive Server if the default of 7000 is not used. If the default is used, do not list port number.
- **Path**: Complete location of AVI file of Video Server **or** channel name of broadcast on Broadcast Server.
    - Use either "|" (a WWW convention) or ":" used after a drive letter.
    - VDOLive Server for Windows 95 or NT: the path must include the **drive and directory** location.
        - Windows NT only : If a location for movie files was defined in the *Absolute Path Directory for Movies* box of the Windows NT Server Configuration window, the location of movie files relative to the absolute path can be used in the URL of the VDO file.
    - VDOLive Server for UNIX: use the */path*, relative to the virtual root set by the -d option of the Server if used, or the system root,

as well as the filename. Slashes within a path may be forward (/)or backward (\).

Examples

- VDOLive Video Server for Windows NT:

For the file *birds.avi* located in c:\movies on server17.coolvideo.com, using port 6500 instead of the default setting of 7000:

```
vdo://server17.coolvideo.com:6500/c:\movies\birds.avi
```

- VDOLive Video Server for UNIX:

To play the file *birds.avi* located in /movies on server24.coolvideo.com using the default TCP port of 7000, the VDO file would look like this:

```
vdo://server24.coolvideo.com/movies/birds.avi
```

- VDOLive Broadcast Server for Windows:

For the broadcast on channel1 from the Broadcast Server called server17.coolvideo.com the VDO file would look like this:

```
vdo://server17.coolvideo.com/channel1
```

Case Sensitive File Names

Since UNIX is case-sensitive, so are names of files created on UNIX systems.

For example, if a file was named *CAPTURE.AVI* on Windows 95, it can be referred to on a Windows NT VDOLive Server as either *capture.avi* or *CAPTURE.AVI* but on a UNIX VDOLive Server only *CAPTURE.AVI* is valid.

VDOLive Servers for UNIX can serve content created on UNIX or non-UNIX systems.

HTML Links to VDO Files

About HTML links

HTML is case **in**sensitive on any platform and so the same HTML code can be used on computers running UNIX or Windows operating systems. Filenames, however, will remain case-sensitive (see section above).

About HTML links to VDOLive video

HTML links can be used to display VDOLive video clips or broadcast content in one of the following ways:

- By a **direct reference** to the VDO file.
- By use of the <`embed>` tag; this is called **Plug-in** or **embedded** video. More information is in the section beginning on page 16.

Each HTML link, whether a direct reference or for Plug-in video, points to one VDO file.

Using direct reference to the VDO file for a clip or broadcast displays the video in one of two ways:

- in the window of the external VDOLive Player
- in the user's browser window, in full-screen "theater" mode.

The display depends on the user's choice. Users can select, in their Player Setup, to display non-embedded video in the external Player. In this case, any video not coded to display as Plug-in video will display in the user's

Player window. If the user leaves this box blank, non-embedded video will display in their browser window inside a large frame representing a theater.

Syntax of HTML links to VDOLive video

The syntax of an HTML link with a direct reference to the VDO file for a clip or broadcast is:

```
<a href="URL of VDO file"> Your Link Text Here</a>
```

Whereas the VDO file is a URL of the VDOLive compressed AVI file or broadcast channel, this URL specifies the location of the VDO file on the HTTP Server.

The correct syntax of the URL for the VDO file is:

```
protocol://[serveraddress/]Path/filename.vdo
```

Where:

- **Protocol**: the protocol used here is http://
  The colon and the forward slashes must be included. Backslashes (\ ) do not work.
- **Server address**: identification of the Web (HTTP) Server where the VDO file is located by the Server's host name **or** IP address. (use only one of the two possible addresses).
  - Advantage of using host name to identify server address: minimizes time until video reaches end user.
  - Advantage of using IP address to identify server address: easier to maintain VDO files, since you can change the IP of the host computer where the AVI files are stored and still use the same host name.

  - Server address (of either type) must be followed by a forward slash (/) not a backslash (\ ). If the location includes the server port number (see below), the slash should follow the port number.
  - If the VDO files are stored on the **same** HTTP as your web page, you do not need to specify the address of the Web server and can begin with the directory location of the VDO file.
- **Path**: Location of VDO file on HTTP Server.

