VESA®

VDIF Standard

Video Electronics Standards Association

2150 North First Street, Suite 440
San Jose, CA 95131-2029

Phone: (408)435-0333
FAX: (408)435-8225

VESA Display Information Format
VDIF

Standard

Version Number: 1.0
Date: August 23, 1993

Purpose
To standardize the format for a machine readable display specification. A uniform display specification enables video controllers from various manufactures to provide correct monitor timing and video signals to any connected VDIF compliant display, without prior knowledge about the displays timing requirements.

Summary
This standard defines a common set of parameters that defines a display sufficiently for a video controller to configure itself for optimal monitor timing. The parameters are stored in ASCII format and or Binary format files, that the video controller can read at configuration time.
VDIF also provides information about the display geometry and color characteristics, thus enabling true WYSIWYG both when it comes to geometry and color characteristics.

Document revision history

Version 1.0, Aug. 23, 1993

First issue

Acknowledgments

This document was made possible by the joint efforts of the members of the VESA Monitor Committee 1991 / 1992 / 1993. In particular the following individuals and their companies contributed with significant amount of time and knowledge.
Edward D Anwyl, IBM Personal Computer Company: Future communication channel extensions and in depth review.
Richard Atanus, NEC Technologies: Display File information and definitions
Anders Frisk, ICL Personal Systems: Display / Controller Compatibility Specification. Editor of the document
Thomas Roell, Snitily Graphics Consulting Services: Color definitions. ASCII and Binary file format. "C" code parser.
Mark W Snitily, Snitily Graphics Consulting Services: Color definitions. ASCII file format
Andre Vachon, Microsoft Corporation: Binary file format. In depth review
Jeff Zalewski, Diamond Computer Systems: Configuration Process.

VDIF Document Content

1. Purpose and General Description

VDIF provides a way for a display to communicate its specification in machine readable format to the display controller system. This will enable the creation of display controller systems that can adapt dynamically to a wide range of displays without having prior detailed knowledge about the specification of the attached display.

Properly implemented VDIF will eliminate the need for future "discrete monitor timing standards". Hence any display manufacturer can move forward with incremental improvements in monitor timings without having to agree with every controller manufacturer in order to get support for a new timing.

In the same way controller manufacturers will be allowed to implement new resolutions without having to agree in advance on the details with the monitor manufacturer.

This will benefit end users as well since they can invest in leading edge technology without fear that the display and or the controller becomes out dated as soon as a new monitor timing standard is released.

VDIF will allow for greater market diversification, essentially users can select a controller based on controller qualities (e.g. drawing speed, number of colors, etc.). In the same way, users will be able to select the display based on display qualities (e.g. refresh rate, screen utilization, overscan, size, etc.).

VDIF will specify how the information is structured and how it shall be communicated to the host system. VDIF will not specify how the information about the display is to be used.

The information can be provided by the Display manufacturer in the form of a disk file.

2. Display Capability File Information and Definitions

The information in the file is intended to fully describe a display unit. The current revision is primarily addressing information pertinent to CRT based displays. It is the ambition of the work group to expand this to also cover other types of displays. Please direct any suggestions for future additions to the VESA office.

The Display information is grouped in three sections

1. Description
2. Operational Limits
3. Pre-Adjusted Timings

The Operational Limits and the Pre-Adjusted Timings sections may occur several times depending on the type of monitor. Each occurrence describes the limits for one of the operational ranges the monitor is capable of operation within.

The Pre-Adjusted Timings section describes sets of frequencies that have been adjusted and verified by the manufacturer. Each occurrence describes one possible mode of operation. The monitor may also be able to operate in modes not described in the Pre-Adjusted Timings section. In such case the configuration SW has to verify that the intended mode does not violate the Operational Limits specification.

Tolerances for Pre-Adjusted timing values are +/- 0.5%.

A supplier may opt not to provide the information in some fields. See the ASCII file format definition for details on how to create a field without such information.

PARAMETER DEFINITIONS

VERSION:

Version Major and minor version of the VDIF format.

MONITOR DESCRIPTION SECTION:

File date "Date"
Date when the file was created

File Revision "FileRevision"
Major and minor file revision.

Manufacturer "Manufacturer"
String selected by the manufacturer, identifying the manufacturer of the display. This field might be used when presenting the user with selection alternatives at configuration time.

Model "ModelNumber"
String selected by the manufacturer, identifying the model of the display. This field might be used when presenting the user with selection alternatives at configuration time.

Min. VDIF Index for max. resolution
"MinVDIFIndex"
A string stating the maximum resolution and the VDIF Index.

Version "Version"
Version of the Model

Serial Number "SerialNumber"
String assigned by the manufacturer to identify the individual units.

Date Manufactured "DateManufactured"
Date when the monitor was manufactured.

Monitor Type "MonitorType"
Monochrome or Color. A monitor identifying itself as "Color" will be assumed to have three signal inputs and three independently-excitable phosphors, which will be identified as "Red", "Green", and "Blue" in this file, in this order corresponding to the order of the three signal inputs, regardless of the actual colors of the three phosphors.

CRT Size "CRTSize" This is the actual CRT size measured diagonally in inches and not the CRT display area. This parameter is provided as an aide to the consumer and the display manufacturers support staff to easily determine if the correct file is being used. The information is not envisioned being used for configuration of the display controller.

The following parameters provide information relating to the phosphor and color characteristics of the monitor. Note, that in the case of phosphor information, three values are ALWAYS supplied, for both monochrome and color monitors. In the case of a monitor which identifies itself as "Color" in the "Monitor Type" field, this should be assumed to correspond to the Red, Green, and Blue phosphors. If the monitor identifies itself as "Monochrome", the additional two fields can be used to give more complete information, if desired, in the case of a multi-component monochrome phosphor. If only a single set of phosphor parameters is given in the monochrome case, all entries for the second and third phosphors should be zero. A monochrome monitor MUST at least provide information in the first set of fields for each parameter.

Phosphor Decay

`"RedPhosphorDecay", "GreenPhosphorDecay", "BluePhosphorDecay" or "Phosphor1Decay", "Phosphor2Decay", "Phosphor3Decay".`
The time in microseconds for each phosphor to decay from 100% to 10% luminance.
Phosphor 2 and 3 decay are optional for monochrome monitors.

Border Color

`"BorderRed", "BorderGreen", "BorderBlue" or "Border".`
This is the percentages of the Red, Green and Blue border colors that are recommended by the manufacturer of the display. This is a linear number. 100% for all the three guns would represent white and 0% for all three guns would represent black.

White Point

`"WhitePointx", "WhitePointy", "WhitePointCapY"`
This is the white point described as x and y coordinates in the CIE 1931 Chromaticity chart. The luminance Y is stated in cd/m2.

