

[MS-WFDPE]:

Wi-Fi Display Protocol Extension

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation (“this documentation”) for protocols, file formats, data portability, computer languages, and standards support. Additionally, overview documents cover inter-protocol relationships and interactions.
- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you can make copies of it in order to develop implementations of the technologies that are described in this documentation and can distribute portions of it in your implementations that use these technologies or in your documentation as necessary to properly document the implementation. You can also distribute in your implementation, with or without modification, any schemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications documentation.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that might cover your implementations of the technologies described in the Open Specifications documentation. Neither this notice nor Microsoft's delivery of this documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specifications document might be covered by the Microsoft [Open Specifications Promise](#) or the [Microsoft Community Promise](#). If you would prefer a written license, or if the technologies described in this documentation are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **License Programs.** To see all of the protocols in scope under a specific license program and the associated patents, visit the [Patent Map](#).
- **Trademarks.** The names of companies and products contained in this documentation might be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.
- **Fictitious Names.** The example companies, organizations, products, domain names, email addresses, logos, people, places, and events that are depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

Reservation of Rights. All other rights are reserved, and this notice does not grant any rights other than as specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications documentation does not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments, you are free to take advantage of them. Certain Open Specifications documents are intended for use in conjunction with publicly available standards specifications and network programming art and, as such, assume that the reader either is familiar with the aforementioned material or has immediate access to it.

Support. For questions and support, please contact dochelp@microsoft.com.

Revision Summary

Date	Revision History	Revision Class	Comments
6/30/2015	1.0	New	Released new document.
10/16/2015	2.0	Major	Significantly changed the technical content.
7/14/2016	3.0	Major	Significantly changed the technical content.
3/16/2017	4.0	Major	Significantly changed the technical content.
6/1/2017	4.1	Minor	Clarified the meaning of the technical content.

Table of Contents

1	Introduction	5
1.1	Glossary	5
1.2	References	6
1.2.1	Normative References	6
1.2.2	Informative References	7
1.3	Overview	7
1.4	Relationship to Protocols and Other Structures	8
1.5	Applicability Statement	8
1.6	Versioning and Localization	8
1.7	Vendor-Extensible Fields	8
2	Structures	9
2.1	Device Metadata	9
2.1.1	Elements	9
2.1.1.1	intel_friendly_name	9
2.1.1.2	intel_sink_device_URL	9
2.1.1.3	intel_sink_manufacturer_logo	9
2.1.1.4	intel_sink_manufacturer_name	10
2.1.1.5	intel_sink_model_name	10
2.1.1.6	intel_sink_version	10
2.1.2	Attributes	10
2.1.3	Complex Types	10
2.1.4	Simple Types	11
2.2	Enhanced Diagnostics	11
2.2.1	Elements	11
2.2.1.1	microsoft_diagnostics_capability	11
2.2.1.2	microsoft_tear_down_reason	11
2.2.2	Attributes	12
2.2.3	Complex Types	12
2.2.4	Simple Types	12
2.3	Dynamic Resolution and Refresh Rate	12
2.3.1	Elements	12
2.3.1.1	microsoft_format_change_capability	12
2.3.2	Attributes	13
2.3.3	Complex Types	13
2.3.4	Simple Types	13
2.4	Latency Management	13
2.4.1	Elements	13
2.4.1.1	microsoft_latency_management_capability	13
2.4.2	Attributes	14
2.4.3	Complex Types	14
2.4.4	Simple Types	14
2.5	Display Source Identification	15
2.5.1	Elements	15
2.5.1.1	Server header	15
2.5.2	Attributes	15
2.5.3	Complex Types	15
2.5.4	Simple Types	15
2.6	Device Capabilities	16
2.6.1	Elements	16
2.6.1.1	wfd_idr_request_capability	16
2.6.2	Attributes	16
2.6.3	Complex Types	16
2.6.4	Simple Types	16
2.7	Video Formats	16

