Video Electronics Standards Association

VESA DisplayID Standard

Version 2.1

18 November, 2021

vesa.org

Purpose

This Standard defines flexible data formats that organize interface and display configuration information in a structure stored within a display product. A host has access to the structure over the video interface connection. The host uses this data to automatically setup and optimize the video interface and image signals while facilitating plug and play operation with minimal or no user intervention.

Summary

DisplayID v2.1 describes the second-generation version of the VESA *DisplayID Standard*. Use of DisplayID structures is intended to eventually replace use of EDID structures to describe capabilities of new display devices. *DisplayID v2.1* contains several new features that better allow its use in a wide range of applications, including PC monitors, consumer television products, embedded displays (e.g., laptop LCD panels), AR/VR headsets, and other display products. The DisplayID structure definition is intended to be a non-ambiguous, simple, and compact data declaration that provides information about modern displays that are used by the system. This information includes model and specific unit identification information, colorimetry, color depth, color space, native gamma EOTF, audio feature support, supported timings, and supported formats that are to be supplied to the host video source over an appropriate communications channel (e.g., VESA *E-DDC*). The Source device may then use this information to automatically configure itself to optimally support the display in use.

Notes: *DisplayID v2.1 retains the same structure as DisplayID v2.0, but has been updated with new data blocks that are designed to better accommodate modern display mandates and capabilities.*

*DisplayID structures are **not** directly backward-compatible with previous EDID structure definitions, but do include many of the same and/or similar data field definitions from those previous Standards. In many cases, similar data fields have been extended to provide greater magnitude and/or precision than was possible with EDID.*

*DisplayID structure Extension Sections may be used within EDID Extension Blocks to help facilitate backward compatibility and accommodate devices that are **not** yet capable of directly supporting DisplayID structures (see Section 2.1).*

Contents

Tables

Figures

Preface

Intellectual Property

While every precaution has been taken in the preparation of this Standard, the Video Electronics Standards Association and its contributors assume no responsibility for errors or omissions, and make no warranties, expressed or implied, of functionality or suitability for any purpose.

Trademarks

DisplayPort is a trademark and VESA is a registered trademark of the Video Electronics Standards Association.

CTA is a trademark of the Consumer Technology Association.

CIE is a trademark of the International Commission on Illumination.

DVI is a trademark of the Digital Display Working Group.

HDMI and High-Definition Multimedia Interface are trademarks or registered trademarks of HDMI Licensing Administrator, Inc. in the United States and other countries.

I^2C is a trademark of NXP Semiconductors (formerly Philips Semiconductors).

All other trademarks, registered trademarks, or servicemarks used within this Standard are the property of their respective owners.

Patents

VESA® draws attention to the fact that compliance with this Standard might involve the use of a patent or other intellectual property right (collectively, "IPR") concerning DisplayID. VESA takes no position concerning the evidence, validity, and/or scope of this IPR. At the time of publication, there are currently no IPR specific to DisplayID to list in this Standard.

Attention is drawn to the possibility that some of the elements of this VESA Standard might be the subject of IPR external to this Standard other than those identified above. VESA shall not be held responsible for identifying any or all such IPR, and has made no inquiry into the possible existence of any such IPR.

THIS STANDARD IS BEING OFFERED WITHOUT ANY WARRANTY WHATSOEVER, AND IN PARTICULAR, ANY WARRANTY OF NON-INFRINGEMENT IS EXPRESSLY DISCLAIMED. ANY IMPLEMENTATION OF THIS STANDARD SHALL BE MADE ENTIRELY AT THE IMPLEMENTER'S OWN RISK, AND NEITHER VESA, NOR ANY OF ITS MEMBERS OR SUBMITTERS, SHALL HAVE ANY LIABILITY WHATSOEVER TO ANY IMPLEMENTER OR THIRD PARTY FOR ANY DAMAGES OF ANY NATURE WHATSOEVER DIRECTLY OR INDIRECTLY ARISING FROM THE IMPLEMENTATION OF THIS STANDARD.

Support for this Standard

Clarifications and application notes to support this Standard may have been written. To obtain the latest Standard and any support documentation, contact VESA.

If you have a product that incorporates DisplayID, ask the company that manufactured your product for assistance. If you are a manufacturer, VESA can assist you with any clarification you might need.

Submit all comments or reported errors to support@vesa.org.

Acknowledgments

This Standard would not have been possible without the efforts of VESA's DisplayID Task Group. In particular, Table 1 lists the individuals and their companies that contributed significant time and knowledge to this version of the Standard.

Table 1: Main Contributors to DisplayID v2.1

Company	Name	Contribution
Advanced Micro Devices, Inc.	Bruce Cui	Technical Contributor
	Aric Cyr	Technical Contributor
	Syed A. Hussain	Task Group Vice Chair and Document Editor
	Shaylesh Mehta	Technical Contributor
Analog Devices, Inc.	Gyudong Kim	Technical Contributor
Analogix Semiconductor, Inc.	Mehran Badii	Technical Contributor
	Greg Stewart	Technical Contributor
Apple, Inc.	Bob Ridenour	Task Group Chair
	Alan Kobayashi	Technical Contributor
Avatar Tech Pubs	Trish McDermott	Technical Writer
Broadcom Corp.	Chris Pasqualino	Technical Contributor
Dell, Inc.	Van Martin	Technical Contributor
	Pavel Olchovik	Technical Contributor
EIZO Corp.	Yoshinobu Bamba	Technical Contributor
Innolux Corp.	Jeff Chen	Technical Contributor
Intel Corp.	Chandra Konduru	Technical Contributor
JVC Kenwood Corp.	Takayuki Kashahara	Technical Contributor
LG Electronics, Inc.	DK (Do Kyun) Kim	Technical Contributor
	Seunghyun Yoo	Technical Contributor
LuxShare ICT	Daniel Chen	Technical Contributor
MediaTek, Inc.	Tung-Sheng Lin	Technical Contributor
Microsoft Corp.	Roger Coote	Technical Contributor
	Norm LeMieux	Technical Contributor

Table 1: Main Contributors to DisplayID v2.1 (Continued)

Company	Name	Contribution
NVIDIA Corp.	Kai Chen	Technical Contributor
	Robert Schutten	Technical Contributor
Parade Technologies, Ltd.	Craig Wiley	Technical Contributor
RealTek Semiconductor Corp.	Jay Lin	Technical Contributor
Samsung Electronics Co., Ltd.	Dale Stolitzka	Technical Contributor
Simula Technology, Inc.	Alice Yu	Technical Contributor
Sony Corp.	Robert Blanchard	Technical Contributor
	Peter Shintani	Technical Contributor
Teledyne LeCroy	Jeff Stenhouse	Technical Contributor
VESA Compliance Program Manager	Jim Choate	Technical Contributor

Revision History

Table 2: Revision History

Date	Version	Description
November 18, 2021	2.1	• Added changes made to version 2.0 of this Standard through ratification of the following DisplayID v2.0 SCRs: • *Adaptive-Sync Data Block Definition (update after 2nd adoption)* (Includes new Adaptive-Sync data block) • *Appendix A Color Corrections* • *Appendix A Color Signaling Capabilities* • *Appendix A Detailed Timing Corrections* • *Color Depth Mode Support and YCbCr 420 clarification* • *Created a new type X timing data block to support CVT RB v3 and supportrefresh rate field up to 1024 Hz (update after 2nd adoption)* (Includes new Type X Formula-based Timing data block) • *Dark Theme preference with exposure of optional Display DeviceTheme Preference information/DisplayID v2.0 SCR_Display Device Theme Preference/Display Device Power Characteristics* • *DSC pass-through timing support* • *Mandatory DisplayID Data Blocks and Structure Coexistence* • *Maximum Vertical Refresh Rate field extension of Dynamic Video Timing Range Limit Data Block* • *Missing Data Block Payload Length in Table B-1* • *SCR_DisplayIDv2.0_Dynamic Video Timing Range Limits Data Block_r2_proposal_AMD* • *Support DisplayID version 2 section as a base EDID extension* • *Timing Formula Field Clarification in Type IX Timing Descriptor* • *Updated Type VII timing data block to add future extensibility and accommodate CTA request to explicitly indicate YCC 420 support in each timing descriptor* • *Updated Type VIII Timing data block to accommodate CTA request to explicitly indicate YCC 420 support.* • Added changes made to version 2.0 of this Standard through the following proposed SCR: • *DisplayID V2.0 SCR ARVR Related Data Blocks* (Includes new ARVR_HMD and ARVR_Layer data blocks) • *SCR_DisplayIDv2.0_Dynamic Video Timing Range Limits Data Block_r2_proposal_AMD*

Table 2: Revision History (Continued)

Date	Version	Description

November 5, 2021	2.1 d9	**Continued:**
		• Global:
		  • Updated frontmatter specific to *DisplayID v2.1*
		  • Clarified **optional** and **mandatory** data blocks
		  • Removed the Flag and Tag columns from any tables that had them
		  • Removed "Detailed" from any Timing descriptor descriptions that are not EDID Detailed Timing descriptor blocks (*CTA-861-B* or higher) or Type VII
		  • Renamed "CTA DisplayID data block" to "CTA-861 Data Block Encapsulation data block"
		  • Streamlined how Section, Base Section, Extension Sections, EDID, structure, and CVT standard and reduced blanking timings are mentioned, and how data block and field tables are introduced
		  • Added clarification that references to some specified ranges that differ from others are intentional
		  • Removed mention of "format" with respect to data block and field definitions
		  • Applied minor grammatical edits, corrected typos, and VESA template style changes
		• Table 1-4 – New, Unit of Measure Symbols table
		• Section 3 – Reworded 2nd sentence of first paragraph
		• Table 3-3 – Added Adaptive-Sync data block, removed Dynamic Video Timing Range Limits data block
		• Tables 4-31 and 4-32 – 4195: Clarified "*N*" number of bytes
		• Table 4-41 – 4183: Removed extraneous Offset 03h[4:0] row/entry
		• Table 4-57 – Corrected the name of Offset 00h; Replaced all offsets after 03h (removed descriptors and showed payload instead)
		• Section 4.3.1.1 – 4036: Removed "Support for Interlaced Video Timing Modes" subsection; subsequent subsections renumbered accordingly
		• Section 4.6 – 4143: Replaced 2nd and 3rd paragraphs/removed reference to data block Revision 1
		• Table A-1 – Updated Offset 4Ch to include new bits 6:4, and updated its RESERVED bit information; Updated checksum value
		• Table B-1 – Clarified Offset 07h information related to Multi-SST operation and external *DisplayPort*; Table B-1 – 4201: Updated payload length definition in VESA Organization Vendor-specific Data Block (Offset 02h[7:0])

September 11, 2017	2.0	Initial release.

1 Introduction (Informative)

1.1 Summary

This Standard, henceforth referred to as *DisplayID v2.1*, defines a framework and data structures for describing physical and performance attributes and supported timings. *DisplayID v2.1* also provides other relevant information regarding a display product to enable the video source, and driving that display to automatically configure itself for optimum display usage. At least one variable-length data structure (i.e., the "DisplayID Base" or "DisplayID Base Section") shall be provided under this system to provide basic identification, features, supported timings, and other information regarding a given display product to its host and/or video source device.

A method shall also be described for extending this basic information set through additional variable-length structures, referred to as "DisplayID Extension Sections," that comprise various forms of data blocks. The DisplayID data structure shall be independent of the communication protocol (i.e., video interface) that is used between the host and display product.

1.2 Background and Changes from Previous EDID Definitions

Note: *The following is provided as an informative summary only; this section does not establish mandates and/or definitions under DisplayID v2.1.*

The VESA Extended Display Identification Data (*E-EDID*) Standard (globally referenced herein as *EDID*), first introduced in 1994, established the most widely used data format for conveying display identification and description information to video sources (e.g., PCs and set-top boxes), which permitted such devices to configure themselves for optimum display use. *EDID* continues to be widely used within the display industry, particularly for computer displays and digital television, despite an implicit assumption in the EDID Base Section definition that the display is a direct-view CRT type.

DisplayID v2.1, intended as a true second-generation *EDID*, defines a more-flexible and extensible display identification data format that is designed to meet the needs of a wide range of display types, technologies, and applications. The most obvious difference between this Standard and its *EDID* predecessors is the use of a modular structure that is based on the concept of "data blocks," individually defined and self-contained data formats that each provide a specific set of related display information.

Data block definitions may be modified or added to the overall DisplayID system, as needed. This Standard also allows for data blocks to be defined by other organizations, in particular *CTA*. In addition, these data blocks may, within certain restrictions, be "mixed and matched," as needed, throughout the overall DisplayID structure. Use of this system also means that there are no longer separately defined and unique extensions, as was the case with *EDID*. Like the DisplayID Base Section, DisplayID Extension Sections are created, as needed, from the defined data blocks.

A number of data block definitions have been developed to support *DisplayID v2.1*. In addition, the DisplayID system expects to leverage CTA-861 data blocks, previously developed for use in the CTA EDID Extension as defined by *CTA-861-H*. These are included using the CTA-861 Data Block Encapsulation DisplayID data block (see Section 4.12). With the range of available data blocks, it is expected that the basic DisplayID system shall be applicable to desktop monitors, television monitors, receivers, and display devices (e.g., providing ID information on the LCD panel used in a laptop), as well as non-display devices that might be connected to video sources (e.g., repeaters or interface-translator products).

1.3 Objectives

VESA developed this Standard to meet, exceed, and/or complement certain criteria to provide information in a compact format that can allow a graphics subsystem to be configured based on the attached display's capabilities.

1.4 Document Organization

This Standard is organized as follows:

- Section 1 – Introduction (Informative)
- Section 2 – DisplayID Structure
- Section 3 – Data Blocks – Definitions and Use
- Section 4 – Data Block Definitions
- Appendix A – Fixed-length DisplayID Section Example (Informative)
- Appendix B – VESA Organization Vendor-specific Data Block (Normative)
- Appendix C – Scan Orientation Clarification (Informative)
- Appendix D – Main Contributor History (Previous Revisions)

1.5 Document Conventions

1.5.1 Precedence

If there is a conflict between text, figures, and tables, the precedence shall be tables, figures, and then text.

1.5.2 Keywords

Table 1-1 lists keywords that differentiate between the levels of mandates and options within this Standard.

Table 1-1: Keywords

Keyword	Definition
informative	Information that discusses and clarifies mandates and features.
may	Indicates a choice that does not have an implied preference.
N/A	Indicates that a field or value is **not** applicable, does not have a defined value, and shall **not** be checked or used by the recipient.
normative	Features that are mandated by this Standard.
optional	Features that are **not** mandated by this Standard. However, if an **optional** feature is implemented, the feature shall be implemented as defined by this Standard (**optional** normative).
RESERVED	Indicates RESERVED bits, bytes, words, fields, and code values that are set aside for future standardization. Their use and interpretation may be specified by future extensions to this Standard and, unless otherwise stated, shall **not** be used or adapted by vendor implementation. A RESERVED bit, byte, word, or field shall be cleared by the transmitter and if received, ignored by the receiver.
shall	Indicates a mandate. Designers shall implement all such mandates to ensure interoperability with other compliant devices.
should	Indicates flexibility of choice with a preferred alternative. Equivalent to the phrases “is recommended” and “it is recommended that.”

1.5.3 Numbering

Table 1-2 lists the different types of numbering that is used within this Standard.

Table 1-2: Numbering

Numbering Format	Definition
Binary	Numbers that are immediately followed by a lowercase “b” (e.g., 01b). Also used for individual bit values (i.e., 0, 1).
Decimal	Numbers that are not immediately followed by either a “b” or “h” (e.g., 5).
Hexadecimal	Numbers that are immediately followed by a lowercase “h” (e.g., 10h).
	Numbers that start with “0x” (e.g., 0x00).

1.5.4 Data Format Conventions

The DisplayID structure is designed to be compact in its representation of data to fit the most information into a limited space. To accomplish this, variable data lengths are used according to the particular element's needs. These include fields from a single bit up to 256 bytes in length. Except where explicitly stated, the conventions listed in Table 1-3 shall be used in all cases.

Table 1-3: Data Format Conventions

Data Length	Convention Used	Example
1 to 7 bits	Stored in the order stated.	
8 bits (1 byte)	Stored at the location stated.	
16 bits (2 bytes)	Bytes are in a binary format (not binary-coded decimal format) that is stored at locations that are specified with the least significant byte (LSB) in the first location.	1,280 decimal = 0500h. Stored 00h at the first location and 05h at the next location.
Character string	Bytes are ASCII, stored in the order in which they appear within the string.	"ACED" stored 41h at the first location, 43h at the second location, 45h at the third location, and 44h at the fourth location.

1.5.5 Unit of Measure Symbols

Table 1-4 lists unit of measure symbols that are used throughout this Standard.

Table 1-4: Unit of Measure Symbols

Symbol	Unit of Measure	Symbol	Unit of Measure
bpc	bits per (color) component	ms	millisecond
cd/m^2	candela per square meter (formerly referred to as "nits")	us	microsecond
cm	centimeter	Mbps	megabits per second
Hz	hertz	MP/s	MegaPixels/second
kHz	kilohertz	%	percent
mm	millimeter	ps	picosecond

1.5.6 Acronyms and Abbreviations

Table 1-5 lists acronyms and abbreviations that are used throughout this Standard. Capitalization is applied in the definition to indicate the letters used in acronyms and abbreviations.

Table 1-5: Acronyms and Abbreviations

Term	Definition
ARVR AR/VR	Augmented Reality and/or Virtual Reality
AVT	Adaptive VTotal
CTA	Consumer Technology Association (*CTA-861*)
CTS	Compliance Test Specification
CVT	Coordinated Video Timings (*CVT Standard*)
DisplayID	Display Identification Data (this Standard)
DP	*DisplayPort* (*DP Standard*)
EDID	Extended Display Identification Data (legacy VESA structure, superseded by the DisplayID structure; *E-EDID Standard*)
E-EDID	Enhanced Extended Display Identification Data (legacy VESA structure, superseded by the DisplayID structure; *E-EDID Standard*)
FAVT	Fixed-Average VTotal
FOV	Field Of View
GPU	Graphics Processing Unit
HBlank	Horizontal Blank
HMD	Head Mounted Display
IEEE	Institute of Electrical and Electronics Engineers
IPD	InterPupillary Distance
LSB	Least Significant Byte
MSB	Most Significant Byte
OUI	Organizational Unique ID
RB	Reduced Blanking (*CVT Standard*)
SST	Single-Stream Transport
VBI	Vertical Blanking Interval
VTotal	Vertical Total
VESA	Video Electronics Standards Association

1.5.7 Glossary

This Standard uses slightly different terminology to refer to the overall DisplayID structure's components than previous *EDID* Standards. Table 1-6 defines several key terms.

Table 1-6: Glossary

Term	Definition
base video frame period	Frame duration of a timing as specified by the timing descriptor. Adaptive-Sync operation may be used to extend the video frame period.
chromatic aberration	Caused by the optical property of a lens that varies with the light's wavelength. In RGB color space, the **red**, **green**, and **blue** light are deformed differently as they pass through the lens.
data block or block	Defined set of related information that is used to construct the DisplayID Sections. Except for a few bytes at the beginning and end of each DisplayID Section (e.g., the Checksum bytes or the Structure Version/Revision byte), all DisplayID Structure v2.0 information shall be supplied as part of a data block. ***Note:*** *Although DisplayID Structure v2.0 defines many of the data blocks that can be used in DisplayID, other data blocks, including those defined by other standards organizations (such as CTA), may be encountered within a given DisplayID structure.*
descriptor(s)	Collection of fields, strings, elements, flags, and/or tags.
display	Full display assembly including monitor, and drive electronics in either a laptop (this would typically comprise components within the laptop lid), or a monitor where this would comprise the display-related components of a monitor, excluding the chassis, speakers, *USB* ports, etc.
element(s)	Key piece(s) of information regarding the display. May refer to a single bit flag or an entire DisplayID Section. See Section 4.1.1 for an example.
field	One or more contiguous bits or bytes within a data block that are used to convey a particular piece of information. For example, the Horizontal Pixel Count field within the Display Parameters data block comprises two bytes that specify the total number of physical pixels that the display device provides (see Section 4.2.2). May refer to a group of related flags within a given byte, –or– group of bits within a given byte, that provide a numeric value.
flag(s)	Single bits that, when set or cleared, provide information regarding a given parameter or feature in "yes/no," "present/not present," terms, etc. Indicate a Boolean choice of support or non-support. Typical elements refer to a single flag or collection of flags, any or all of which may be supported in DisplayID. See Section 4.1.4 for an example.
IEEE OUI	For DisplayID purposes, the assignment of an IEEE identifier for use in MAC addresses (MA-L, MA-M, MA-S) or an IEEE Company Identifier (CID) may be used wherever the Standard specifies use of an IEEE OUI.
lens distortion	Optical property of a lens that distorts the image. In this case, it does **not** include the color separation known as chromatic aberration. Sometimes referred to as "geometric aberration" or "monochromatic aberration." In the context of RGB color space, each of the color segments – **red**, **green**, and **blue** – are treated as if they are **green**.
monitor	Complete standalone desktop monitor, including the display plus chassis, power, and potentially *USB* hubs and/or speakers.
multi-function display	Display that includes additional USB functions, within the same enclosure, that are associated with the display function (e.g., speakers, cameras, microphones, sensors, etc.).
screen	Graphic portion of a display.

Table 1-6: Glossary (Continued)

Term	Definition
Section	Collection of variable-length data blocks of up to 256-byte portions of the DisplayID structure. Comprises the DisplayID Base Section, as well as Display ID Extension Sections if extensions are provided. All DisplayID structures shall have a Base Section, and may also include one or more DisplayID Extension Sections.
string	Contiguous bytes that are to be interpreted as text information (i.e., a series of characters that are read in the order in which they appear within the DisplayID block). Unless otherwise specified, all strings provided under DisplayID are provided in standard ASCII format (*ISO/IEC 8859-1*; see Section 1.3).
structure	Body of information provided per DisplayID Structure v2.0 (i.e., the DisplayID Base Section of up to 256 bytes of information, plus any and all Extension Sections, each of which may be up to 256 bytes in length). Provided as one contiguous space that is calculated from the sum of all individual DisplayID Section sizes.
tag(s)	Used when only a single choice is supported or referenced within a block (e.g., block identifiers). See Section 4.1.1 for an example.

1.6 Reference Documents

Tables 1-7 and 1-8 list the various reference documents that are used within this Standard. Users of this Standard are advised to ensure that they have the latest versions/revisions of reference Standards/Specifications and documents.

Table 1-7: Reference Documents (Normative)

Document	Version/ Revision[a]	Publication Date	Referenced As
Adobe® RGB (1998) Color Image Encoding	Version 2005-05	May 2005	*Adobe RGB*
BT.601, Studio encoding parameters of digital television for standard 4:3 and wide screen 16:9 aspect ratios[b]	Version 7	March 2011	*ITU-R BT.601*
BT.709, Parameter values for the HDTV standards for production and international programme exchange[b]	Version 6	June 2015	*ITU-R BT.709*
BT.1886, Reference electro-optical transfer function for flat panel displays used in HDTV studio production[b]	Version 0	March 2011	*ITU-R BT.1886*
BT.2020, Parameter values for ultra-high definition television systems for production and international programme exchange[b]	Version 2	October 2015	*ITU-R BT.2020*
CIE 15.2, Colorimetry	1986	1986	*CIE 15.2*
CIE 1931, x, y Chromaticity Diagram	1931	1931	*CIE 1931*
*CIE ISO 11664-5, Colorimetry – Part 5: CIE 1976 L*u*v* colour space and u', v' uniform chromaticity scale diagram*	2016	2016	*CIE 1976 L*u*v and u', v'*
ANSI/CTA-861, A DTV Profile for Uncompressed High Speed Digital Interfaces[c] (formerly known as *CEA-861*)	H	January 2021	*CTA-861-H* CTA-861 *CTA*
IEC 61966-2-1:1999, Multimedia systems and equipment – Colour measurement and management – Part 2-1: Colour management – Default RGB colour space – sRGB[d]	1999	October 18, 1999 and updates through January 2014	*IEC 61966-2-1*
IEEE 754-2019, IEEE Standard for Floating-Point Arithmetic[e]		July 22, 2019	*IEEE 754*
ISO/IEC 8859-1:1998, Information Technology – 8-bit single-byte coded graphic character sets – Part 1: Latin alphabet No. 1[d]		April 16, 1998	*ISO/IEC 8859-1*
SMPTE RP 431-2, D-Cinema Quality – Reference Projector and Environment[f]	2011	April 6, 2011	*SMPTE RP 431-2*
SMPTE ST 2084, High Dynamic Range Electro-Optical Transfer Function of Mastering Reference Displays[f]	2014	August 16, 2014	*SMPTE ST 2084*

Table 1-7: Reference Documents (Normative) (Continued)

Document	Version/ Revision[a]	Publication Date	Referenced As
VESA and Industry Standards and Guidelines for Computer Display Monitor Timing (DMT) Standard[g]	Version 1.13	February 8, 2013	*DMT Standard* *DMT*
VESA Coordinated Video Timings (CVT) Standard[g]	Version 1.2[h] Version 2.0[i]	February 8, 2013 September 27, 2021	*CVT v1.2* *CVT v2.0* *CVT Standard* *CVT*
VESA Display Stream Compression (DSC) Standard[g]	Version 1.1 Version 1.2a Version 1.2b[j]	August 1, 2014 January 18, 2017 August 12, 2021	*DSC v1.1* *DSC v1.2a* *DSC v1.2b* *DSC Standard* *DSC*
VESA Enhanced Extended Display Identification Data Standard (E-EDID)[g k]	Release A.2	September 25, 2006	*E-EDID Standard* *E-EDID* *EDID*

a. All references include subsequently published errata, specification change notices or engineering change notices, etc.
b. Published by International Telecommunication Union (ITU). See itu.int.
c. Available for download from global.ihs.com.
d. Published by the International Electrotechnical Commission (IEC). See webstore.iec.ch.
e. Published by IEEE Computer Society (IEEE). See standards.ieee.org.
f. Published by Society of Motion Picture and Television Engineers.
g. See vesa.org/vesa-standards/.
h. For standard blanking and reduced blanking timing v1 (RB Timing v1), see CVT v1.2.
i. For RB Timing v2 and v3, see CVT v2.0.
j. DSC v1.2b is functionally identical to DSC v1.2a. DSC v1.2b is a maintenance release that includes only minor usability enhancements to the C model, editorial updates, and integration of DSC v1.2a-related SCRs in the Standard.
k. Legacy VESA Standard, superseded by this Standard.

Table 1-8: Reference Documents (Informative)

Document	Version/ Revision[a]	Publication Date	Referenced As
Digital Visual Interface (DVI)	1.0	April 2, 1999	*DVI r1.0* *DVI*
High-Definition Multimedia Interface (HDMI) Specification[b]	Version 1.4b Version 2.1	October 11, 2011 November 28, 2017	*HDMI v1.4b* *HDMI v2.1* *HDMI Specification* *HDMI*
IETF RFC 4122, A Universally Unique IDentifier (UUID) URN Namespace[c]		March 2, 2013	*IETF RFC 4122*
LVDS (defined by *ANSI/TIA/EIA-644-A*)		2002	
Universal Serial Bus 4 (USB4™) Specification[b]	Version 1.0	August 29, 2019	*USB4* *USB*
VESA Adaptive-Sync Display Certification Test Specification (Adaptive-Sync Display CTS)[d]	Revision 1.0	***TBD***	*Adaptive-Sync Display CTS*
VESA Adaptive-Sync Link Layer Certification Test Specification (Adaptive-Sync Source CTS)[d]	Revision 1.0	***TBD***	*Adaptive-Sync Link Layer CTS*
VESA Augmented Reality/Virtual Reality (ARVR) Standard[d]	Version 1.0	***TBD***	*ARVR Standard*
VESA DisplayPort 2.x DisplayID EDID Compliance Test Specification (DisplayID EDID CTS)[e]	DP 2.*x* Revision 1.0	***TBD***	*DisplayID EDID CTS*
VESA DisplayPort (DP) Standard[d]	Version 2.0[f]	June 26, 2019	*DP v2.0* *DP Standard* *DisplayPort* *DP*
DisplayPort v2.0 Errata E11[d]	E11	October 12, 2021	*DisplayPort v2.0 Errata E11*
VESA Embedded DisplayPort (eDP) Standard[d]	Version 1.5	August 26, 2021	*eDP*
VESA Enhanced Display Data Channel (E-DDC) Standard[e]	Version 1.3	September 11, 2017	*E-DDC Standard* *E-DDC*
VESA Glossary of Terms[g]	Current	Current	
VESA Intellectual Property Rights (IPR) Policy[d]	200D	March 27, 2017	

a. All references include subsequently published errata, specification change notices or engineering change notices, etc.
b. See hdmi.org.
c. See ietf.org/rfc/rfc4122.txt.
d. See vesa.org/join-vesamemberships/member-downloads/.
e. See vesa.org/vesa-standards/.
f. This version of the referenced Standard is correct at the time of publication of this Standard. In the event that a later version of the referenced Standard is published, reference should be made to the latest published version.
g. Available in the VESA Public Standards folder in Box.

2 DisplayID Structure

DisplayID, like previous *EDID* Standards, is based on the concept that essential display identification and configuration data is declared in one or more contiguous Sections. DisplayID is divided into 256 variable-length Sections of up to 256 bytes each. The first DisplayID Section is referred to as the EDID Base Section in the original *EDID* nomenclature, –or– DisplayID Base Section as of DisplayID Structure v2.0. Subsequent DisplayID Sections are referred to as "DisplayID Extension Sections."

DisplayID Sections are collections of related elements, listed individually or grouped into data blocks and further into sub-blocks, referred to as "descriptors."

Addresses for each descriptor within a data block are provided based on the beginning of that block, in hexadecimal format. Addresses for individual field offsets within a descriptor are provided in decimal-format bytes or bits.

Under DisplayID Structure v2.0, however, the entire content, including the DisplayID Base Section and any DisplayID Extension Sections, may be constructed in a flexible manner from any number of elements and/or predefined data blocks or descriptors. This method is derived from, and intended to be compatible with, a method first developed by *CTA* in its CTA EDID Extension, Version 3 (see *CTA-861*). In defining that extension, *CTA* developed a powerful model in which related display data elements may be collected into variable-length data blocks, each with a unique identifying tag, which may then be mixed and matched, as needed, in the creation of a CTA Extension. VESA has since adopted this model for its DisplayID structure. The entire DisplayID structure provided here, including the DisplayID Base Section and any and all Extension Sections that may appear following the DisplayID Base Section, can be allocated, as needed, using defined data block structures (both VESA and *CTA* definitions) that have only a few restrictions.

A display product's DisplayID structure shall identify at least one Video Timing mode, provide details on that timing, and declare that Video Timing mode as the preferred mode for that display product. The identification of a timing mode as preferred, the definition of preferred, and the prioritizing of Video Timing modes within the DisplayID structure, are described in Section 4.3.

Certain elements are needed in the DisplayID Base Section, and certain elements may appear only once within the entire DisplayID structure. These restrictions are detailed within the individual block definitions. Table 2-1 defines the variable-length DisplayID Structure v2.0 format, in which the Checksum byte immediately follows the data block's end.