Chromaticity

"RedChromaticityx", "RedChromaticityy", "GreenChromaticityx", "GreenChromaticityy", "BlueChromaticityx", "BlueChromaticityy" or "Phosp1Chromaticityx", "Phosp1Chromaticityy" "Phosp2Chromaticityx", "Phosp2Chromaticityy" "Phosp3Chromaticityx", "Phosp3Chromaticityy"
This is the x and y values of the Red, Green and Blue phosphors at 25 deg. centigrade and at full illumination (max. video level on all inputs) per the CIE 1931 Chromaticity Chart.For a monochrome monitor the color points for the different phosphors can be specified as an option. In case they are not specified they should be stated as 0,0.

Gamma

"RedGamma", "GreenGamma", "BlueGamma" or " Gamma"
The exponent function of the nonlinear characteristic of the video input versus the brightness on the CRT used for the purpose of correcting the system input output transfer characteristics.
output= constant *input ^ gamma
The gamma information may also be provided in the form of an optional gamma table, should such table be provided, it shall take precedence over the gamma figure described here.

OPERATIONAL LIMITS SECTION:

Video Type

"VideoType"
The type of Video Signal applied to the display unit. Allowed types: "Analog", "TTL", "ECL" and "DECL", and "Other". "DECL" refers to differential ECL inputs. If "Other", the interface is proprietary and a further description can be made in an optional VDIF section.

Video Input Termination Resistance

"TerminationResistance"
The nominal termination resistance of the video signal inputs.

White Level

"WhiteLevel"
The level in volts above the blank level that corresponds to 100% luminance.

Black Level

"Black Level"
The level in volts above the blank level, which corresponds to the darkest signal which should be provided during the active time. (This correspond to "set up" in an analog video system.)

Blank Level

"BlankLevel"
Normally zero for a standard AC-coupled video input system. If non zero, the value gives the offset, in volts, for the blank level in a DC-coupled system.

Sync Level "SyncLevel"
The level in volts BELOW the blank level, which corresponds to the expected level of the sync pulses in the video signal. If composite sync on video is not used, this parameter should take the value zero.

Sync. Type "SyncType"
The type of Sync. Signal applied to the display unit. Accepted values are "Analog", "TTL", "ECL", "DECL" or "Other". The type "Other" indicate that the sync may be described in an optional VDIF section.

Sync. Configuration "SyncConfiguration"
The type of Sync. Configuration applied to the display unit. Accepted values:
"Separate"
"C" = Composite without serration pulses
"CP" = Composite with serration pulses
"G" = Green without serration pulses
"GP" = Green with serration pulses
"Other" = Indicate that the sync may be described in an optional VDIF section. This include displays that require equalization pulses.

Min. Hor. Freq. "MinHorFrequency"
The minimum horizontal frequency in kiloHertz.

Max. Hor. Freq. "MaxHorFrequency"
The maximum horizontal frequency in kiloHertz.

Min. Ver. Freq. "MinVerFrequency"
The minimum vertical frequency in Hertz. For interlaced scanning this is the minimum field rate.

Max. Ver. Freq. "MaxVerFrequency"
The maximum vertical frequency In Hertz. For interlaced scanning this is the maximum field rate.

Max. Pixel Clock "MaxPixelClock
The maximum pixel clock in MegaHertz.

Max. Hor. Pixels "MaxHorPixel"
Maximum number of horizontal pixels that can be addressed.

Max. Ver. Pixels "MaxVerPixel"
Maximum number of vertical pixels (lines) that can be addressed. This is the number of lines

containing addressable pixels per frame.

Hor. Line Dimension	"MaxHorAddrLength"	The maximum horizontal line length in millimeters of addressable pixels.
Ver. Height Dimension	"MaxVerAddrHeight"	The maximum vertical height in millimeters of addressable pixels.
Min. Hor. Retrace	"MinHorRetrace"	The minimum time, in microseconds, required for horizontal retrace. The Hor. Blank. Time must be equal or greater than this value.
Min. Ver. Retrace	"MinVerRetrace"	The minimum time, in milliseconds, needed for vertical retrace. Ver. Blanking. Time must be equal or greater than this value.

PRE-ADJUSTED TIMINGS SECTION:

Pre-Adjusted Timing Name `"PreAdjustedTimingname"`
A string used by the manufacturer and end users to easily identify the Pre-Adjusted Timing. Can be either special names derived by the manufacturer or standard names such as VESA 800 x 600 at 60 Hz.

Hor. Pixels `"HorPixel"`
The Horizontal Addressability measured in Pixels.This number divided into the Horizontal Addressable Time and the result divided into 1 will produce the Pixel Clock Frequency.

Ver. Pixels `"VerPixel"`
The Vertical Addressability measured in Lines. This is the number of addressable lines per frame, regardless of scan type.

Hor. Freq. `"HorFrequency"`
The Horizontal scanning Frequency measured in kiloHertz. The inverse of this number will produce the Horizontal Total Time.

Ver. Freq. `"VerFrequency"`
The Vertical scanning Frequency measured in Hertz .The inverse of this number will produce the Vertical Total Time.
This is the field rate in case of interlaced scanning.

Pixel Clock `"PixelClock"`
The pixel clock in MegaHertz. This parameter can be calculated as Hor.Pixels / Hor.Addressable.Time.

Character Width `"CharacterWidth"`
In case all the horizontal timings are designed to be multiples of a certain character time (defined as: CharacterWidth / PixelClock), then that character width is specified in this field. Monitor designers should not assume that controllers can supply a timing with finer resolution than the pixel clock, this translates to a character width of "1", which is the default value for this parameter. (Note, the name of this parameter refers to a term commonly used to describe a limitation in a display controller, that may limit the timing a particular controller is capable of producing.)

Hor. Addressable Line Length `"HorAddrLength"`
The Horizontal Addressable Line Length measured in millimeters (mm). This is the actual length of the image during the Horizontal Addressable period.

Ver. Addressable Height `"VerAddrHeight"` The Vertical Addressable Height measured in millimeters (mm). This is the actual height of the image during the Vertical Addressable period .

Pixel Aspect Ratio — "PixelWidthRatio"and "PixelHeightRatio"
The Aspect Ratio of a single Pixel. The first number is the pixel width and the second number is the pixel height.

Scan Type — "ScanType"
The type of Scan Type applied to the display unit. Allowed types are: "Non-Interlaced", "Interlaced" and "Other". The type Other indicates that the scan type is described in an optional manufacturer specific VDIF section.

Sync. Polarity — "HorSyncPolarity", "VerSyncPolarity"
The Polarities of the horizontal and vertical sync signals applied to the display unit. Polarities may be either negative or positive. To describe the polarity for Sync. Configuration Composite Sync., only the Horizontal Sync. Polarity parameter is used and the Vertical Sync. Polarity value is ignored.