Example

- Example of an HTML direct reference link to VDO file for a VDOLive video clip **or** Broadcast:

For the file *travel.vdo* located on the HTTP Server “www.coolvideo.com” in the directory */vdofiles* and where the hyperlink displayed on the web page appears as “Click to play”:

```
<a href="http://www.coolvideo.com/vdofiles/travel.vdo">Click to play</a>
```

Using the VDOLive Plug-in

About Plug-in Video

- Plug-in video allows you to display video **inline** on a web page, in the user's browser window.
- Creating Plug-in video is different from creating direct reference HTML links to VDOLive video in that it requires use of the **`<embed>`** HTML tag.
- Currently, users with one of two supported browsers can view VDOLive Plug-in video:
    - Netscape Navigator, version 2.0 or higher.
    - Microsoft Internet Explorer, version 3.0 or higher.

Users of other browsers who click on a Plug-in link will see a "broken image" icon.

Creating a VDOLive Plug-in

➢ To create a Plug-in:

1. Create a VDO file for each clip.
2. Store the VDO file on an HTTP (web) server.

3. Create an HTML link to the VDO file, using the syntax defined below.

When specifying the height of the Plug-in, add 16 pixels to the height, to allow for display of messages and Plug-in control.

Syntax of the Embed Tag

Below is the syntax of the Embed tags **as defined by Netscape™**:

<EMBED>

This tag allows different kinds of documents of varying data types to be embedded into an HTML document. It has three default attributes: SRC, WIDTH, and HEIGHT. It may also contain optional parameters that can be sent to the plug-in handling the embedded data type.

SRC=<URL>

The URL of the source document.

WIDTH=<size in pixels>

The WIDTH attribute specifies the width of the embedded document, in pixels.

HEIGHT=<size in pixels>

The HEIGHT attribute specifies the height of the embedded document, in pixels.

PARAMETER_NAME=<PARAMETER_VALUE>

Optional parameters to send to plug-ins. There can be an infinite number of parameters passed to a plug-in. Examples of parameters are `PLAY_LOOP=TRUE`, or `CONTROLS=FALSE`. Parameters are specific to each plug-in.

Examples:

<EMBED SRC="MyMovie.mov", WIDTH=150, HEIGHT=250 CONTROLS=TRUE>

<EMBED SRC="DoomGame.ids", WIDTH=400, HEIGHT=300 SPEED=SLOW LEVEL=12>

Additional Embed Options valid for use with VDOLive Plug-ins

In addition to the <src> <width> and <height> tags described above, following are additional options that can be used with VDOLive Plug-in video, or information relating to the above tags that is specific to their use with VDOLive Plug-ins:

Tag	Description
src	Defines name and location of VDO file.
autostart	▪ Valid values for this tag are true or false. The default value = true. ▪ When autostart = true, Plug-in video begins playing automatically when user reaches web page. ▪ When autostart = false, Plug-in video only starts with user’s mouse-click on video link.
stretch	▪ Use of this tag is not recommended, since it produces a poor quality video image. ▪ This tag is valid only for VDOLive video, not for Lite VDO content. ▪ Valid values for this tag are true or false. ▪ When stretch = true, size of Plug-in video matched to size of Plug-in window, even if this requires distortion of the true image size. ▪ When stretch = false, video is displayed at true size even if it does not match size of Plug-in window. If the area of the Plug-in window is larger than the dimensions of the video image, extra space is shown at the perimeter; if

stretch, cont.	the Plug-in window is smaller than the true image size, the image is cut off at the top and right borders of the Plug-in window.
Width, height	When stretch = true, use these tags to scale video to designated section. When stretch = false, video is centered in designated area. When you specify width=1 and height=1, Netscape makes the video fill the page, whatever the page size is.

For the video to accommodate features of the plug-in, such as messages, specify a Plug-in size larger than the actual video image (frame) size, as follows:

To Avoid...	**Do This…**
Truncation at lower border of image	Add 16 pixels to the height of the Plug-in.
Frame not displayed	Add 8 pixels to the height and 8 pixels to the width of the Plug-in.

HTML Syntax for VDOLive Plug-ins

```
<embed src = “protocol://[serveraddress/][directory
name/]filename.vdo” width=x height=y autostart={true
or false} stretch={true or false}>
```

Where:

- **Protocol**: the protocol for these links must be HTTP. The colon and the forward slashes must be included. Backslashes (\ ) do not work.

- **Server address**: identification of the HTTP Server where the VDO file is located by the Server’s host name **or** IP address. (use only one of the two possible addresses). This location, must be followed by a forward slash (/) not a backslash (\ ).
- **Path**: Complete location of VDO file on HTTP (Web) Server.
  - Use either “|” (a WWW convention) or “:” used after a drive letter.