2.7.1	Elements	17
2.7.1.1	wfdx_video_formats	17
2.7.1.1.1	Tables	18
2.7.1.2	microsoft_video_formats	20
2.7.1.2.1	Tables	21
2.7.2	Attributes	22
2.7.3	Complex Types	22
2.7.4	Simple Types	22
2.8	RTCP	22
2.8.1	Elements	22
2.8.1.1	microsoft_rtcp_capability	22
2.8.2	Attributes	22
2.8.3	Complex Types	22
2.8.4	Simple Types	23
2.9	High-Fidelity Color Space Conversion	23
2.9.1	Elements	23
2.9.1.1	microsoft_color_space_conversion	23
2.9.2	Attributes	24
2.9.3	Complex Types	24
2.9.4	Simple Types	25
2.10	Maximum Supported Bitrate	25
2.10.1	Elements	25
2.10.1.1	microsoft_max_bitrate	25
2.10.2	Attributes	25
2.10.3	Complex Types	25
2.10.4	Simple Types	25
3	Structure Examples	26
4	Security	31
4.1	Security Considerations for Implementers	31
4.2	Index Of Security Fields	31
5	Appendix A: Product Behavior	32
6	Change Tracking	33
7	Index	34

1 Introduction

The Wi-Fi Display Protocol Extension extends the Wi-Fi Display Technical Specification v1.1 [\[WFDTS1.1\]](#) with a set of extensions. This protocol extension set enables latency control, extended diagnostic information, and dynamic format changes on Wi-Fi Display Devices. When implemented, these extensions provide an improved and more consistent Wi-Fi Display experience for a variety of wireless display scenarios, including word processing, web browsing, gaming, and video projection.

Sections 1.7 and 2 of this specification are normative. All other sections and examples in this specification are informative.

1.1 Glossary

This document uses the following terms:

4:2:0 color space: A color space where only half of the vertical and horizontal color information is stored for each pixel.

4:4:4 color space: A color space where the full vertical and horizontal color information is stored for each pixel.

Augmented Backus-Naur Form (ABNF): A modified version of Backus-Naur Form (BNF), commonly used by Internet specifications. ABNF notation balances compactness and simplicity with reasonable representational power. ABNF differs from standard BNF in its definitions and uses of naming rules, repetition, alternatives, order-independence, and value ranges. For more information, see [\[RFC5234\]](#).

base64 encoding: A binary-to-text encoding scheme whereby an arbitrary sequence of bytes is converted to a sequence of printable ASCII characters, as described in [\[RFC4648\]](#).

bit rate: A measure of the average bandwidth that is required to deliver a track, in bits per second (bps).

context-adaptive binary arithmetic coding (CABAC): A form of entropy encoding used in H.264.

instantaneous decoder refresh (IDR): A video frame that can be decoded without reference to previous video frames.

Portable Network Graphics (PNG): A bitmap graphics file format that uses lossless data compression and supports variable transparency of images (alpha channels) and control of image brightness on different computers (gamma correction). PNG-format files have a .png file name extension.

Real-Time Streaming Protocol (RTSP): A protocol used for transferring real-time multimedia data (for example, audio and video) between a server and a client, as specified in [\[RFC2326\]](#). It is a streaming protocol; this means that **RTSP** attempts to facilitate scenarios in which the multimedia data is being simultaneously transferred and rendered (that is, video is displayed and audio is played).

Real-Time Transport Control Protocol (RTCP): A network transport protocol that enables monitoring of Real-Time Transport Protocol (RTP) data delivery and provides minimal control and identification functionality, as described in [\[RFC3550\]](#).

Real-Time Transport Protocol (RTP): A network transport protocol that provides end-to-end transport functions that are suitable for applications that transmit real-time data, such as audio and video, as described in [\[RFC3550\]](#).

sequence parameter set/picture parameter set (SPS/PPS): Data units in an H.264 stream that include metadata about the stream.

Uniform Resource Identifier (URI): A string that identifies a resource. The URI is an addressing mechanism defined in Internet Engineering Task Force (IETF) Uniform Resource Identifier (URI): Generic Syntax [\[RFC3986\]](#).

User Datagram Protocol (UDP): The connectionless protocol within TCP/IP that corresponds to the transport layer in the ISO/OSI reference model.

video frame: A single still image that is shown as part of a quick succession of images in a video.

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as defined in [\[RFC2119\]](#). All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2 References

Links to a document in the Microsoft Open Specifications library point to the correct section in the most recently published version of the referenced document. However, because individual documents in the library are not updated at the same time, the section numbers in the documents may not match. You can confirm the correct section numbering by checking the [Errata](#).

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information.