Table 2-1: Variable-length DisplayID Section Structure

Offset	Value	Definition	Mandatory
00h	20h	**DisplayID Structure Version/Revision** Version 2, Revision 0.	✓
01h	00h – FBh	**Bytes in Section** Section length, excluding the five mandatory bytes. Value is $N - 5$.[a]	✓
03h – 02h		**Display Product Primary Use Case and Extension Count**	
02h	0h – Fh	**Display Product Primary Use Case**	✓
03h	00h – FFh	**Extension Count** RESERVED. Cleared to all 0s for the DisplayID Extension Section and total Extension Count within the DisplayID Base Section.	✓
04h	Block	**Data Block** See Section 4.	See Table 3-1
…		**…**	
$(N - 2)$h[a]	Block	**Last Byte of Last Valid Data Block**	
$(N - 1)$h[a]	00h – FFh	**Checksum**	✓

a. N represents the total number of bytes within the DisplayID Section.

The primary difference between DisplayID Base and Extension Sections is that in an Extension Section, the Extension Count bytes are cleared to all 0s. The Display Product Primary Use Case field shall be declared only in the first DisplayID Section.

The DisplayID framework allows a DisplayID Section to be defined as a fixed size with unused bytes that are filled with dummy values (0s) between the DisplayID Section's last valid data block and its last byte, the Checksum byte. Table 2-2 defines the fixed-length DisplayID Structure v2.0 format.

Note: *See Appendix A for a fixed-length DisplayID Section example.*

Table 2-2: Fixed-length DisplayID Section Structure

Offset	Value	Definition	Mandatory
00h	20h	**DisplayID Structure Version/Revision** Version 2, Revision 0.	✓
01h	00h – FBh	**Bytes in Section** Section length, excluding the five mandatory bytes. Value is $N - 5$.[a]	✓
03h – 02h		**Display Product Primary Use Case and Extension Count**	
02h	0h – Fh	**Display Product Primary Use Case**	✓
03h	00h – FFh	**Extension Count** RESERVED. Cleared to all 0s for the DisplayID Extension Section and total Extension Count within the DisplayID Base Section.	✓
04h	Block	**Data Block** See Section 4.	See Table 3-1
…		**…**	
…h	Block	**Last Byte of Last Valid Data Block**	
(… + 1)h	00h	**Fill Data**	
…		**Fill Data**	
(N – 2)h[a]	00h	**Fill Data**	
(N – 1)h[a]	00h – FFh	**Checksum**	✓

a. N represents the total number of bytes within the DisplayID Section.

Tables 2-3 and 2-4 define the DisplayID Base and Extension Section elements, respectively.

Table 2-3: DisplayID Base Section

Offset	Bit #	Field
00h	**DisplayID Structure Version/Revision**	
	3:0	**Revision** 0h = Revision 0.
	7:4	**Version** 0h = RESERVED. 2h = Version 2.
01h	7:0	**Bytes in Section** Section length, excluding the five mandatory bytes. Value is $N - 5$.[a]
02h	**Display Product Primary Use Case**	
	3:0	**Display Product Primary Use Case** Use case ranges from 0 through 15. 0h = Used for DisplayID Extension Sections that have the same primary use case as the DisplayID Base Section. 1h = Test Structure; test equipment only. 2h = None of the listed primary use cases; generic display. 3h = Television (TV) display. 4h = Desktop productivity display. 5h = Desktop gaming display. 6h = Presentation display. 7h = Head-mounted Virtual Reality (VR) display. 8h = Head-mounted Augmented Reality (AR) display. 9h through Fh = RESERVED.
	7:4	RESERVED, cleared to all 0s.
03h	7:0	**Extension Count** RESERVED. Cleared to all 0s for the DisplayID Extension Section and total Extension Count within the DisplayID Base Section.
04h	7:0	**Data Block** See Section 3.
…		**…**
$(N - 1)$h[a]		**Checksum** Checksum ranges from 0 through 255. 00h = 0. … FFh = 255.

a. N represents the total number of bytes within the DisplayID Section.

Table 2-4: DisplayID Extension Section

Offset	Bit #	Field
00h	**DisplayID Structure Version/Revision**	
	3:0	**Revision** 0h = Revision 0.
	7:4	**Version** 0h = RESERVED. 2h = Version 2.
01h	7:0	**Bytes in Section** Section length, excluding the five mandatory bytes. Value is $N - 5$.[a]
02h	**Display Product Primary Use Case**	
	3:0	**Display Product Primary Use Case** Value shall be 0h.[b]
	7:4	**RESERVED** Cleared to all 0s.
03h	7:0	**Extension Count** Value shall be 00h.
04h	7:0	**Data Block** See Section 3.
…		**…**
$(N - 1)$h[a]		**Checksum** Value ranges from 00h through FFh. 00h = 0. … FFh = 255.

a. N represents the total number of bytes within the DisplayID Section.

b. A native DisplayID structure is composed of a DisplayID Base Section and zero or more DisplayID Extension Sections. However, when DisplayID is used as an EDID Extension block, the first DisplayID Section that is contained in an EDID Extension block is a Base Section. Hence, the Display Product Primary Use Case field is nonzero in that particular Base Section.

2.1 DisplayID Structure v2.0 as an EDID Extension Block

To facilitate use of DisplayID Structure v2.0 Section(s) as EDID Extension Block(s), a new framework is defined. EDIDs containing Extension Sections that use DisplayID Structure v2.0 shall include all **mandatory** data blocks, as defined in Table 3-1, and populate the listed fields as follows:

- EDID Extension block tag shall be programmed to 70h to denote the existence of a DisplayID Section.
- Following the block tag byte, a regular DisplayID Structure v2.0 fixed-length Section shall be used to map to the EDID Extension Section mandate of 128 bytes/Extension Section.
- DisplayID Structure Version/Revision byte shall be programmed to 20h, corresponding to DisplayID Structure v2.0.
- First DisplayID Structure v2.0 EDID Extension Section shall populate the Display Product Primary Use Case field with a value of 01h to 08h based on the Sink device's intended primary use case.[a] If the Primary Use Case is unknown, the Sink device shall program the value to 02h. Any subsequent DisplayID Structure v2.0 EDID Extension Section(s) shall clear the Display Product Primary Use Case field to 00h.
- Extension Count byte shall be cleared to 00h.
- Bytes in Section byte shall be programmed to 79h (121). This corresponds to five additional bytes for the DisplayID Section framework + EDID Extension Section block tag for DisplayID and EDID Extension Section block checksum, for a total of 128 bytes.
- EDID Extension Section block checksum shall be calculated using the DisplayID Section Checksum and Extension Section block tag byte. Because the DisplayID Section Checksum already ensures that all DisplayID Section bytes add up so that modulo 256 of the sum of all bytes is 0h, only the Data Block tag byte needs to be considered for the EDID Extension Section block checksum. Consequently, the EDID Extension Section block checksum shall be programmed to 90h.

a. *A native DisplayID structure is composed of a DisplayID Base Section and zero or more DisplayID Extension Sections. However, when DisplayID is used as an EDID Extension block, the first DisplayID Section that is contained in an EDID Extension block is a Base Section. Hence, the Display Product Primary Use Case field is nonzero in that particular DisplayID Base Section.*

To facilitate compatibility with legacy Source devices that may not have been updated to be able to parse DisplayID Structure v2.0, the Sink device may expose separate DisplayID Extension Sections with DisplayID Structure v1.2 and DisplayID Structure v2.0. If the Sink device exposes Extension Sections with both DisplayID Structure v1.2 and DisplayID Structure v2.0, the following mandates shall be met in addition to those listed for DisplayID Structure v2.0 Extension Sections:

- DisplayID EDID Extension Section with DisplayID Structure v1.2 shall meet *DisplayID v1.3* mandates. In particular, note that all data blocks are **optional** for DisplayID EDID Extension Sections with DisplayID Structure v1.2 – which data blocks to include is an implementation-specific choice.
- DisplayID EDID Extension Sections with DisplayID Structure v2.0 shall be exposed at higher EDID block numbers than any **optional** DisplayID EDID Extension Sections with DisplayID Structure v1.2.
- DisplayID EDID Extension Sections (both DisplayID Structure v1.2 and DisplayID Structure v2.0) shall be exposed at higher EDID block numbers than any **optional** CTA Extension Sections.

Table 2-5 lists examples of how the EDID Extension Section blocks are to be ordered.

Table 2-5: Example of Mandatory Ordering of EDID Extension Section Blocks

Segment Pointer/ Address	EDID Block Number	Block Type
00h / 00h through 7Fh	0	EDID Base Section
00h / 80h through FFh	1	Extension Section Block – *CTA-861*
01h / 00h through 7Fh	2	Extension Section Block – DisplayID Structure v1.2
01h / 80h through FFh	3	Extension Section Block – DisplayID Structure v2.0
02h / 00h through 7Fh	4	Extension Section Block – DisplayID Structure v2.0

A Source device that supports parsing EDID Extension Sections that use DisplayID Structure v2.0 shall parse EDID Extension Sections regardless of where the Extension Sections appear within a valid *EDID* structure. If similar capabilities are described in multiple EDID blocks, the Source device shall use the information exposed in the DisplayID Structure v2.0 Extension Section and ignore the duplicate or conflicting information that is exposed in other EDID blocks. Examples of similar capabilities include the following:

- Display Parameters data block that is present in both a DisplayID Structure v1.2 Extension Section and DisplayID Structure v2.0 Extension Section
- Chromaticity information that is present in both the EDID Base block and a Display Parameters data block in a DisplayID Structure v2.0 Extension Section

Figure 2-1 illustrates an example of the DisplayID Structure v2.0 Section being used as an EDID Extension Section.

DisplayID-to-EDID Extension Section Example – With Fill Data for Fixed-length Section							
			Field Name	Offset (Dec)	Offset (Hex)	Value (Hex)	Value (Dec)
EDID Extension Section	DisplayID Structure v2.0 EDID Extension Section	EDID Extension Section Block Tag	Tag	0	0	70	112
	DisplayID Extension Section	DisplayID Structure v2.0 Extension Section Header	DisplayID Structure Version/Revision	1	0	20	32
			# of Bytes in Section	2	1	79	121
			Display Product Primary Use Case	3	2	2	2
			Extension Count RESERVED (00h)	4	3	00	00
		Section Data Block(s) (Example)	Data Block Tag	5	4	ID	ID
			Data Block Revision	6	5	0	0
			# of Payload Bytes	7	6	6	6
			Payload Byte 1	8	7	PB1	PB1
			Payload Byte 2	9	8	PB2	PB2
			Payload Byte 3	10	9	PB3	PB3
			Payload Byte 4	11	0A	PB4	PB4
			Payload Byte 5	12	0B	PB5	PB5
			Payload Byte 6	13	0C	PB6	PB6
		Fill Data for Fixed Length (Example)	Fill Data #1	14	0D	0	0
			Fill Data #2	15	0E	0	0
			Fill Data #3…125	125	7D	0	0
			Section Checksum	126	7E	CHK_DID	CHK_DID
			EDID Extension Section Block Checksum	127	7F	90	144

Figure 2-1: DisplayID Structure v2.0 as an EDID Extension Section Example (Informative)

3 Data Blocks – Definitions and Use

As noted in Section 2, the DisplayID structure differs from the previous EDID structure. All information provided under DisplayID, except for primary use case-related information, is in the form of data blocks. Data blocks are predefined modules containing a given set of related information. With certain restrictions, data blocks may be used as needed and at the manufacturer's discretion throughout the DisplayID structure.

Typically, data blocks do not have a fixed length, although some specific block definitions may establish a fixed length for that type of block. Data blocks shall be distinguished through a block tag, which is always the first byte. The second byte contains block revision- and data block-related information. Following that byte shall be a single byte that defines the block length, as the remaining number of bytes to follow in that block up to a maximum of 248 bytes, while excluding the mandatory tag, block revision, and length bytes. This shall permit a single data block to fill the remainder of any 256-byte Section within the DisplayID structure – a single 251-byte data block, plus the mandatory five bytes – headers and checksum – for that DisplayID Section.

VESA and *CTA* both have defined data blocks. To simplify distinguishing between them, the range of possible block tags has been evenly divided so that the most significant bit (bit 7) serves as a VESA or *CTA* flag. VESA controls tags within the range of 0 through 127 decimal. *CTA* or other organizations control tags within the range of 128 through 255 decimal. Although VESA and *CTA* may both provide information describing the other's data blocks, as is done within this section and Section 4, the information is provided only for reference. To ensure compliance, all DisplayID structure implementers shall consult the relevant standards of the appropriate organization for the data block(s) in question.

Notes: *VESA-defined data block specifications are provided in Section 4. For editing and revision purposes, however, each shall be considered as a separate specification and assigned a revision number that applies only to that data block. The overall DisplayID version/revision number, as indicated on the cover page of this Standard, applies to all DisplayID Structure v2.0 Sections except for the data block specifications. Readers are cautioned to note the document revision date on the cover page, and to check with the VESA office to ensure that they are using the current revision/version of all related standards/specifications.*

Section 4.12 describes how a CTA-861 Data Block Encapsulation DisplayID data block can be mapped within a DisplayID data block.

Table 3-1 lists the DisplayID Structure v2.0 data blocks.

Table 3-1: Data Block Tag Allocation

Block Tag	Data Block Name	Mandatory Block[a] [b]		Defined in
		Native Structure	EDID Extension Section	
1Fh – 00h	**RESERVED** RESERVED for legacy data blocks for DisplayID Structure v1.*x*. and shall **not** be used with DisplayID Structure v2.*x*.			
20h	**Product Identification Data Block**	Yes, for Display Devices	Yes, for Display Devices if Tiled Display Topology –or– Vendor-specific data block (using other than VESA OUI) is present	4.1
21h	**Display Parameters Data Block**	Yes, for Display Devices	Yes, for Display Devices	4.2
22h	**Type VII Timing – Detailed Timing Data Block**	Yes, for Display Devices; shall contain at least one timing	Yes, for Display Devices; shall contain at least one timing	4.3.1
23h	**Type VIII Timing – Enumerated Timing Code Data Block**	No	No	4.3.2
24h	**Type IX Timing – Formula-based Timing Data Block**	No	No	4.3.3
25h	**Dynamic Video Timing Range Limits Data Block**	No	No	4.4
26h	**Display Interface Features Data Block**	Yes, for Display Devices	Yes, for Display Devices	4.5
27h	**Stereo Display Interface Data Block**	Yes, for ARVR HMDs Otherwise no, but shall be present whenever the 3D Stereo Support flags in any Timing descriptor indicate that the display is capable of displaying stereoscopic images	No, but shall be present whenever the 3D Stereo Support flags in any Timing descriptor indicate that the display is capable of displaying stereoscopic images	4.6
28h	**Tiled Display Topology Data Block**	Yes, for Display Devices that support Tiled Display	Yes, for Display Devices that support Tiled Display	4.7
29h	**ContainerID Data Block**	Yes, for Multi-function Display Devices	Yes, for Multi-function Display Devices	4.8

Table 3-1: Data Block Tag Allocation (Continued)

Block Tag	Data Block Name	Mandatory Block[a] [b]		Defined in
		Native Structure	EDID Extension Section	
2Ah	**Type X Timing – Formula-based Timing Data Block**	Yes, for Display Devices that support multiple *CVT v2.0* RB Timing v3 timings	No	4.3.4
2Bh	**Adaptive-Sync Data Block**	Yes, for Display Devices that support Adaptive-Sync	Yes, for Display Devices that support Adaptive-Sync	4.9
2Ch	**ARVR_HMD Data Block**	Yes, for ARVR HMDs	Not allowed for ARVR HMDs	4.10.1
2Dh	**ARVR_Layer Data Block**	Yes, for ARVR HMDs	Not allowed for ARVR HMDs	4.10.2
7Dh – 2Eh	**RESERVED for Additional VESA-defined Data Blocks**			
7Eh	**Vendor-specific Data Block**	No	No	4.11
80h – 7Fh	**RESERVED**			
81h[c]	**CTA Audio Data Block encapsulated in CTA-861 Data Block Encapsulation DisplayID Data Block**	Yes, for Display Devices with in-band Interface Audio support beyond what can be exposed in DisplayID data blocks	Yes (unless CTA Extension is present), for Display Devices with in-band Interface Audio support beyond what can be exposed in DisplayID data blocks	4.12
	CTA Speaker Allocation Block encapsulated in CTA-861 Data Block Encapsulation DisplayID Data Block			
	CTA HDR Static Metadata Data Block encapsulated in CTA-861 Data Block Encapsulation DisplayID Data Block	Yes, for Display Devices with HDR support beyond what can be exposed in DisplayID data blocks	Yes (unless CTA Extension is present), for Display Devices with HDR support beyond what can be exposed in DisplayID data blocks	
	Other CTA-related Data Blocks encapsulated in CTA-861 Data Block Encapsulation DisplayID Data Block	No	No	

Table 3-1: Data Block Tag Allocation (Continued)

Block Tag	Data Block Name	Mandatory Block[a b]		Defined in
		Native Structure	EDID Extension Section	
FFh – 82h	**RESERVED** RESERVED for additional data blocks related to external standards organization(s).			

a. "Yes, for Display Devices" indicates that the Display Product Primary Use Case field of the DisplayID structure for the relevant product shall be populated with a value within the range of 2h through 8h.

*b. A Source device that supports DisplayID Structure v2.0 shall be able to parse all **mandatory** data blocks and enable relevant features that are based on the Source device's Capability Declaration Form as part of* DisplayID EDID CTS.

c. CTA-861 Data Block Encapsulation DisplayID data blocks are defined in CTA-861-H. *A Source device shall be able to parse encapsulated versions of the CTA data blocks, as well as CTA data blocks that exist within the CTA EDID Extension.*

3.1 Data Blocks

All DisplayID Structure v2.0 data blocks follow the basic format listed in Table 3-2. The block's first byte contains an 8-bit tag (see Table 3-1), the second byte contains the block revision and data block, and the third byte contains the data block's length. Three bits of the data block revision number are provided as the lower three bits of the second byte; the byte's remaining five bits may be used as specified in the particular data block definition in question. Following these three mandatory bytes shall be the actual payload of information conveyed by that block, per its specification as provided here or elsewhere. Eight payload length bits are provided in the third byte of each data block. However, the maximum length of any data block is 251 bytes, –or– a full 256-byte DisplayID Section. (For example, in this case, there would be 251 bytes for the data block itself, comprising a 3-byte data block header plus 248-byte payload, plus the four mandatory DisplayID Section header bytes, plus the Checksum byte that is needed for all DisplayID Sections.)

Note: *Under this system, 00h shall be a valid payload length. It is possible that a data block may be defined such that its mere presence conveys significant information to the Source device, –or– that all necessary information is conveyed in the second byte's five block-specific bits.*

Table 3-2: DisplayID Data Block

Offset	Bit #	Definition
00h	7:0	**Data Block Identification** Value ranges from 00h through FFh.
01h	**Block Revision and Other Data**	
	2:0	**Block Revision** Revision ranges from 0 through 7. 000b = Revision 0 (default).
	7:3	**RESERVED** Block-specific. Cleared to all 0s.
02h	7:0	**Number of Payload Bytes** Number of payload bytes within the data block ranges from 0 through 248. 00h = 0 bytes. … F8h = 248 bytes.
03h	7:0	**First Data Payload Byte** Descriptor.
04h	7:0	**Second Data Payload Byte** Descriptor, if present.
…		**…**

The Block Revision field shall be incremented whenever a new data block field is defined that does not break compatibility with legacy revisions. In case backward-compatibility cannot be achieved, a new block tag shall be used to define the new data block. The following are cases in which the Block Revision field shall be incremented:

- Adding a new field at the end of the data block (the Number of Payload Bytes field shall also be updated for this case)
- Redefining previously RESERVED bit(s)
- Extending the range of a previously RESERVED field

Any variable-length field within a data block shall be set up such that the field's length can be determined, without ambiguity. This allows for future expansion of data blocks without breaking compatibility.

3.2 DisplayID Section – A Visual Representation

Table 3-3 visually describes DisplayID Section format and packing with common data blocks.

Table 3-3: DisplayID Section Visual Representation

Header Name			Field Name
DisplayID Section	Section Header		DisplayID Structure Version/Revision
			Section Size
			Display Product Primary Use Case
			Extension Count
	Product Identification Data Block	Block Header	Block Tag
			Block Revision and Other Data
			Number of Payload Bytes
		Block Payloads	Block Payload Byte 0
			Block Payload Byte 1
			Block Payload Byte 2
			Block Payload Byte 3
			…
			Block Payload Last Byte *N*
	Display Parameters Data Block	Block Header	Block Tag
			Block Revision and Other Data
			Number of Payload Bytes
		Block Payloads	Block Payload Byte 0
			Block Payload Byte 1
			Block Payload Byte 2
			Block Payload Byte 3
			…
			Block Payload Last Byte *N*

Table 3-3: DisplayID Section Visual Representation (Continued)

Header Name			Field Name
DisplayID Section	Display Interface Features Data Block	Block Header	Block Tag
			Block Revision and Other Data
			Number of Payload Bytes
		Block Payloads	Block Payload Byte 0
			Block Payload Byte 1
			Block Payload Byte 2
			Block Payload Byte 3
			…
			Block Payload Last Byte *N*
	Video Timing Mode Data Block(s)	Block Header	Block Tag
			Block Revision and Other Data
			Number of Payload Bytes
		Block Payloads	Block Payload Byte 0
			Block Payload Byte 1
			Block Payload Byte 2
			Block Payload Byte 3
			…
			Block Payload Last Byte *N*
	Adaptive-Sync Data Block	Block Header	Block Tag
			Block Revision and Other Data
			Number of Payload Bytes
		Payloads	Block Payload Byte 0
			Block Payload Byte 1
			…
			Block Payload Last Byte *N*
	Checksum		Checksum

4 Data Block Definitions

This section provides specifications for all VESA-defined data blocks that are defined for use within the DisplayID structure. As previously noted, valid DisplayID data blocks may be defined by other organizations, most notably *CTA*, and individual VESA data block definitions shall be considered as separate Standards by VESA and carry their own revision number and revision date.

The overall DisplayID version and revision number shall not be incremented for revisions to the individual data block definitions; these shall increment only for revisions applied to previous DisplayID Sections, including changes to the overall DisplayID structure definition or the basic data block structure (e.g., location and/or size of the tag code or length fields, etc.). The overall DisplayID version/revision numbers shall also **not** be incremented in the case of adding new tag code assignments. The document revision date, however, shall be updated for any changes to one or more data block definitions, –or– the addition or deletion of data block definitions and/or tag code numbers.

The mandated definition for each data block (i.e., whether a field is mandatory) is relevant for the DisplayID Base Section.

Tables summarizing the mandates for each display product primary use case defined by Table 3-1, along with the tag code, latest revision, and any restrictions, are provided at the beginning of each data block's definition.

4.1 Product Identification Data Block

The DisplayID Base Section shall include a mandatory Product Identification data block as its first data block. The Product Identification data block is composed of several fields that shall be used to uniquely identify the monitor.

For Native structures, Product Identification is a **mandatory** data block for the product primary use case as a display device; otherwise, the data block is **optional**.

For EDID Extension Sections, Product Identification is a **mandatory** data block for the Tiled Display Topology or Vendor-specific data blocks (using other than VESA OUI is present); otherwise, the data block is **optional**.

Table 4-1 defines the size and order of each field, with all addresses relative to the beginning of the data block.

Note: *See Appendix A for an example of how the Product Identification data block is used in a fixed-length DisplayID Section.*

Table 4-1: Product Identification Data Block

Offset	Bit #	Definition	Defined in
00h	7:0	**Product Identification Data Block** 20h.	
01h	**Block Revision and Other Data**		
	2:0	**Block Revision** Revision ranges from 0 through 7. 000b = Revision 0 (default).	
	7:3	**RESERVED** Cleared to all 0s.	
02h	7:0	**Number of Payload Bytes in Block** Number of payload bytes within the data block ranges from 12 through 248. 0Ch = 12 bytes. … F8h = 248 bytes.	
05h – 03h	23:0	**Manufacturer/Vendor ID** Descriptor.	4.1.1
07h – 06h	15:0	**Product ID Code** Descriptor.	4.1.2
0Bh – 08h	31:0	**Serial Number** Optional Descriptor.	4.1.3
0Ch	7:0	**Week of Manufacture/Model Tag** Descriptor.	4.1.4
0Dh	7:0	**Year of Manufacture/Model Year** Descriptor.	4.1.4
0Eh	7:0	**Size of Product Name String** Descriptor.	4.1.5
FBh – 0Fh		**Product Name String** Optional Descriptor.	4.1.5

4.1.1 Manufacturer/Vendor ID Field

The Manufacturer/Vendor ID field listed in Table 4-1 is a **mandatory** element of the Product Identification data block. The Manufacturer/Vendor ID field contains the display manufacturer's 3-byte IEEE Organizationally Unique Identifier (OUI) code, which identifies the display's manufacturer or vendor.

The Institute of Electrical and Electronics Engineers (IEEE) issues IEEE OUIs. Contact the IEEE at standards.ieee.org/products-services/regauth/oui/index.html for registration and/or further details.

Table 4-2 defines the Manufacturer/Vendor ID field.

Table 4-2: Manufacturer/Vendor ID Field

Offset	Byte #	Bit #	Definition
03h	0	7:0	**IEEE OUI First Byte** Byte code.
04h	1	7:0	**IEEE OUI Second Byte** Byte code.
05h	2	7:0	**IEEE OUI Third Byte** Byte code.

4.1.2 Product ID Code Field

The Product ID Code field listed in Table 4-1 is a **mandatory** element of the Product Identification data block. The Product ID Code field contains a 2-byte vendor-assigned Product ID code. The field's length helps to differentiate between multiple models from the same manufacturer. If this field is used to represent a Product ID code (e.g., a model number), the number shall be stored in hexadecimal format with the least significant byte (LSB) listed first. Table 4-3 defines the Product ID Code field.

Table 4-3: Product ID Code Field

Offset	Byte #	Bit #	Definition
06h	0	7:0	**Product ID Code7:0** Value ranges from 00h through FFh.
07h	1	7:0	**Product ID Code15:8** Value ranges from 00h through FFh.

4.1.3 Serial Number Field

The Serial Number field listed in Table 4-1 is an **optional** element of the Product Identification data block. The Serial Number field is a 32-bit serial number that shall be used to differentiate between individual instances of the same model of display product. When used, this field's bit order follows the order defined in Table 4-4. The four bytes of the serial number are listed with the LSB first. The serial number ranges from 0 through 4,294,967,295. This serial number shall be a number only – it shall **not** represent an ASCII code. If this field is not used, however, a value of "00h, 00h, 00h, 00h" shall be entered.

Table 4-4: Serial Number Field

Offset	Byte #	Bit #								Definition
		7	6	5	4	3	2	1	0	
08h	0	7	6	5	4	3	2	1	0	**ID Serial Number**
09h	1	15	14	13	12	11	10	9	8	
0Ah	2	23	22	21	20	19	18	17	16	
0Bh	3	31	30	29	28	27	26	25	24	

4.1.4 Week of Manufacture/Model Tag and Year of Manufacture/Model Year Fields

The Week of Manufacture/Model Tag and Year of Manufacture/Model Year fields listed in Table 4-1 are **mandatory** Product Identification data block elements. The Week of Manufacture/Model Tag field, when used to indicate the week, shall be programmed to a value within the range of 1 through 52. If a week is not declared, the value shall be cleared to all 0s. If a model year is to be declared, the value shall be programmed to FFh and the Year of Manufacture/Model Year field shall indicate the model year.

Otherwise, the Year of Manufacture/Model Year field shall be used to represent the Gregorian calendar year in which the display was manufactured. In either case, the value used to indicate the year shall be stored as an offset from the year 2000, as derived from the following equation:

Stored Value = (Year of Manufacture – 2000)

For example, a display manufactured in 2021 would contain value 15h in the Year of Manufacture/Model Year field. If the Year of Manufacture/Model Year field is not used, the byte would be cleared to all 0s.

Table 4-5 defines the Week of Manufacture/Model Tag and Year of Manufacture/Model Year fields.

Table 4-5: Week of Manufacture/Model Tag and Year of Manufacture/Model Year Field

Offset	Byte #	Bit #	Definition
0Ch	0	7:0	**Week of Manufacture/Model Tag** 00h = No week specified. 01h through 34h = Week number within the year of manufacture. Week number ranges from 1 through 52. 35h through FEh = RESERVED. FFh = Model year is specified in Byte 0Dh.
0Dh	1	7:0	**Year of Manufacture/Model Year** 00h through 0Eh = RESERVED. 0Fh through FFh = Year of manufacture: • Model year when Byte 0Ch = FFh • Gregorian year when Byte 0Ch ≠ FFh

4.1.5 Size of Product Name String and Product Name String Fields

The Size of Product Name String field listed in Table 4-1 is a **mandatory** element of the Product Identification data block. The Size of Product Name String field provides the Product Name String field length, in bytes. If the length is cleared to all 0s, the Product Name String field does **not** exist.

The Product Name String field listed in Table 4-1 is an **optional** element of the Product Identification data block. The Product Name String field may be used to provide additional model and/or product identification information in the form of an ASCII character string of variable length, up to the DisplayID Section's maximum size. The string's first character is stored at Offset 0Fh, the second character is stored at Offset 10h, etc.

Table 4-6 defines the Size of Product Name String and Product Name String fields.

Table 4-6: Size of Product Name String and Product Name String Field

Offset	Byte #	Bit #	Definition
0Eh	0	7:0	**Size of Product Name String** 00h = No string specified (not typical use). 01h through ECh = Number of bytes within the Product Name String field ranges from 1 through 236.
FBh – 0Fh	236:1		**Product Name String** Value ranges from 00h through FFh.

4.2 Display Parameters Data Block

The Display Parameters data block is composed of several fields that shall define the monitor's global parameters. No more than one Display Parameters data block shall be exposed within the DisplayID Section. Display Parameters is a **mandatory** data block for product primary use case as a display device; otherwise, the data block is **optional**.

Table 4-7 defines the size and order of each Display Parameters data block field, with all addresses relative to the beginning of the data block.