Examples

Example of an HTML link for VDOLive Plug-in video (clip **or** broadcast) :

For the file *travel.vdo* located on the HTTP Server “www.coolvideo.com” in the directory */vdofiles*:

```
<embed src="http://www.coolvideo.com/vdofiles/travel.vdo"
width=160 height=128 stretch=”false” autostart=”false”>
```

The HTML link should be written with no line breaks.

Reaching Audiences Behind Firewalls

Introduction

Firewalls present a problem to content providers since many firewalls do not allow video content to reach users. Several alternatives are available to you if you are providing video content to users whose computers are behind a firewall.

- If the firewall in use is a **packet filtering firewall**, the firewall administrator can open a port for incoming and outgoing VDOLive data. Refer the system or firewall administrator to the VDOLive Server manual, or to VDOnet Corp. for information and guidelines.
- Some **proxy firewalls** support VDOLive video streaming; if this type of firewall is in use, your content can reach the users, provided they properly configure their VDOLive Players.
- If neither of the above solutions is available, create **Lite VDO** movies for this audience. See the section below for more information.

Serving Audiences behind Firewalls

You can serve audiences behind firewalls that do not support video by using either of the first two methods defined above, each of which involves working with the firewall administrator to allow VDOLive content through while maintaining the integrity of the firewall, or by using the **Lite VDO Plug-in** developed by VDOnet for situations in which VDOLive content cannot pass through the firewall.

Creating a Lite VDO Plug-in

➢ To create a Lite VDO Plug-in:

1. Create an AVS file for each clip. Use the AVI-AVS Converter tool, included with the VDOLive Tools, to do this. (see below)
2. Store the AVS file on an HTTP (web) server.
3. Create an HTML link to the AVS file using the syntax described below.

When specifying the height of the Plug-in, add 30 pixels to allow for display of the Plug-in control.

Using the AVI-AVS Converter

Overview

Use this tool to convert AVI files compressed in Lite VDO mode into AVS files for direct streaming from an HTTP server (Web) to VDOLive clients.

If you do not have Lite VDO content, but want to use your VDOLive content for direct streaming from an HTTP (Web) server to VDOLive clients, you can also use VDOLive Movies for this. The best method is to use the uncompressed source video to create Lite VDO AVI files (by using

Lite VDO encoding mode). If you do not have the uncompressed video, then use this Converter to create AVS files from VDOLive Movies.

AVS files created from Lite VDO encoded files have better video quality than AVS created from VDOLive Movies using this Converter.

The Converter can also convert AVS files into Lite VDO AVI files.

Procedures

➢ To convert Lite VDO AVI to AVS:

1. Open the Converter from the VDOLive Tools program folder.
2. In the Input File box, browse for, or type in, the name of the Lite VDO AVI file to be converted.
3. In the Output File box, the same file name appears, with the AVS extension. You can change the file name or directory.
4. Click Convert. Conversion begins, and stops automatically when completed. Depending on the size of the clip you are compressing, this may take a while; the percentage of the conversion completed is displayed on screen.

➢ To convert VDOLive Movies to AVS:

1. Open the Converter from the VDOLive Tools program folder.
2. In the Input File box, browse for, or type in, the name of the VDOLive Movie AVI file to be converted.
3. In the Output File box, the same file name appears, with the AVS extension. You can change the file name or directory.
4. Click Convert. The AVS Encoding dialog opens.

5. Use the slider to select the AVS Encoding rate. Select the rate that matches the expected rate at which your audience connects to the Internet, for example, set the rate to 28 for an audience of users with a modem connection to the Internet.
6. Click OK. Conversion begins, and stops automatically when completed. Depending on the size of the clip you are compressing, this may take a while; the percentage of the conversion completed is displayed on screen.

The maximum AVS Encoding rate is 128 kbps. The minimum is 14 kbps.

➢ To convert AVS to Lite VDO AVI:

1. Open the Converter from the VDOLive Tools program folder.
2. In the Input File box, browse for, or type in, the name of the AVS file to be converted.
3. In the Output File box, the same file name appears, with the AVI extension. You can change the file name or directory.
4. Click Convert. Conversion begins, and stops automatically when completed.