[CEA-861-F] Consumer Electronics Association (CEA), "A DTV Profile for Uncompressed High Speed Digital Interfaces", CEA-861-F (ANSI), August 2013, <https://www.cta.tech/Research-Standards/Standards-Listing/DTV-Interface-Standards.aspx/>

Note There is a charge to download this specification.

[ISO/IEC-13818-1] International Organization for Standardization, "Information Technology -- Generic Coding of Moving Pictures and Associated Audio Information: Systems", ISO/IEC 13818-1:2007, http://www.iso.org/iso/catalogue_detail?csnumber=44169%20

Note There is a charge to download the specification.

[ITU-H.264-201201] ITU-T, "Advanced video coding for generic audiovisual services", Recommendation H.264, January 2012, <http://www.itu.int/rec/T-REC-H.264-201201-S/en>

[MS-ERREF] Microsoft Corporation, "[Windows Error Codes](#)".

[PNG] ISO/IEC 15948:2004., "Portable Network Graphics PNG", http://www.iso.org/iso/catalogue/catalogue_tc/catalogue_detail.htm?csnumber=29581

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, <http://www.rfc-editor.org/rfc/rfc2119.txt>

[RFC2250] Hoffman, D., Fernando, G., Goyal, V., and Civanlar, M., "RTP Payload Format for MPEG1/MPEG2 Video", RFC 2250, January 1998, <http://www.ietf.org/rfc/rfc2250.txt>

[RFC2326] Schulzrinne, H., Rao, A., and Lanphier, R., "Real Time Streaming Protocol (RTSP)", RFC 2326, April 1998, <http://www.ietf.org/rfc/rfc2326.txt>

[RFC2616] Fielding, R., Gettys, J., Mogul, J., et al., "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999, <http://www.rfc-editor.org/rfc/rfc2616.txt>

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and Jacobson, V., "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003, <http://www.ietf.org/rfc/rfc3550.txt>

[RFC3629] Yergeau, F., "UTF-8, A Transformation Format of ISO 10646", STD 63, RFC 3629, November 2003, <http://www.ietf.org/rfc/rfc3629.txt>

[RFC3986] Berners-Lee, T., Fielding, R., and Masinter, L., "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005, <http://www.rfc-editor.org/rfc/rfc3986.txt>

[RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, October 2006, <http://www.rfc-editor.org/rfc/rfc4648.txt>

[RFC5234] Crocker, D., Ed., and Overell, P., "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008, <http://www.rfc-editor.org/rfc/rfc5234.txt>

[VESA-CVT] Video Electronics Standards Association (VESA), "Coordinated Video Timings (CVT) v1.2", <http://vesa.org/vesa-standards/standards-summaries>

[WF-DTS1.1] Wi-Fi Alliance, "Wi-Fi Display Technical Specification v1.1", April 2014, <https://www.wi-fi.org/file/wi-fi-display-technical-specification-v11>

Note There is a charge to download the specification.

1.2.2 Informative References

None.

1.3 Overview

The Wi-Fi Display protocol [\[WF-DTS1.1\]](#) is used for a variety of scenarios; however, the Wi-Fi Display protocol does not allow for a Wi-Fi Display Source [WF-DTS1.1] to communicate the user's intent to the Wi-Fi Display Sink [WF-DTS1.1]. For example, a Wi-Fi Display Sink can be designed for watching movies, with a high display latency that facilitates smooth playback and post-processing. A user who wants to play a game will find the latency disturbing because the game requires real-time responses. Similarly, a user who wants to watch a full-screen movie might find the jitter and artifacts of a lower-latency Wi-Fi Display Sink to be distracting because it can affect the display quality of the movie. Ideally, the display is able to match the frame rate and frame size of the video content. The Wi-Fi Display Protocol Extension facilitates communication of the user's intent from the Wi-Fi Display Source to the Wi-Fi Display Sink. Once the user intent is known, the latency can be changed according to the scenario needs.

Additionally, because Wi-Fi Display connections can spontaneously disconnect for a variety of reasons, it is difficult to determine the reason for such connection failures after the fact. This protocol extension set enables a Wi-Fi Display Sink to communicate additional information about itself and the reason for a disconnection, when applicable. This information helps Wi-Fi Display Device vendors and implementers to resolve problems and improve usability.