Hor. Total Time — "HorTotalTime" The time measured in microseconds of the total Horizontal period. This is the period measured from the beginning of one Horizontal Addressable period to the beginning of the next Horizontal Addressable period. This number divided into 1 will produce the Horizontal Frequency.

Hor. Addressable Time — "HorAddrTime"
The time period, measured in microseconds, during which the addressable data is displayed. The beginning of this time is the reference point for the Horizontal Sync. Start and Horizontal Blank Start parameters.

Hor. Blank Start — "HorBlankStart" The time period measured in microseconds starting at the beginning of the Horizontal Addressable Time and ending at the beginning of the Horizontal Blanking Time. The actual horizontal right border time (if present) can be calculated by subtracting the Horizontal Addressable Time parameter from the Horizontal Blanking Start parameter.

Hor. Blank Time — "HorBlankTime" The time period measured in microseconds starting at the beginning of the Horizontal Blanking and ending at the end of the Horizontal Blanking period. The horizontal back porch (whether negative or positive)can be calculated from this field by adding the Horizontal Blanking Start parameter and the Horizontal Blanking Time period together and subtracting the Horizontal Sync. Start period and the Horizontal Sync. Time period from them. In addition, the horizontal left border (if present) can be calculated by subtracting the Horizontal Blanking Start parameter and the Horizontal Blanking Time parameters from the Horizontal Total parameter.

Hor.Sync. Start — `"HorSyncStart"`
The time period measured in microseconds starting at the beginning of the Horizontal Addressable period and ending at the beginning of the Horizontal Sync. Time. This time period includes the Horizontal Addressable Time, the horizontal right border (if present) and the horizontal front porch (whether negative or positive).

Hor. Sync. Time — `"HorSyncTime"`
The time measured in microseconds, during which the sync. pulse is active. . This is the period of time starting at the end of the front porch and ending at the beginning of the back porch.

Ver. Total Time — `"VerTotalTime"` The time measured in milliseconds of the total Vertical period. This is the time measured from the beginning of one Vertical Addressable period to the beginning of the next Vertical Addressable period. This number divided into 1 will produce the Vertical Frequency. In case of interlaced scanning, all vertical timings refer to the timing for each field.

Ver. Addressable Time — `"VerAddrTime"`
The time period measured in milliseconds . This is the period of time during which the addressable data is displayed.

Ver. Blank Start — `"VerBlankStart"` The time period measured in milliseconds starting at the beginning of the Vertical Addressable period and ending at the beginning of the Vertical Blanking period. The vertical bottom border time (if present) can be calculated by subtracting the Vertical Addressable Time parameter from the Vertical Blanking Start parameter.

Ver. Blank Time — `"VerBlankTime"` The time period measured in milliseconds starting at the beginning of the Vertical Blanking and ending at the end of the Vertical Blanking period. The vertical back porch (whether negative or positive)can be calculated by adding the Vertical Blanking Start parameter and the Vertical Blanking Time period together and subtracting the Vertical Sync. Start period and the Vertical Sync. Time period from them. In addition, the vertical top border (if present) can be calculated by subtracting the Vertical Blanking Start parameter and the Vertical Blanking Time parameters from the Vertical Total parameter.

Ver. Sync. Start — `"VerSyncStart"`
The time period measured in milliseconds starting at the beginning of the Vertical Addressable Time and ending at the beginning of the Vertical

Sync. Time. This time period includes the Vertical Addressable Time, the vertical bottom border (if present) and the vertical front porch (whether negative or positive).

Ver. Sync. Time

`"VerSyncTime"`

The time period of the Vertical Sync. Time measured in milliseconds. This is the period of time starting at the end of the front porch and ending at the beginning of the back porch.

GAMMA TABLE SECTION: (Optional)

Gamma Table

An optional table consisting of values that represent the luminance output for a given input voltage.

The table consists of two parts, the first part lists the number of sample points in the table and the second part is the table with output values.

The sample points represent equally spaced values of the video level input voltage, ranging from 0 to "White Level" defined above.

Depending on if the monitor is color or monochrome the table has 3 or 1 luminance value(s) per sample point, one for each primary color.

GRAPHICAL REPRESENTATION OF TIMING PARAMETERS

3. FILE FORMAT

The VDIF file information shall be distributed in two different formats, ASCII and Binary. It is anticipated that each format will appeal to different markets.

A parser for both formats in the form of C source code is available from VESA.

In the event that the ASCII and Binary files don't match, the ASCII file takes precedence.

ASCII format

The ASCII format is specified in appendix A.

Binary format

The binary format is specified in appendix B.

4. STORAGE LOCATION

It is expected that the VDIF file will not be used in it's 'as supplied' form but will be compiled on a one-time basis at system installation to a controller-specific format. Although the VDIF file(s) might therefore reside on the system, it is not a requirement, however in order to provide for this situation both a naming and storage convention have been established.

DEFINITION

The definition file will be stored in a VDIF subdirectory of the directory appropriate to the installed operating system. The ASCII format file will have a file extension of ".VDA". The Binary format file will have a file extension of ".VDB".

MANUFACTURER CODE

The first three characters in the file name are defined as appropriate for the manufacturer of the display from the agreed list of display manufacturer identifiers.

A directory over the assigned codes is kept by VESA. Manufacturers are requested to register selected code with the VESA office in order to avoid naming conflicts.

The next five characters are defined by the display manufacturer.

NOTE! The file name and extension can be uppercase or lowercase characters. For operating systems that are case sensitive lowercase should be used.

STORAGE LOCATION

The directory referenced below should be located on the same disk drive as the currently loaded operating system uses to store its other configuration files.

DOS

File stored in \DOS\VDIF directory

Microsoft WINDOWS

File stored in \WINDOWS\SYSTEM

Microsoft Windows NT

File stored in NT\SYSTEM\DRIVER

OS/2

File stored in \OS2\SYSTEM\VDIF

UNIX

/usr/lib/vesa/vdif

Apple

File stored in the "SYSTEM FOLDER"

DISKETTE LABELING

If the VDIF file is supplied on a diskette, the labeling should contain the following information:

VDIF File(s), use this diskette for configuration of the display system.

MULTIPLE VDIF FILES IN ONE SYSTEM

More than one VDIF file may reside in the storage location described above. This allows VDIF files to reside on a file server that handle multiple monitors and/or provides systems that have multiple monitors access to multiple VDIF files. It is the responsibility of the operating system (or application software) to determine the active VDIF file.