Note: *See Appendix A for an example of how the Display Parameters data block is used in a fixed-length DisplayID Section.*

Table 4-7: Display Parameters Data Block

Offset	Bit #	Definition	Defined in
00h	7:0	**Display Parameters Data Block** 21h.	
01h	**Block Revision and Other Data**		
	2:0	**Block Revision** Revision ranges from 0 through 7. 001b = Revision 1. Added the Display Device Theme Preference bit to the data block (Offset 1Eh[7]).	
	6:3	**RESERVED** Cleared to all 0s.	
	7	**Image Size Multiplier** 0 = Horizontal Image Size and Vertical Image Size fields (Offsets 03h through 06h) have 0.1-mm precision (default). 1 = Horizontal Image Size and Vertical Image Size fields have 1.0-mm precision.	4.2.1
02h	7:0	**Number of Payload Bytes in Block** 1Dh = Data block is composed of 29 payload bytes.	
04h– 03h	15:0	**Horizontal Image Size** Descriptor.	4.2.1
06h– 05h	15:0	**Vertical Image Size** Descriptor.	4.2.1
08h– 07h	15:0	**Horizontal Pixel Count** Descriptor.	4.2.2
0Ah– 09h	15:0	**Vertical Pixel Count** Descriptor.	4.2.2
0Bh	7:0	**Feature Support Flags** Descriptor.	4.2.3
0Eh – 0Ch	23:0	**Native Color Chromaticity (Primary Color 1 Chromaticity)** Descriptor.	4.2.4

Table 4-7: Display Parameters Data Block (Continued)

Offset	Bit #	Definition	Defined in
11h – 0Fh	23:0	**Native Color Chromaticity (Primary Color 2 Chromaticity)** Descriptor.	4.2.4
14h – 12h	23:0	**Native Color Chromaticity (Primary Color 3 Chromaticity)** Descriptor.	4.2.4
17h – 15h	23:0	**Native Color Chromaticity (White Point Chromaticity)** Descriptor.	4.2.4
19h – 18h	15:0	**Native Maximum Luminance (Full Coverage)** Descriptor.	4.2.5
1Bh – 1Ah	15:0	**Native Maximum Luminance (10% Rectangular Coverage)** Descriptor.	4.2.5
1Dh – 1Ch	15:0	**Native Minimum Luminance** Descriptor.	4.2.5
1Eh	7:0	**Native Color Depth and Display Device Technology** Descriptor.	4.2.6
1Fh	7:0	**Native Gamma EOTF** Descriptor.	4.2.7

4.2.1 Horizontal Image Size and Vertical Image Size Fields

The Horizontal Image Size and Vertical Image Size fields of the Display Parameters data block, along with the Image Size Multiplier bit (Offset 01h[7]), shall define the active image area's display size. Two bytes are provided for the horizontal dimensions, two bytes are provided for the vertical dimensions, and one bit is provided for the multiplier, in that order. The values stored here shall be interpreted as 16-bit numbers that provide the size along the horizontal or vertical axis, respectively, in units of 0.1 or 1.0 mm, based on the Image Size Multiplier bit value:

- **Image Size Multiplier bit = 0** – Range of sizes that may be stored for either field is 0.1 through 6,553.5 mm, in each direction
- **Image Size Multiplier bit= 1** – Range of sizes that may be stored for either field is 1.0 through 65,535.0 mm, in each direction

The multiplier's minimum value should be used, when possible, to achieve maximum precision.

For display technologies that do **not** have a precisely fixed image size or native pixel format (e.g., CRT displays), the values stored here shall comprise the maximum image size for any supported format or timing. This is not necessarily the physical screen's limitation – the maximum image size should be the maximum portion of the screen that the manufacturer has designated as usable for image display. If the Horizontal Image Size and Vertical Image Size fields are cleared to all 0s, the system shall **not** make any assumptions regarding display size.

Note: *If a projection display provides an image of indeterminate size, it shall be appropriate to clear both fields to all 0s.*

Table 4-8 defines the Horizontal Image Size and Vertical Image Size fields.

Table 4-8: Horizontal Image Size and Vertical Image Size Field

Offset	Byte #	Bit #	Definition[a] [b]
04h – 03h	1:0	15:0	**Horizontal Image Size** • Range shall be 0.1 through 6,553.5 mm when Image Size Multiplier = 0 • Range shall be 1.0 through 65,535.0 mm when Image Size Multiplier = 1
06h – 05h	1:0	15:0	**Vertical Image Size** • Range shall be 0.1 through 6,553.5 mm when Image Size Multiplier = 0 • Range shall be 1.0 through 65,535.0 mm when Image Size Multiplier = 1

a. If a projection display provides an image of indeterminate size, it shall be appropriate to clear both fields to all 0s.
b. Note that this range is different from other similar ranges defined within this Standard.

4.2.2 Horizontal Pixel Count and Vertical Pixel Count Fields (Native Format)

The 2-byte Horizontal Pixel Count and Vertical Pixel Count fields of the Display Parameters data block shall define the display device's native format, in pixels. For display technologies that do **not** provide a fixed native format (e.g., CRT displays), all four bytes shall contain all 0s. The image or device aspect ratio may **not** be inferred from this information because there is no mandate that the device's pixels shall be square or that the horizontal and vertical directions shall have an equal number of pixels per unit distance.

Table 4-9 defines the Horizontal Pixel Count and Vertical Pixel Count fields.

Table 4-9: Horizontal Pixel Count and Vertical Pixel Count Field (Native Format)

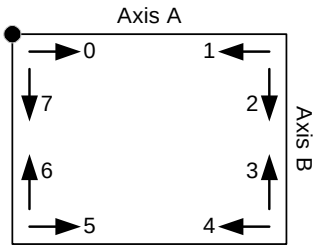
Offset	Byte #	Bit #	Definition
08h – 07h	1:0	15:0	**Horizontal Pixel Count** Native format, horizontal pixels. Pixel quantities range from 0 through 65,535. Note that this range is different from other similar ranges defined within this Standard. 0000h = 0 pixels. … FFFFh = 65,535 pixels.
0Ah – 09h	1:0	15:0	**Vertical Pixel Count** Native format, vertical pixels. Pixel quantities range from 0 through 65,535. Note that this range is different from other similar ranges defined within this Standard. 0000h = 0 pixels. … FFFFh = 65,535 pixels.

4.2.3 Feature Support Flags Field

The Feature Support Flags field of the Display Parameters data block is a single byte that provides flags that shall be used to indicate the display's support for various features or functions.

Table 4-10 defines the Feature Support Flags field.

Table 4-10: Feature Support Flags Field

Offset	Bit #	Definition
0Bh	2:0	**Scan Orientation** Scan Orientation with respect to normal viewing position. Axis A and B map to the horizontal and vertical pixels as seen from the normal viewing position, and may not map to the horizontal and vertical dimensions in the specified timing declarations (see Appendix C for further clarification). Unless otherwise governed by display interface-specific rules (e.g., *DisplayPort* or *HDMI*), the Source device shall support Scan Orientation 0, and may also support other scan orientations. 000b = Left to right, top to bottom. 001b = Right to left, top to bottom. 010b = Top to bottom, right to left. 011b = Bottom to top, right to left. 100b = Right to left, bottom to top. 101b = Left to right, bottom to top. 110b = Bottom to top, left to right. 111b = Top to bottom, left to right. 
	4:3	**Luminance Information** When Native Luminance-related fields in the Display Parameters data block contain positive values, this field indicates how to interpret those fields. If the Native Luminance-related fields contain a negative 0 value, -0, this field shall be cleared to all 0s. 00b = Nonzero maximum luminance information contained within this block is exposed as a minimum guaranteed value. 01b = Nonzero maximum luminance information contained within this block is provided as a guidance for the Source device. ***Note:*** *The Sink device has an option to use -0 values for luminance fields that indicate to the Source device to not use Native Luminance-related fields.* All other values are RESERVED.
	5	**RESERVED** Cleared to 0.
	6	**Color Information** 0 = All color information contained within this block shall be provided in terms of *CIE 1931 x, y* coordinates (default). 1 = All color information contained within this block shall be provided in terms of *CIE 1976 L*u*v and u', v'* coordinates.
	7	**Audio Speaker Information** 0 = Audio speakers shall be integrated into the display (default). 1 = Audio speakers shall not be integrated into the display. Separate speaker connection shall be supplied by an external jack.

4.2.4 Native Color Chromaticity Fields

The Native Color Chromaticity fields shall provide information regarding the display's native color characteristics, including the primary colors' chromaticity aspects and display's **white** points, as defined by the 3-byte fields that are listed for each in Table 4-11. Values may be provided as 12-bit (x, y) color coordinates per *CIE 1931*, –or– as 12-bit (u', v') color coordinates per *CIE 1976 L*u*v and u', v'*. The Color Information bit of the Display Parameters data block's Feature Support Flags field (Offset 0Bh[6]) shall determine whether the colors are indicated in (x, y) or (u', v') coordinate format.

Table 4-11: Native Color Chromaticity Field

Offset	Byte #	Bit #	Definition
0Eh – 0Ch	**Native Color Chromaticity (Primary Color 1 Chromaticity)**		
0Ch	0	7:0	**Primary Color 1 *x* or *u'* Value7:0 or Standard Color Space Identifier** Value ranges from 00h through FFh.
0Dh	1	11:8	**Primary Color 1 *x* or *u'* Value11:8** Value ranges from 0h through Fh.
		15:12	**Primary Color 1 *y* or *v'* Value3:0** Value ranges from 0h through Fh.
0Eh	2	23:16	**Primary Color 1 *y* or *v'* Value11:4** Value ranges from 00h through FFh.
11h – 0Fh	**Native Color Chromaticity (Primary Color 2 Chromaticity)**		
0Fh	0	7:0	**Primary Color 2 *x* or *u'* Value7:0 or Standard Color Space Identifier** Value ranges from 00h through FFh.
10h	1	11:8	**Primary Color 2 *x* or *u'* Value11:8** Value ranges from 0h through Fh.
		15:12	**Primary Color 2 *y* or *v'* Value3:0** Value ranges from 0h through Fh.
11h	2	23:16	**Primary Color 2 *y* or *v'* Value11:4** Value ranges from 00h through FFh.
14h – 12h	**Native Color Chromaticity (Primary Color 3 Chromaticity)**		
12h	0	7:0	**Primary Color 3 *x* or *u'* Value7:0 or Standard Color Space Identifier** Value ranges from 00h through FFh.
13h	1	11:8	**Primary Color 3 *x* or *u'* Value11:8** Value ranges from 0h through Fh.
		15:12	**Primary Color 3 *y* or *v'* Value3:0** Value ranges from 0h through Fh.
14h	2	23:16	**Primary Color 3 *y* or *v'* Value11:4** Value ranges from 00h through FFh.

Table 4-11: Native Color Chromaticity Field (Continued)

Offset	Byte #	Bit #	Definition
17h – 15h	**Native Color Chromaticity (White Point Chromaticity)**		
15h	0	7:0	**White Point *x* or *u′* Value7:0 or Standard Color Space Identifier** Value ranges from 00h through FFh.
16h	1	11:8	**White Point *x* or *u′* Value11:8** Value ranges from 0h through Fh.
		15:12	**White Point *y* or *v′* Value3:0** Value ranges from 0h through Fh.
17h	2	23:16	**White Point *y* or *v′* Value11:4** Value ranges from 00h through FFh.

4.2.4.1 Primary Color Definition and Ordering

"Primary colors" are defined as those that correspond to physical sub-pixels or other sources of pure color, controlled by a given physical or logical channel of the device interface or video signal. The ordering of the primary colors, as provided in this data block, shall correspond to standard ordering of the primary colors as they correspond to the video signal channels as given in the applicable interface/video signal standard. For example, in a typical RGB color representation, "**R**" (**red**), "**G**" (**green**), and "**B**" (**blue**) are the primary colors and are described in that order.

4.2.4.2 White Point Definition and Ordering

"**White** point" describes the color that the display produces when all primary channels are driven to their maximum saturation value (e.g., in an 8-bit color RGB display, **white** is the color that results from all three R, G, and B channels each being driven to a value of 255).

4.2.4.3 Data Format

The chromaticity and **white** point values to be stored in the Native Color Chromaticity fields shall be initially expressed as decimal-format fractional numbers, accurate to the thousandths place.

Each value shall be stored in this data block as a 12-bit binary-format fraction. In such a fractional representation, a value of 1 for the bit immediately to the right of the decimal point (the most significant bit of the value provided, in this case bit 11) represents a value of 2^{-1}. A value of 1 in the rightmost bit (i.e., the least significant bit, or bit 0) represents a value of 2^{-12}.

Using this representation, all color coordinates provided in these bytes should be accurate within ±0.0001 of the actual value. Table 4-12 lists three example 12-bit binary-format fraction representations.

Table 4-12: 12-bit Binary-format Fraction Representation (Examples)

Actual Value	Binary-format Value	Converted Back to Decimal Format
0.610	100111000011b	0.610107
0.307	010011101001b	0.306884
0.150	001001100110b	0.149902

4.2.5 Native Luminance-related Fields

The Native Luminance-related fields provide information related to the display's native luminance in various data transmission conditions with differing maximum and 0 code coverage. The information shall be consistent with the physical display such that the display's actual measurement in the described condition shall result in a similar value to that exposed in DisplayID, within a reasonable tolerance. If luminance information is provided, all Native Luminance-related fields shall contain positive values. If luminance information is **not** provided, all Native Luminance-related fields shall contain a negative 0 value, -0 (0 × 8,000).

Table 4-13 defines the Native Luminance-related fields.

Table 4-13: Native Luminance-related Field

Offset	Byte #	Bit #	Definition
19h – 18h	1:0	15:0	**Native Maximum Luminance (Full Coverage)** Native maximum luminance (in cd/m^2) that shall be physically possible to attain on the display with all pixels programmed to maximum code. Value shall be encoded in *IEEE 754* half-precision binary floating-point format. A positive value indicates that the field contains valid luminance data. A negative 0 value, -0, indicates that the field does not contain valid luminance data and shall not be used. All other negative values are RESERVED.
1Bh – 1Ah	1:0	15:0	**Native Maximum Luminance (10% Rectangular Coverage)** Native maximum luminance (in cd/m^2) that shall be physically possible to attain on the display with a 10% rectangular patch programmed to maximum code in the middle of the screen, while the remainder of the screen is programmed to 0 code. Value shall be encoded in *IEEE 754* half-precision binary floating-point format. A positive value indicates that the field contains valid luminance data. A negative 0 value, -0, indicates that the field does not contain valid luminance data and shall not be used. All other negative values are RESERVED.
1Dh – 1Ch	1:0	15:0	**Native Minimum Luminance** Native minimum luminance (in cd/m^2) that shall be physically possible to attain on the display, using a corner box pattern in which a Source device transmits a code of 0 everywhere except at the 2.5% rectangular patches at the corners of the screen, which are programmed to maximum code. Value shall be encoded in *IEEE 754* half-precision binary floating-point format. A positive value indicates that the field contains valid luminance data. A negative 0 value, -0, indicates that the field does not contain valid luminance data and shall not be used. All other negative values are RESERVED.

4.2.6 Native Color Depth and Display Device Technology Fields

The Native Color Depth and Display Device Technology fields provide information related to the display's native color depth and the technology used by the display panel.

Table 4-14 defines the Native Color Depth and Display Device Technology fields. In Block Revision 1 of the Display Parameters data block, Display Device Theme Preference (Offset 1Eh[7]) was added to **optionally** expose theme preference.

Table 4-14: Native Color Depth and Display Device Technology Field

Offset	Bit #	Definition
1Eh	2:0	**Native Color Depth** Value shall be selected based on the Sink device's ability to use all available bits per component (bpc) for display output rendering. 000b = Not defined. Source device shall apply display interface-specific rules. 001b = 6 bpc. 100b = 12 bpc. 010b = 8 bpc. 101b = 16 bpc. 011b = 10 bpc. 110b and 111b = RESERVED.
	3	**RESERVED** Cleared to 0.
	6:4	**Display Device Technology** Defines the technology used by the display panel. The Source device may optionally use the information provided by this field as a hint to optimize the Source device's content for the relevant technology. The Sink device shall **not** expect a guaranteed behavior of the Source device based on the information provided in this field. 000b = Technology type is not specified. 001b = Active Matrix LCD technology. 010b = Organic LED technology. All other values are RESERVED.
	7	**Display Device Theme Preference** The Source device may **optionally** use the information provided by this field to optimize the Source device's content according to the preference expressed by the Sink device. The Sink device shall **not** expect a guaranteed behavior of the Source device based on the information provided in this field. 0 = No preference. 1 = Dark Theme preferred (lower average-frame brightness is preferred).

4.2.7 Native Gamma EOTF Field

The Native Color Depth field provides information related to the native gamma Electro-Optical Transfer Function (EOTF) that the display supports. Exposing a specific gamma value indicates that the Sink device shall be able to decode content generated by the Source device at that gamma value.

Table 4-15 defines the Native Color Depth field.

Table 4-15: Native Gamma EOTF Field

Offset	Bit #	Definition
1Fh	7:0	**Native Gamma EOTF** Defines the gamma range, from 1.00 through 3.754, as follows: *Field Value = (Gamma × 100) – 100* Value ranges from 00h through FFh. FFh = No gamma information shall be provided.

4.3 Video Timing Mode-related Data Blocks

The Video Timing Mode-related data blocks can declare support for multiple timings, using three different descriptor formats.

Table 4-16 defines the size and order of each Video Timing Mode-related data block field, with all addresses relative to the beginning of the data block.

Table 4-16: Video Timing Mode-related Data Blocks

Offset	Bit #	Definition/Priority	Defined in
00h	7:0	**Video Timing Mode-related Data Blocks** 22h = Type VII Timing – Detailed. 23h = Type VIII Timing – Enumerated Timing Code. 24h = Type IX Timing – Formula-based. 2Ah = Type X Timing – Formula-based.	 4.3.1 4.3.2 4.3.3 4.3.4
01h	**Block Revision and Other Data**		
	2:0	**Block Revision** Revision ranges from 0 through 7. 000b = Revision 0 (default).	
	7:3	**RESERVED** Timing-specific. Cleared to all 0s.	
02h	7:0	**Number of Payload Bytes** Number of payload bytes ranges from 0 through 248. 00h = 0 payload bytes. … F8h = 248 payload bytes.	
03h	7:0	**First Video Timing Mode Byte** Descriptor. Priority 1.	
03h + *N*[a]	7:0	**Second Video Timing Mode Byte** Descriptor, if present. Priority 2.	
…		**…**	

a. N represents the total number of bytes within the DisplayID Section.

Notes: Preferred Timings and Timing Prioritization

As noted in Section 2, most DisplayID structures shall identify at least one timing as the preferred timing for that product; in addition, other timings may also be identified as preferred or supported elsewhere within the DisplayID structure.

A preferred timing is defined as one that shall, in the manufacturer's opinion, result in the product's optimum performance within its intended use or application. If two or more timings are identified as preferred or supported within the DisplayID structure, the Source device shall use the following priorities when selecting which timing to use:

1 *First timing identified in the DisplayID Base Section, either as the first timing block (of any type), within a Video data block, –or– as a timing code within the appropriate data block.*

2 *Other timing (if any) provided in the DisplayID structure that is identified as preferred.*

3 *Other timing (if any) indicated as supported by way of one of the methods listed in item 1.*

4 *Standard 640 × 480, 60-Hz progressive-scan timing that is to be used as a base video mode for any display that does* ***not*** *otherwise identify a supported timing, –or– whose DisplayID information* ***cannot*** *be read.*

4.3.1 Type VII Timing – Detailed Timing Data Block

The Type VII Detailed Timing data block shall be composed of single or multiple 20-byte Detailed Timing descriptors. Type VII Detailed Timing is a **mandatory** data block for product primary use case as a display device and shall contain at least one timing; otherwise, the data block is **optional**.

The DisplayID Structure v2.0 definition of the Type VII Detailed Timing data block is similar to that of the DisplayID Structure v1.2 Type I Detailed Timing data block, except for Bytes 0, 1, 2 of the descriptor fields, which carry the pixel clock information. With Type VII, higher-precision pixel clock support can be exposed. The field's range allows the pixel clock rate to be defined as 0.001 through 16,777.216 MP/s.

Tables 4-17 and 4-18 define the Type VII Detailed Timing data block's structure and descriptor formats, respectively.

Note: *See Appendix A for an example of how the Type VII Detailed Timing data block is used in a fixed-length DisplayID Section.*

Table 4-17: Type VII Timing – Detailed Timing Data Block[a] [b]

Offset	Bit #	Definition/Priority
00h	7:0	**Type VII Timing – Detailed Timing Data Block** 22h.
01h	**Block Revision and Other Data**	
	2:0	**Block Revision** Revision ranges from 0 through 7. 000b = Revision 0. 001b = Revision 1 (redefined Offset 01h[3] as a flag that indicates *DSC* pass-through timing support for all listed timing within the block). 010b = Revision 2 (should be used for new designs).
	3	**For Block Revision 0** **RESERVED** Cleared to all 0s. **For Block Revisions 1 and 2** **Pass-through Timing Support for Target DSC Bits per Pixel** 0 = Same as Revision 0 (standard timing support declaration as *DisplayID v2.0*). 1 = Indicates that the data block's listed timing descriptors are supported with *DSC* pass-through with *DSC* RGB encoding and specific target *DSC* bits per pixel only. For *DisplayPort*, the VESA Organization Vendor-specific data block (see Appendix B) shall exist with relevant pass-through integer and fractional *DSC* bits per pixel declaration when this bit is set to 1; otherwise, the Source device shall disregard the timings. **Use of the bit with *CTA-861-H*** – When the Type VII Detailed Timing data block is packed in a *CTA* data block, shall be cleared to 0.
	6:4	**For Block Revision 0** **RESERVED** Cleared to all 0s. **For Block Revisions 1 and 2** **Number of Payload Bytes (M) in a Timing Descriptor** 000b = 20 + 0 bytes/descriptor, where *M* (bytes) = 20 (initial descriptor size) + field value. All other values are RESERVED.
	7	**RESERVED** Cleared to 0.

Table 4-17: Type VII Timing – Detailed Timing Data Block[a] [b] (Continued)

Offset	Bit #	Definition/Priority
02h	7:0	**Number of Payload Bytes in Block** Number of payload bytes within the block is based on the number of descriptors (*N*) × size of each descriptor (*M*) bytes. **Use of the bit with *CTA-861-H*** – When the Type VII Detailed Timing data block is packed in a *CTA* data block, which separately indicates the *CTA* data block payload length, Offset 02h shall not be present and the size of the Type VII Detailed Timing data block is reduced by 1 byte. All other values are RESERVED.
03h through 03h + *M* – 1	(*M* × 8 – 1):0	**First Type VII Detailed Timing Descriptor** *M*-byte descriptor. Priority 1.
03h + *M* through 03h + 2*M* – 1	(*M* × 8 – 1):0	**Second Type VII Detailed Timing Descriptor** *M*-byte descriptor, if present. Priority 2.
…		**…**
03h + (*N* – 1) × *M* through 03h + (*N* × *M*) – 1	(*M* × 8 – 1):0	**Nth Type VII Detailed Timing Descriptor** *M*-byte descriptor, if present. Priority *N*.

a. N represents the number of Detailed Timing descriptors in the data block.
b. M represents the value of Offset 01h[6:4]. Source device implementation shall accommodate varying field for future extensibility.

Table 4-18: Type VII Detailed Timing Descriptor

Byte #	Bit #	Definition
2:0	**Pixel Clock ÷ 1000**	Rate range is defined as 0.001 through 16,777.216 MP/s. 000000h = 0.001 MP/s. … FFFFFFh = 16,777.216 MP/s.
0	7:0	**Pixel Clock ÷ 1000 7:0**
1	7:0	**Pixel Clock ÷ 1000 15:8**
2	7:0	**Pixel Clock ÷ 1000 23:16**
3	**Timing Options**	
	3:0	**Aspect Ratio** 0h = 1:1. 4h = 16:9. 1h = 5:4. 5h = 16:10. 2h = 4:3. 6h = 64:27. 3h = 15:9. 7h = 256:135. 8h = Aspect ratio shall be calculated by using Bytes 5:4 and 13:12. 9h through Fh = RESERVED.
	4	**Interface Frame Scanning Type** 0 = Progressive scan frame. 1 = Interlaced scan frame.
	6:5	**3D Stereo Support** 00b = This timing shall always be displayed in mono (no stereo). 01b = This timing shall always be displayed in stereo. 10b = This timing shall be displayed in mono or stereo, depending on a user action (such as wearing stereo glasses). 11b = RESERVED
	7	**For Block Revisions 0 and 1** **Preferred Detailed Timing** 1 = Preferred Detailed timing. **For Block Revision 2** **YCC 420 Support** 0 = Explicit YCbCr 4:2:0 support is not indicated for the current timing (other data blocks may be used to indicate support). 1 = Explicit YCbCr 4:2:0 support is indicated for the current timing.

Table 4-18: Type VII Detailed Timing Descriptor (Continued)

Byte #	Bit #	Definition
5:4	**Horizontal Active Image Pixels** Pixel quantities range from 1 through 65,536. 0000h = 1 pixel. … FFFFh = 65,536 pixels.	
4	7:0	**Horizontal Active Image Pixels7:0**
5	7:0	**Horizontal Active Image Pixels15:8**
7:6	**Horizontal Blank Pixels** Pixel quantities range from 1 through 65,536. 0000h = 1 pixel. … FFFFh = 65,536 pixels.	
6	7:0	**Horizontal Blank Pixels7:0**
7	7:0	**Horizontal Blank Pixels15:8**
9:8	**Horizontal Offset (Front Porch)** Pixel quantities range from 1 through 32,768. 0000h = 1 pixel. … 7FFFh = 32,768 pixels.	
8	7:0	**Horizontal Offset (Front Porch)7:0**
9	6:0	**Horizontal Offset (Front Porch)14:8**
	7	**Horizontal Sync Polarity** 0 = Negative. 1 = Positive.
11:10	**Horizontal Sync Width** Pixel quantities range from 1 through 65,536. 0000h = 1 pixel. … FFFFh = 65,536 pixels.	
10	7:0	**Horizontal Sync Width7:0**
11	7:0	**Horizontal Sync Width15:8**

Table 4-18: Type VII Detailed Timing Descriptor (Continued)

Byte #	Bit #	Definition
13:12	**Vertical Active Image Lines** Line quantities range from 1 through 65,536. 0000h = 1 line. … FFFFh = 65,536 lines.	
12	7:0	**Vertical Active Image Lines7:0**
13	7:0	**Vertical Active Image Lines15:8**
15:14	**Vertical Blank Lines** Line quantities range from 1 through 65,536. 0000h = 1 line. … FFFFh = 65,536 lines.	
14	7:0	**Vertical Blank Lines7:0**
15	7:0	**Vertical Blank Lines15:8**
17:16	**Vertical Sync Offset (Front Porch)** Line quantities range from 1 through 32,768. 0000h = 1 line. … 7FFFh = 32,768 lines.	
16	7:0	**Vertical Sync Offset (Front Porch)7:0**
17	6:0	**Vertical Sync Offset (Front Porch)14:8**
	7	**Vertical Sync Polarity** 0 = Negative. 1 = Positive.
19:18	**Vertical Sync Width** Line quantities range from 1 through 65,536. 0000h = 1 line. … FFFFh = 65,536 lines.	
18	7:0	**Vertical Sync Width7:0**
19	15:8	**Vertical Sync Width15:8**

4.3.1.1 Additional Mandates and Information Regarding Borders

This section provides additional information and mandates regarding the use of borders in the Type VII Detailed Timing data block.

Note: *The concept of border areas dates back to the early days of CRT-based computer monitors, and refers to a portion of the displayed image that is outside the addressable area (i.e., the area that the host system's graphics controller uses for displaying information). The video source may set these border areas to* ***white*** *or some other solid color to provide a visible frame around the active video area. Border use is rare in current PC industry practice, and is of questionable value for non-CRT display types.*

For these reasons, borders have been removed from the detailed timing structure and shall be implicitly considered to be cleared to all 0s.

Figure 4-1 provides additional information on the relationship between border times and other defined horizontal and vertical timing parameters.

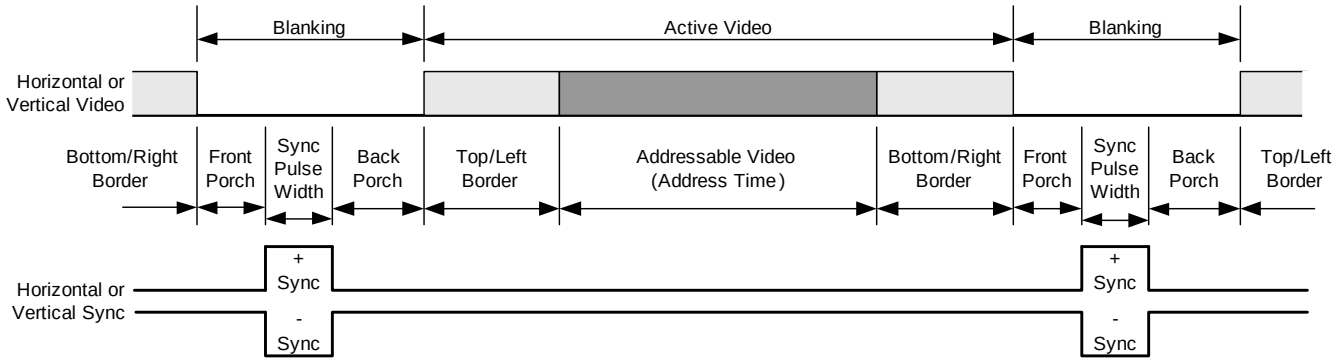


Figure 4-1: Video Timing Parameter Definitions

4.3.2 Type VIII Timing – Enumerated Timing Code Data Block

The Type VIII Enumerated Timing Code data block provides a means for exposing support for timings based on enumerating timing codes within the data block. Timing codes shall be defined by a relevant standards organization such as VESA, *CTA*, or *HDMI*, in *DMT* Timings (*DMT Standard*), *CTA* Video Information Code (VIC) Timings, and/or *HDMI* VIC Timings, respectively. Timing priority shall be in the order listed. A DisplayID Section can have multiple Type VIII Enumerated Timing Code data blocks. Each Type VIII Enumerated Timing Code data block shall include only timings based on a single Timing Code type.

Type VIII Enumerated Timing Code is an **optional** data block.

Note: *Type VIII Enumerated Timing Code data blocks contain CTA VIC timing codes, rather than Short Video Descriptors (SVDs). Native formats shall be inferred from information in the Display Parameters data block, primarily the native format Horizontal Pixel Count and Vertical Pixel Count fields (Offsets 08h – 07h and 0Ah – 09h, respectively) of the Display Parameters data block.*

Table 4-19 defines the Type VIII Enumerated Timing Code data block's structure.

Table 4-19: Type VIII Timing – Enumerated Timing Code Data Block

Offset	Bit #	Definition/Priority
00h	7:0	**Type VIII Timing – Enumerated Timing Code Data Block** 23h.
01h	**Block Revision and Other Data**	
	2:0	**Block Revision** Revision ranges from 0 through 7. 000b = Revision 0. 001b = Revision 1 (should be used for new designs).
	3	**Timing Code Size** 0 = 1-byte descriptor timing code. 1 = 2-byte descriptor timing code.
	4	**RESERVED** Cleared to 0.
	5	**For Block Revision 0** **RESERVED** Cleared to 0. **For Block Revision 1** **YCC 420 Support** 0 = Explicit YCbCr 4:2:0 support is not indicated (other data blocks may be used to indicate support). 1 = Explicit YCbCr 4:2:0 support is indicated.
	7:6	**Timing Code Type** 00b = *DMT* timing code. 01b = *CTA* VIC timing code. 10b = *HDMI* VIC timing code. 11b = RESERVED timing code type.
02h	7:0	**Number of Payload Bytes in Block** Number of payload bytes within the data block ranges from 1 through 248. 01h = 1 byte. … F8h = 248 bytes. All other values are RESERVED. **Use of the bit with *CTA-861-H*** – When the Type VIII Enumerated Timing Code data block is packed in a *CTA* data block, which separately indicates the *CTA* data block payload length, Offset 02h shall not be present and the size of the Type VIII Enumerated Timing Code data block is reduced by 1 byte.