This protocol extension set enables a Wi-Fi Display Source to negotiate resolutions other than those that are supported in [WF-DTS1.1]. This capability enables support for 4K resolution and for resolutions with a 3:2 picture aspect ratio, for example. [<1>](#)

This protocol extension uses **Real-Time Transport Control Protocol (RTCP)** messages between the Wi-Fi Display Source and the Wi-Fi Display Sink to analyze the user's current throughput and modify the encoding parameters from [WF-DTS1.1], to improve the user's current experience. [<2>](#)

This protocol extension adds a high-fidelity color space conversion extension that enables the Wi-Fi Display Source to encode additional color information in repeat frames, which produces a **4:4:4 color space** quality frame by using a **4:2:0 color space** encoder. [<3>](#)

This protocol extension enables Wi-Fi Display Sinks to specify their maximum supported bitrate, which is required because some Wi-Fi Display Sinks are unable to handle high encoding bitrates due to hardware limitations. <4>

1.4 Relationship to Protocols and Other Structures

The Wi-Fi Display Protocol Extension extends the Wi-Fi Display Technical Specification v1.1 [\[WF-DTS1.1\]](#).

1.5 Applicability Statement

This protocol extension is intended for any Wi-Fi Display Device.

1.6 Versioning and Localization

None.

1.7 Vendor-Extensible Fields

This protocol uses HRESULT values as defined in [\[MS-ERREF\]](#) section 2.1. Vendors can define their own HRESULT values, provided they set the **C** bit (0x20000000) for each vendor-defined value, indicating the value as a customer code.

2 Structures

Protocol extensions are defined using **Augmented Backus-Naur Form (ABNF)**, as specified in [\[RFC5234\]](#).

A Wi-Fi Display Sink implementing any of the protocol extensions defined in this specification MUST support **context-adaptive binary arithmetic coding (CABAC)** in both the Baseline Profile and High Profile as specified in [\[ITU-H.264-201201\]](#) sections 9.3, A.2.1, and A.2.4.

A Wi-Fi Display Device implementing any of the protocol extension defined in this specification MUST use the M bit of the **Real-Time Transport Protocol (RTP)** packet header in the manner prescribed for video data in [\[RFC2250\]](#) section 3.3.

2.1 Device Metadata

This section extends [\[WF-DTS1.1\]](#) section 6.1, with additional data structures for device metadata.

2.1.1 Elements

The following subsections provide details about the device metadata data structures.

2.1.1.1 intel_friendly_name

The **intel_friendly_name** parameter specifies a human-readable name of the Wi-Fi Display Sink.

The **ABNF** syntax is as follows:

```
intel-friendly-name = "intel friendly name:" SP friendly-name CRLF
friendly-name = 1*18(utf8byte-no-hyphen)
utf8byte-no-hyphen = %x00-2C / %x2E-FF
```

The **friendly-name** parameter MUST be formatted as specified in [\[RFC3629\]](#). The hyphen code point ("-") MUST NOT be included in the **friendly-name** parameter.

2.1.1.2 intel_sink_device_URL

The **intel_sink_device_URL** parameter specifies a **Uniform Resource Identifier (URI)** for the product information of the Wi-Fi Display Sink.

The **ABNF** syntax is as follows:

```
intel-sink-device-URL = "intel_sink_device_URL:" SP uri CRLF
uri = 1*256(VCHAR) / "none"
```

The **intel_sink_device_URL** parameter specifies a URI as specified in [\[RFC3986\]](#). A value of "none" means that no product information is available.

2.1.1.3 intel_sink_manufacturer_logo

The **intel_sink_manufacturer_logo** parameter specifies an image file representing the manufacturer of the Wi-Fi Display Sink. The image MUST be in **Portable Network Graphics (PNG)** format with the following specifications: 96 dots-per-inch, 24 bits per pixel, 160 pixels wide, and 120 pixels high [\[PNG\]](#). The image MUST be **base64**-encoded, as specified in [\[RFC4648\]](#).

The **ABNF** syntax is as follows:

```
intel-sink-manufacturer-logo = "intel_sink_manufacturer_logo:" SP logo CRLF
logo = "none" / base64-logo
base64-logo = 464*76800 (BASE64CHAR)
```

A value of "none" means that no image is available.

2.1.1.4 intel_sink_manufacturer_name

The **intel_sink_manufacturer_name** parameter specifies the name of the manufacturer of the Wi-Fi Display Sink.