5. CONFIGURATION PROCESS

The VDIF Configuration Process allows user to select and store the best match between a monitor and a Video Graphics Controller (Controller) installed on a user's system. A VDIF file that is provided by a display manufacturer contains the monitor information and Pre-Adjusted Timing "tables" in a fixed format. This file is used by a Video Graphics Controller (Controller) installation or a setup program supplied by the Controller manufacturer. It is not the intention of VESA to suggest a method of installation procedure but rather to provide a detailed guide for that process.

The configuration process takes place after initial start up procedure has been completed and control is transferred to the operating system. For sake of simplicity of this chapter all references are made to the ISA/EISA/MC platform. The VDIF installation process can be broken down into four steps:

Step 1: Read and Display the VDIF file information.

VDIF file is read from the floppy drive A(B).

Monitor information is presented to the user.

Manufacturer

Compatibility

Model

Monitor type

Serial #

CRT size

VDIF index

List of pre-adjusted timings matched and supported by the monitor and the Controller is displayed.

640x480 60Hz

800x600 75Hz

1024x768 60Hz

1024x768 75Hz

etc.

Backward compatibility timing supported is displayed.

All IBM VGA modes

800x600 72Hz VESA

VDIF index for each resolution may also be displayed.

Step 2: Resolution selection

User can select one or more resolutions from the list presented in the step 1.

Step 3: Convert and Save

Before selected resolutions are saved the installation program converts ".VDA" text file into a binary file. A format of the binary file must be the same as the corresponding ".VDA" file. This file can be stored either in a non-volatile memory located on the Controller or in the system as a TSR file for MSDOS operating system or as a ".VDB" file for OS/2 or Windows NT operating systems.

Unlike DOS system that uses video BIOS calls, Windows NT, UNIX and OS/2 operating systems program the Controller directly through a video driver. The CRTC programming parameters are read by the video driver from the ".VDB" file or from a non-destructive memory (if accessible). The ".VDB" file(s) are stored in the same subdirectory as the ".VDA" files.

6. Display / Controller Compatibility Specification

To ensure that a non technical end user can select matching Controller and Display, it is imperative that both Displays and Controllers are specified in the same way.

The VDIF specification calls for a parameter "VDIF Index".

Definition for displays:

VDIF Index = Horizontal addressable time (us)

Definition for controllers:

VDIF Index = Horizontal addressable pixels / Pixel clock frequency (MHz)

To be VDIF compliant, data sheets and "on the box" specifications for supported resolutions and timings shall contain the statements:

SUPPORTED RESOLUTIONS

Horizontal x Vertical pixels, VDIF Index (2), Vertical Refresh Rate (1)

The above line shall be repeated for each supported resolution and index.

NOTES

(1) Vertical refresh rate should preferably not be specified for controllers since this is a parameter that is defined by the display timing and controllers are assumed to be able to handle a range of timings. .

Interlaced display modes shall specify both frame rate and vertical frequency.

For backwards compatibility, references regarding support for existing industry standard timings can be specified as they have been in the past (resolution and refresh rate) . However it is important that the VDIF Index be listed for all supported resolutions.

(2) The VDIF index can be stated as a range.

Examples showing how VDIF compliant specifications can look

Example of a controller specification:

This controller supports the VESA Display File

To select a suitable monitor for this controller, follow this procedure:

1. Determine which is the highest resolution that you are planning to use.

2. Note the VDIF Index range that this controller support for that particular resolution.

3. Look for a monitor that has a VDIF Index that falls within this range for the selected resolution. Generally a lower index indicates a higher quality monitor with better ergonomics.

In case you are planning to use several different resolutions, you should verify that the selected monitor support these resolutions and that it has a VDIF Index that falls within the range of this controller (same procedure as above).

SUPPORTED RESOLUTIONS

H x V Pixels	Colors	RAM	VDIF Index	Monitor type
1280 x 1024	16	1MB	10.67 - 20.00	VDIF Comp.
1024 x 768	256 (16)	1MB (512KB)	8.53 - 21.00	VDIF Comp. VESA 70Hz
800 x 600	256	512KBor1MB	6.67 - 22.22	VDIF Comp. VESA 56Hz VESA 60Hz VESA 72Hz

Example of a display specification:

This display supports the VESA Display File.

To select a suitable controller for this display follow this procedure:

1. Determine which is the highest resolution that you are planning to use.

2. Note the VDIF Index or Index range that this display support for that particular resolution.

3. Look for a controller that has a VDIF Index that falls within this range for the selected resolution. A controller that supports the lowest VDIF Index specified for this display will allow you to run this display at its highest refresh rate.

In case you are planning to use several different resolutions, you should verify that the selected controller support those resolutions and that it has a VDIF Index that falls within the range of this display (same procedure as above).

SUPPORTED RESOLUTIONS*:

H x V pixels	VDIF Index	Refresh rate	DPI (dots/inch)
1280 x 1024	9.50	76Hz	105
1150 x 900	9.50	86.5Hz	94
1024 x 768	11.20 - 12.80	86 - 75Hz	84
1024 x 768	15.00	64Hz	84
800 x 600	14.20 - 18.00	86 - 68Hz	66
640 x 480	18.00	85.5Hz	52

*
Resolutions that are displayed as interlaced shall be followed by the word "Interlaced" or an "I*" and a statement below the table "I* = Interlaced".
Examples "1024x768 I*" or "1024x768 Interlaced"
Refresh rates for interlaced modes shall state both the frame rate and the vertical frequency. Example "43.5/87 Hz".

NOTE! THE VALUES IN THE EXAMPLE ABOVE ARE NOT NECESSARILY REALISTIC.

Appendix A

ASCII File Format

The purpose of this section is to formally specify the layout and form of an ASCII VDIF file. The VDIF syntax is compatible with the syntax used by SVPMI files (see VESA Standard #VS911020, Super VGA Protected Mode Interface).

1. Overall Structure:

A VDIF file has six different section types:

Section Name	Required/Optional
[VERSION] VDIF version	Required
[MONITOR_DESCRIPTION] Basic information about the monitor	Required
[OPERATIONAL_LIMITS] Operational limits of the monitor	Required (one or more)
[PREADJUSTED_TIMING] Pre-adjusted timing for a specific operational limits	Required (one or more per [OPERATION_LIMITS])
[GAMMA_TABLE_ENTRIES] Number of entries in the gamma table	Optional
[GAMMA_TABLE] A table of display-luminance values	Optional

There is one [VERSION] and one [MONITOR_DESCRIPTION] section at the beginning of the VDIF file. The [VERSION] section simply contains the current VDIF version and revision. The [MONITOR_DESCRIPTION] section contains general monitor information.

An [OPERATIONAL_LIMITS] section defines a set of limits, a range, an abstract "space" where the monitor works. A VDIF file may have multiple [OPERATIONAL_LIMITS] sections.