Table 4-19: Type VIII Timing – Enumerated Timing Code Data Block (Continued)

Offset	Bit #	Definition/Priority
03h through 03h + (*N* – 1)[a]	(*N* × 8 – 1):0	**First N-byte Timing Code** *N*-byte descriptor. Priority 1.
03h + *N* through 03h + (2*N* – 1)[a]	(*N* × 8 – 1):0	**Second N-byte Timing Code** *N*-byte descriptor, if present. Priority 2.
03h + 2*N* through 03h + (3*N* – 1)[a]	(*N* × 8 – 1):0	**Third N-byte Timing Code** *N*-byte descriptor, if present. Priority 3.
…		**…**

a. N represents the timing code length that is specified in the Timing Code Size field (i.e., one or two bytes).

4.3.3 Type IX Timing – Formula-based Timing Data Block

The Type IX Formula-based Timing data block provides a means for a Sink device to expose timing support that is derived from an industry-standard formula. Block Revision 0 supports *CVT v1.2* standard blanking timing, *CVT v1.2* reduced blanking timing v1 (RB Timing v1), and *CVT v2.0* RB Timing v2. Timing priority shall be in the order listed.

Type IX Formula-based Timing is an **optional** data block.

Tables 4-20 and 4-21 define the Type IX Formula-based Timing data block's structure and descriptor formats, respectively.

Table 4-20: Type IX Timing – Formula-based Timing Data Block

Offset	Bit #	Definition/Priority
00h	7:0	**Type IX Timing – Formula-based Timing Data Block** 24h.
01h	**Block Revision and Other Data**	
	2:0	**Block Revision** Revision ranges from 0 through 7. 000b = Revision 0 (default).
	7:3	**RESERVED** Cleared to all 0s.
02h	7:0	**Number of Payload Bytes in Block** Number of payload bytes within the data block ranges from 6 through 246 ($1 \le N \le 41$).[a] Value is based on $N \times 6$. 06h = 6 bytes. … F6h = 246 bytes. All other values are RESERVED.
08h – 03h	47:0	**First Type IX Formula-based Timing Descriptor** 6-byte descriptor. Priority 1.
0Eh – 09h	47:0	**Second Type IX Formula-based Timing Descriptor** 6-byte descriptor, if present. Priority 2.
…		**…**

a. N represents the number of formula-based timing descriptors in the data block.

Table 4-21: Type IX Formula-based Timing Descriptor

Byte #	Bit #	Definition
0	**Timing Options**	
	2:0	**Timing Formula/Algorithm** 000b = *CVT* standard blanking timing. 001b = *CVT v1.2* RB Timing v1. 010b = *CVT v2.0* RB Timing v2. All other values are RESERVED.
	3	**RESERVED** Cleared to 0.
	4	**Video Refresh Rate Option** 0 = Descriptor indicates support for one timing with the integer Refresh Rate specified in Byte 5. 1 = Descriptor indicates support for two timings, one with the integer Refresh Rate specified in Byte 5, and one with a fractional refresh rate calculated as Refresh Rate × (1,000 / 1,001).
	6:5	**3D Stereo Support** 00b = This timing shall always be displayed in mono (no stereo). 01b = This timing shall always be displayed in stereo. 10b = This timing shall be displayed in mono or stereo, depending on a user action (such as wearing stereo glasses). 11b = RESERVED.
	7	**RESERVED** Cleared to 0.
2:1	**Horizontal Active Image Pixels** Pixel quantities range from 1 through 65,536. 0000h = 1. … FFFFh = 65,536.	
1	7:0	**Horizontal Active Image Pixels7:0**
2	7:0	**Horizontal Active Image Pixels15:8**

Table 4-21: Type IX Formula-based Timing Descriptor (Continued)

Byte #	Bit #	Definition
4:3	**Vertical Active Image Lines** Line quantities range from 1 through 65,536. 0000h = 1. … FFFFh = 65,536.	
3	7:0	**Vertical Active Image Lines7:0**
4	7:0	**Vertical Active Image Lines15:8**
5	7:0	**Refresh Rate** Refresh rates range from 1 through 256 Hz. 00h = 1 Hz. … FFh = 256 Hz.

4.3.4 Type X Timing – Formula-based Timing Data Block

The Type X Formula-based Timing data block provides a means for a Sink device to expose timing support that is derived from an industry-standard formula by *DisplayID v2.0* (and higher). Block Revision 0 supports *CVT v1.2* with standard blanking timing and RB Timing v1, and *CVT v2.0* RB Timing v2 and v3. The 7-byte descriptor variant of the block also supports a refresh rate field that extends to 1,024 Hz. Note that the Type X Formula-based Timing data block is a superset of the older Type IX Formula-based Timing data block and may accommodate all the Type IX capability exposure, using a 6-byte descriptor variant. Newer Sink devices should use the Type X Formula-based Timing data block.

The Sink device may list its timing priority with respect to timing exposure as a hint. Note, however, that a Source device might not use the Sink device-exposed priority, and instead implement a Source device implementation-specific policy based on link capability and the display's native resolution.

For Native structures, Type X Formula-based Timing is a **mandatory** data block for product primary use case as a display device that supports multiple *CVT v2.0* RB Timing v3 timings; otherwise, the data block is **optional**.

For EDID Extension Sections, Type X Formula-based Timing is an **optional** data block.

Tables 4-22 and 4-23 define the Type X Formula-based Timing data block's structure and descriptor formats, respectively.

Table 4-22: Type X Timing – Formula-based Timing Data Block[a] [b]

Offset	Bit #	Definition/Priority
00h	7:0	**Type X Timing – Formula-based Timing Data Block** 2Ah.
01h	**Block Revision and Other Data**	
	2:0	**Block Revision** Revision ranges from 0 through 7. 000b = Revision 0. All other values are RESERVED.
	3	**RESERVED** Cleared to 0.
	6:4	**Number of Payload Bytes (M) in a Timing Descriptor** Where *M* (bytes) = 6 (initial descriptor size) + field value. 000b = 6 + 0 bytes/descriptor. 001b = 6 + 1 bytes/descriptor. All other values are RESERVED.
	7	**RESERVED** Cleared to 0.

Table 4-22: Type X Timing – Formula-based Timing Data Block[a] [b] (Continued)

Offset	Bit #	Definition/Priority
02h	7:0	**Number of Payload Bytes in Block** Number of payload bytes within the block is based on the number of descriptors (*N*) × size of each descriptor (*M*) bytes. **Use of the bit with *CTA-861-H*** – When the Type X Formula-based Timing data block is packed in a *CTA* data block, which separately indicates the *CTA* data block payload length, Offset 02h shall not be present and the size of the Type X Formula-based Timing data block is reduced by 1 byte. All other values are RESERVED.
03h through 03h + *M* – 1	(*M* × 8 – 1):0	**First Type X Formula-based Timing Descriptor** *M*-byte descriptor. Priority 1.
03h + *M* through 03h + 2*M* – 1	(*M* × 8 – 1):0	**Second Type X Formula-based Timing Descriptor** *M*-byte descriptor, if present. Priority 2.
…		**…**
03h + (*N* – 1) × *M* through 03h + (*N* × *M*) – 1	(*M* × 8 – 1):0	**Nth Type X Formula-based Timing Descriptor** *M*-byte descriptor, if present. Priority *N*.

a. N represents the number of formula-based timing descriptors in the data block.
b. M represents the value of Offset 01h[6:4]. Source device implementation shall accommodate varying field for future extensibility.

Table 4-23: Type X Formula-based Timing Descriptor

Byte #	Bit #	Definition	Byte Length[a]	
			6	7
0	**Timing Options**		✓	✓
	2:0	**Timing Formula/Algorithm** 000b = *CVT v1.2* standard blanking timing. 001b = *CVT v1.2* RB Timing v1. 010b = *CVT v2.0* RB Timing v2. 011b = *CVT v2.0* RB Timing v3. All other values are RESERVED.	✓	✓
	3	The definition of this bit is dependent on Byte 0[2:0]. **For *CVT v1.2* Standard CRT-based timing and RB Timing v1 (Byte 0[2:0] = 000b or 001b, respectively), and *CVT v2.0* RB Timing v2 (Byte 0[2:0] = 010b)** **RESERVED** Cleared to 0. **For *CVT v2.0* RB Timing v3 (Byte 0[2:0] = 011b)** **Early VSync** 0 = VSync location is late (near the end of the VBlank period). 1 = VSync location is early (near the middle of the VBlank period).	✓	✓

Table 4-23: Type X Formula-based Timing Descriptor (Continued)

Byte #	Bit #	Definition	Byte Length[a]	
			6	7
0	4	The definition of this bit is dependent on Byte 0[2:0]. **For *CVT v1.2* Standard CRT-based timing and RB Timing v1 (Byte 0[2:0] = 000b or 001b, respectively)** **RESERVED** Cleared to 0. **For *CVT v2.0* RB Timing v2 (Byte 0[2:0] = 010b)** **Video Refresh Rate Option** 0 = Descriptor indicates support for one timing with the integer Refresh Rate specified in Byte 5. 1 = Descriptor indicates support for two timings, one with the integer Refresh Rate specified in Byte 5, and one with a fractional refresh rate calculated as Refresh Rate × (1,000 / 1,001). **For *CVT v2.0* RB Timing v3 (Byte 0[2:0] = 011b)** **Horizontal Blank Option** 0 = 80-pixel HBlank. 1 = 160-pixel HBlank.	✓	✓
	6:5	**3D Stereo Support** 00b = This timing shall always be displayed in mono (no stereo). 01b = This timing shall always be displayed in stereo. 10b = This timing shall be displayed in mono or stereo, depending on a user action (such as wearing stereo glasses). 11b = RESERVED.	✓	✓
	7	**YCC 420 Support** 0 = Explicit YCbCr 4:2:0 support is not indicated for the current timing (other data blocks may be used to indicate support). 1 = Explicit YCbCr 4:2:0 support is indicated for the current timing.	✓	✓
2:1	**Horizontal Active Image Pixels** Pixel quantities range from 1 through 65,536. 0000h = 1. … FFFFh = 65,536.		✓	✓
1	7:0	**Horizontal Active Image Pixels7:0**	✓	✓
2	7:0	**Horizontal Active Image Pixels15:8**	✓	✓

Table 4-23: Type X Formula-based Timing Descriptor (Continued)

Byte #	Bit #	Definition	Byte Length[a]	
			6	7
4:3	**Vertical Active Image Lines** Line quantities range from 1 through 65,536. 0000h = 1. … FFFFh = 65,536.		✓	✓
3	7:0	**Vertical Active Image Lines7:0**	✓	✓
4	7:0	**Vertical Active Image Lines15:8**	✓	✓
6:5	**Refresh Rate** Refresh rates range from 1 through 1,024 Hz. 0000h = 1 Hz. … 00FFh = 256 Hz.		✓	✓
	… 03FFh = 1,024 Hz.			✓
5	7:0	**Refresh Rate7:0** Refresh rates range from 1 through 256 Hz. ***Note:*** *The 6-byte Type X Formula-based Timing Descriptor ends at this offset and limits the refresh rate to 256 Hz.*	✓	✓

Table 4-23: Type X Formula-based Timing Descriptor (Continued)

<table>
<tr><th>Byte #</th><th>Bit #</th><th>Definition</th><th colspan="2">Byte Length[a]</th></tr>
<tr><th></th><th></th><th></th><th>6</th><th>7</th></tr>
<tr><td>6</td><td>1:0</td><td>Refresh Rate9:8
Valid only for the 7-byte Type X Formula-based Timing Descriptor.
When used in combination with Byte 5[7:0], extends the refresh rate range to 1,024 Hz.</td><td></td><td>✓</td></tr>
<tr><td></td><td>4:2</td><td>Delta Horizontal Blank
Valid only for CVT v2.0 RB Timing v3 (Byte 0[2:0] = 011b).
Provides the delta HBlank pixels that are to be added in addition to the value reported in CVT v2.0 RB Timing v3 Byte 0[4]. Note that the delta value is dependent on the number of pixels indicated by Byte 0[4] (i.e., 80 or 160 pixels) to accommodate the flexibility range of 80 to 200 HBlank pixels.
When CVT v2.0 RB Timing v3 Byte 0[4] = 0 (80 pixels):
HBlank (in pixels) = Byte 6[4:2] value × 8 + 80
When CVT v2.0 RB Timing v3 Byte 0[4] = 1 (160 pixels):
If Byte 6[4:2] ≤ 5:
HBlank (in pixels) = Byte 6[4:2] value × 8 + 160
Else, if 5 < Byte 6[4:2] ≤ 7:
HBlank (in pixels) = (5 – (Byte 6[4:2] value)) × 8 + 160
The following table lists the available values when this field is used in combination with Byte 0[4].
<table>
<tr><th rowspan="2">Field Value</th><th colspan="2">HBlank (pixels)</th></tr>
<tr><th>Byte 0[4] = 0</th><th>Byte 0[4] = 1</th></tr>
<tr><td>000b</td><td>0 × 8 + 80 = 80</td><td>0 × 8 + 160 = 160</td></tr>
<tr><td>001b</td><td>1 × 8 + 80 = 88</td><td>1 × 8 + 160 = 168</td></tr>
<tr><td>010b</td><td>2 × 8 + 80 = 96</td><td>2 × 8 + 160 = 176</td></tr>
<tr><td>011b</td><td>3 × 8 + 80 = 104</td><td>3 × 8 + 160 = 184</td></tr>
<tr><td>100b</td><td>4 × 8 + 80 = 112</td><td>4 × 8 + 160 = 192</td></tr>
<tr><td>101b</td><td>5 × 8 + 80 = 120</td><td>5 × 8 + 160 = 200</td></tr>
<tr><td>110b</td><td>6 × 8 + 80 = 128</td><td>(5 – 6) × 8 + 160 = 152</td></tr>
<tr><td>111b</td><td>7 × 8 + 80 = 136</td><td>(5 – 7) × 8 + 160 = 144</td></tr>
</table></td><td></td><td>✓</td></tr>
</table>

Table 4-23: Type X Formula-based Timing Descriptor (Continued)

<table>
<tr><th>Byte #</th><th>Bit #</th><th>Definition</th><th colspan="2">Byte Length[a]</th></tr>
<tr><th></th><th></th><th></th><th>6</th><th>7</th></tr>
<tr><td>6</td><td>7:5</td><td>Additional Vertical Blank Time
Valid only for CVT v2.0 RB Timing v3 (Byte 0[2:0] = 011b). For all other timing formulas, a Vertical Blank Time of 460 us shall be used.
Additional Vertical Blank Time = [Field Value] × 35 us
Total Vertical Blank Time to be used = 460 us + Additional Vertical Blank Time
The following table lists the available values.
<table>
<tr><th>Field Value</th><th>Additional Vertical Blank Time (us)</th><th>Total Vertical Blank Time (us)</th></tr>
<tr><td>0</td><td>0 × 35 = 0</td><td>460</td></tr>
<tr><td>1</td><td>1 × 35 = 35</td><td>495</td></tr>
<tr><td>2</td><td>2 × 35 = 70</td><td>530</td></tr>
<tr><td>3</td><td>3 × 35 = 105</td><td>565</td></tr>
<tr><td>4</td><td>4 × 35 = 140</td><td>600</td></tr>
<tr><td>5</td><td>5 × 35 = 175</td><td>635</td></tr>
<tr><td>6</td><td>6 × 35 = 210</td><td>670</td></tr>
<tr><td>7</td><td>7 × 35 = 245</td><td>705</td></tr>
</table></td><td></td><td>✓</td></tr>
</table>

a. Bytes 0 through 5 are used for the 6-byte descriptor. Bytes 0 through 6 are used for the 7-byte descriptor.

4.4 Dynamic Video Timing Range Limits Data Block

The Dynamic Video Timing Range Limits data block shall be used to convey the supported timing range of vertical frequencies. Any Sink device that exposes this data block shall indicate support for a seamless, continuous vertical frequency device that can support the seamless timing change for any timing that lies within the specified minimum and maximum pixel rates. Dynamic Video Timing Range Limits is an **optional** data block in the DisplayID Base Section.

The Dynamic Video Timing Range Limits data block is considered to be a legacy data block and should **not** be used for new designs. If support for dynamic video timing range limits is needed in new designs, the Adaptive-Sync data block (see Section 4.9) should be used in its place.

Table 4-24 defines the size and order of each Dynamic Video Timing Range Limits data block field, with all addresses relative to the beginning of the data block.

Table 4-24: Dynamic Video Timing Range Limits Data Block

Offset	Bit #	Definition
00h	7:0	**Dynamic Video Timing Range Limits Data Block** 25h.
01h	**Block Revision and Other Data**	
	2:0	**Block Revision** Revision ranges from 0 through 7. 000b = Revision 0 (default). 001b = Revision 1.
	7:3	**RESERVED** Cleared to all 0s.
02h	7:0	**Number of Payload Bytes in Block** 09h = Data block is composed of nine payload bytes. All other values are RESERVED.
05h – 03h	23:0	**Minimum Pixel Clock** Shall be populated. Specifies the minimum pixel rate above which dynamic video timing changes shall be supported, as per the reported scheme. 000000h = 0.001 MP/s. … FFFFFFh = 16,777.216 MP/s.
08h – 06h	23:0	**Maximum Pixel Clock** Shall be populated. Specifies the maximum pixel rate below which dynamic video timing changes shall be supported, as per the reported scheme. 000000h = 0.001 MP/s. … FFFFFFh = 16,777.216 MP/s.

Table 4-24: Dynamic Video Timing Range Limits Data Block (Continued)

Offset	Bit #	Definition
09h	7:0	**Minimum Vertical Refresh Rate** Shall be populated. Specifies the display device's minimum dynamic vertical frequency. Vertical frequency shall be defined as follows: *(pixel_clock / horizontal_total) / vertical_total* 00h = 0 Hz. … FFh = 255 Hz.
0Ah	7:0	**Maximum Vertical Refresh Rate7:0** Shall be populated. Specifies the display device's maximum dynamic vertical frequency. **For Block Revision 0 (Includes only 0Ah)** 00h = 0 Hz. … FFh = 255 Hz. **For Block Revision 1 (Used in Combination with 0Bh)** Note that this range is different from other similar ranges defined within this Standard. 000h = 0 Hz. … 3FFh = 1,023 Hz.
0Bh	**Dynamic Video Timing Range-related and Support Flags**	
	1:0	**For Block Revision 0** **RESERVED** Cleared to all 0s. **For Block Revision 1 (Used in Combination with 0Ah)** **Maximum Vertical Refresh Rate9:8** Note that this range is different from other similar ranges defined within this Standard. 000h = 0 Hz. … 3FFh = 1,023 Hz.
	6:2	**RESERVED** Cleared to all 0s.
	7	**Seamless Dynamic Video Timing Support** 0 = Seamless Dynamic Video Timing change shall not be supported with a fixed horizontal pixel rate and dynamic vertical blanking. 1= Seamless Dynamic Video Timing change shall be supported with a fixed horizontal pixel rate and dynamic vertical blanking.

4.5 Display Interface Features Data Block

The Display Interface Features data block shall provide information about the supported pixel encoding format along with the supported color depths. The data block also provides information about the supported basic audio rates. Display Interface Features is a **mandatory** data block for product primary use case as a display device; otherwise, the data block is **optional**.

Table 4-25 defines the size and order of each Display Interface Features data block field, with all addresses relative to the beginning of the data block.

Note: *See Appendix A for an example of how the Display Interface Features data block is used in a fixed-length DisplayID Section.*

Table 4-25: Display Interface Features Data Block

Offset	Bit #	Definition	Defined in
00h	7:0	**Display Interface Features Data Block** 26h.	
01h	**Block Revision and Other Data**		
	2:0	**Block Revision** Revision ranges from 0 through 7. 000b = Revision 0 (default).	
	7:3	**RESERVED** Cleared to all 0s.	
02h	7:0	**Number of Payload Bytes in Block** 09h + *N* = Data block is composed of nine + *N* payload bytes.[a]	
03h	7:0	**Supported Interface Color Depth for RGB Encoding** Descriptor.	4.5.1
04h	7:0	**Supported Interface Color Depth for YCbCr 4:4:4 Encoding** Descriptor.	4.5.1
05h	7:0	**Supported Interface Color Depth for YCbCr 4:2:2 Encoding** Descriptor.	4.5.1
06h	7:0	**Supported Interface Color Depth for YCbCr 4:2:0 Encoding** Descriptor.	4.5.1
07h	7:0	**Minimum Pixel Rate at Which YCbCr 4:2:0 Encoding Is Supported** Descriptor.	4.5.2
08h	7:0	**Supported Interface Audio Capability and Feature Flags** Descriptor.	4.5.3
09h	7:0	**Supported Interface Color Space and EOTF Standard Combination 1** Descriptor.	4.5.4

Table 4-25: Display Interface Features Data Block (Continued)

Offset	Bit #	Definition	Defined in
0Ah	7:0	**Supported Interface Color Space and EOTF Standard Combination 2** Descriptor. RESERVED. Cleared to all 0s.	
0Bh	7:0	**Number of Additional Supported Interface Color Space and EOTF (*N*)** Descriptor.[a]	4.5.4
0Bh + 1	7:0	**Additional Supported Interface Color Space and EOTF #1** Descriptor.	4.5.4
…		**…**	
0Bh + *N*[a]	7:0	**Additional Supported Interface Color Space and EOTF #*N*** Descriptor.	4.5.4

a. N represents the number of additional supported interface color space and EOTF combinations.

4.5.1 Supported Interface Color Depth-related Fields

The four Supported Interface Color Depth-related fields – one byte each for RGB, YCbCr 4:4:4, YCbCr 4:2:2, and YCbCr 4:2:0 pixel encodings, as listed in Table 4-27 – define the supported color depths for the corresponding pixel encoding defined in Table 4-26.

Table 4-26: Supported Interface Color Depth-related Fields

Support	Definition
Supported	Color depth is supported for all timings. Supported timing includes all DisplayID exposed timings (i.e., timing exposed using DisplayID timing types and *CTA* VICs) unless a CTA block is exposed that restricts the support to a limited set of timing for a particular pixel encoding (e.g., the YCbCr 4:2:0 capability map data block and/or YCbCr 4:2:0 video data block). If those CTA blocks are exposed, support for that pixel encoding would be limited only to the timings exposed within the restricted set.
No support indicated	Sink device has not explicitly reported a color-depth capability in the Display Interface Features data block; thus, the Source device shall rely on other DisplayID data blocks, –or– an interface-specific rule, to determine color-depth support. Examples of other data blocks that may report color-depth capability include the *HDMI* vendor-specific and *HDMI* Forum vendor-specific data blocks.

Table 4-27: Supported Interface Color Depth-related Field

Offset	Bit #	Definition
03h	**Supported Interface Color Depth for RGB Encoding** 0 = No support indicated. 1 = Supported.	
	0	**6 bits per Primary Color**
	1	**8 bits per Primary Color**
	2	**10 bits per Primary Color**
	3	**12 bits per Primary Color**
	4	**14 bits per Primary Color**
	5	**16 bits per Primary Color**
	7:6	**RESERVED** Cleared to all 0s.

Table 4-27: Supported Interface Color Depth-related Field (Continued)

Offset	Bit #	Definition
04h	**Supported Interface Color Depth for YCbCr 4:4:4 Encoding** 0 = No support indicated. 1 = Supported.	
	0	**6 bits per Primary Color**
	1	**8 bits per Primary Color**
	2	**10 bits per Primary Color**
	3	**12 bits per Primary Color**
	4	**14 bits per Primary Color**
	5	**16 bits per Primary Color**
	7:6	**RESERVED** Cleared to all 0s.
05h	**Supported Interface Color Depth for YCbCr 4:2:2 Encoding** 0 = No support indicated. 1 = Supported.	
	0	**8 bits per Primary Color**
	1	**10 bits per Primary Color**
	2	**12 bits per Primary Color**
	3	**14 bits per Primary Color**
	4	**16 bits per Primary Color**
	7:5	**RESERVED** Cleared to all 0s.
06h	**Supported Interface Color Depth for YCbCr 4:2:0 Encoding** 0 = No support indicated. 1 = Supported.	
	0	**8 bits per Primary Color**
	1	**10 bits per Primary Color**
	2	**12 bits per Primary Color**
	3	**14 bits per Primary Color**
	4	**16 bits per Primary Color**
	7:5	**RESERVED** Cleared to all 0s.

4.5.2 Minimum Pixel Rate at Which YCbCr 4:2:0 Encoding Is Supported Field

The Minimum Pixel Rate at Which YCbCr 4:2:0 Encoding Is Supported field indicates the minimum pixel rate at which the Sink device shall support YCbCr 4:2:0 encoding, as indicated in the Supported Interface Color Depth for YCbCr 4:2:0 Encoding field (Offset 06h). The Source device shall use YCbCr 4:2:0 encoding only for supported modes that meet or exceed the listed pixel rate. A value of 00h in this field shall indicate that the Sink device can support YCbCr 4:2:0 encoding at all supported modes. If the YCbCr 4:2:0 capability map data block and YCbCr 4:2:0 video data block are used to limit the YCbCr 4:2:0 timing support, the field shall be cleared to all 0s and support for YCbCr 4:2:0 pixel encoding is then limited to the timings exposed in the restricted set exposed in the CTA data block (for further details of supported mode mandates, see Section 4.5.1).

Table 4-28 defines Minimum Pixel Rate at Which YCbCr 4:2:0 Encoding Is Supported field.

Table 4-28: Minimum Pixel Rate at Which YCbCr 4:2:0 Encoding Is Supported Field

Offset	Bit #	Definition
07h	7:0	**Minimum Pixel Rate at Which YCbCr 4:2:0 Encoding Is Supported** YCbCr 4:2:0 Minimum Pixel Rate Mandate = 74.25 MP/s × field value 00h = Sink device can support YCbCr 4:2:0 encoding at all supported modes.

4.5.3 Supported Interface Audio Capability and Feature Flags Field

The Supported Interface Audio Capability and Feature Flags field is a single byte that provides single-bit flags, as defined in Table 4-29, that shall be used to indicate the display's support for various features or functions. Bits 7, 6, and 5 define the display's basic audio capabilities.

Table 4-29: Supported Interface Audio Capability and Feature Flags Field

Offset	Bit #	Definition
08h	4:0	**RESERVED** Cleared to all 0s.
	5	**48-kHz Sample Rate Supported** 0 = Audio shall not be supported at the 48-kHz sample rate. 1 = Audio shall be supported at the 48-kHz sample rate.
	6	**44.1-kHz Sample Rate Supported** 0 = Audio shall not be supported at the 44.1-kHz sample rate. 1 = Audio shall be supported at the 44.1-kHz sample rate.
	7	**32-kHz Sample Rate Supported** 0 = Audio shall not be supported at the 32-kHz sample rate. 1 = Audio shall be supported at the 32-kHz sample rate.

4.5.4 Supported Interface Color Space and EOTF-related Fields

The Supported Interface Color Space and EOTF-related fields list the supported standard Color Space and EOTF that the display can decode.

Table 4-30 defines the Supported Interface Color Space and EOTF-related fields.

Table 4-30: Supported Interface Color Space and EOTF-related Fields

Offset	Bit #	Definition
09h	**Supported Interface Color Space and EOTF Standard Combination 1** 0 = No support indicated. 1 = Supported.	
	0	**Color Space and EOTF as Defined in sRGB Specification** Support as defined in *IEC 61966-2-1*.
	1	**Color Space and EOTF as Defined in ITU-R BT.601 Specification** Support as defined in *ITU-R BT.601*.
	2	**Color Space as Defined in ITU-R BT.709 Specification and EOTF as Defined in ITU-R BT.1886 Specification** Support as defined in *ITU-R BT.709* and *ITU-R BT.1886*.
	3	**Color Space and EOTF as Defined in Adobe RGB Specification** Support as defined in *Adobe RGB*.
	4	**Color Space and EOTF as Defined in DCI-P3 Specification** Support as defined in *SMPTE RP 431-2*.
	5	**Color Space and EOTF as Defined in ITU-R BT.2020 Specification** Support as defined in *ITU-R BT.2020*.
	6	**Color Space as Defined in ITU-R BT.2020 Specification and EOTF as Defined in SMPTE ST 2084 Specification** Support as defined in *ITU-R BT.2020* and *SMPTE ST 2084*.
	7	**RESERVED** Cleared to 0.
0Ah	**Supported Interface Color Space and EOTF Standard Combination 2**	
	7:0	**RESERVED** Cleared to all 0s.

Table 4-30: Supported Interface Color Space and EOTF-related Fields (Continued)

Offset	Bit #	Definition
0Bh	**Number of Additional Supported Interface Color Space and EOTF (*N*)**	
	2:0	**Number of Additional Supported Interface Color Space and EOTF (*N*)** Indicates how many additional interface color space and EOTF combinations (*N*) are supported. A corresponding number, *N*, of Additional Supported Interface Color Space and EOTF bytes shall be exposed. This field shall be cleared to all 0s if additional supported interface color space and EOTF bytes do **not** need to be exposed. Value ranges from 000b through 111b.
	7:3	**RESERVED** Cleared to all 0s.
0Bh + 1	**Additional Supported Interface Color Space and EOTF #1**	
	3:0	**Supported Interface EOTF** 0h = Supported EOTF is not defined. Source device shall apply display interface-specific rules. 1h = EOTF shall be as defined in sRGB color space (*IEC 61966-2-1*). 2h = EOTF shall be as defined in *ITU-R BT.601* color space. 3h = EOTF shall be as defined in *ITU-R BT.1886* for *ITU-R BT.709* color space. 4h = EOTF shall be as defined in *Adobe RGB* color space. 5h = EOTF shall be as defined in DCI-P3 color space (*SMPTE RP 431-2*). 6h = EOTF shall be as defined in *ITU-R BT.2020* color space. 7h = Gamma function EOTF within the range of 1.00 through 3.54 (gamma value as per the Native Gamma EOTF field (Offset 1Fh) in the Display Parameters data block). 8h = *SMPTE ST 2084* EOTF. 9h = Hybrid Log EOTF. Ah = Custom EOTF (details shall be defined in another data block). All other values are RESERVED.
	7:4	**Supported Interface Color Space** 0h = Supported color space is not defined in this field. Source device shall apply display interface-specific rules. 1h = sRGB color space (*IEC 61966-2-1*). 2h = *ITU-R BT.601* color space. 3h = *ITU-R BT.709* color space. 4h = *Adobe RGB* color space. 5h = DCI-P3 color space (*SMPTE RP 431-2*). 6h = *ITU-R BT.2020* color space. 7h = Custom color space (details shall be defined in another data block). All other values are RESERVED.
…		**…**

Table 4-30: Supported Interface Color Space and EOTF-related Fields (Continued)

Offset	Bit #	Definition
0Bh + *N*	**Additional Supported Interface Color Space and EOTF #*N***	
	3:0	**Supported Interface EOTF** 0h = Supported EOTF is not defined. Source device shall apply display interface-specific rules. 1h = EOTF shall be as defined in sRGB color space (*IEC 61966-2-1*). 2h = EOTF shall be as defined in *ITU-R BT.601* color space. 3h = EOTF shall be as defined in *ITU-R BT.1886* for *ITU-R BT.709* Color Space. 4h = EOTF shall be as defined in *Adobe RGB* color space. 5h = EOTF shall be as defined in DCI-P3 color space (*SMPTE RP 431-2*). 6h = EOTF shall be as defined in *ITU-R BT.2020* color space. 7h = Gamma function EOTF within the range of 1.00 through 3.54 (gamma value as per the Native Gamma EOTF field (Offset 1Fh) in the Display Parameters data block). 8h = *SMPTE ST 2084* EOTF. 9h = Hybrid Log EOTF. Ah = Custom EOTF (details shall be defined in another data block). All other values are RESERVED.
	7:4	**Supported Interface Color Space** 0h = Supported color space is not defined in this field. Source device shall apply display interface-specific rules. 1h = sRGB color space (*IEC 61966-2-1*). 2h = *ITU-R BT.601* color space. 3h = *ITU-R BT.709* color space. 4h = *Adobe RGB* color space. 5h = DCI-P3 color space (*SMPTE RP 431-2*). 6h = *ITU-R BT.2020* color space. 7h = Custom color space (details shall be defined in another data block). All other values are RESERVED.