The **ABNF** syntax is as follows:

```
intel-sink-manufacturer-name = "intel_sink_manufacturer_name:" SP manufacturer-
name CRLF
manufacturer-name = 1*32 (VCHAR) / "none"
```

A value of "none" means that the manufacturer name is not available.

2.1.1.5 intel_sink_model_name

The **intel_sink_model_name** parameter specifies the model name assigned by the manufacturer of the Wi-Fi Display Sink.

The **ABNF** syntax is as follows:

```
intel-sink-model-name = "intel sink model name:" SP model-name CRLF
model-name = 1*32 (VCHAR) / "none"
```

A value of "none" means that the model name is not available.

2.1.1.6 intel_sink_version

The **intel_sink_version** parameter specifies the product identifier, hardware version, and software version of the Wi-Fi Display Sink.

```
intel-sink-version = "intel_sink_version:" SP product-id SP hw-version SP sw-version CRLF
product-id = "product ID=" 1*16 (VCHAR)
hw-version = "hw_version=" version-tag
sw-version = "sw_version=" version-tag
version-tag = major "." minor "." sku "." build
major = 1*2 (DIGIT)
minor = 1*2 (DIGIT)
sku = 1*2 (DIGIT)
build = 1*4 (DIGIT)
```

2.1.2 Attributes

Not applicable.

2.1.3 Complex Types

Not applicable.

2.1.4 Simple Types

Not applicable.

2.2 Enhanced Diagnostics

The enhanced diagnostics protocol extension enables the Wi-Fi Display Sink to report error codes and error reasons to the Wi-Fi Display Source.

The extension consists of a data structure to first negotiate whether or not the diagnostics extension is supported by the Wi-Fi Display Sink. The data structure extends [\[WF-DTS1.1\]](#) section 6.1. Additionally, the **M8** message, as specified in [\[WF-DTS1.1\]](#) section 6.4.8, is extended with a payload.

2.2.1 Elements

2.2.1.1 microsoft_diagnostics_capability

The **microsoft_diagnostics_capability** parameter specifies whether a Wi-Fi Display Sink supports including the **microsoft_teardown_reason** parameter (section [2.2.1.2](#)) in the message body of the **M8** request.

The **ABNF** syntax is as follows:

```
microsoft-diagnostics-capability = "microsoft_diagnostics_capability:" SP
                                   diagnostics-capability CRLF
diagnostics-capability = "supported" / "none"
```

If the **diagnostics-capability** parameter has the value "supported", it means that the **M8** request ([\[WF-DTS1.1\]](#) section 6.4.8) that was sent by the Wi-Fi Display Sink includes the **microsoft_teardown_reason** parameter in the message body of that request. If the **diagnostics-capability** parameter has the value "none", it means that there are no changes to the **M8** request.

2.2.1.2 microsoft_teardown_reason

The **microsoft_teardown_reason** parameter is included in the message body of the **M8** message ([\[WF-DTS1.1\]](#) section 6.4.8) when the Wi-Fi Display Sink sets the **diagnostics-capability** parameter of the **microsoft_diagnostics_capability** parameter (section [2.2.1.1](#)) to "supported".

The **ABNF** syntax is as follows:

```
microsoft-teardown-reason = "microsoft_teardown_reason:" SP error-code SP error-reason
                           CRLF
error-code = 8*8HEXDIG
error-reason = *VCHAR
```

The 8 hexadecimal digit value of the error-code parameter is an HRESULT value as specified in [\[MS-ERREF\]](#) section 2.1. The following predefined error codes SHOULD be preferred over custom error codes when the reason for the failure applies.

Return value/code	Failure condition
MF_E_CANNOT_PARSE_BYTESTREAM C00D36F0	Cannot parse byte stream; the incoming data is not a valid MPEG-2 stream.
MF_E_INVALID_FORMAT C00D3E8C	The stream is valid, but the format (resolution, frame rate, channel count, and so on) cannot be handled.
MF_E_TRANSFORM_CANNOT_CHANGE_MEDIA_TYPE_WHILE_PROCESSING C00D6D74	The format of the H.264, AAC, or AC3 ITU-H.264-201201 bit streams changed, but the change cannot be handled.
MF_E_INVALID_STREAM_DATA C00D36CB	The H.264, AAC, or AC3 bit stream is not valid and cannot be decoded.
MF_E_NET_TIMEOUT C00D4278	The Wi-Fi Display Sink timed out waiting for a keep-alive or RTP data.
MF_E_INVALID_TIMESTAMP C00D36C0	The presentation time stamps in the MPEG-2 Packetized Elementary Stream packets are corrupted, and the Wi-Fi Display Sink can no longer render the stream.