HTML Syntax for Lite VDO Plug-ins

Use HTML to create links to Lite VDO clips; the syntax of the HTML string should be one of the following:

```
<embed src = “[serveraddress/][directory
name/]movie1.avs” width =x height=y autostart={true or
false}>
```

–or–

```
<a href = "http://[serveraddress/][directory
name/]filename.avs">Click to play video</a>
```

where:

- The server address is required only if the AVS file is on an HTTP server other than the one containing the web page including this Plug-in. If used, the HTTP server address should be specified by its complete hostname or IP address.

- The directory name or full path of the AVS file is required only when the AVS file is located in a different directory than the HTML page from which the user accesses the movie (i.e. the page with the link on it).

- The width and height must be specified, in pixels. Dimensions of the clip (frame size) must be evenly divisible by 16.

- Allow an extra 30 pixels to the height of the Plug-in to allow for the control buttons that users see when playing Lite VDO clips.

- Autostart can be **either** true **or** false.

- Loop and Stretch options do not work for the Lite VDO Plug-in.

- Using the **`<embed>`** tag will display the clip inline.

- Using the **`<a href>`** tag will display the clip in full page or “theater” mode.

Do not put more than one Lite VDO (AVS) plug-in on a web page; this is not a supported option in this version of the Plug-in.

For example:

```
<embed src="http://www.mydomain.com/files/clips
/lite/nice_movie.avs" width=160 height=145
autostart="true">
```

To create a page that shows several possible clips from which the user can play one at a time, by clicking on images:

```
<html>

<head> <title>Lite VDO Clips </title> </head>

<body>

Click the name of a movie to see a Lite VDO
clip....

<OL>

<LI><A HREF=1.html><img
src="../img/gifs/pic1.gif">Nice movie</A><br>

<LI><A HREF=2.html><img
src="../img/gifs/pic2.gif">Old movie</A><br>

<LI><A HREF=3.html><img
src="../img/gifs/pic3.gif">New movie</A><br>

</OL>

</body></html>
```

For details of the syntax of the Embed tag, refer to page 17.

Additional Embed Options valid for use with Lite VDO Plug-ins

In addition to the <src> <width> and <height> tags described above, following are additional options that can be used with Lite VDO Plug-in video, or information relating to the above tags that is specific to their use with Lite VDO Plug-ins:

Tag	Description
src	Defines name and location of VDO file.
autostart	▪ Valid values for this tag are true or false. The default value = true. ▪ When autostart = true, Plug-in video begins playing automatically when user reaches web page. ▪ When autostart = false, Plug-in video only starts with user’s mouse-click on video link.
stretch	▪ this tag invalid for use with Lite VDO Plug-in

Configuring the MIME type on your HTTP (Web) Server

For VDOLive Video

The following settings should be used to set up the MIME type for the VDO file type/s in use when you are serving VDOLive video:

- MIME type: **video**
- Sub type: **vdo**
- File extension: **vdo**

For further details, please refer to the documentation for your HTTP server.

For Lite VDO™

When you are serving Lite VDO video, your Web Server is serving AVS files directly, not using VDO files. Use these MIME settings:

- MIME type: **video**
- Sub type: **avs**
- File extension: **avs**

Please refer to the documentation for your HTTP server for further details.

Testing Content

Testing VDOLive Content

You can test compressed clips locally using the Windows Media Player or using the Playback option from VDO Clip. These methods of playback, however, are only useful for evaluating video image quality; they do not simulate network conditions such as those your users will have. Refer to the Guide to the VDOLive Tools for additional information.

To test both video image quality and also reception when playing clips from a VDOLive Server, over a network, you can use the VDOLive Player in standalone mode to test compressed VDOLive clips. The clips must first be uploaded to the VDOLive Server.

➢ To use the Player to test playback quality of compressed clips:

1. Open the VDOLive Player and click on the Play button; a dialog box opens.
2. Enter the complete location and name of the clip you want to test, using the same syntax as for a VDO file:

```
vdo://VDOLiveServerHostName|Server IP
address[:Server port]/FileNameOnServer.avi
```

Testing Lite VDO Content

You can test AVS content before uploading it to the web server.

➢ To test AVS clips:

1. Install the Lite VDO Plug-in. Installing the VDOLive Player version 2.1b or higher from the VDOnet web site (http://www.vdo.net) will also install this Plug-in.
2. Write an HTML file using either the **`a href`** or the **`embed src`** syntax described above.
3. Open the HTML in your browser, or drag and drop the HTML file into your browser.