4.6 Stereo Display Interface Data Block

For Native structures, Stereo Display Interface is a **mandatory** data block for ARVR HMDs; otherwise, the data block is **optional**, but shall be present whenever the 3D Stereo Support flags in any Timing descriptor indicate that the display is capable of displaying stereoscopic images. The data block provides information regarding the data format that is needed for transmitting stereo image pairs across the interface to the display.

The Stereo Display Interface data block contains one sub-block that defines the interface method to the stereoscopic display. Byte 03h specifies the sub-block's length so that other sub-blocks can be added in the future. The sub-block contains a 1-byte Stereo Interface Method Code field (Offset 04h) and a set of Interface Method-specific parameters. A provision for optionally exposing a supported timing descriptor for a given 3D stereo interface descriptor is also included. The 3D Stereo Timing Support field (Offset 01h[7:6]) shall be used to indicate which stereo method to apply to the specified timing.

Table 4-31 defines the size and order of each Stereo Display Interface data block field, including the corresponding timing support, with all addresses relative to the beginning of the data block. Whenever the 3D Stereo Timing Support field indicates that Timing Code shall be listed as part of the Stereo Display Interface data block (Offset 01h[6] = 1), the data block shall have additional timing descriptor(s) following the Interface Method-specific Parameters field (Offset 05h).

Table 4-31: Stereo Display Interface Data Block[a]

Offset	Bit #	Definition
00h	7:0	**Stereo Display Interface Data Block** 27h.
01h	**Block Revision and Other Data**	
	2:0	**Block Revision** Revision ranges from 0 through 7. 000b = Revision 0 (default).
	5:3	**RESERVED** Cleared to all 0s.
	7:6	**3D Stereo Timing Support** 00b = Data block shall apply only to timing descriptors that explicitly report 3D capability. 01b = Data block shall apply only to timing descriptors that explicitly report 3D capability, and the Timing Code shall be listed as part of the Stereo Display Interface data block. 10b = Data block shall apply to all listed timing descriptors within any of the timing blocks. 11b = Only the Timing Code shall be listed as part of the Stereo Display Interface data block.
02h	7:0	**Number of Payload Bytes in Block** Number of payload bytes is based on $N + 2$ (N is the number of bytes in the stereo interface method-specific parameters; see Table 4-32), and ranges from 02h through FFh.

Table 4-31: Stereo Display Interface Data Block[a] (Continued)

Offset	Bit #	Definition
03h	7:0	**Number of Bytes in Stereo Interface Method** Number of bytes is *N* + 1 (*N* is the number of bytes in the stereo interface method-specific parameters; see Table 4-32).
04h	7:0	**Stereo Interface Method Code** Descriptor. See Section 4.6.1.
05h	7:0	**Interface Method-specific Parameters** Descriptor for interface method-specific parameters (size is *N* bytes; see Table 4-32).
…		**…** Descriptor.
N + 04h	7:0	Descriptor.
N + 05h	**3D Timing Descriptor** Exists only the 3D Stereo Timing Support bit is set (Offset 01h[6] = 1).	
	4:0	**Number of Supported Timing Code** Value ranges from 00h through 1Fh (e.g., *M*).
	5	**RESERVED** Cleared to 0.
	7:6	**Timing Code Type** 00b = *DMT* timing code. 01b = *CTA* VIC timing code. 10b = *HDMI* VIC timing code. 11b = RESERVED timing code type.
N + 06h	7:0	**1-byte Timing Code – First Code** 1-byte descriptor.
…		**…** 1-byte descriptor.
N + *M* + 06h	7:0	**1-byte Timing Code – Mth Code** 1-byte descriptor.

Table 4-31: Stereo Display Interface Data Block[a] (Continued)

Offset	Bit #	Definition
N + *M* + 07h	**3D Timing Descriptor** Exists only the 3D Stereo Timing Support bit is set (Offset 01h[6] = 1).	
	4:0	**Number of Supported Timing Code** Value ranges from 00h through 1Fh (e.g., *P*).
	5	**RESERVED** Cleared to 0.
	7:6	**Timing Code Type** 00b = *DMT* timing code. 01b = *CTA* VIC timing code. 10b = *HDMI* VIC timing code. 11b = RESERVED timing code type.
N + *M* + 08h	7:0	**1-byte Timing Code** 1-byte descriptor.
…		**…** 1-byte descriptor.
N + *M* + *P* + 08h	7:0	**1-byte Timing Code – Pth Code** 1-byte descriptor.

a. M represents the number of timing codes in the first example 3D timing descriptor.
N represents the number of payload bytes in the stereo interface method-specific parameters.
P represents the number of timing codes in the second example 3D timing descriptor.

4.6.1 Stereo Interface Method Code Field

The 1-byte Stereo Interface Method Code field shall encode the interface method, as defined in Table 4-32. Depending on the Code value, there are one, two, or eight bytes of additional parameters that further define the interface data format.

Table 4-32: Stereo Interface Method Codes

Code (Offset 04h Field Value)	Interface Method	# of Parameter Bytes (*N*)	Defined in
00h	Frame/Field Sequential Stereo.	1	4.6.1.1
01h	Side-by-side Stereo.	1	4.6.1.2
02h	Pixel-interleaved Stereo.	8	4.6.1.3
03h	Dual Interface, Left and Right Separate.	1	4.6.1.4
04h	Multi-view.	2	4.6.1.5
05h	Stacked Frame Stereo.	1	4.6.1.6
FEh – 06h	RESERVED.	–	
FFh	Proprietary Stereo Interface Methods.	0	4.6.1.7

4.6.1.1 Stereo Interface Method – Frame/Field Sequential Stereo

The Frame/Field Sequential stereo interface method indicates that the display shall expect an image sequence that consists of left, right, left, etc. The display includes a selection device (e.g., shutter glasses or Z-screen) that derives left/right information through a VESA Standard Connector for Stereoscopic Display Hardware or through another standard method. (Other proprietary interfaces shall be supported only for legacy devices.)

Table 4-33 defines the format of the Frame/Field Sequential stereo interface method in the Stereo Display Interface data block.

Table 4-33: Frame/Field Sequential Stereo Interface Method Stereo Display Interface Data Block

Offset	Bit #	Definition
00h	7:0	**Stereo Display Interface Data Block** 27h.
01h	**Block Revision and Other Data**	
	2:0	**Block Revision** Revision ranges from 0 through 7. 000b = Revision 0 (default).
	6:3	**RESERVED** Cleared to all 0s.
	7	**3D Stereo Timing Support** 0 = Data block shall apply only to timing descriptors that explicitly report 3D capability. 1 = Data block shall apply to all listed timing descriptors within any of the timing blocks.
02h	7:0	**Number of Payload Bytes in Block** 03h = Data block is composed of three payload bytes.
03h	7:0	**Number of Bytes in Stereo Interface Method** 02h = Two bytes.
04h	7:0	**Stereo Interface Method Code** 00h = Frame/Field Sequential Stereo.

Table 4-33: Frame/Field Sequential Stereo Interface Method Stereo Display Interface Data Block (Continued)

Offset	Bit #	Definition
05h	**Stereo Polarity Descriptor**	
	0	**Stereo Polarity Descriptor** ***Note:*** *The polarity shall be 0 for devices that use a VESA Standard Connector for Stereoscopic Display Hardware. For other methods, the polarity should also be 0 (i.e., the Stereo Sync signal is 1 when transmitting a left-eye image).* 0 = Stereo Sync shall be a logical 1 when transmitting a left-eye image, and a logical 0 when transmitting a right-eye image. 1 = Stereo Sync shall be a logical 1 when transmitting a right-eye image, and a logical 0 when transmitting a left-eye image.
	7:1	**RESERVED** Cleared to all 0s.

4.6.1.2 Stereo Interface Method – Side-by-side Stereo

The Side-by-side stereo interface method indicates that the display shall expect a single image, the left half of which corresponds to the first of the stereo pair and the right half of which corresponds to the second of the stereo pair.

No scaling is assumed. For example, an 800 × 600 display that takes side-by-side data would declare **a** Timing descriptor of resolution 1,600 × 600 marked as always being displayed in stereo.

Table 4-34 defines the format of the Side-by-side stereo interface method in the Stereo Display Interface data block.

Table 4-34: Side-by-side Stereo Interface Method Stereo Display Interface Data Block

Offset	Bit #	Definition
00h	7:0	**Stereo Display Interface Data Block** 27h.
01h	**Block Revision and Other Data**	
	2:0	**Block Revision** Revision ranges from 0 through 7. 000b = Revision 0 (default).
	6:3	**RESERVED** Cleared to all 0s.
	7	**3D Stereo Timing Support** 0 = Data block shall apply only to timing descriptors that explicitly report 3D capability. 1 = Data block shall apply to all listed timing descriptors within any of the timing blocks.
02h	7:0	**Number of Payload Bytes in Block** 03h = Data block is composed of three payload bytes.
03h	7:0	**Number of Bytes in Stereo Interface Method** 02h = Two bytes.
04h	7:0	**Stereo Interface Method Code** 01h = Side-by-side Stereo.
05h	**View Identity Descriptor**	
	0	**View Identity Descriptor** 0 = Left half of the image represents the left-eye view. Right half of the image represents the right-eye view. 1 = Left half of the image represents the right-eye view. Right half of the image represents the left-eye view.
	7:1	**RESERVED** Cleared to all 0s.

4.6.1.3 Stereo Interface Method – Pixel-interleaved Stereo

The Pixel-interleaved stereo interface method indicates that the display shall expect a single image in which pixels from the left- and right-eye views are interleaved in a repeating pattern that is defined by the eight parameter bytes.

Table 4-35 defines the format of the Pixel-interleaved stereo interface method in the Stereo Display Interface data block.

Table 4-35: Pixel-interleaved Stereo Interface Method Stereo Display Interface Data Block

Offset	Bit #	Definition
00h	7:0	**Stereo Display Interface Data Block** 27h.
01h	**Block Revision and Other Data**	
	2:0	**Block Revision** Revision ranges from 0 through 7. 000b = Revision 0 (default).
	6:3	**RESERVED** Cleared to all 0s.
	7	**3D Stereo Timing Support** 0 = Data block shall apply only to timing descriptors that explicitly report 3D capability. 1 = Data block shall apply to all listed timing descriptors within any of the timing blocks.
02h	7:0	**Number of Payload Bytes in Block** 0Ah = Data block is composed of 10 payload bytes.
03h	7:0	**Number of Bytes in Stereo Interface Method** 09h = Nine bytes.
04h	7:0	**Stereo Interface Method Code** 02h = Pixel-interleaved Stereo.
0Ch – 05h	63:0	**8x8 Interleave Pattern Descriptor** Defines an 8x8 pixel pattern that starts at the display's top-left pixel, which is then horizontally and vertically repeated until the display's bottom-right pixel is reached. The interleave pattern's first byte defines the topmost line, the next byte the second line, etc. (e.g., bit 7 of each byte defines the leftmost image pixel, bit 6 defines the first pixel from the left, etc.). 0 = Pixel position for the descriptor bit shall be a right-eye image pixel. 1 = Pixel position for the descriptor bit shall be a left-eye image pixel. ***Note:*** *2-way interleaved stereo can easily be represented in the Pixel-interleaved stereo interface method. For example, the 2-way interleaved stereo, right-eye image on even lines can be represented as FF00FF00FF00FF00h, with the first byte (FFh) placed at Offset 05h, the second byte (00h) placed at Offset 06h, etc., with the last byte (00h) placed at Offset 0Ch.*

4.6.1.4 Stereo Interface Method – Dual Interface, Left and Right Separate

The Dual Interface, Left and Right Separate stereo interface method applies to display devices that have dual interfaces. One interface shall carry the image sequence for the left-eye view, and the other interface shall carry the image sequence for the right-eye view. To support a wide variety of stereo display techniques, this interface method shall support optional mirroring.

Table 4-36 defines the format of the Dual Interface, Left and Right Separate stereo interface method in the Stereo Display Interface data block.

Table 4-36: Dual Interface, Left and Right Separate Stereo Interface Method Stereo Display Interface Data Block

Offset	Bit #	Definition
00h	7:0	**Stereo Display Interface Data Block** 27h.
01h	**Block Revision and Other Data**	
	2:0	**Block Revision** Revision ranges from 0 through 7. 000b = Revision 0 (default).
	6:3	**RESERVED** Cleared to all 0s.
	7	**3D Stereo Timing Support** 0 = Data block shall apply only to timing descriptors that explicitly report 3D capability. 1 = Data block shall apply to all listed timing descriptors within any of the timing blocks.
02h	7:0	**Number of Payload Bytes in Block** 03h = Data block is composed of three payload bytes.
03h	7:0	**Number of Bytes in Stereo Interface Method** 02h = Two bytes.
04h	7:0	**Stereo Interface Method Code** 03h = Dual Interface, Left and Right Separate.

Table 4-36: Dual Interface, Left and Right Separate Stereo Interface Method Stereo Display Interface Data Block (Continued)

Offset	Bit #	Definition
05h	**Left and Right Polarity and Mirroring Descriptors**	
	0	**Left and Right Polarity Descriptor** ***Note:*** *If the Source device encounters two left- or two right-eye view devices, the Source device shall arbitrarily assign left-eye view to one device and right-eye view to the other device.* 0 = This interface carries the right-eye view. 1 = This interface carries the left-eye view.
	2:1	**Mirroring Descriptor** 00b = No mirroring. 01b = Left/right are mirrored. 10b = Top/bottom are mirrored. 11b = RESERVED.
	7:3	**RESERVED** Cleared to all 0s.

4.6.1.5 Stereo Interface Method – Multi-view

The Multi-view stereo interface method indicates a display with more than two views. For example, in addition to the standard display, a lenticular or barrier-type auto-stereoscopic display is also included.

Table 4-37 defines the format of the Multi-view stereo interface method in the Stereo Display Interface data block.

Table 4-37: Multi-view Stereo Interface Method Stereo Display Interface Data Block

Offset	Bit #	Definition
00h	7:0	**Stereo Display Interface Data Block** 27h.
01h	**Block Revision and Other Data**	
	2:0	**Block Revision** Revision ranges from 0 through 7. 000b = Revision 0 (default).
	6:3	**RESERVED** Cleared to all 0s.
	7	**3D Stereo Timing Support** 0 = Data block shall apply only to timing descriptors that explicitly report 3D capability. 1 = Data block shall apply to all listed timing descriptors within any of the timing blocks.
02h	7:0	**Number of Payload Bytes in Block** 04h = Data block is composed of four payload bytes.
03h	7:0	**Number of Bytes in Stereo Interface Method** 03h = Three bytes.
04h	7:0	**Stereo Interface Method Code** 04h = Multi-view.
05h	7:0	**Number of Views Descriptor** Indicates the number of discrete full color views that shall be needed to compose the image to drive this display. Number of Views descriptors shall be greater than 2 (value of 03h or higher).
06h	7:0	**View Interleaving Method Code Descriptor** VESA shall assign the view-interleaving method code at the request of a display manufacturer. The manufacturer shall provide VESA with a description of the view-interleaving method at the time of the request. Number of View-interleaving Method Code descriptors ranges from 0 through 255. 00h = 0 descriptors. … FFh = 255 descriptors.

4.6.1.6 Stereo Interface Method – Stacked Frame Stereo

The Stacked Frame stereo interface method indicates that the display shall expect a single image, the top and bottom portions of which correspond to the first and second of the stereo pair, respectively. Between the two image portions are additional unused image lines whose count is equal to the timing's vertical blank.

No scaling is assumed. For example, an 800 × 600 display with a vertical blank of 36 lines that takes stacked frame stereo would declare a Timing descriptor of resolution 800 × 600. This timing shall be used as-is for monoscopic timings. For stereo timings, resolution output to the display should be increased to 800 × 1,236 (i.e., 800 × 600 for the top and bottom portions, plus an additional 36 unused lines between them).

Table 4-38 defines the format of the Stacked Frame stereo interface method in the Stereo Display Interface data block.

Table 4-38: Stacked Frame Stereo Interface Method Stereo Display Interface Data Block

Offset	Bit #	Definition
00h	7:0	**Stereo Display Interface Data Block** 27h.
01h	**Block Revision and Other Data**	
	2:0	**Block Revision** Revision ranges from 0 through 7. 000b = Revision 0 (default).
	6:3	**RESERVED** Cleared to all 0s.
	7	**3D Stereo Timing Support** 0 = Data block shall apply only to timing descriptors that explicitly report 3D capability. 1 = Data block shall apply to all listed timing descriptors within any of the timing blocks.
02h	7:0	**Number of Payload Bytes in Block** 03h = Data block is composed of three payload bytes.
03h	7:0	**Number of Bytes in Stereo Interface Method** 02h = Two bytes.
04h	7:0	**Stereo Interface Method Code** 05h = Stacked Frame Stereo.
05h	**View Identity Descriptor**	
	0	**View Identity Descriptor** 0 = Top and bottom portions of the image represent the left- and right-eye views, respectively. 1 = RESERVED. **Shall not be used.**
	7:1	**RESERVED** Cleared to all 0s. **Shall not be used.**

4.6.1.7 Stereo Interface Method – Proprietary Stereo Interface Methods

For stereo displays that use a proprietary interface method, a Stereo Display Interface data block is needed and the Stereo Interface Method Code should be programmed to FFh. Elsewhere in the DisplayID structure, vendors should include a Vendor-specific data block that defines the stereo parameters for the display's proprietary stereo interface method.

Table 4-39 defines the format of the Proprietary stereo interface method in the Stereo Display Interface data block.

Table 4-39: Proprietary Stereo Interface Methods Stereo Display Interface Data Block

Offset	Bit #	Definition
00h	7:0	**Stereo Display Interface Data Block** 27h.
01h	**Block Revision and Other Data**	
	2:0	**Block Revision** Revision ranges from 0 through 7. 000b = Revision 0 (default).
	6:3	**RESERVED** Cleared to all 0s.
	7	**3D Stereo Timing Support** 0 = Data block shall apply only to timing descriptors that explicitly report 3D capability. 1 = Data block shall apply to all listed timing descriptors within any of the timing blocks.
02h	7:0	**Number of Payload Bytes in Block** 02h = Data block is composed of two payload bytes.
03h	7:0	**Number of Bytes in Stereo Interface Method** 01h = One byte.
04h	7:0	**Stereo Interface Method Code** FFh = Proprietary Stereo Interface Methods.

4.7 Tiled Display Topology Data Block

The Tiled Display Topology data block shall be used to define whether tiled displays are implemented in a single enclosure, –or– as a collection of separate physical displays configured within a tiled display topology. The Source device may use information provided by the data block to determine the best way to automatically configure the display and/or to provide useful information to the user with regard to display configuration. Tiled Display Topology is a **mandatory** data block for product primary use case as a display device that supports tiled display; otherwise, the data block is **optional**.

Table 4-40 defines the size and order of each Tiled Display Topology data block field, with all addresses relative to the beginning of the data block.

Table 4-40: Tiled Display Topology Data Block

Offset	Bit #	Definition	Defined in
00h	7:0	**Tiled Display Topology Data Block** 28h.	
01h	**Block Revision and Other Data**		
	2:0	**Block Revision** Revision ranges from 0 through 7. 000b = Revision 0 (default).	
	7:3	**RESERVED** Cleared to all 0s.	
02h	7:0	**Number of Payload Bytes in Block** 16h = Data block is composed of 22 payload bytes.	
03h	7:0	**Tiled Display and Tile Capabilities**	4.7.1
06h, 04h	7:4, 7:0	**Tiled Display Topology**	4.7.2
06h – 05h	3:0, 7:0	**Tile Location**	4.7.2
0Ah – 07h	31:0	**Tile Size**	4.7.3
0Fh – 0Bh	39:0	**Tile Pixel Multiplier and Tile Bezel-related Information**	4.7.4
18h – 10h	71:0	**Tiled Display Topology ID**	4.7.5

4.7.1 Tiled Display and Tile Capabilities Field

The Tiled Display and Tile Capabilities field details the tiled display behavior of various tile configurations (driven by the Source device) that shall expose the tiled display physical enclosure information.

Table 4-41 defines the Tiled Display and Tile Capabilities field in the Tiled Display Topology data block.

Table 4-41: Tiled Display and Tile Capabilities Field

Offset	Bit #	Definition
03h	2:0	**Tile Behavior when It Is the Only Tile Receiving an Image from the Source** 000b = Behavior cannot be described by other values defined in this revision of the data block. 001b = Image is displayed at the location specified by the Tile Location field (Offsets 06h – 05h). 010b = Image is scaled to fit the entire tiled display. 011b = Image is cloned to all other tiles within the entire tiled display. All other values are RESERVED.
	4:3	**Tile Behavior when More than One Tile and Fewer than Total Number of Tiles are Driven by the Source** ***Note:*** *When the total number of tiles is two, this field (bits 4:3) becomes* ***don't care*** *because the number of tiles driven by the Source device cannot be more than one and fewer than the total number of tiles.* 00b = Behavior cannot be described by other values defined in this revision of the data block. 01b = When this tile is receiving an image, the image is displayed at the location specified by the Tile Location field (Offsets 06h – 05h). All other values are RESERVED.
	5	**RESERVED** Cleared to 0.
	6	**Tile Bezel Descriptor** 0 = Tile Bezel Information descriptor is not available. Offsets 0Ch through 0Fh shall be cleared to all 0s. ***Note:*** *The Tile Pixel Multiplier shall be cleared to all 0s for this case.* 1 = Offsets 0Ch through 0Fh contain the Tile Bezel Information descriptor. The Tile Pixel Multiplier in the Tile Pixel Multiplier and Tile Bezel-related Information field at Offset 0Bh shall be a nonzero value. When bit 7 is set to 1, this bit shall also be set to 1.
	7	**Physical Display Enclosure** 0 = Tiled display consists of multiple physical display enclosures. 1 = Tiled display is within a single physical display enclosure.

Note: *If a Sink device exposes a non-native mode in DisplayID, the Source device may transmit that mode as-is and the Sink device shall appropriately handle stitching of the tiles across the seam for a single enclosure display case. The stitching shall work across all supported scaling options that the Sink device exposes for that resolution in the on-screen display. If the Sink device cannot handle stitching across the seam with a scaled image for a given resolution, the Sink device should do one of the following:*

- ***Not*** *report the mode in DisplayID, –or–*
- *If the mode is reported in DisplayID, the Sink device should center the image and* ***not*** *expose the scaling option in the on-screen display for that resolution*

4.7.2 Tiled Display Topology and Tile Location Fields

The Tiled Display Topology and Tile Location fields define the tiled display topology and tile location. The lower four bits and upper two bits of the Tiled Display Topology fields are located at Offsets 04h and 06h, respectively. The lower four bits and upper two bits of the Tile Location fields are located at Offsets 05h and 06h, respectively.

Table 4-42 defines the Tiled Display Topology and Tile Location fields in the Tiled Display Topology data block.

Table 4-42: Tiled Display Topology and Tile Location Fields

Offset	Bit #	Definition
04h	**Total Number of Tiles (Low Bits)**	
	3:0	**Total Number of Vertical Tiles3:0** Value ranges from 0h through Fh. Used in combination with Offset 06h[5:4] (high bits) to define the total number of vertical tiles, which ranges from 1 through 64.
	7:4	**Total Number of Horizontal Tiles3:0** Value ranges from 0h through Fh. Used in combination with Offset 06h[7:6] (high bits) to define the total number of horizontal tiles, which ranges from 1 through 64.
05h	**Tile Location (Low Bits)**	
	3:0	**Vertical Tile Location3:0** Value ranges from 0h through Fh. Used in combination with Offset 06h[1:0] (high bits) to define the vertical tile location, which ranges from 1 through 64.
	7:4	**Horizontal Tile Location3:0** Value ranges from 0h through Fh. Used in combination with Offset 06h[3:2] (high bits) to define the horizontal tile location, which ranges from 1 through 64.

Table 4-42: Tiled Display Topology and Tile Location Fields (Continued)

Offset	Bit #	Definition
06h	**Tile Location and Total Number of Tiles (High Bits)**	
	1:0	**Vertical Tile Location5:4** Value ranges from 0h through 3h. Used in combination with Offset 05h[3:0] (low bits) to define the vertical tile location, which ranges from 1 through 64.
	3:2	**Horizontal Tile Location5:4** Value ranges from 0h through 3h. Used in combination with Offset 05h[7:4] (low bits) to define the horizontal tile location, which ranges from 1 through 64.
	5:4	**Total Number of Vertical Tiles5:4** Value ranges from 0h through 3h. Used in combination with Offset 04h[3:0] (low bits) to define the total number of vertical tiles, which ranges from 1 through 64.
	7:6	**Total Number of Horizontal Tiles5:4** Value ranges from 0h through 3h. Used in combination with Offset 04h[7:4] (low bits) to define the total number of horizontal tiles, which ranges from 1 through 64.

4.7.3 Tile Size Field

The Tile Size field defines the horizontal and vertical size of individual tiles (and **not** the size of the entire tiled display) at native resolution.

Table 4-43 defines the Tile Size field in the Tiled Display Topology data block.

Table 4-43: Tile Size Field

Offset	Byte #	Bit #	Definition
07h	0	7:0	**Horizontal Size7:0** Value ranges from 00h through FFh. Used in combination with Offset 08h[7:0] (high bits) to define the horizontal tile size, which ranges from 1 through 65,536 pixels.
08h	1	7:0	**Horizontal Size15:8** Value ranges from 00h through FFh. Used in combination with Offset 07h[7:0] (low bits) to define the horizontal tile size, which ranges from 1 through 65,536 pixels.
09h	2	7:0	**Vertical Size7:0** Value ranges from 00h through FFh. Used in combination with Offset 0Ah[7:0] (high bits) to define the vertical tile size, which ranges from 1 through 65,536 lines.
0Ah	3	7:0	**Vertical Size15:8** Value ranges from 00h through FFh. Used in combination with Offset 09h[7:0] (low bits) to define the vertical tile size, which ranges from 1 through 65,536 lines.

4.7.4 Tile Pixel Multiplier and Tile Bezel-related Information Fields

The Tile Pixel Multiplier and Tile Bezel-related Information fields define the Tile Pixel Multiplier and Tile Bezel-related information. If bezel information is indicated to be present (i.e., bit 6 of the Tiled Display and Tile Capabilities field is set (Offset 03h[6] = 1), the Tile Pixel Multiplier value shall be nonzero.

Table 4-44 defines the Tile Pixel Multiplier and Tile Bezel-related Information fields in the Tiled Display Topology data block.

Note: *The bezel size in pixels calculation was originally based on the Tile Pixel Multiplier representing a unit of pixels per centimeter and bezel size in millimeters; however, if needed, any generic multiplier can be used to achieve higher pixel precision.*

Table 4-44: Tile Pixel Multiplier and Tile Bezel-related Information Fields

Offset	Byte #	Bit #	Definition
0Bh	0	7:0	**Tile Pixel Multiplier** Pixel multiplier value ranges from 0 through 255. 00h = 0. … FFh = 255.
0Ch	1	7:0	**Tile Top Bezel Size** *Top Bezel in pixels = (Tile Pixel Multiplier × Tile Top Bezel Size × 0.1)* Bezel size ranges from 0 through 255. 00h = 0. … FFh = 255.
0Dh	2	7:0	**Tile Bottom Bezel Size** *Bottom Bezel in pixels = (Tile Pixel Multiplier × Tile Bottom Bezel Size × 0.1)* Bezel size ranges from 0 through 255. 00h = 0. … FFh = 255.

Table 4-44: Tile Pixel Multiplier and Tile Bezel-related Information Fields (Continued)

Offset	Byte #	Bit #	Definition
0Eh	3	7:0	**Tile Right Bezel Size** *Right Bezel in pixels = (Tile Pixel Multiplier × Tile Right Bezel Size × 0.1)* Bezel size ranges from 0 through 255. 00h = 0. … FFh = 255.
0Fh	4	7:0	**Tile Left Bezel Size** *Left Bezel in pixels = (Tile Pixel Multiplier × Tile Left Bezel Size × 0.1)* Bezel size ranges from 0 through 255. 00h = 0. … FFh = 255.

4.7.5 Tiled Display Topology ID-related Fields

The Tiled Display Manufacturer/Vendor ID, Tiled Display Product ID Code, and Tiled Display Serial Number fields comprise the Tiled Display Topology ID field descriptors. The Source device uses these descriptors to uniquely identify the topology of single enclosure and multiple enclosure (i.e., a group of discrete displays) topologies. These fields shall be populated with valid, unique values. For single enclosure topologies, these fields should match the Product Identification data block fields (although even if the Serial Number field in the Product Identification data block is unused, the Tiled Display Serial Number shall be populated with a unique value). For multiple enclosure topologies, these fields may differ from the Product Identification data block fields because the fields are expected to be unique to each discrete display.

Table 4-45 defines the Tiled Display Topology ID-related fields in the Tiled Display Topology data block.

Note: *Associating tiles with a specific tiled display is necessary to be able to produce expected behavior when more than one tiled display is connected to a Source device. Without this correct association, tiles from different tiled displays may be grouped together by the Source device, thereby resulting in tiled images potentially being transmitted to, and displayed on, the wrong physical display device.*

Table 4-45: Tiled Display Topology ID-related Fields

Offset	Byte #	Bit #	Definition
12h – 10h	2:0	23:0	**Tiled Display Manufacturer/Vendor ID** Descriptor.
14h – 13h	4:3	15:0	**Tiled Display Product ID Code** Descriptor.
18h – 15h	8:5	31:0	**Tiled Display Serial Number** Descriptor.

4.7.5.1 Tiled Display Manufacturer/Vendor ID Field

The Tiled Display Manufacturer/Vendor ID field listed in Table 4-46 is a **mandatory** element of the Tiled Display Topology data block. The Tiled Display Manufacturer/Vendor ID field contains the display manufacturer's 3-byte IEEE OUI code, which identifies the display's manufacturer or vendor.

Table 4-46 defines the Tiled Display Manufacturer/Vendor ID field in the Tiled Display Topology data block.

The IEEE issues IEEE OUIs. Contact the IEEE at standards.ieee.org/products-services/regauth/oui/index.html for registration and/or further details.

Table 4-46: Tiled Display Manufacturer/Vendor ID Field

Offset	Byte #	Bit #	Definition
10h	0	7:0	**IEEE OUI First Byte** Byte code.
11h	1	7:0	**IEEE OUI Second Byte** Byte code.
12h	2	7:0	**IEEE OUI Third Byte** Byte code.