If the Wi-Fi Display Sink encounters an error that does not correspond to any of the failure conditions in the previous table, it SHOULD use a custom error code. A custom error code MUST NOT be any of the predefined error codes, and MUST set the **C** bit (0x20000000) for each vendor-defined value, indicating the value as a customer code.

Error reasons are implementation-defined and provide a human-readable explanation of the error condition. A Wi-Fi Display Source vendor and a Wi-Fi Display Sink vendor working together can use the error reason to identify and potentially correct the error condition.

2.2.2 Attributes

Not applicable.

2.2.3 Complex Types

Not applicable.

2.2.4 Simple Types

Not applicable.

2.3 Dynamic Resolution and Refresh Rate

The dynamic resolution and refresh rate extension allows the Wi-Fi Display Source to change the video resolution or video refresh rate of the video stream without sending an additional **Real-Time Streaming Protocol (RTSP)** message to the Wi-Fi Display Sink.

The extension consists of a data structure to negotiate the support for dynamic changes to the video resolution and video refresh rate. The data structure extends [\[WF-DTS1.1\]](#) section 6.1.

2.3.1 Elements

2.3.1.1 microsoft_format_change_capability

The **microsoft_format_change_capability** parameter specifies whether the Wi-Fi Display Sink supports dynamic changes to the video resolution and video refresh rate.

The **ABNF** syntax is as follows:

```
microsoft-format-change-capability = "microsoft format change capability:" SP
    format-change-caps CRLF
format-change-caps = "supported" / "none"
```

If the **format-change-caps** parameter equals the value "supported", it means that the Wi-Fi Display Sink monitors the **sequence parameter set/picture parameter set (SPS/PPS)** in the H.264 stream for changes to the video resolution or **video frame** rate. And it adapts to such changes without displaying any visible changes such as flicker or a black screen.

The H.264 video bit stream that is sent by a Wi-Fi Display Source MUST include an **instantaneous decoder refresh (IDR)** frame as the first video frame after changing either the video resolution or the video frame rate. The bit stream MUST also include the SPS/PPS with the new resolution and frame rate. The new video resolution MUST belong to the set of video resolutions included in the Wi-Fi Display Sink's **M3** message response. The video frame rate MUST be less than or equal to the maximum frame rate that the Wi-Fi Display Sink claims support for at that resolution.

For details about how the Wi-Fi Display Sink specifies which video resolutions and video frame rates it supports, see [\[WF-DTS1.1\]](#) section 6.1.3.

If the **format-change-caps** parameter has the value "none", the frame rate and video resolution of the H.264 bit stream cannot be changed unless the Wi-Fi Display Source sends an **M4** message ([\[WF-DTS1.1\]](#) section 6.4.4).

2.3.2 Attributes

Not applicable.

2.3.3 Complex Types

Not applicable.

2.3.4 Simple Types

Not applicable

2.4 Latency Management

The latency management extension allows the Wi-Fi Display Source to inform the Wi-Fi Display Sink of the user intent regarding the display latency (section [2.4.1.1](#)) of the currently streaming content.

The extension consists of a data structure to negotiate support and changes for the video display latency. The data structure extends [\[WF-DTS1.1\]](#) section 6.1.

2.4.1 Elements

2.4.1.1 microsoft_latency_management_capability

The **microsoft_latency_management_capability** parameter specifies whether the Wi-Fi Display Sink is capable of dynamically changing the display latency of the video bit stream. When sent by the Wi-Fi Display Sink, the parameter specifies the desired latency mode.

This parameter is included by the Wi-Fi Display Source in the M3 request to specify support for latency management, by the Wi-Fi Display Sink in the M3 response to specify support for latency management, and by the Wi-Fi Display Source in a SET_PARAMETER request to set the latency mode to a new value.

For the purposes of this section, "latency" is defined as the time from the moment the last **RTP** packet is received for a **video frame**, until the time that the Wi-Fi Display Sink renders the frame to the output device.