A [PREADJUSTED_TIMING] section defines a specific setting, a sample point, a "point" within the abstract "space" defined by an [OPERATIONAL_LIMITS] section.
Multiple [PREADJUSTED_TIMING] sections can be specified for a given [OPERATION_LIMITS].

The optional [GAMMA_TABLE_ENTRIES] and [GAMMA_TABLE] sections allows one to specify a

table of display-luminance output measurements for the monitor. The information in this section is normally obtained from a calibration device that measures display-luminance. The intent is that the gamma table would be added automatically by a software program controlling the calibration device. Use of this information
provides a precise gamma correction and the ability of mapping the monitor's precise color gamut to and from device independent color spaces.

Future VESA extensions, as well as manufacturer proprietary extensions, to the VDIF file format will be allowed by introducting new sections. In the binary file, the `structure _VDIFScnHdr` is a generic section header that contains a tag field named `ScnTag`. In VDIF version 1.0, the only VESA tags that are defined correspond to the Operational Limits, Preadjusted Timings, and Gamma Table sections. VESA reserves the tag values up through 127 for future extensions. A tag values of 128 or higher will be interpreted as a manufacturer's proprietary extension.

An example of how the VDIF file is structured is as follows:

```
[VERSION]
[MONITOR_DESCRIPTION]

[OPERATIONAL_LIMITS]    <-----------------------,
   [PREADJUSTED_TIMING]                          |
   [PREADJUSTED_TIMING]    2 PTs for this OL. --'

[OPERATIONAL_LIMITS]    <-----------------------,
   [PREADJUSTED_TIMING]    1 PT   for this OL. --'

[OPERATIONAL_LIMITS]    <-----------------------,
   [PREADJUSTED_TIMING]                          |
   [PREADJUSTED_TIMING]                          |
   [PREADJUSTED_TIMING]    3 PTs for this OL. --'

[GAMMA_TABLE_ENTRIES]
[GAMMA_TABLE]
```

Graphically, one can think of the VDIF file as being a tree structure where the [MONITOR_DESCRIPTION] is the root of the tree, the [OPERATIONAL_LIMITS] sections and [GAMMA_TABLE] section are the children of the root, and the [PREADJUSTED_TIMING] sections are the children of the [OPERATION_LIMITS].

The structure used in the above example could conceptually be thought of as the following tree:

```
                        Monitor
                           |
      +--------------+------+---------+--------------+
      |              |                |              |
    limit          limit            limit          gamma
      |              |                |
  +---+---+          +        +------+------+
  |       |          |        |      |      |
timing  timing     timing   timing timing timing
```

2. Top Level Formal Grammar.

In formal context free grammars, a rule of the form

LIST = ELEMENT LIST | ELEMENT

is the way one expresses a list of 1 or more elements. That is,

"LIST" is an "ELEMENT" followed recursively by a "LIST"
or (which is the "|" symbol)
"LIST" is a single "ELEMENT".

And a rule of the form

A = B | <empty>

is the way one says that "A" is optional. It'll either be whatever "B" is defines as, or else it won't exist at all.

The top level formal grammar of the ASCII VDIF file is as follows:

(Ellipses (...) are used when we reach one of the VDIF sections. These section are described in more detail later on. In the following grammar, "ol" is short for "operational limits' and "pt" is short for "preadjusted timing'.)

```
vdif                    = version   monitor_description   ol_pt_list   gamma_option
version                 = ...
monitor_description     = ...
ol_pt_list              = ol_pt   ol_pt_list   |   ol_pt
ol_pt                   = operational_limits   pt_list
operational_limits      = ...
pt_list                 = preadjusted_timing   pt_list  |   preadjusted_timing
preadjusted_timing      = ...
gamma_option            = gamma   |   <empty>
gamma                   = gamma_table_entries   gamma_table
gamma_table_entries     = ...
gamma_table             = ...
```

3. Types in ASCII File:

The core types in the ASCII file are:

```
<integer>   non-negative numbers
<date>      <integer>/<integer>/<integer>  (year/month/day)
<string>    strings
<scaled>    <integer> or
            <integer>.<integer>
```

The <scaled> type is used for representing what appear to be floating point values in the ASCII file. Note that there are NO floating point values in the binary file. When a <scaled> type is encountered in the ASCII file, the resulting value is multiplied by 1000 before being placed into the binary file. Therefore, when the form <integer>.<integer> is encountered, the <integer> to the right of the decimal point has at most three decimal digits of accuracy. As an example, specifying 1 or 1.000 in the ASCII file would result in a value of 1000 in the binary file, specifying 1.234 would result in 1234.

For parameters that have the <scaled> type, the units defined in the ASCII file will be 1000 times the units defined in the binary file. For example, a parameter that is defined in terms of milliseconds in the ASCII file will be defined in terms of microseconds in the binary file.

The <scaled> type is also used for parameters that do not have units, namely the Chromaticity and Gamma exponent parameters. For these cases, the value is still multiplied by 1000 and is thought of as representing a floating point value in the binary file.

The <integer> type is used for all other numerical parameters.

Strings are composed of 7-bit ASCII characters terminated with the NULL character (i.e. zero).

4. Optional Parameter Values:

ALL parameters are REQUIRED. But it is permissible for the values of certain parameters to left unspecified. These parameters are enumerated below. (One specifies a null string with two double-quotes "".)

```
Date                        0/0/0
FileRevision                null string
Manufacturer                null string
Modelnumber                 null string
Version                     null string
SerialNumber                null string
DateManufactured            0/0/0
StartupCompatibility        null string

RedGamma                    1.000
GreenGamma                  1.000
BlueGamma                   1.000
WhiteGamma                  1.000
```

If a Gamma parameter is specified as 1 then the gamma function is interpreted to be linear (i.e. signal-input amplitude verses display luminance is linear.)

This is not the common characteristic for a CRT monitor.

5. Section Contents:

The entire ASCII file is case insensitive (except for the characters specified within strings). It doesn't matter if uppercase or lowercase characters are used for any of the reserved words.

For example DATE, Date, date, dATE are all parsed the same.

Comments may appear anywhere in the file. Following C++ language conventions, comments begin with a // and continue to the end of the current line.

The end of line is denoted by a <CR> and/or <LF> character. That is, the combinations <CR>, <CR><LF>, <LF><CR> and <LF> are all valid end of line markers.

A semicolon <;> character is used to indicate the end of a parameter field.