4.7.5.2 Tiled Display Product ID Code Field

The Tiled Display Product ID Code field listed in Table 4-45 is a **mandatory** element of the Tiled Display Topology data block. The Tiled Display Product ID Code field contains a 2-byte vendor-assigned product code. The field's length helps to differentiate between multiple models from the same manufacturer. If this field is used to represent a Product ID code (e.g., a model number), the number shall be stored in hexadecimal format with the LSB listed first.

Table 4-47 defines the Tiled Display Product ID Code field in the Tiled Display Topology data block.

Table 4-47: Tiled Display Product ID Code Field

Offset	Byte #	Bit #	Definition
13h	0	7:0	**Product ID Code7:0** Value ranges from 00h through FFh.
14h	1	7:0	**Product ID Code15:8** Value ranges from 00h through FFh.

4.7.5.3 Tiled Display Serial Number Field

The Tiled Display Serial Number field defined in Table 4-45 is a **mandatory** element of the Tiled Display Topology data block. The Tiled Display Serial Number field is a 32-bit serial number that shall be used to differentiate between individual instances of the same display product model. This field's bit order follows the order listed in Table 4-48. The four bytes of the serial number are listed with the LSB first. The serial number ranges from 1 through 4,294,967,295. This serial number shall be a number only – it shall **not** represent an ASCII code. Value 00000000h is RESERVED and shall **not** be used.

Note: *Unlike the Serial Number field in the Product Identification data block, a value of 0 does* ***not*** *indicate that this field is unused.*

Table 4-48: Tiled Display Serial Number Field

Offset	Byte #	Bit #								Definition
		7	6	5	4	3	2	1	0	
15h	0	7	6	5	4	3	2	1	0	**ID Serial Number**
16h	1	15	14	13	12	11	10	9	8	
17h	2	23	22	21	20	19	18	17	16	
18h	3	31	30	29	28	27	26	25	24	

4.8 ContainerID Data Block

The ContainerID data block provides a unique identifier that shall be used to associate the display device, and other features (e.g., audio devices) described in the DisplayID structure, with instances of other related devices (notably *USB* devices). Related devices shall typically reside within the same physical enclosure (e.g., a display device and camera device that are part of the same monitor enclosure). ContainerID is a **mandatory** data block for product primary use case as a multi-function display device; otherwise, the data block is **optional**.

Table 4-49 defines the size and order of each ContainerID data block field, with all addresses relative to the beginning of the data block.

Table 4-49: ContainerID Data Block

Offset	Bit #	Definition	Defined in
00h	7:0	**ContainerID Data Block** 29h.	
01h	**Block Revision and Other Data**		
	2:0	**Block Revision** Revision ranges from 0 through 7. 000b = Revision 0 (default).	
	7:3	**RESERVED** Cleared to all 0s.	
02h	7:0	**Number of Payload Bytes in Block** 10h = Data block is composed of 16 payload bytes.	
12h – 03h	127:0	**ContainerID** 16-byte Universally Unique Identifier (UUID).	4.8.1

4.8.1 ContainerID Field

The ContainerID field is a 128-bit UUID that is unique to an individual device instance. The same ContainerID value provided in the ContainerID data block should be provided by other instances of devices that:

- Reside within the same enclosure, –or–
- Are otherwise to be associated with the specific device instance that is described by the ContainerID data block

The 16-byte UUID is typically displayed in human-readable form as a series of hexadecimal characters and hyphens that separate the bytes into fields, such as:

11223344-5566-7788-99AA-BBCCDDEEFF00h

The individual fields, bytes, or bits of a UUID may have different meanings, depending on how the UUID is constructed; however, the 16-byte UUID itself shall be interpreted as a 16-byte binary-format value with no magnitude. The UUID's first byte (11h in the example) is placed at Offset 03h, the second byte (22h in the example) is placed at Offset 04h, etc., with the last byte (00h in the example) placed at Offset 12h.

Note: *For details regarding how to generate a UUID, see* IETF RFC 4122.

Table 4-50 defines the ContainerID field in the ContainerID data block.

Table 4-50: ContainerID Field

Offset	Byte #	Bit #	Definition
03h	0	7:0	**UUID First Byte** Byte code.
04h	1	7:0	**UUID Second Byte** Byte code.
…			**…**
12h	15	7:0	**UUID Sixteenth Byte** Byte code.

4.9 Adaptive-Sync Data Block

The Adaptive-Sync data block shall be used only by a Sink device to convey Adaptive-Sync operation modes and supported refresh rate range capability on a VESA CERTIFIED Adaptive-Sync Sink device by *DisplayID v2.0* (and higher). The Sink device shall present a maximum of one Adaptive-Sync data block. Adaptive-Sync is a **mandatory** data block for product primary use case as a display device that supports Adaptive-Sync; otherwise, the data block is **optional**.

Table 4-51 defines the size and order of each Adaptive-Sync data block field, with all addresses relative to the beginning of the data block. Table 4-52 defines the data block's descriptor formats.

Table 4-51: Adaptive-Sync Data Block[a] [b]

Offset	Bit #	Definition/Priority
00h	7:0	**Adaptive-Sync Data Block** 2Bh.
01h	**Block Revision and Other Data**	
	2:0	**Block Revision** Revision ranges from 0 through 7. 000b = Revision 0. All other values are RESERVED.
	3	**RESERVED** Cleared to 0.
	6:4	**Number of Payload Bytes (M) in an Adaptive-Sync Operation Mode and Range Descriptor** Where *M* (bytes) = 6 (initial descriptor size) + field value. 000b = 6 + 0 bytes/descriptor (defined as part of Adaptive-Sync, Revision 0). All other values are RESERVED.
	7	**RESERVED** Cleared to 0.
02h	7:0	**Number of Payload Bytes in Block** Number of payload bytes within the block is based on the number of descriptors (*N*) × size of each descriptor (*M*) bytes. All other values are RESERVED.

Table 4-51: Adaptive-Sync Data Block[a] [b] (Continued)

Offset	Bit #	Definition/Priority
03h through 03h + *M* – 1	(*M* × 8 – 1):0	**First Adaptive-Sync Operation Mode and Range Descriptor** *M*-byte descriptor.
03h + *M* through 03h + 2*M* – 1	(*M* × 8 – 1):0	**Second Adaptive-Sync Operation Mode and Range Descriptor** *M*-byte descriptor, if present.
…		**…**
03h + (*N* – 1) × *M* through 03h + (*N* × *M*) – 1	(*M* × 8 – 1):0	**Nth Adaptive-Sync Operation Mode and Range Descriptor** *M*-byte descriptor, if present.

a. N represents the number of Detailed Timing descriptors in the data block.
b. M represents the value of Offset 01h[6:4]. Source device implementation shall accommodate varying field for future extensibility.

Table 4-52: Adaptive-Sync Operation Mode and Range Descriptor

Byte #	Bit #	Definition
0	**Adaptive-Sync Operation and Range Information**	
	0	**Adaptive-Sync Range** 0 = Non-native panel range. (The display implements buffering to support the declared Adaptive-Sync range, and may repeat frames as necessary.) 1 = Native panel range. (The display does not implement buffering to support the declared Adaptive-Sync range, and does not repeat frames.)
	1	**Successive Frame Duration Increase Tolerance for Meeting VESA Adaptive Sync Flicker Performance** 0 = Flicker performance is met in any duration increase within the refresh rate range, but may cause up to a single base video frame period jitter impact. 1 = Flicker performance is met in any duration increase within Byte 1. ***Note:*** *Flicker performance is met in any duration increase within the refresh rate range without jitter impact when either of the following conditions are met:* • *Byte 1 = 00h, –or–* • *Byte 1 ≥ the delta between the maximum frame duration (= minimum refresh rate) and the minimum frame duration (= maximum refresh rate)*
	3:2	**Supported Adaptive-Sync Modes** 00b = Fixed-Average VTotal (FAVT) mode is supported. 01b = Both Fixed-Average VTotal and Adaptive VTotal modes (FAVT and AVT, respectively) are supported. All other values are RESERVED.

Table 4-52: Adaptive-Sync Operation Mode and Range Descriptor (Continued)

Byte #	Bit #	Definition
0	4	**Seamless Transition of Adaptive-Sync Mode and Range Not Supported** 0 = Seamless transition to and from current Adaptive-Sync mode and range is supported. 1 = Seamless transition to and from current Adaptive-Sync mode and range is not supported.
	5	**Successive Frame Duration Decrease Tolerance for Meeting VESA Adaptive Sync Flicker Performance** 0 = Flicker performance is met in any duration decrease within the refresh rate range, but may cause up to a single base video frame period jitter impact. 1 = Flicker performance is met in any duration decrease within Byte 5. ***Note:*** *Flicker performance is met in any duration decrease within the refresh rate range without jitter impact when either of the following conditions are met:* • *Byte 5 = 00h, –or–* • *Byte 5 ≥ the delta between the maximum frame duration (= minimum refresh rate) and the minimum frame duration (= maximum refresh rate)*
	7:6	**RESERVED** Cleared to all 0s.
1	7:0	**Maximum Single Frame Duration Increase Allowed for Meeting VESA Adaptive Sync Flicker Performance** 6.2 format (six integer bits and two fractional bits) that results in a value range of 0.00 to 63.75 ms, inclusive. 00h = Flicker performance is met in any duration increase within the refresh rate range without jitter impact.
2	7:0	**Minimum Refresh Rate** Minimum refresh rate ranges from 0 through 255 Hz, divided by 1.001. 00h = 0 Hz. 01h = (1 / 1.001) Hz. … FFh = (255 / 1.001) Hz.

Table 4-52: Adaptive-Sync Operation Mode and Range Descriptor (Continued)

Byte #	Bit #	Definition
4:3	**Maximum Refresh Rate** Maximum refresh rate ranges from 1 through 1,024 Hz, plus 350 ppm. Note that the value stored in this field shall match that of at least one *CVT v2.0* RB Timing v3 timing supported by the Sink device. 000h = (1 × 1.00035) Hz. … 3FFh = (1,024 × 1.00035) Hz.	
3	7:0	**Maximum Refresh Rate7:0**
4	1:0	**Maximum Refresh Rate9:8**
	7:2	**RESERVED** Cleared to all 0s.
5	7:0	**Maximum Single Frame Duration Decrease Allowed for Meeting VESA Adaptive Sync Flicker Performance** 6.2 format (six integer bits and two fractional bits) that results in a value range of 0.00 to 63.75 ms, inclusive. 00h = Flicker performance is met in any duration decrease within the refresh rate range without jitter impact.

4.10 AR/VR-related Data Blocks

Two data blocks define the capabilities of Augmented Reality and Virtual Reality (ARVR) Head Mounted Displays (HMDs). Two data blocks are defined – ARVR_HMD, in Section 4.10.1 and ARVR_Layer in Section 4.10.2. One ARVR_HMD data block is presented by the HMD, and one ARVR_Layer data block is presented by each stream endpoint contained within the HMD.

HMDs shall use native DisplayID Structure v2.0 and shall **not** use DisplayID Extension Sections to *EDID*

For Native structures, ARVR_HMD and ARVR_Layer are **mandatory** data blocks for ARVR HMDs; otherwise, the data blocks are **optional**.

For EDID Extension Sections, ARVR_HMD and ARVR_Layer data blocks are **not** allowed for ARVR HMDs; otherwise, the data blocks are **optional**.

4.10.1 ARVR_HMD Data Block

All ARVR_HMD data block fields are considered to be **mandatory** elements of the data block; however, not all features must be supported. If a feature is **not** supported there is a way to state this, typically by clearing the bit(s) to 0.

Table 4-53 defines the size and order of each ARVR_HMD data block field, with all addresses relative to the beginning of the data block.

Table 4-53: ARVR_HMD Data Block

Offset	Bit #	Definition	Defined in
00h	7:0	**ARVR_HMD Data Block** 2Ch.	
01h	**Block Revision and Other Data**		
	2:0	**Block Revision** Revision ranges from 0 through 7. 000b = Revision 0 (default).	
	7:3	**RESERVED** Block-specific. Cleared to all 0s.	
02h	7:0	**Number of Payload Bytes** 4Dh (77) for Block Revision 000b.	

Table 4-53: ARVR_HMD Data Block (Continued)

Offset	Bit #	Definition	Defined in
03h	**Dual Layer Single Stream Transport Support**		
	1:0	**Single_Stream_Transport_Dual_Layer** 00b = Dual Layer with single-stream transport is not supported by the HMD. 01b = Dual Layer Interleaving with single-stream transport is supported by the HMD. 10b = Dual Layer Extended Frame with single-stream transport is supported by the HMD. 11b = Both Extended Frame and Interleaving methods of Dual Layer with single-stream transport are supported by the HMD.	
	7:2	**RESERVED** Cleared to all 0s.	
04h	**Number of Displays and Streams**		
	3:0	**Num_Displays** Defines how many physical displays exist in the HMD. This includes displays that can accept an individual stream or shall always be combined in a 3D stereo stream. 0h = One display. 1h = Two displays. 2h = Three displays. 3h = Four displays. All other values are RESERVED.	
	7:4	**Num_Streams** Designates how many *DP* streams are supported by the HMD using one or two single-stream links –or– a single link with multi-stream transport. HMDs that support multiple streams shall also support a single stream. 0h = One stream. 1h = Two streams. All other values are RESERVED.	

Table 4-53: ARVR_HMD Data Block (Continued)

Offset	Bit #	Definition	Defined in
05h	**Layers**		
	3:0	**Num_Layers** Designates the maximum number of layers that are supported by the HMD. Reduced number of layers shall be supported in accordance with *ARVR Standard*. 0h = Single layer. 1h = Support for two layers. All other values are RESERVED.	4.10.1.1.1
	5:4	**Layer_Metadata** Designates what types of layer metadata are supported by the HMD. Two possibilities exist, using the 1st line of Layer 0, and the AR/VR SDP. 00b = No support for layer metadata. 01b = Support for 1st line of Layer 0 to be used for metadata. 10b = Support for the AR/VR SDP. 11b = Support for both the 1st line and the AR/VR SDP.	4.10.1.1.2
	7:6	**Replication_Factor** If the HMD supports Layer Metadata, denoted by bit 0 of the Layer_Metadata field is set to 1 (Offset 05h[4] = 1) and the Source device has programmed the Metadata_Config DPCD register to 01b, denoting that the 1st line of Layer 0 is used, the HMD uses this field to declare the replication factor used to decode the metadata. This field shall be cleared to 00b when the Layer_Metadata value is 00b –or– 10b (Offset 05h[6:4] = 00b –or– 10b). Used by the HMD to define the replication factor that the HMD expects. The Source device cannot change this, and shall replicate the number of times set by this field. 00b = Replication factor of 1 (i.e., no replication). 01b = Replication factor of 4. 10b = Replication factor of 8. 11b = Replication factor of 16.	
Area of Low Distortion Field Set			4.10.1.2
07h:06h	15:0	**Right_Low_Distortion_Area_X_Coordinate** The number of pixels from the upperleft corner of the Right (or only) display to the upperleft corner of an area that is usable when there is no lens distortion correction applied to the image. Only this area should be used before there is lens correction applied to the image being displayed in the HMD. ***Note:*** *If the HMD can support lens correction and it is enabled, this field can be ignored by setting it to 0h.* A 16-bit unsigned integer number of pixels from the upperleft corner of the Right (or only) display to the upperleft corner of the low distortion area that is usable without lens correction being applied to the Image Stream.	

Table 4-53: ARVR_HMD Data Block (Continued)

Offset	Bit #	Definition	Defined in
09h:08h	15:0	**Right_Low_Distortion_Area_Y_Coordinate** The number of lines from the upperleft corner of the Right (or only) display to the upperleft corner of an area that is usable when there is no lens distortion correction applied to the image. A 16-bit unsigned integer number of lines from the upperleft corner of the Right (or only) display to the upperleft corner of the low distortion area that is usable without lens correction being applied to the Image Stream.	
0Bh:0Ah	15:0	**Left_Low_Distortion_Area_X_Coordinate** The number of pixels from the upperleft corner of the Left (or only) display to the upperleft corner of an area that is usable when there is no lens distortion correction applied to the image. Only this area should be used before there is lens correction applied to the image being displayed in the HMD. ***Note:*** *If the HMD can support lens correction and it is enabled, this field can be ignored by setting it to 0h.* A 16-bit unsigned integer number of pixels from the upperleft corner of the Left (or only) display to the upperleft corner of the low distortion area that is usable without lens correction being applied to the Image Stream.	
0Dh:0Ch	15:0	**Left_Low_Distortion_Area_Y_Coordinate** The number of lines from the upperleft corner of the Left (or only) display to the upperleft corner of an area that is usable when there is no lens distortion correction applied to the image. A 16-bit unsigned integer number of lines from the upperleft corner of the Left (or only) display to the upperleft corner of the low distortion area that is usable without lens correction being applied to the Image Stream.	
0Fh:0Eh	15:0	**Low_Distortion_Area_Width** The number of pixels wide that the usable image area is. The same width is used for both Right and Left image streams. A 16-bit unsigned integer number of pixels that defines the width of the usable area.	
11h:10h	15:0	**Low_Distortion_Area_Height** The number of lines high of the usable image area. The same height is used for both the Right and Left image streams. A 16-bit unsigned integer number of lines that defines the height of the usable area.	
End of Area of Low Distortion Field Set			

Table 4-53: ARVR_HMD Data Block (Continued)

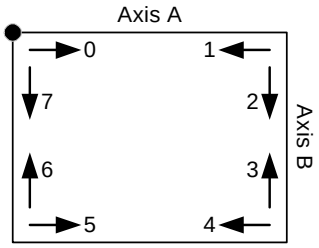
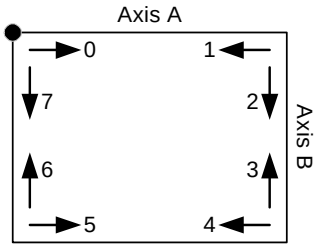
Offset	Bit #	Definition	Defined in
12h	**Eye Rotation Orientation**		
	2:0	**Right_Eye_Rotation_Orientation** 000b = HMD does not require rotation and the standard orientation #0 is used. 001b = HMD requires rotation in the #1 orientation. 010b = HMD requires rotation in the #2 orientation. 011b = HMD requires rotation in the #3 orientation. 100b = HMD requires rotation in the #4 orientation. 101b = HMD requires rotation in the #5 orientation. 110b = HMD requires rotation in the #6 orientation. 111b = HMD requires rotation in the #7 orientation. 	
	3	**RESERVED** Cleared to 0.	
	6:4	**Left_Eye_Rotation_Orientation** 000b = HMD does not require rotation and the standard orientation #0 is used. 001b = HMD requires rotation in the #1 orientation. 010b = HMD requires rotation in the #2 orientation. 011b = HMD requires rotation in the #3 orientation. 100b = HMD requires rotation in the #4 orientation. 101b = HMD requires rotation in the #5 orientation. 110b = HMD requires rotation in the #6 orientation. 111b = HMD requires rotation in the #7 orientation. 	
	7	**RESERVED** Cleared to 0.	

Table 4-53: ARVR_HMD Data Block (Continued)

Offset	Bit #	Definition	Defined in
Optics Field Set			
14h:13h	15:0	**Right_Lens_Diameter** The distance in cm across the right lens from edge to edge, assuming a circular lens, or a partially circular lens with portions of the lens being this diameter. This is intended to be used in HMDs that do not mechanically adjust the distance between the lenses for varying interpupillary distance (IPD). This along with the distortion and focal length fields should allow the rendering to compensate for the center of the eyes not aligning with the center of the lenses. Expressed in cm with a fixed point with three bits of integer, and 13 bits of fractional values, allowing for lenses up to 9 cm, or 3.5 inches in diameter. +3.13 Fixed Point	
16h:15h	15:0	**Left_Lens_Diameter** The distance in cm across the left lens from edge to edge, assuming a circular lens, or a partially circular lens with portions of the lens being this diameter. This is intended to be used in HMDs that do not mechanically adjust the distance between the lenses for varying IPDs. This along with the distortion and focal length fields should allow the rendering to compensate for the center of the eyes not aligning with the center of the lenses. Expressed in cm with a fixed point with three bits of integer, and 13 bits of fractional values, allowing for lenses up to 9 cm, or 3.5 inches in diameter. +3.13 Fixed Point	
18h:17h	15:0	**Interocular Angle** Angle between the center of projection lines of standard orientation #0 for the left and right eye. The angle between the screens measured between the lines from the center of each lens to the center of projection. Expressed in radians with three bits for the integer portion, and 13 bits for the fractional portion of a radian, or +3.13 Fixed Point. A number around 1 radian starts to become unusable.	4.10.1.3
End of Optics Field Set			

Table 4-53: ARVR_HMD Data Block (Continued)

Offset	Bit #	Definition	Defined in
		Lens Adjustment Field Set	4.10.1.4
19h	**Lenses**		
	0	**Lens_Adjust** Designates whether the distance between the lenses physically adjust for varying IPD. 0 = Distance between the lenses cannot be adjusted for varying IPD. 1 = Distance between the lenses can be adjusted for varying IPD.	
	2:1	**Lens_Adjust_Motion** Designates whether the lens and the display move together along a straight line and by the same amount and the interocular angle does not change with IPD change. 00b = Lenses do not move in a straight line. 01b = Lenses and displays move together and the movement is in a straight line. 10b = Lenses move in a straight line, parallel to the displays which remain stationary. 11b = RESERVED.	
	3	**Lens_Distance_Available** Designates whether the HMD provides a way to read the inter-lens distance that the lenses are adjusted to and present that to the Source device. This field is not applicable if Lens_Adjust is cleared to 0. 0 = Inter-lens Distance value that the HMD is adjusted to is not reported back to the Host. In this case the Lens_Distance DPCD is a static register set to the fixed Inter-lens Distance of the HMD. 1 = Inter-lens Distance value that the HMD is adjusted to is reported back to the Host. The HMD will write this value in the Lens_Distance DPCD register and is will be updated if the user adjusts the lens.	
	4	**IPD_Useful_To_HMD** Designates that the HMD cannot measure IPD; however, if the Source device can supply the IPD to the HMD, the HMD can use it. If this bit is set and the Source device can provide the IPD, the Source device sets DPCD register “Interpupillary Distance (IPD) Measurement made by the Source” to a nonzero value.	
	7:5	**RESERVED** Cleared to all 0s.	
1Bh:1Ah	15:0	**Lens_Adjust_Minimum** Minimum Lens separation in the adjustment range (in cm). Distance in centimeters that the lens centers are apart when adjusted to the minimum distance, measured in centimeters (cm). Expressed in a positive fixed-point representation with three integer bits and 13 fractional bits. If the lenses do not move, this field should be set to the fixed distance between the lens centers. +3.13 Fixed Point	

Table 4-53: ARVR_HMD Data Block (Continued)

Offset	Bit #	Definition	Defined in
1Dh:1Ch	15:0	**Lens_Adjustment_Range** Distance in centimeters (cm) that the lenses can move in or out, measured from the closest together point to the farthest apart point. If this value is zero, the lenses cannot move. Expressed in a positive fixed-point representation with three integer bits and 13 fractional bits. If the lenses do not move, this field should be cleared to all 0s. +3.13 Fixed Point	
1Fh:1Eh	15:0	**IPD_Center_Offset** Offset from center that allows the Left and Right IPD measurements to be at different locations from the center. If the HMD cannot support a center offset, it implies that the Left and Right IPD are always symmetric and this field shall be cleared to all 0s. Expressed as a 2's complement fixed point representation with two integer bits and 13 fractional bits and a sign bit (MSB). ±2.13 Fixed Point	
21h:20h	15:0	**IPD_Meas_Min** The minimum IPD distance that the HMD can measure. Note that this can be smaller than the Lens_Adjust_Minimum, which defines how far the lens can move. Measured in centimeters (cm). Expressed in a positive fixed-point representation with three integer bits and 13 fractional bits. If the lenses do not move, or the measurement is not made by the HMD, this field should be cleared to all 0s. +3.13 Fixed Point	
23h:22h	15:0	**IPD_Meas_Range** The range of IPD distance in cm that the HMD can measure. Note that this can be larger than the Lens_Adjustment_Range. Expressed in a positive fixed-point representation with three integer bits and 13 fractional bits. If the lenses do not move, or the measurement is not made by the HMD, this field should be cleared to all 0s. +3.13 Fixed Point	

Table 4-53: ARVR_HMD Data Block (Continued)

Offset	Bit #	Definition	Defined in
24h	**Lens Adjustments Available in HMD**		
	0	**Distance to Right Display Available** HMD has the ability to report the distance from the nodal point of the lens to the surface of the Right Display in the Distance to Right Display DPCD register. 0 = HMD cannot report this value. 1 = HMD can report this information in the Distance to Right Display DPCD register.	
	1	**Distance to Left Display Available** HMD has the ability to report the distance from the nodal point of the lens to the surface of the Left Display in the Distance to Left Display DPCD register. 0 = HMD cannot report this value. 1 = HMD can report this information in the Distance to Left Display DPCD register.	
	2	**Distance to Right Eye Available** HMD has the ability to report the distance from the nodal point of the lens to the surface of the Right Eye in the Distance to Right Eye DPCD register. 0 = HMD cannot report this value. 1 = HMD can report this information in the Distance to Right Eye DPCD register.	
	3	**Distance to Left Eye Available** HMD has the ability to report the distance from the nodal point of the lens to the surface of the Left Eye in the Distance to Left Eye DPCD register. 0 = HMD cannot report this value. 1 = HMD can report this information in the Distance to Left Eye DPCD register.	
	7:4	**RESERVED** Cleared to all 0s.	
25h	**Foveated Rendering Support**		
	1:0	**Foveated Rendering Support** Defines whether the HMD supports particular foveated rendering methods. 00b = HMD cannot support foveated rendering. 01b = HMD can support foveated rendering with a single stream. 10b = HMD can support foveated rendering with dual streams. 11b = HMD can support foveated rendering with either a single stream or dual streams.	
	7:2	**RESERVED** Cleared to all 0s.	
End of Lens Adjustment Field Set			

Table 4-53: ARVR_HMD Data Block (Continued)

Offset	Bit #	Definition	Defined in
Field of View (FoV) for Layer 0 Field Set ***Note:*** *All references to displays refer to Display 0 for the left or right eye, –or– to the single display in single-display HMDs.*			4.10.1.5
27h:26h	15:0	**Horizontal_FoV** The total horizontal Field of View measured in Radians from the leftmost visible edge Horizontally to the rightmost visible edge. Measurement in Radians from the leftmost visible edge to the rightmost edge of the visible area. Expressed in radians with three bits for the integer portion, and 13 bits for the fractional portion of a radian. +3.13 Fixed Point	
29h:28h	15:0	**R_FoV_Right** Field of View measured in Radians from the Center of Projection Horizontally to the rightmost visible edge. Measurement in Radians from the Center of Projection to the rightmost edge of the visible area. Expressed in radians with three bits for the integer portion, and 13 bits for the fractional portion of a radian. +3.13 Fixed Point	
2Bh:2Ah	15:0	**R_FoV_Left** Field of View measured in Radians from the Center of Projection Horizontally to the leftmost visible edge. Measurement in Radians from the Center of Projection to the leftmost edge of the visible area. Expressed in radians with three bits for the integer portion, and 13 bits for the fractional portion of a radian. +3.13 Fixed Point	
2Dh:2Ch	15:0	**R_FoV_Up** Field of View measured in Radians from the Center of Projection Vertically to the uppermost visible edge. Measurement in Radians from the Center of Projection to the uppermost edge of the visible area. Expressed in radians with three bits for the integer portion, and 13 bits for the fractional portion of a radian. +3.13 Fixed Point	
2Fh:2Eh	15:0	**R_FoV_Down** Field of View measured in Radians from the Center of Projection Vertically to the lowermost visible edge. Measurement in Radians from the Center of Projection to the lowermost edge of the visible area. Expressed in radians with three bits for the integer portion, and 13 bits for the fractional portion of a radian. +3.13 Fixed Point	

Table 4-53: ARVR_HMD Data Block (Continued)

Offset	Bit #	Definition	Defined in
31h:30h	15:0	**L_FoV_Right** Field of View measured in Radians from the Center of Projection Horizontally to the rightmost visible edge. Measurement in Radians from the Center of Projection to the rightmost edge of the visible area. Expressed in radians with three bits for the integer portion, and 13 bits for the fractional portion of a radian. +3.13 Fixed Point	
33h:32h	15:0	**L_FoV_Left** Field of View measured in Radians from the Center of Projection Horizontally to the leftmost visible edge. Measurement in Radians from the Center of Projection to the leftmost edge of the visible area. Expressed in radians with three bits for the integer portion, and 13 bits for the fractional portion of a radian. +3.13 Fixed Point	
35h:34h	15:0	**L_FoV_Up** Field of View measured in Radians from the Center of Projection Vertically to the uppermost visible edge. Measurement in Radians from the Center of Projection to the uppermost edge of the visible area. Expressed in radians with three bits for the integer portion, and 13 bits for the fractional portion of a radian. +3.13 Fixed Point	
37h:36h	15:0	**L_FoV_Down** Field of View measured in Radians from the Center of Projection Vertically to the lowermost visible edge. Measurement in Radians from the Center of Projection to the lowermost edge of the visible area. Expressed in radians with three bits for the integer portion, and 13 bits for the fractional portion of a radian. +3.13 Fixed Point	
3Bh:38h	31:0	**R_Focal Length** The angular pixel density in pixels/radian at the center of projection, for each display if there are more than one. Pixels per radian at the center of projection for Right Display 0. The angle used to determine the radians is from the center of the lens to the display. Encoded as a 16-bit digit for the integer portion, and 16 bits for the fractional value. +16.16 Fixed Point	

Table 4-53: ARVR_HMD Data Block (Continued)

Offset	Bit #	Definition	Defined in
3Fh:3Ch	31:0	**L_Focal Length** The angular pixel density in pixels/radian at the center of projection, for each display if there are more than one. Pixels per radian at the center of projection for Left Display 0. The angle used to determine the radians is from the center of the lens to the display. Encoded as a 16-bit digit for the integer portion, and 16 bits for the fractional value. +16.16 Fixed Point	
End of Field of View (FoV) for Layer 0 Field Set			
Center of Projection Field Set			4.10.1.6
RightCenterOfProjection The distance in pixels from the upperleft corner of Display 0 to the center of projection. If Num_Displays = 0h (meaning that there is only one display in the HMD), Figure 4-8 is used to reference the measurements to the upperleft corner of the single display. If Num_Displays ≥ 1h (meaning that there are two or more displays in the HMD), Figure 4-9 is used to reference the measurements to the upperleft corner of Right Display 0.			
43h:40h	31:0	**R_CoPY** The Center of Projection is the distance measured in pixels and lines from the upperleft pixel of Right Display 0 to the Center of Projection through the lens, which is where the light goes through the center of the lens, virtually undistorted. Each component, X and Y, is expressed in a 32-bit IEEE single precision floating point.	
47h:44h	31:0	**R_CoPX** The Center of Projection is the distance measured in pixels and lines from the upperleft pixel of Right Display 0 to the Center of Projection through the lens, which is where the light goes through the center of the lens, virtually undistorted. Each component, X and Y, is expressed in a 32-bit IEEE single precision floating point.	