Note that in the [MONITOR_DESCRIPTION] section some parameters depend upon whether the monitor is color or monochrome. For ease of reading, these "MonitorType" dependent parameters have been partitioned into two groups, color and monochrome. As noted in the following, the lines that partition these groups are not part of the ASCII file.

```
[VERSION]
    <integer>.<integer>;                // version.revision


[MONITOR_DESCRIPTION]
    Date                 = <date>;       // yyyy/mm/dd
    FileRevision         = <string>;
    Manufacturer         = <string>;
    ModelNumber          = <string>;
    MinVDIFIndex         = <string>;
    Version              = <string>;
    SerialNumber         = <string>;
    DateManufactured     = <date>;
    MonitorType          = COLOR | COLOUR | MONOCHROME;
    CRTSize              = <integer>;   // inches

    If Color Monitor:  (this line is not part of the ASCII file)

    RedPhosphorDecay     = <scaled>;    // milliseconds
    GreenPhosphorDecay   = <scaled>;    // milliseconds
    BluePhosphorDecay    = <scaled>;    // milliseconds
    BorderRed            = <integer>;   // percent
    BorderGreen          = <integer>;   // percent
    BorderBlue           = <integer>;   // percent
    WhitePointx          = <scaled>;    // in CIExyY (cd / m2)
    WhitePointy          = <scaled>;
    WhitePointCapY       = <scaled>;
    RedChromaticityx     = <scaled>;
    RedChromaticityy     = <scaled>;
    GreenChromaticityx   = <scaled>;
    GreenChromaticityy   = <scaled>;
    BlueChromaticityx    = <scaled>;
    BlueChromaticityy    = <scaled>;
    RedGamma             = <scaled>;
    GreenGamma           = <scaled>;
    BlueGamma            = <scaled>;

    If Monochrome Monitor:  (this line is not part of the ASCII file)

    Phosphor1Decay       = <scaled>;   // milliseconds
    Phosphor2Decay       = <scaled>;   // milliseconds, optional
    Phosphor3Decay       = <scaled>;   // milliseconds, optional
    Border               = <integer>;   // percent
    WhitePointx          = <scaled>;    // in CIExyY (cd / m2)
    WhitePointy          = <scaled>;
    WhitePointCapY       = <scaled>;
    Phosp1Chromaticityx  = <scaled>;    // optional information
    Phosp1Chromaticityy  = <scaled>;    // optional information
    Phosp2Chromaticityx  = <scaled>;    // optional information
    Phosp2Chromaticityy  = <scaled>;    // optional information
    Phosp3Chromaticityx  = <scaled>;    // optional information
    Phosp3Chromaticityy  = <scaled>;    // optional information
    Gamma                = <scaled>;    // optional information


[OPERATIONAL_LIMITS]
    MaxHorPixel           = <integer>;     // pixels
    MaxVerPixel           = <integer>;     // lines
    MaxHorAddrLength      = <integer>;     // millimeters
    MaxVerAddrHeight      = <integer>;     // millimeters
    VideoType             = ANALOG | TTL | ECL | DECL | OTHER ;
    TerminationResistance = <integer>
    WhiteLevel            = <scaled>;      // Volts
    BlackLevel            = <scaled>;      // Volts
```

```
    BlankLevel            = <scaled>;      // Volts
    SyncType           = ANALOG | TTL | ECL | DECL | OTHER;
    SyncConfiguration  = SEPARATE | C | CP | G | GP | OTHER;
    SyncLevel          = <scaled>;      // Volts
    MaxPixelClock      = <scaled>;      // MegaHertz
    MinHorFrequency    = <scaled>;      // kiloHertz
    MaxHorFrequency    = <scaled>;      // kiloHertz
    MinVerFrequency    = <scaled>;      // Hertz
    MaxVerFrequency    = <scaled>;      // Hertz
    MinHorRetrace      = <scaled>;      // microseconds
    MinVerRetrace      = <scaled>;      // milliseconds


[PREADJUSTED_TIMING]
    Preadjustedtimingname = <string>;
    HorPixel          = <integer>;      // pixels
    VerPixel          = <integer>;      // lines
    HorFrequency      = <scaled>;       // kiloHertz
    VerFrequency      = <scaled>;       // Hertz
    PixelClock        = <scaled>;       // MegaHertz
    CharacterWidth    = <integer>;      // pixel clock cycles
    HorAddrLength     = <integer>;      // millimeters
    VerAddrHeight     = <integer>;      // millimeters
    PixelWidthRatio   = <integer>;      // gives H:V
    PixelHeightRatio  = <integer>;
    ScanType          = INTERLACED | NONINTERLACED | OTHER;
    HorSyncPolarity   = NEGATIVE | POSITIVE;
    VerSyncPolarity   = NEGATIVE | POSITIVE;
    HorTotalTime      = <scaled>;       // microseconds
    HorAddrTime       = <scaled>;       // microseconds
    HorBlankStart     = <scaled>;       // microseconds
    HorBlankTime      = <scaled>;       // microseconds
    HorSyncStart      = <scaled>;       // microseconds
    HorSyncTime       = <scaled>;       // microseconds
    VerTotalTime      = <scaled>;       // milliseconds
    VerAddrTime       = <scaled>;       // milliseconds
    VerBlankStart     = <scaled>;       // milliseconds
    VerBlankTime      = <scaled>;       // milliseconds
    VerSyncStart      = <scaled>;       // milliseconds
    VerSyncTime       = <scaled>;       // milliseconds



[GAMMA_TABLE_ENTRIES]
    <integer>;


[GAMMA_TABLE]
    <integers>
```

6. Gamma Table Details:

The optional gamma table section allows one to specify a table of the display-luminance output measurements for the monitor. The information in this section is normally obtained from a calibration device that measures display-luminance. The intent is that the gamma table would be added automatically by a software program controlling the calibration device. Use of this information provides precise gamma correction and the ability of mapping the monitor's precise color gamut to and from device independent color spaces.

It should be noted that this table describes the relation between the monitor input voltage and the display-luminance output from the monitor. In case the monitor is connected to a controller that has a non linear transfer function or that is not producing exactly the voltage level specified in this file, that has to be factored in when using the gamma table.

The optional gamma table consists of two parts, the first part lists the number of entries in the table and the second part is the table itself.

The syntax of the first part is simply:

```
[GAMMA_TABLE_ENTRIES]

      N;
```

where N is of type <integer> with a range of 0..65535.

The syntax for the table is dependent upon whether the monitor is monochrome or color. For monochrome monitors the syntax is:

```
[GAMMA_TABLE]

      gray0,
      gray1,
      ...
      grayN-1
```

where gray*i* is of type <integer> with a range of 0..65535.

For color monitors the syntax is:

```
[GAMMA_TABLE]

      r0,   g0,   b0,
      r1,   g1,   b1,
      ...   ...   ...
      rN-1, gN-1, bN-1
```

where r*i*, g*i*, b*i* correspond to the red, green and blue components and are of the type <integer> with a range
of 0..65535.

The gamma table is a list of 16-bit unsigned integers (or 3 16-bit unsigned integers for RGB) that define a curve for each primary color. These curves describes the relationship of the signal-input amplitude to the display-luminance output and may have any shape. The number of entries, N, are equally spaced steps of signal-input amplitude starting with no signal at 0 and completing with max. signal at N-1. The display-luminance output value of no intensity, or black, is 0 (or 0,0,0) and the value of full intensity, or white, is 65535 (or 65535,65535,65535).

Appendix B

Binary file format

The binary format directly corresponds to the ASCII format. It has some extension and special semantics in order to fulfill the needs of a wide area of applications (e.g. device drivers and BIOS).

The bit & byte order of the structures are compatible with a little endian architecture, most significant bit first bit order and least significant byte first byte order. The structures are aligned so that they will work with 1/2/4 byte alignment. The fundamental types used are:

unchar	8 bit unsigned integer
ushort	16 bit unsigned integer
ulong	32 bit unsigned integer

Currently there is one optional section, VDIFGammaRec. This is the gamma table. This structure is followed by GammaTableEntries (MONO) or GammaTableEntries * 3 (COLOR) unsigned shorts which represent the gamma table. For color monitors, the order of the RGB values are the same as in the ASCII format, namely r0,g0,b0, r1,g1,b1, ...

All strings are composed of 7-bit ASCII characters terminated with the NULL character (i.e. zero).

The different sections are linked together with pointers encoded as binary offsets. All offsets are relative to the current section and all string offset are relative to the beginning of the string table.
Note that end-of-list marks/tags are not provided. Rather, fields containing counts are provided to specify how many sections are in a list.

The checksum is the sum of all bytes AFTER the checksum field. The whole file should be treated as a series of bytes when it comes to summing the file (regardless whether the structure fields are longs, shorts, or bytes).

The size of the binary file can be calculated by:

```
size = 1      * 116 bytes   +  //      size of(_VDIF       structure)
       M      *  64 bytes   +  //  M * size of(_VDIFLimits structure)
       N      *  68 bytes   +  //  N * size of(_VDIFTiming structure)
    (0 or 1) * (8 + n * 2) +  //  optional gamma table with n entries
       size of string table   //  string table size is variable
```

The minimum size binary file has only one (M=1) _VDIFLimits and one (N=1) _VDIFTiming structure and would therefore be 248 (i.e. 116+64+68) bytes plus the size of the string table. The string table varies in size since it contains all of the strings in the VDIF file. It's size is simply the sum of the lengths of all of the strings in the VDIF file.