Table 4-53: ARVR_HMD Data Block (Continued)

Offset	Bit #	Definition	Defined in
LeftCenterOfProjection The distance in pixels from the upperleft corner of Left Display 0 to the center of projection. If Num_Displays ≥ 1h (meaning that there are two or more displays in the HMD), Figure 4-8 is used to reference the measurements to the upperleft corner of Left Display 0. If Num_Displays = 0h (meaning that there is only one display in the HMD), Figure 4-9 is used to reference the measurements to the upperleft corner of the single display.			
4Bh:48h	31:0	**L_CoPY** The Center of Projection is the distance measured in pixels and lines from the upperleft pixel of Left Display 0 to the Center of Projection through the lens, which is where the light goes through the center of the lens, virtually undistorted. Each component, X and Y, is expressed in a 32-bit IEEE single precision floating point.	
4Fh:4Ch	31:0	**L_CoPX** The Center of Projection is the distance measured in pixels and lines from the upperleft pixel of Left Display 0 to the Center of Projection through the lens, which is where the light goes through the center of the lens, virtually undistorted. Each component, X and Y, is expressed in a 32-bit IEEE single precision floating point.	
End of Center of Projection Field Set			

4.10.1.1 Layers

4.10.1.1.1 Num_Layers Field

ARVR HMDs use the Num_Layers field to declare the number of layers that the ARVR HMD can support. Lower number of layers shall also be supported as defined by *ARVR Standard*. Layers are used to support foveated rendering and other applications, and are defined in *ARVR Standard*. All HMDs shall support a minimum of one layer; thus, the default value of this field is 0h.

4.10.1.1.2 Layer Metadata Field

The Layer metadata is used for the Source device to communicate to the Sink device the location and size of Layers > 0. This can change with each frame. Two methods to transfer the metadata are possible with HMDs with only two layers. The preferred method is to use the ARVR SDP.

An alternate is available that is more restrictive which uses the 1st line of video data in Layer 1.

4.10.1.2 Area of Low Distortion Fields

Before the operating system can install a driver for the Graphics Processing Unit (GPU). it is difficult to pre-distort the image being transferred to the HMD to account for the lenses affect on the image. To have a visible image that is not distorted beyond a comfortable viewing experience, the HMD reports an area to the Source device in which the image is relatively low in distortion and can be comfortably viewed by the user. It is the Source device's responsibility to offset any image to fit within the prescribed area. This area is defined by a rectangle where the upperleft corner is defined by Right_Low_Distortion_Area_X_Coordinate and Right_Low_Distortion_Area_Y_Coordinate for the Right eye, and similarly for the Left eye. The low distortion area's size is defined by Low_Distortion_Area_Width and Low_Distortion_Area_Height. The size is the same for both the right and left eyes; however, the upperleft corner can be different. All horizontal measurements are in pixels. All vertical measurements are in lines.

If the HMD has the lens distortion correction capabilities, as defined by 10b in the Lens_Distortion_Support field, the low distortion area fields will be cleared to all 0s. In this case, the lens distortion capability must be enabled, and the GPU can transmit image data to anywhere on the display.

4.10.1.3 Interocular Angle Field

Noncanted displays in a single plane result in an interocular angle of 0°, as illustrated in Figure 4-2.

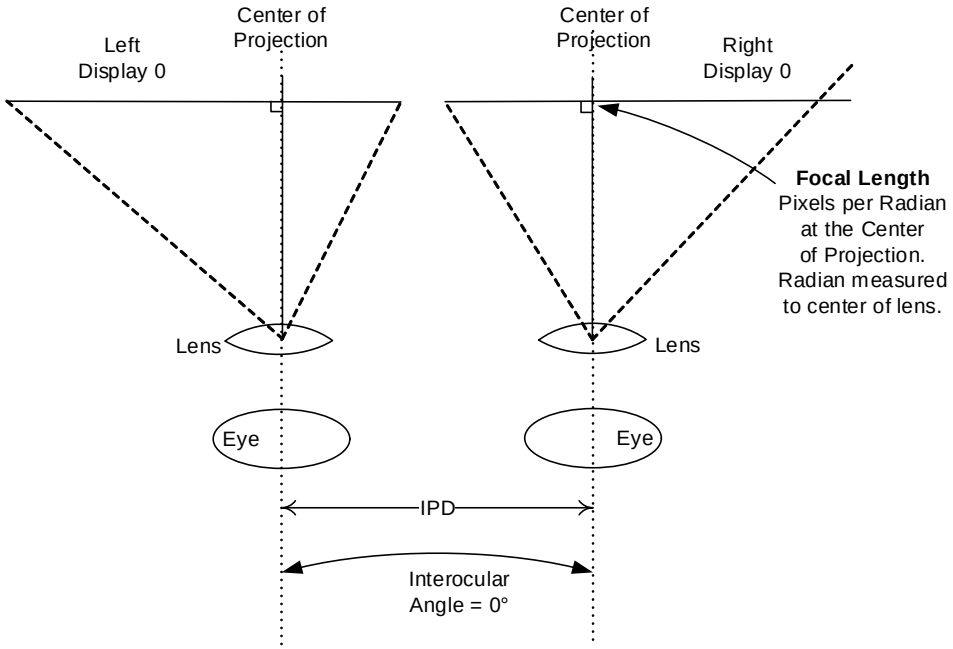


Figure 4-2: Interocular Angle Equals Zero

Canted displays create an interocular angle greater than 0°, as illustrated in Figure 4-3.

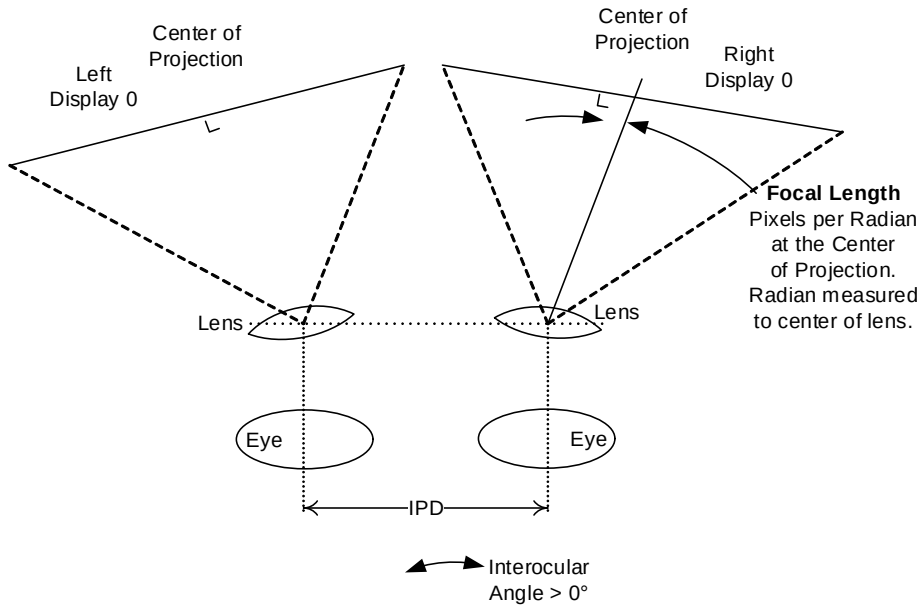


Figure 4-3: Interocular Angle is Greater than Zero

4.10.1.4 Lens Adjustment Fields

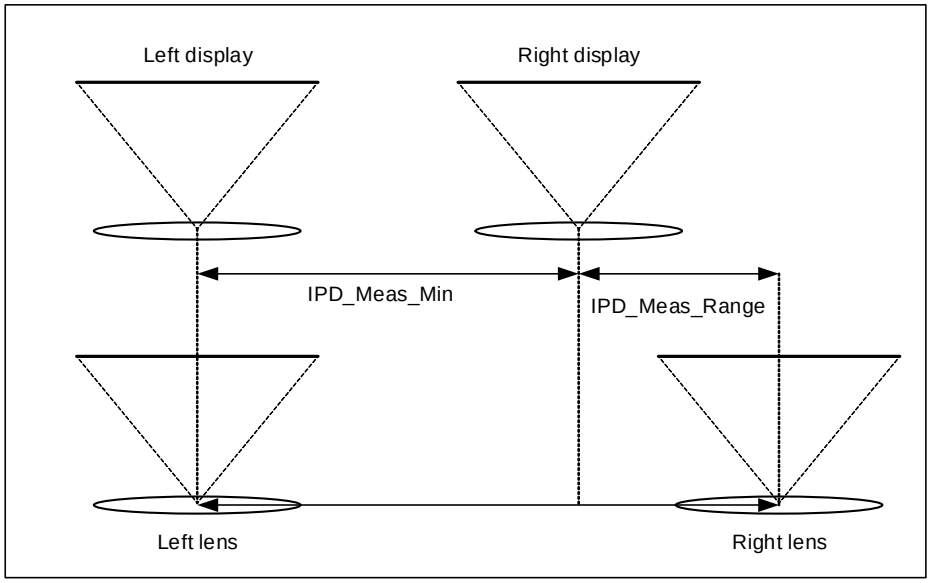


Figure 4-4: IPD Measurement Minimum and Range

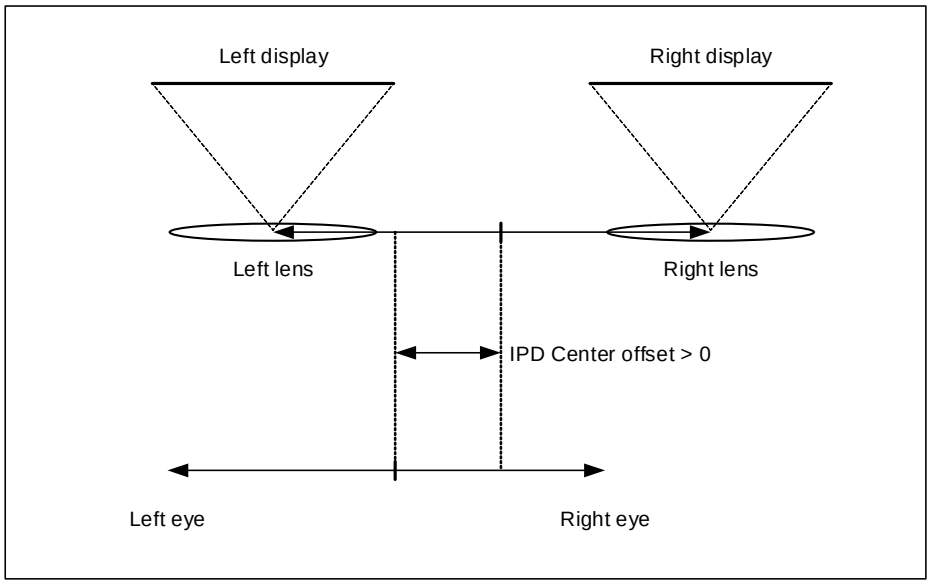


Figure 4-5: IPD Center Offset

4.10.1.5 Field of View Fields

Figures 4-6 and 4-7 show the measurement of various terms related to Field of View (FoV). Note that these measurements always refer to Display 0 for the right and left eye. These measurements are to be made with any adjustments in lens position that effect FoV set to maximize the FoV. For example, if the lens can be moved closer or further from the display, the lens should be set to the location furthest from the display to maximize the FoV before the measurements are made.

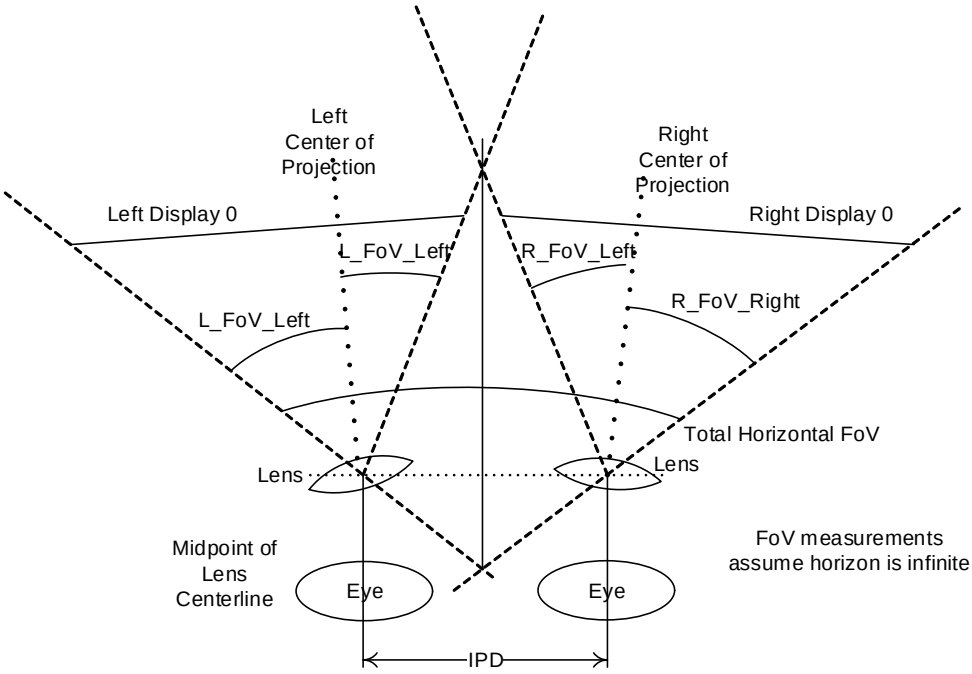


Figure 4-6: Horizontal FOV

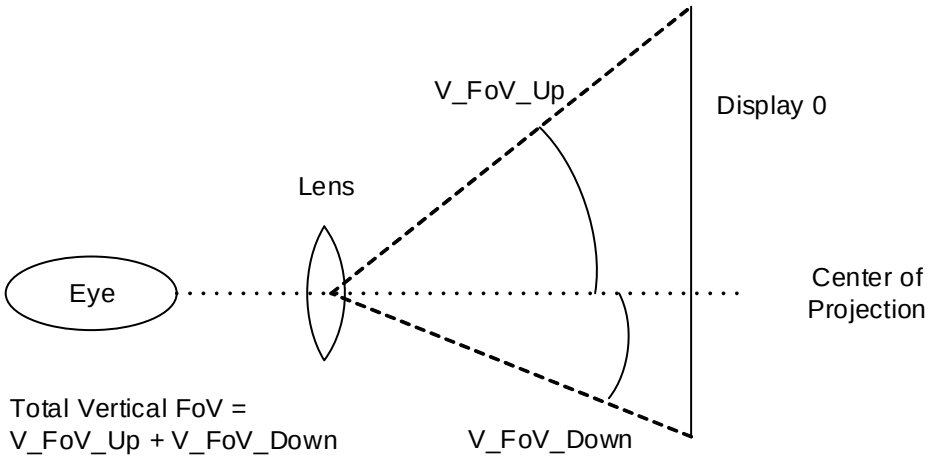


Figure 4-7: Vertical FOV

4.10.1.6 Center of Projection Fields

In an HMD with a single display, the upperleft pixel is used to reference the Center of Projection for both the left and right eyes, as illustrated in Figure 4-8.

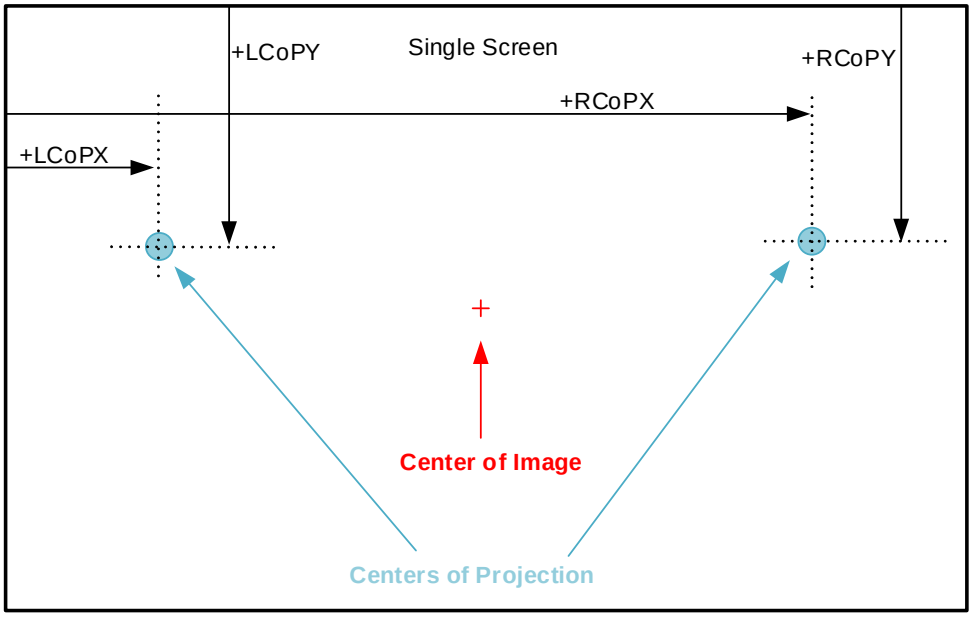


Figure 4-8: Center of Projection for Single-display HMD

In an HMD that has two or more displays, the upperleft corner of Display 0 for each eye is used, as illustrated in Figure 4-8. The left Center of Projection is referenced to the upperleft corner of Left Display 0. The right center of projection is referenced to the upperleft corner of Right Display 0.

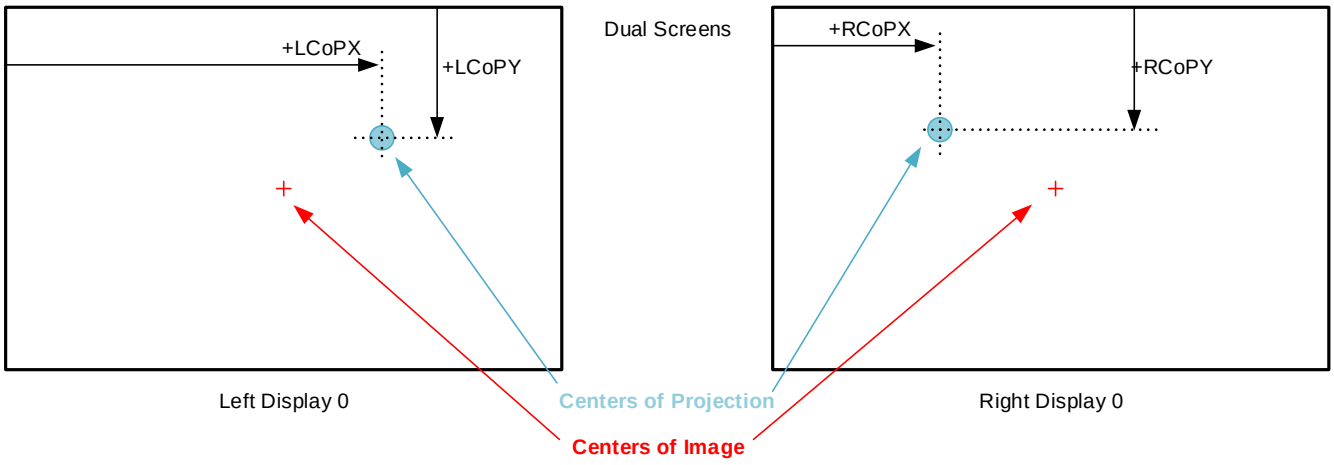


Figure 4-9: Center of Projection for a Dual-display HMD

4.10.2 ARVR_Layer Data Block

All ARVR_Layer data block fields are considered to be **mandatory** elements of the data block; however, not all features must be supported. If a feature is **not** supported there is a way to state this, typically by clearing the bit(s) to 0.

Table 4-54 defines the size and order of each ARVR_Layer data block field, with all addresses relative to the beginning of the data block.

Table 4-54: ARVR_Layer Data Block

Offset	Bit #	Definition	Defined in
00h	7:0	**ARVR_Layer Data Block** 2Dh.	
01h	**Block Revision and Other Data**		
	2:0	**Block Revision** Revision ranges from 0 through 7. 000b = Revision 0 (default).	
	7:3	**RESERVED** Block-specific. Cleared to all 0s.	
02h	7:0	**Number of Payload Bytes** 19h (25) for Block Revision 000b.	
05h:03h	23:0	**HMD_Manufacturer_Vendor ID** (Same numbers for all layers in the HMD.)	4.10.2.1
07h:06h	15:0	**HMD_Product ID Code** (Same numbers for all layers in the HMD.)	4.10.2.2
0Bh:08h	31:0	**HMD_Serial Number** (Same numbers for all layers in the HMD.)	4.10.2.3

Table 4-54: ARVR_Layer Data Block (Continued)

Offset	Bit #	Definition	Defined in
0Ch	**Layers**		
	3:0	**Layer_Number** Denotes the layer number associated with this stream. 0h = Layer 0. 1h = Layer 1. All other values are RESERVED.	4.10.2.4
	4	**Layer_Configurable** 0 = Source device cannot configure (enable or disable) the layer. 1 = Source device may configure (enable and disable) the layer image data using the Layer_Config DPCD field.	
	5	**Cropping Supported for Horizontal and Vertical** 0 = Cropping of the Layer image data is not supported. 1 = Cropping of the Horizontal and Vertical image data is supported and can be changed dynamically using the metadata type defined by the Layer_Metadata field in the ARVR_HMD data block using the ARVR Layer Rectangle Width and ARVR Layer Rectangle Height fields in the metadata.	
	7:6	**RESERVED** Cleared to all 0s.	
0Dh	7:0	**RESERVED (for tile location)**	

Table 4-54: ARVR_Layer Data Block (Continued)

Offset	Bit #	Definition	Defined in
0Eh	**Lens Distortion**		
	1:0	**Lens_Distortion_Support** Defines whether the HMD supports lens distortion and chromatic aberration correction. 00b = HMD does not support any lens distortion correction. 01b = HMD has chromatic aberration correction capabilities. 10b = HMD supports both lens distortion and chromatic aberration correction. 11b = RESERVED.	
	3:2	**Lens_Distortion_Configurable** Defines the configuration flexibility the Source device has over the Lens Distortion Correction capabilities of the HMD by way of the Lens_Distortion_Config DPCD register. 00b = Source device cannot configure the HMD's lens distortion correction capabilities. 01b = Source device can enable or disable the HMD's chromatic aberration correction, but not the HMD's lens distortion correction. 10b = Source device can enable or disable the HMD's chromatic aberration and lens distortion correction. 11b = RESERVED.Col	
	7:4	**RESERVED** Cleared to all 0s.	

Table 4-54: ARVR_Layer Data Block (Continued)

Offset	Bit #	Definition	Defined in
0Fh	**Gamma, Degamma, Mura, VBI**		
	0	**Gamma_Support** Defines support in the HMD for gamma. 0 = HMD does not support gamma. 1 = HMD supports gamma.	
	1	**Gamma_Configurable** Defines whether support in the HMD for gamma can be enabled and disabled by the Source device. 0 = Gamma correction in the HMD cannot be disabled by the Source device. 1 = Gamma correction in the HMD can be disabled by the Source device using the Gamma_Config DPCD field.	
	2	**Degamma_Support** Defines support in the HMD for degamma. 0 = HMD does not support degamma. 1 = HMD supports degamma.	
	3	**Degamma_Configurable** Defines whether support in the HMD for degamma can be enabled and disabled by the Source device. 0 = Degamma correction in the HMD cannot be disabled by the Source device. 1 = Degamma correction in the HMD can be disabled by the Source device.	
	4	**Mura_Support** Defines whether the HMD color space Mura Compensation. 0 = HMD does not support Mura Compensation. 1 = HMD has Mura Compensation capabilities.	
	5	**Mura_Configurable** Defines whether HMD support for Mura Compensation can be configured by the Source device. 0 = Mura Compensation in the HMD cannot be disabled by the Source device. 1 = Mura Compensation in the HMD can be disabled by the Source device using the Mura_Config DPCD field.	
	6	**VBI_Support** For each Image Stream. 0 = HMD does not support Vertical Blanking Interval insertion. 1 = HMD supports Vertical Blanking Interval insertion.	
	7	**VBI_Configurable** For each Image Stream: 0 = Vertical Blanking Interval support in the HMD cannot be disabled by the Source device. 1 = Vertical Blanking Interval support in the HMD can be disabled by the Source device using the VBI_Config DPCD field.	

Table 4-54: ARVR_Layer Data Block (Continued)

Offset	Bit #	Definition	Defined in
10h	**Asynchronous Reprojection**		
	1:0	**Async_Reprojection_Support** Defines whether the HMD supports different types of Asynchronous Reprojection. 00b = HMD does not have 3DoF or 6DoF Asynchronous Reprojection capabilities. 01b = HMD has only 3DoF Asynchronous Reprojection capabilities. 10b = HMD has 3DoF and 6DoF Asynchronous Reprojection capabilities. 11b = RESERVED.	
	3:2	**Async_Reproject_Configurable** Defines whether Asynchronous Reprojection support in the HMD can be configured by the Source device. Leave 0 for fields that do not have capabilities presented in Async_Reprojection_Support. 00b = 3DoF and 6DoF Asynchronous Reprojection capabilities in the HMD cannot be configured by the Source device. 01b = 3DoF Asynchronous Reprojection capabilities in the HMD can be disabled by the Source device using the Async_Reproject_Config DPCD field, but the 6DoF capabilities cannot. 10b = 3DoF Asynchronous Reprojection capabilities in the HMD cannot be disabled by the Source device using the Async_Reproject_Config DPCD field, but the 6DoF capabilities can be disabled by the Source device using the Async_Reproject_Config DPCD field. 11b = Both the 3DoF and 6DoF Asynchronous Reprojection capabilities in the HMD can be disabled by the Source device using the Async_Reproject_Config DPCD field.	
	7:4	**RESERVED** Cleared to all 0s.	

Table 4-54: ARVR_Layer Data Block (Continued)

Offset	Bit #	Definition	Defined in
11h	**Scaling_Support** Defines whether the HMD can upscale the image for this Stream endpoint and if so, what scaling factors are supported. A bit for each scaling factor. Set the appropriate bit to 1 for each scaling factor that the HMD can support, and leave it cleared to 0 if it is not supported.		4.10.2.5
	0	1 = HMD supports a scaling factor of 2.	
	1	1 = HMD supports a scaling factor of 3.	
	2	1 = HMD supports a scaling factor of 4.	
	3	1 = HMD supports a scaling factor of 5.	
	4	1 = HMD supports a scaling factor of 6.	
	5	1 = HMD supports a scaling factor of 8.	
	6	1 = HMD supports a scaling factor of other specific scaling factor listed in Scaling_Nonlisted fields.	
	7	**Scaling_Configurable** Defines whether the Source device can configure the scaling in the HMD. 0 = Source device cannot configure the scaling. 1 = Source device can configure the HMD scaling of the image.	
12h	7:0	**Scaling_NonListed_0** Allows for scaling factors that are not listed in the Scaling_Support field. 8-bit field to define a scaling factor that is supported by the HMD but is not specifically listed in the Scaling_Support field. Expressed in a 3.5 Fixed Point format.	
13h	7:0	**Scaling_NonListed_1** Allows for scaling factors that are not listed in the Scaling_Support field. 8-bit field to define a scaling factor that is supported by the HMD but is not specifically listed in the Scaling_Support field. Expressed in a 3.5 Fixed Point format.	
14h	7:0	**Scaling_NonListed_2** Allows for scaling factors that are not listed in the Scaling_Support field. 8-bit field to define a scaling factor that is supported by the HMD but is not specifically listed in the Scaling_Support field. Expressed in a 3.5 Fixed Point format.	
15h	7:0	**Scaling_NonListed_3** Allows for scaling factors that are not listed in the Scaling_Support field. 8-bit field to define a scaling factor that is supported by the HMD but is not specifically listed in the Scaling_Support field. Expressed in a 3.5 Fixed Point format.	

Table 4-54: ARVR_Layer Data Block (Continued)

Offset	Bit #	Definition	Defined in
16h	**Frame_Rate_Support** Defines the ratio of the Layer's supported refresh rate to Layer 0's refresh rate that the HMD can support. Seven bits are used for fixed ratios, and the 8th bit allows for additional ratios to be expressed in the next field using 3.5 Fixed Point Format. If the bit is set, the HMD can support the refresh rate ratio.		4.10.2.6
	0	2:1 (e.g., 50:25, 59.94:29.97, 60:30, 90:45, 120:60).	
	1	3:2 (e.g., 60:40, 75:50, 90:60, 120:80).	
	2	2.5:1 (e.g., 60:24).	
	3	3:1 (e.g., 75:25, 90:30, 120:40, 150:50, 180:60). 1/3 (e.g., 25/75, 30/90, 40/120, 50/150, 60/180…).	
	4	4:1 (e.g., 100:25, 120:30, 160:40).	
	5	5:1 (e.g., 120:24).	
	6	8:1 (e.g., 240:30).	
	7	Other rate(s) supported as defined in Frame_Rate_NonListed fields.	
17h	7:0	**Frame_Rate_NonListed_0** Defines a Frame Rate ratio that is supported by the HMD that are not specifically listed in the Frame_Rate_Support field. Each field is expressed in a 3.5 Fixed Point format. Cleared to 00h if not supported.	
18h	7:0	**Frame_Rate_NonListed_1** Defines a Frame Rate ratio that is supported by the HMD that are not specifically listed in the Frame_Rate_Support field. Each field is expressed in a 3.5 Fixed Point format. Cleared to 00h if not supported.	
19h	7:0	**Frame_Rate_NonListed_2** Defines a Frame Rate ratio that is supported by the HMD that are not specifically listed in the Frame_Rate_Support field. Each field is expressed in a 3.5 Fixed Point format. Cleared to 00h if not supported.	
1Ah	7:0	**Frame_Rate_NonListed_3** Defines a Frame Rate ratio that is supported by the HMD that are not specifically listed in the Frame_Rate_Support field. Each field is expressed in a 3.5 Fixed Point format. Cleared to 00h if not supported.	

Table 4-54: ARVR_Layer Data Block (Continued)

Offset	Bit #	Definition	Defined in
1Bh	**Frame Rate Multiplication**		
	0	**Frame_Rate_Mult_Configurable** Defines whether the Source device can configure the Frame Rate Multiplier in the HMD. 0 = Source device cannot configure the Frame Rate Multiplier. 1 = Source device can configure the Frame Rate Multiplier in the HMD.	
	7:1	**RESERVED** Cleared to all 0s.	

4.10.2.1 HMD_Manufacturer_Vendor ID Field

The HMD_Manufacturer_Vendor ID field follows the format of the Product Identification data block's Manufacturer/Vendor ID field defined in Section 4.1.1. The difference in the HMD_Layer is that the field shall contain the HMD manufacturer's or vendor's IEEE-assigned OUI. This field shall be identical for all ARVR_Layer data blocks contained within a single HMD.

The IEEE issues IEEE OUIs. Contact the IEEE at standards.ieee.org/products-services/regauth/oui/index.html for registration and/or further details.

4.10.2.2 HMD_Product ID Code Field

The HMD_Product ID Code field follows the format of the Product Identification data block's Product ID Code field defined in Section 4.1.2. The difference in the HMD_Layer is that the field shall contain the HMD's Product ID and **cannot** be the display's Product ID or HMD component's Product ID (such as a scaler). This field shall be identical for all ARVR_Layer data blocks contained within a single HMD. The HMD_Product ID Code shall be changed if a new display or display component such as a scaler is changed.