The following C structures define the VDIF binary format:

```
#define VDIF_MONITOR_MONOCHROME 0
#define VDIF_MONITOR_COLOR      1

#define VDIF_VIDEO_TTL          0
#define VDIF_VIDEO_ANALOG       1
#define VDIF_VIDEO_ECL          2
#define VDIF_VIDEO_DECL         3
#define VDIF_VIDEO_OTHER        4

#define VDIF_SYNC_SEPARATE      0
#define VDIF_SYNC_C             1
#define VDIF_SYNC_CP            2
#define VDIF_SYNC_G             3
#define VDIF_SYNC_GP            4
#define VDIF_SYNC_OTHER         5

#define VDIF_SCAN_NONINTERLACED 0
#define VDIF_SCAN_INTERLACED    1
#define VDIF_SCAN_OTHER         2

#define VDIF_POLARITY_NEGATIVE  0
#define VDIF_POLARITY_POSITIVE  1


typedef struct _VDIF {                 /* Monitor Description:             */
  unchar        VDIFId[4];             /* alway "VDIF"                     */
  ulong         FileLength;            /* lenght of the whole file         */
  ulong         Checksum;              /* sum of all bytes in the file after*/
                                       /* this feeld                       */
  ushort        VDIFVersion;           /* structure version number         */
  ushort        VDIFRevision;          /* structure revision number        */
  ushort        Date[3];               /* file date Year/Month/Day         */
  ushort        DateManufactured[3];   /* date Year/Month/Day              */
  ulong         FileRevision;          /* file revision string             */
  ulong         Manufacturer;          /* ASCII ID of the manufacturer     */
  ulong         ModelNumber;           /* ASCII ID of the model            */
  ulong         MinVDIFIndex;          /* ASCII ID of Minimum VDIF index   */
  ulong         Version;               /* ASCII ID of the model version    */
  ulong         SerialNumber;          /* ASCII ID of the serial number    */
  unchar        MonitorType;           /* Monochrome or Color              */
  unchar        CRTSize;               /* inches                           */
  unchar        BorderRed;             /* percent                          */
  unchar        BorderGreen;           /* percent                          */
  unchar        BorderBlue;            /* percent                          */
  unchar        Reserved1;             /* padding                          */
  ushort        Reserved2;             /* padding                          */
  ulong         RedPhosphorDecay;      /* microseconds                     */
  ulong         GreenPhosphorDecay;    /* microseconds                     */
  ulong         BluePhosphorDecay;     /* microseconds                     */
  ushort        WhitePoint_x;          /* WhitePoint in CIExyY (scale 1000) */
  ushort        WhitePoint_y;
  ushort        WhitePoint_Y;
  ushort        RedChromaticity_x;     /* Red chromaticity in x,y          */
  ushort        RedChromaticity_y;
  ushort        GreenChromaticity_x;   /* Green chromaticity in x,y        */
  ushort        GreenChromaticity_y;
  ushort        BlueChromaticity_x;    /* Blue chromaticity in x,y         */
  ushort        BlueChromaticity_y;
  ushort        RedGamma;              /* Gamme curve exponent (scale 1000) */
  ushort        GreenGamma;
```

```
  ushort        BlueGamma;
  ulong         NumberOperationalLimits;
  ulong         OffsetOperationalLimits;
  ulong         NumberOptions;          /* optinal sections (gamma table)    */
  ulong         OffsetOptions;
  ulong         OffsetStringTable;
} VDIFRec;

typedef enum {                          /* Tags for section identification   */
  VDIF_OPERATIONAL_LIMITS_TAG = 1,
  VDIF_PREADJUSTED_TIMING_TAG,
  VDIF_GAMMA_TABLE_TAG
} VDIFScnTag;

typedef struct _VDIFScnHdr {            /* Generic Section Header:           */
  ulong         ScnLength;              /* lenght of section                 */
  ulong         ScnTag;                 /* tag for section identification    */
} VDIFScnHdrRec;

typedef struct _VDIFLimits {            /* Operational Limits:               */
  VDIFScnHdrRec Header;                 /* common section info               */
  ushort        MaxHorPixel;            /* pixels                            */
  ushort        MaxVerPixel;            /* lines                             */
  ushort        MaxHorActiveLength;     /* millimeters                       */
  ushort        MaxVerActiveHeight;     /* millimeters                       */
  unchar        VideoType;              /* TTL / Analog / ECL / DECL         */
  unchar        SyncType;               /* TTL / Analog / ECL / DECL         */
  unchar        SyncConfiguration;      /* separate / composite / other      */
  unchar        Reserved1;              /* padding                           */
  ushort        Reserved2;              /* padding                           */
  ushort        TerminationResistance;  /*                            */
  ushort        WhiteLevel;             /* millivolts                        */
  ushort        BlackLevel;             /* millivolts                        */
  ushort        BlankLevel;             /* millivolts                        */
  ushort        SyncLevel;              /* millivolts                        */
  ulong         MaxPixelClock;          /* kiloHertz                         */
  ulong         MinHorFrequency;        /* Hertz                             */
  ulong         MaxHorFrequency;        /* Hertz                             */
  ulong         MinVerFrequency;        /* milliHertz                        */
  ulong         MaxVerFrequency;        /* milliHertz                        */
  ushort        MinHorRetrace;          /* nanoseconds                       */
  ushort        MinVerRetrace;          /* microseconds                      */
  ulong         NumberPreadjustedTimings;
  ulong         OffsetNextLimits;
} VDIFLimitsRec;

typedef struct _VDIFTiming {            /* Preadjusted Timing:               */
  VDIFScnHdrRec Header;                 /* common section info               */
  ulong         PreadjustedTimingName;  /* SVGA/SVPMI mode number            */
  ushort        HorPixel;               /* pixels                            */
  ushort        VerPixel;               /* lines                             */
  ushort        HorAddrLength;          /* millimeters                       */
  ushort        VerAddrHeight;          /* millimeters                       */
  unchar        PixelWidthRatio;        /* gives H:V                         */
  unchar        PixelHeightRatio;
  unchar        Reserved1;              /* padding                           */
  unchar        ScanType;               /* noninterlaced / interlaced / other*/
  unchar        HorSyncPolarity;        /* negative / positive               */
  unchar        VerSyncPolarity;        /* negative / positive               */
  ushort        CharacterWidth;         /* pixels                            */
  ulong         PixelClock;             /* kiloHertz                         */
  ulong         HorFrequency;           /* Hertz                             */
  ulong         VerFrequency;           /* milliHertz                        */
  ulong         HorTotalTime;           /* nanoseconds                       */
```

```
  ulong         VerTotalTime;           /* microseconds                         */
  ushort        HorAddrTime;            /* nanoseconds                          */
  ushort        HorBlankStart;          /* nanoseconds                          */
  ushort        HorBlankTime;           /* nanoseconds                          */
  ushort        HorSyncStart;           /* nanoseconds                          */
  ushort        HorSyncTime;            /* nanoseconds                          */
  ushort        VerAddrTime;            /* microseconds                         */
  ushort        VerBlankStart;          /* microseconds                         */
  ushort        VerBlankTime;           /* microseconds                         */
  ushort        VerSyncStart;           /* microseconds                         */
  ushort        VerSyncTime;            /* microseconds                         */


} VDIFTimingRec;

typedef struct _VDIFGamma {             /* Gamma Table:                         */
  VDIFScnHdrRec Header;                 /* common section info                  */
  ushort        GammaTableEntries;      /* count of grays or RGB 3-tuples       */
  ushort        Unused1;
} VDIFGammaRec;

/* access macros */
#define VDIF_OPERATIONAL_LIMITS(vdif) \
        ((VDIFLimitsPtr)((char*)(vdif) + (vdif)->OffsetOperationalLimits))

#define VDIF_NEXT_OPERATIONAL_LIMITS(limits) \
        ((VDIFLimitsPtr)((char*)(limits) + (limits)->OffsetNextLimits))

#define VDIF_PREADJUSTED_TIMING(limits) \
        ((VDIFTimingPtr)((char*)(limits) + (limits)->Header.ScnLength))

#define VDIF_NEXT_PREADJUSTED_TIMING(timing) \
        ((VDIFTimingPtr)((char*)(timing) + (timing)->Header.ScnLength))

#define VDIF_OPTIONS(vdif) \
        ((VDIFScnHdrPtr)((char*)(vdif) + (vdif)->OffsetOptions))

#define VDIF_NEXT_OPTIONS(options) \
        ((VDIFScnHdrPtr)((char*)(options) + (options)->ScnLength))

#define VDIF_STRING(vdif, string) \
        ((char*)((char*)vdif + vdif->OffsetStringTable + (string)))
```

A graphical representation of the binary VDIF format follows on the next page:

```
vdif         +-------------------------------+
             | FileLength                    |---------,
             |                               |         |
             |                               |         |
             |                               |         |
             | OffsetOperationalLimits       |-,       |
             | OffsetOptionalSections        |-+---,   |
             | OffsetStringTable             |-+---+-, |
             +-------------------------------+ |   | | |
                                             <-'   | | |
limits 1     +-------------------------------+     | | |
           ,-| ScnLength                     |     | | |
           | |                               |     | | |
           | |       OffsetNextLimits        |---, | | |
           | +-------------------------------+   | | | |
           '->                                   | | | |
            timing 1a +----------------------+   | | | |
                      |ScnLength             |-, | | | |
                      |                      | | | | | |
                      |                      | | | | | |
                      +----------------------+ | | | | |
                                             <-' | | | |
            timing 1b +----------------------+   | | | |
                      |ScnLength             |-, | | | |
                      |                      | | | | | |
                      |                      | | | | | |
                      +----------------------+ | | | | |
                                             <-' | | | |
                     ... more timings ...        | | | |
                                               <-' | | |
limits 2     +-------------------------------+     | | |
           ,-| ScnLength                     |     | | |
           | |                               |     | | |
           | |       OffsetNextLimits        |---, | | |
           | +-------------------------------+   | | | |
           '->                                   | | | |
            timing 2a +----------------------+   | | | |
                      |ScnLength             |-, | | | |
                      |                      | | | | | |
                      |                      | | | | | |
                      +----------------------+ | | | | |
                                             <-' | | | |
            timing 2b +----------------------+   | | | |
                      |ScnLength             |-, | | | |
                      |                      | | | | | |
                      |                      | | | | | |
                      +----------------------+ | | | | |
                                             <-' | | | |
                     ... more timings ...        | | | |
                                               <-' | | |
                     ... more limits ...           | | |
                                                   | | |
                                             <-----' | |
optional     +-------------------------------+       | |
          ,- | ScnLength                     |       | |
          |  |                               |       | |
          |  +-------------------------------+       | |
          '->                                        | |
                     ... more options ...            | |
                                             <-------' |
strings      +-------------------------------+         |
             | Length unimportant            |         |
             |                               |         |
             +-------------------------------+         |
```

<---------'