4.10.2.3 HMD_Serial Number Field

The HMD_Serial Number field follows the format of the Product Identification data block's Serial Number field defined in Section 4.1.3. The difference in the HMD_Layer is that the field shall contain the HMD's Serial Number. This field shall be identical for all ARVR_Layer data blocks contained within a single HMD. The HMD_Serial Number shall be unique to each HMD of the same HMD_Manufacturer_Vendor ID and HMD_Product ID Code.

4.10.2.4 Layer_Number Field

The Layer_Number field denotes the layer number that is used by this Stream.

Image stream data from higher layer numbers replaces data from lower layer numbers. The HMD can do this optically or digitally. If optical, the streams go to different displays within the HMD and optical techniques are used to cause the higher layer number to superimpose over lower layer numbers. If digital, the HMD will replace lower numbered layer data with higher layer numbered data for the pixels where higher layer numbered image data is provided.

4.10.2.5 Scaling_Support Field

The Scaling_Support field defines the HMD's scaling capabilities. The scaling factor is the same in the Horizontal and Vertical directions. For example, if bit 0 is set to 1, the HMD can support a scaling factor of 2 and the incoming image data is scaled to fill an area on the display that is twice as many pixels wide and high as the incoming image data. The scaling factor is described for each Layer. The right and left eyes for each layer are scaled the same.

4.10.2.6 Frame_Rate_Support Field

The Frame_Rate_Support field defines the ratio of the layer's supported refresh rate as a ratio to the Layer 0 refresh rate. For example, if bit 0 is set to 1, denoting that the HMD can support a 2:1 frame rate ratio, that would mean that this layer's refresh rate is 2× the Layer 0 refresh rate. If the Layer 0 refresh rate is 60 Hz, this layer's refresh rate would be 120 Hz.

4.11 Vendor-specific Data Block

The Vendor-specific data block shall be used for proprietary implementations that are not supported by DisplayID Structure v2.0. The data block can also be used for a standards organization-specific data block. This block should be used only when the required data cannot be conveyed by a standard data block that has already been defined by VESA or *CTA*. The vendor is then encouraged to propose a new standard data block to VESA. Vendor-specific is an **optional** data block in the DisplayID Base Section.

The Vendor-specific data block shall conform to the Table 4-55 definition to ensure that the DisplayID structure's generic parser works correctly. The Number of Payload bytes for this block shall include the three bytes needed for a unique Manufacturer/Vendor ID in addition to the size of the vendor-specific data.

Appendix B shows the VESA Organization Vendor-specific data block, which exposes the embedded DisplayPort-specific data structure.

Note: *See Appendix A for an example of how the VESA Organization Vendor-specific data block is used in a fixed-length DisplayID Section.*

Table 4-55: Vendor-specific Data Block

Offset	Bit #	Definition	Defined in
00h	7:0	**Vendor-specific Data Block** 7Eh.	
01h	**Block Revision and Other Data**		
	2:0	**Block Revision** Revision ranges from 0 through 7. 000b = Revision 0 (default).	
	7:3	**RESERVED** Cleared to all 0s.	
02h	7:0	**Number of Payload Bytes in Block** Number of payload bytes within the data block ranges from 3 through 248. 03h = 3 bytes. … F8h = 248 bytes.	
05h – 03h	23:0	**Manufacturer/Vendor ID** Descriptor.	4.11.1
06h – …		**Vendor-specific Data**	

4.11.1 Manufacturer/Vendor ID Field

The Manufacturer/Vendor ID field defined in Table 4-55 is a **mandatory** element of the Vendor-specific data block. The Manufacturer/Vendor ID field contains the display manufacturer's 3-byte IEEE OUI code, which identifies the display's manufacturer or vendor. Table 4-56 defines the field.

The IEEE issues IEEE OUIs. Contact the IEEE at standards.ieee.org/products-services/regauth/oui/index.html for registration and/or further details.

Table 4-56: Manufacturer/Vendor ID Field

Offset	Byte #	Bit #	Definition
03h	0	7:0	**IEEE OUI First Byte** Byte code. Byte length ranges from 0 through 255. 00h = 0. … FFh = 255.
04h	1	7:0	**IEEE OUI Second Byte** Byte code. Byte length ranges from 0 through 255. 00h = 0. … FFh = 255.
05h	2	7:0	**IEEE OUI Third Byte** Byte code. Byte length ranges from 0 through 255. 00h = 0. … FFh = 255.

4.12 CTA-861 Data Block Encapsulation DisplayID Data Block

The CTA-861 Data Block Encapsulation DisplayID data block provides the ability to encapsulate CTA data blocks in a DisplayID data block framework. See Table 3-1 for details regarding when this data block is **mandatory** and **optional**.

Table 4-57 defines how to map CTA-861 data blocks with an appropriate block tag code, associated block length, and the block payload on the CTA Data Block Encapsulation DisplayID data block. This shall allow leverage of *CTA-861* speaker allocation, audio descriptor, colorimetry data block, and others, as-is, without redefining them in DisplayID.

Table 4-57: CTA-861 Data Block Encapsulation DisplayID Data Block

Offset	Bit #	Definition
00h	7:0	**CTA-861 Data Block Encapsulation DisplayID Data Block** 81h.
01h	**Block Revision and Other Data**	
	2:0	**Block Revision** Revision ranges from 0 through 7. 000b = Revision 0 (default).
	7:3	**RESERVED** Block-specific. Cleared to all 0s.
02h	7:0	**Number of Payload Bytes** Number of payload bytes ranges from 0 through 248. 00h = 0 bytes. … F8h = 248 bytes.
03h	**CTA Block 1 Tag Code and Block 1 Length**	
	4:0	**Block Length** Associated data block length (e.g., L1). Value ranges from 00h through 1Fh.
	7:5	**Tag Code** Based on the latest *CTA-861*. Value ranges from 000b through 111b.
04h – 3 + L1	7:0	**CTA Block 1 Payload**

Table 4-57: CTA-861 Data Block Encapsulation DisplayID Data Block (Continued)

Offset	Bit #	Definition
4 + L1	**CTA Block 2 Tag Code and Block 2 Length**	
	4:0	**Block Length** Associated data block length (e.g., L2). Value ranges from 00h through 1Fh.
	7:5	**Tag Code** Based on the latest *CTA-861*. Value ranges from 000b through 111b.
5 + L1 – 4 + L1 + L2	7:0	**CTA Block 2 Payload**

A Fixed-length DisplayID Section Example (Informative)

Table A-1: Fixed-length DisplayID Section Example (Informative)

Offset	Raw Value	Field Name	Interpretation
0h	0x20	**DisplayID Structure Version/Revision**	Version 2, Revision 0.
1h	0x86	**Bytes in Section**	DisplayID Section length = 134 bytes.
2h	0x04	**Display Product Primary Use Case**	Desktop Productivity Display.
3h	0x00	**Extension Count**	Extension Section count = 0.
4h	0x20	**Data Block Name**	Product Identification data block.
5h	0x00	**Block Revision and Other Data**	Bits 7:3 = RESERVED = 00h. Bits 2:0 = Block Revision 0 = 000b.
6h	0x18	**Number of Payload Bytes in Block**	Payload length = 24 bytes.
7h 8h 9h	0x12 0x34 0x56	**Manufacturer/Vendor ID**	IEEE OUI = 12-34-56.
Ah Bh	0x34 0x12	**Product ID Code**	Vendor-assigned Product ID code = 0x1234.
Ch Dh Eh Fh	0x78 0x56 0x34 0x12	**Serial Number**	Vendor-assigned Serial Number = 0x12345678.
10h	0x01	**Week of Manufacture/Model Tag**	Week of manufacture = 1.
11h	0x15	**Year of Manufacture/Model Year**	Year of manufacture = 2021.
12h	0x0C	**Size of Product Name String**	String length = 12 bytes.

Table A-1: Fixed-length DisplayID Section Example (Informative) (Continued)

Offset	Raw Value	Field Name	Interpretation
13h	0x44	**Product Name String**	Vendor-assigned Product Name String = “Display Name”.
14h	0x69		
15h	0x73		
16h	0x70		
17h	0x6C		
18h	0x61		
19h	0x79		
1Ah	0x20		
1Bh	0x4E		
1Ch	0x61		
1Dh	0x6D		
1Eh	0x65		
1Fh	0x21	**Data Block Name**	Display Parameters data block.
20h	0x00	**Block Revision and Other Data**	Bit 7 = Image Size Multiplier = 0 (0.1-mm precision). Bits 6:3 = RESERVED = 0h. Bits 2:0 = Block Revision 1 = 001b.
21h	0x1D	**Number of Payload Bytes in Block**	Data block is composed of 29 payload bytes.
22h	0x00	**Horizontal Image Size**	409.6 mm.
23h	0x10		
24h	0x70	**Vertical Image Size**	216.0 mm.
25h	0x08		
26h	0x00	**Horizontal Pixel Count**	4,096 horizontal pixels.
27h	0x10		
28h	0x70	**Vertical Pixel Count**	2,160 vertical pixels.
29h	0x08		
2Ah	0x00	**Feature Support Flags**	Bit 7 = Audio Speaker Information = 0 (Audio speakers shall be integrated in the display). Bit 6 = Color Information = 0 (specified in terms of *CIE 1931* *x*, *y* coordinates). Bit 5 = RESERVED = 0. Bits 4:3 = Luminance Information = 00b (minimum guaranteed value). Bits 2:0 = Scan Orientation = 000b (left to right, top to bottom).
2Bh	0xCC	**Native Color Chromaticity (Red)**	**Red** X = 0.675, Y = 0.320.
2Ch	0xEA		
2Dh	0x51		

Table A-1: Fixed-length DisplayID Section Example (Informative) (Continued)

Offset	Raw Value	Field Name	Interpretation
2Eh 2Fh 30h	0x45 0x64 0xA6	**Native Color Chromaticity (Green)**	**Green** X = 0.267, Y = 0.650.
31h 32h 33h	0x66 0x42 0x0D	**Native Color Chromaticity (Blue)**	**Blue** X = 0.150, Y = 0.052.
34h 35h 36h	0xFD 0x34 0x54	**Native Color Chromaticity (White)**	**White** X = 0.312, Y = 0.329.
37h 38h	0x40 0x5E	**Native Maximum Luminance (Full Coverage)**	400 cd/m^2.
39h 3Ah	0x40 0x5E	**Native Maximum Luminance (10% Rectangular Coverage)**	400 cd/m^2.
3Bh 3Ch	0x00 0x38	**Native Minimum Luminance**	0.5 cd/m^2.
3Dh	0x13	**Native Color Depth and Display Device Technology**	Bit 7 = Display Device Theme Preference = 0 (No preference). Bits 6:4 = Display Device Technology = 001b (Active Matrix LCD). Bit 3 = RESERVED = 0. Bits 2:0 = Native Color Depth = 011b (10 bpc).
3Eh	0x78	**Native Gamma EOTF**	Gamma = 2.2.
3Fh	0x26	**Data Block Name**	Display Interface Features data block.
40h	0x00	**Block Revision and Other Data**	Bits 7:3 = RESERVED = 00h. Bits 2:0 = Block Revision 0 = 000b.
41h	0x09	**Number of Payload Bytes in Block**	Data block is composed of nine + *N* payload bytes.
42h	0x06	**Supported Interface Color Depth for RGB Encoding**	Supports 10 and 8 bits per primary color.
43h	0x00	**Supported Interface Color Depth for YCbCr 4:4:4 Encoding**	Not supported.
44h	0x00	**Supported Interface Color Depth for YCbCr 4:2:2 Encoding**	Not supported.
45h	0x00	**Supported Interface Color Depth for YCbCr 4:2:0 Encoding**	Not supported.
46h	0x00	**Minimum Pixel Rate at Which YCbCr 4:2:0 Encoding Is Supported**	Minimum pixel rate is 0 MP/s.

Table A-1: Fixed-length DisplayID Section Example (Informative) (Continued)

Offset	Raw Value	Field Name	Interpretation
47h	0x60	**Supported Interface Audio Capability and Feature Flags**	Bit 7 = 32-kHz Sample Rate Supported = 0 (not supported). Bit 6 = 44.1-kHz Sample Rate Supported = 1 (supported). Bit 5 = 48-kHz Sample Rate Supported = 1 (supported). Bits 4:0 = RESERVED = 00h.
48h	0x01	**Supported Interface Color Space and EOTF Standard Combination 1**	Bit 7 = RESERVED = 0. Bit 6 = *ITU-R BT.2020* / *SMPTE ST 2084* = 0 (not supported). Bit 5 = *ITU-R BT.2020* = 0 (not supported). Bit 4 = DCI-P3 (*SMPTE RP 431-2*) = 0 (not supported). Bit 3 = *Adobe RGB* = 0 (not supported). Bit 2 = *ITU-R BT.709* / *ITU-R BT.1886* = 0 (not supported). Bit 1 = *ITU-R BT.601* = 0 (not supported). Bit 0 = sRGB (*IEC 61966-2-1*) = 1 (supported).
49h	0x00	**Supported Interface Color Space and EOTF Standard Combination 2**	Bits 7:0 = RESERVED = 00h.
4Ah	0x00	**Number of Additional Supported Interface Color Space and EOTF (*N*)**	Bits 7:3 = RESERVED = 00h. Bits 2:0 = 000b = Zero additional supported combinations.

Table A-1: Fixed-length DisplayID Section Example (Informative) (Continued)

Offset	Raw Value	Field Name	Interpretation
4Bh	0x22	**Data Block Name**	Type VII Detailed Timing data block.
4Ch	0x00	**Block Revision and Other Data**	Bits 7 = RESERVED = 0. Bits 6:4 = Number of Payload Bytes (M) in a Timing Descriptor = 000b. Bit 3 = Pass-through Timing Support for Target DSC Bits per Pixel = 0. Bits 2:0 = Block Revision 2 = 010b.
4Dh	0x3C	**Number of Payload Bytes in Block**	Data block is composed of 60 payload bytes.
4Eh	0xC7	**Timing 1**	Pixel clock rate = 556.744 MP/s.
4Fh	0x7E		
50h	0x08		Bit 7 = 0 (4:2:0 support is **not** indicated).
51h	0x88		Bits 6:5 = 00b (Timing always mono – no stereo).
52h	0xFF		Bit 4 = 0 (Progressive scan frame).
53h	0x0F		Bits 3:0 = 8h (Aspect Ratio calculated from pixel dimensions).
54h	0x4F		
55h	0x00		
56h	0x07		Horizontal Active Image Pixels = 4,096.
57h	0x80		Horizontal Blank Pixels = 80.
58h	0x1F		Horizontal Sync Offset (front porch) = 8.
59h	0x00		Horizontal Sync Polarity = 1 (positive).
5Ah	0x6F		Horizontal Sync Width = 32.
5Bh	0x08		
5Ch	0x3D		Vertical Active Image Lines = 2,160.
5Dh	0x00		Vertical Blank Lines = 62.
5Eh	0x2F		Vertical Sync Offset (front porch) = 48.
5Fh	0x00		Vertical Sync Polarity = 0 (negative).
60h	0x07		Vertical Sync Width = 8.
61h	0x00		

Table A-1: Fixed-length DisplayID Section Example (Informative) (Continued)

Offset	Raw Value	Field Name	Interpretation
62h	0x5D	**Timing 2**	Pixel clock rate = 234.590 MP/s.
63h	0x94		
64h	0x03		Bit 7 = 0 (4:2:0 support is **not** indicated).
65h	0x08		Bits 6:5 = 00b (Timing always mono – no stereo).
66h	0xFF		Bit 4 = 0 (Progressive scan frame).
67h	0x09		Bits 3:0 = 8h (Aspect Ratio calculated from pixel dimensions).
68h	0x4F		
69h	0x00		
6Ah	0x07		Horizontal Active Image Pixels = 2,560.
6Bh	0x80		Horizontal Blank Pixels = 80.
6Ch	0x1F		Horizontal Sync Offset (front porch) = 8.
6Dh	0x00		Horizontal Sync Polarity = 1 (positive).
6Eh	0x9F		Horizontal Sync Width = 32.
6Fh	0x05		
70h	0x28		Vertical Active Image Lines = 1,440.
71h	0x00		Vertical Blank Lines = 41.
72h	0x1A		Vertical Sync Offset (front porch) = 27.
73h	0x00		Vertical Sync Polarity = 0 (negative).
74h	0x07		Vertical Sync Width = 8.
75h	0x00		

Table A-1: Fixed-length DisplayID Section Example (Informative) (Continued)

Offset	Raw Value	Field Name	Interpretation
76h	0xC7	**Timing 3**	Pixel clock rate = 133.320 MP/s.
77h	0x08		
78h	0x02		Bit 7 = 0 (4:2:0 support is **not** indicated).
79h	0x08		Bits 6:5 = 00b (Timing always mono – no stereo).
7Ah	0x7F		Bit 4 = 0 (Progressive scan frame).
7Bh	0x07		Bits 3:0 = 8h (Aspect Ratio calculated from pixel dimensions).
7Ch	0x4F		
7Dh	0x00		
7Eh	0x07		Horizontal Active Image Pixels = 1,920.
7Fh	0x80		Horizontal Blank Pixels = 80.
80h	0x1F		Horizontal Sync Offset (front porch) = 8.
81h	0x00		Horizontal Sync Polarity = 1 (positive).
82h	0x37		Horizontal Sync Width = 32.
83h	0x04		
84h	0x1E		Vertical Active Image Lines = 1,080.
85h	0x00		Vertical Blank Lines = 31.
86h	0x10		Vertical Sync Offset (front porch) = 17.
87h	0x00		Vertical Sync Polarity = 0 (negative).
88h	0x07		Vertical Sync Width = 8.
89h	0x00		

Table A-1: Fixed-length DisplayID Section Example (Informative) (Continued)

Offset	Raw Value	Field Name	Interpretation
8Ah	0x7E	**Data Block Name**	VESA Organization Vendor-specific data block.
8Bh	0x00	**Block Revision and Other Data**	Bits 7:3 = RESERVED = 00h. Bits 2:0 = Block Revision 0 = 000b.
8Ch	0x05	**Number of Payload Bytes**	Data block is composed of five payload bytes.
8Dh 8Eh 8Fh	0x3A 0x02 0x92	**Manufacturer Vendor ID**	IEEE OUI = 3A-02-92.
90h	0x81	**VESA Data Structure Type**	Bit 7 = Default Color Space and EOTF = 1 (Sink device interprets RGB unspecified color space as native color space and EOTF as specified in the Display Parameters data block; see Table 4-7). Bits 6:3 = RESERVED = 0h. Bits 2:0 = Data Structure Type = 001b (External *DisplayPort*). ***Note:*** *The Sink device represented by this example structure interprets pixel data as follows:* • *As being in the sRGB color space and transfer function when receiving a* DisplayPort *VSC SDP that indicates sRGB* • *As being in the Native color space and transfer function indicated in the Display Parameters data block (see Table 4-7) when receiving a* DisplayPort *VSC SDP that indicates Custom Color RGB, –or– when receiving a* DisplayPort *MSA packet that indicates RGB Unspecified*
91h	0x00	**Multi-SST Operation**	Bit 7 = RESERVED = 0. Bits 6:5 = Multi-SST Operation = 00b (Not supported). Bit 4 = RESERVED = 0. Bits 3:0 = Number of Pixels in Horizontal Pixel Count Overlapping an Adjacent Panel Segment = 0h.
92h	0x80	**Checksum**	

B VESA Organization Vendor-specific Data Block (Normative)

Note: *See Appendix A for an example of how the VESA Organization Vendor-specific data block is used in a fixed-length DisplayID Section.*

Table B-1: VESA Organization Vendor-specific Data Block (Normative)

Offset	Bit #	Definition
00h	7:0	**Vendor-specific Data Block** 7Eh.
01h	**Block Revision and Other Data**	
	2:0	**Block Revision** Revision ranges from 0 through 7. 000b = Revision 0 (default).
	7:3	**RESERVED** Cleared to all 0s.
02h	7:0	**Number of Payload Bytes in Block** 05h = Data block is composed of five bytes (if DSC pass-through is not supported). 07h = Data block is composed of seven bytes (if DSC pass-through is supported).
03h	7:0	**IEEE OUI First Byte** Byte length ranges from 0 through 255. 3Ah = First VESA IEEE OUI byte.
04h	7:0	**IEEE OUI Second Byte** Byte length ranges from 0 through 255. 02h = First VESA IEEE OUI byte.
05h	7:0	**IEEE OUI Third Byte** Byte length ranges from 0 through 255. 92h = First VESA IEEE OUI byte.

Table B-1: VESA Organization Vendor-specific Data Block (Normative) (Continued)

Offset	Bit #	Definition
06h	**VESA Data Structure Type**	
	2:0	**Data Structure Type** 000b = Embedded DisplayPort (*eDP*). 001b = External *DisplayPort*. All other values are RESERVED for future use.
	6:3	**RESERVED** Cleared to all 0s.
	7	**Default Color Space and EOTF Handling** 0 = MISC signaling of "RGB unspecified color space" by a Source device is interpreted by a Sink device as sRGB color space and EOTF. 1 = MISC signaling of "RGB unspecified color space" by a Source device is interpreted by a Sink device as native color space and EOTF as specified in the Display Parameters data block.
07h	3:0	**Number of Pixels in Horizontal Pixel Count Overlapping an Adjacent Panel Segment** If Offset 06h[2:0] is programmed to 001b (External *DisplayPort*), this field shall be cleared to 00b (Not supported). Pixel quantities range from 0 through 8. Value range of 9 through 15 is RESERVED.
	4	**RESERVED** Cleared to 0.
	6:5	**Multi-SST Operation** If Offset 06h[2:0] is programmed to 001b (External *DisplayPort*), this field shall be cleared to 00b (Not supported). 00b = Not supported (Conventional Single-Stream Transport). 01b = Two streams (number of links shall be two or four). 10b = Four streams (number of links shall be four). 11b = RESERVED for future use.
	7	**RESERVED** Cleared to 0.

Table B-1: VESA Organization Vendor-specific Data Block (Normative) (Continued)

Offset	Bit #	Definition
08h	5:0	**Pass-through Timing's Integer Target DSC Bits per Pixel** Integer portion of the target *DSC* ***bits_per_pixel*** for pass-through timing descriptors that are declared when the Pass-through Timing Support for Target DSC Bits per Pixel bit of the Type VII Timing data block, Revision 1, is set to 1 (Offset 01h[3] = 1). The DPTX and DPRX shall follow *DP Standard* mandates that pertain to other *DSC* capability support. 0 = Number of integer target *DSC* ***bits_per_pixel*** ranges from 8 through 16. All other values are RESERVED.
	7:6	**RESERVED** Cleared to all 0s.
09h	3:0	**Pass-through Timing's Fractional Target DSC Bits per Pixel** Fractional portion of target *DSC* ***bits_per_pixel*** for pass-through timing descriptors that are declared when the Pass-through Timing Support for Target DSC Bits per Pixel bit of the Type VII Timing data block, Revision 1, is set to 1 (Offset 01h[3] = 1). The DPTX and DPRX shall follow *DP Standard* mandates that pertain to other *DSC* capability support. Number of fractional target *DSC* bits per pixel ranges from 0 through 15, corresponding to fraction *DSC* ***bits_per_pixel*** of (1/16) × value. 0 = No fractional *DSC* ***bits_per_pixel*** exists.
	7:4	**RESERVED** Cleared to all 0s.

C Scan Orientation Clarification (Informative)

This appendix clarifies use of the Scan Orientation bits of the Feature Support Flags field (Offset 0Bh[2:0]) in the Display Parameters data block.

Figures C-1 and C-2 illustrate the various scan orientation settings for example 1,920 × 1,080 landscape and 1,080 × 1,920 portrait panels, respectively, at various viewing configurations. In this example, the native timing exposed in the DisplayID structure is a normal 1,920 × 1,080 or 1,080 × 1,920 timing, respectively, that represents a left-to-right scan panel.

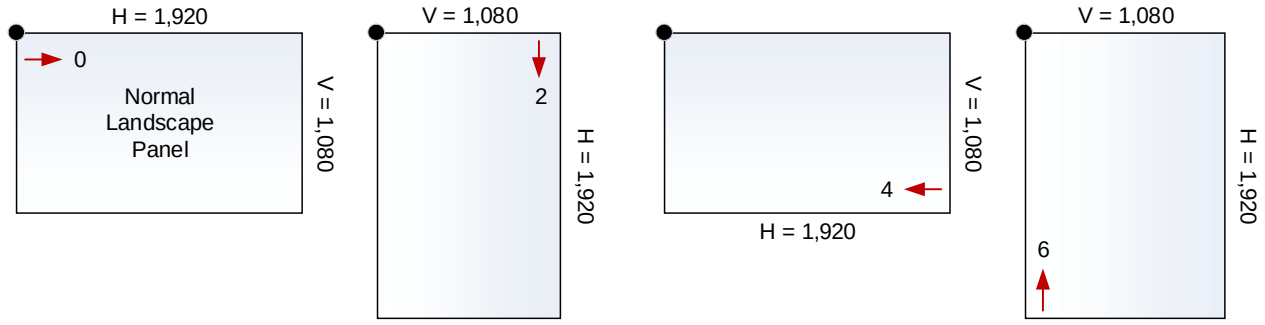


Figure C-1: Scan Orientation Settings for Example 1,920 × 1,080 Landscape Panel at Various Viewing Configurations

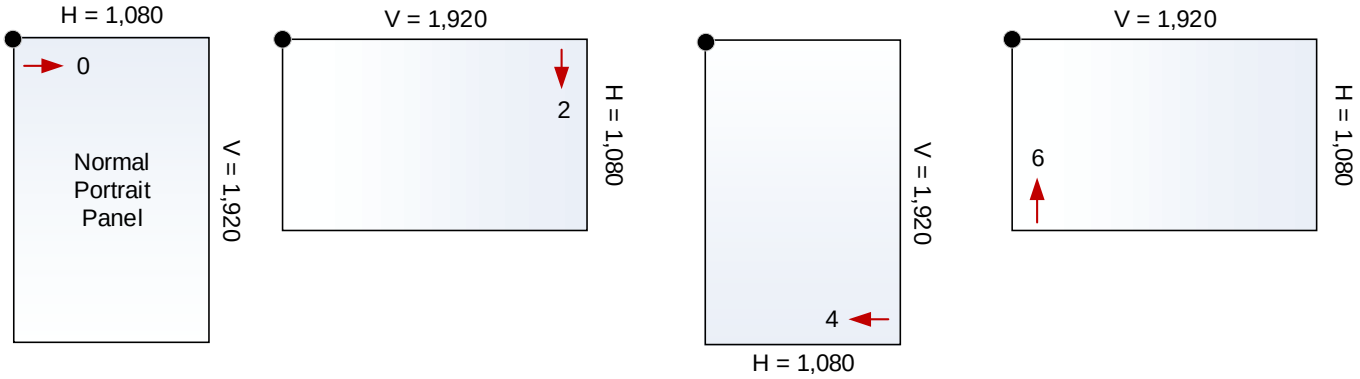


Figure C-2: Scan Orientation Settings for Example 1,080 × 1,920 Portrait Panel at Various Viewing Configurations

Note: *From the viewer's perspective, these examples represent only two display orientations. A landscape display can be implemented as shown in Figure C-1, Orientations 0 and 4, –or– as shown in Figure C-2, Orientations 2 and 6. A portrait display can be implemented as shown in Figure C-1, Orientations 2 and 6, –or– as shown in Figure C-2, Orientations 0 and 4.*

Figures C-3 and C-4 illustrate all possible scan orientations for a landscape display device. Figure C-3 shows the orientations used with landscape timing (e.g., 1,920 × 1,080). Orientations 0 and 4 are "normal" left-to-right scan panels, while Orientations 1 and 5 are "flipped" right-to-left scan panels. Figure C-4 shows the orientations used with portrait timing (e.g., 1,080 × 1,920). Orientations 2 and 6 are "normal" left-to-right scan panels, while Orientations 3 and 7 are "flipped" right-to-left scan panels.

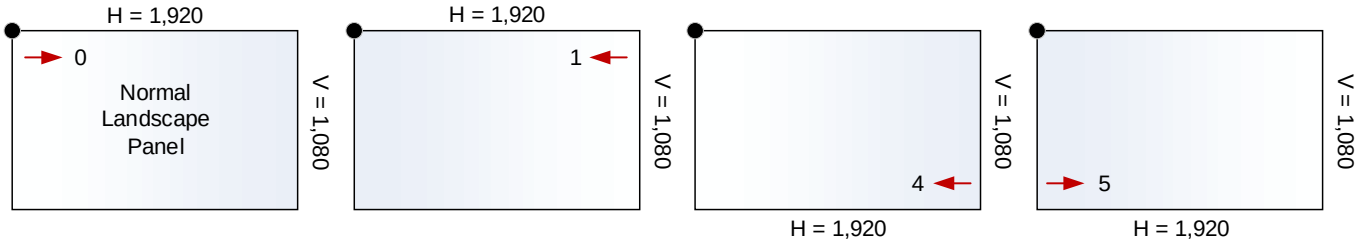


Figure C-3: Scan Orientation Settings Used with an Example Landscape 1,920 × 1,080 Timing to Implement Landscape Display Devices

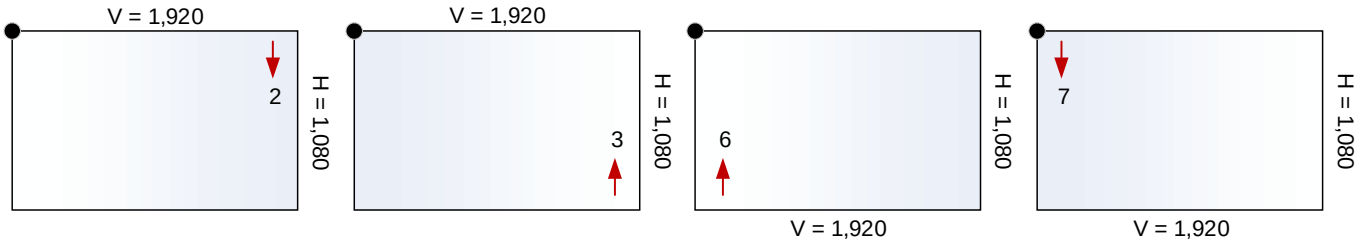


Figure C-4: Scan Orientation Settings Used with an Example Portrait 1,080 × 1,920 Timing to Implement Landscape Display Devices

D Main Contributor History (Previous Revisions)

Table D-1: Main Contributor History (Previous Revisions)

Company	Name	Contribution	Revision
Advanced Micro Devices, Inc.	Aric Cyr	Technical Contributor	2.0
	Syed A. Hussain	Task Group Vice Chair and Document Editor	2.0
Analogix Semiconductor, Inc.	Mehran Badii	Technical Contributor	2.0
Apple, Inc.	Bob Ridenour	Task Group Chair	2.0
Avatar Tech Pubs	Trish McDermott	Technical Writer	2.0
Broadcom Corp.	Chris Pasqualino	Technical Contributor	2.0
EIZO Corp.	Yoshinobu Bamba	Technical Contributor	2.0
JVC Kenwood Corp.	Takayuki Kashahara	Technical Contributor	2.0
NVIDIA Corp.	Kai Chen	Technical Contributor	2.0
	Robert Schutten	Technical Contributor	2.0
Samsung Electronics Co., Ltd.	Dale Stolitzka	Technical Contributor	2.0
Sony Corp.	Robert Blanchard	Technical Contributor	2.0