The **ABNF** syntax is as follows:

```
microsoft-latency-management-capability = "microsoft_latency_management_capability:" SP  
    ( latency-management-cap / latency-mode ) CRLF  
latency-management-cap = "supported" / "none"  
latency-mode = "low" / "normal" / "high"
```

Possible values for the **latency-management-cap** parameter are as follows:

Value	Meaning
supported	Specifies that the Wi-Fi Display Sink is capable of changing the latency mode.
none	Specifies that the Wi-Fi Display Sink is not capable of changing the latency mode.

Possible values for the **latency-mode** parameter are as follows:

Value	Meaning
low	Specifies that the Wi-Fi Display Sink SHOULD keep latency under 50 milliseconds. <5>
normal	Specifies that the Wi-Fi Display Sink SHOULD keep latency under 100 milliseconds. <6>
high	Specifies that the Wi-Fi Display Sink SHOULD buffer additional frames in order to ensure smooth playback, as long as the latency stays under 500 milliseconds.

2.4.2 Attributes

Not applicable.

2.4.3 Complex Types

Not applicable.

2.4.4 Simple Types

Not applicable.

2.5 Display Source Identification

The Display Source Identification protocol extension enables the Wi-Fi Display Source, as specified in [\[WF-DTS1.1\]](#), to provide the version number of the Wi-Fi Display software used by the Wi-Fi Display Source. The protocol extension also enables the Display Source to specify a unique identifier that identifies the current Wi-Fi Display connection.

This information can aid users of the Wi-Fi Display Sink, as specified in [\[WF-DTS1.1\]](#), in troubleshooting connectivity issues.

For example, if a certain version of the Wi-Fi Display software has a defect that causes connections to fail, the version number in the Display Source Identification can identify if that version of the software is being used by the Wi-Fi Display Source.

2.5.1 Elements

2.5.1.1 Server header

The **Server header** field appears in any **RTSP** responses generated by the Wi-Fi Display Source. The **Server header** field provides the version number of the Wi-Fi Display software used by the Wi-Fi Display Source. The protocol extension also enables the Display Source to specify a unique identifier that identifies the current Wi-Fi Display connection.

The **ABNF** syntax of the **Server header** field is as follows.

```
source-product-id      = "MSMiracastSource"
connection-id          = 8HEXDIG "-" 4HEXDIG "-" 4HEXDIG "-" 4HEXDIG "-" 12HEXDIG
connection-id-token    = "guid/" connection-id
server-header-data     = source-product-id "/" product-version [ SP connection-id-token ]
                        *( SP product )
Server                 = "Server:" SP server-header-data
```

The **product** and **product-version** syntax elements are specified in [\[RFC2616\]](#) section 3.8.

The **connection-id** syntax element is an identifier that uniquely identifies the current Wi-Fi Display connection.

To ensure a high probability of uniqueness, the Wi-Fi Display Source SHOULD use a random number generator to generate the syntax element, as shown in the following example.

```
Server: MSMiracastSource/10.00.10011.0000 guid/be113d06-9e40-43e4-98e6-540a325e9ced
```

2.5.2 Attributes

Not applicable.

2.5.3 Complex Types

Not applicable.

2.5.4 Simple Types

Not applicable.

2.6 Device Capabilities

The device capabilities protocol extension enables the Wi-Fi Display Source to determine whether the Wi-Fi Display Sink supports certain capabilities.

The extension consists of a data structure that specifies whether a given capability is supported. There is a separate data structure for each capability.

The data structures extend [\[WF-DTS1.1\]](#) section 6.1.

Currently, the extension defines the **wfd_idr_request_capability** data structure (section [2.6.1.1](#)).

2.6.1 Elements

2.6.1.1 wfd_idr_request_capability

The **wfd_idr_request_capability** parameter specifies whether a Wi-Fi Display Sink supports sending an **RTSP** M13 message as specified in [\[WF-DTS1.1\]](#) section 6.4.13.

Since the purpose of the RTSP M13 message is to request that the Wi-Fi Display Source generate an **instantaneous decoder refresh (IDR)** picture, it might be useful for a Wi-Fi Display Source to determine a case where the Wi-Fi Display Sink never sends an RTSP M13 message.

A Wi-Fi Display Source that determines that it will never receive any RTSP M13 messages can insert instantaneous decoder refresh (IDR) pictures more frequently in the video bit stream, to compensate for the Wi-Fi Display Sink's inability to request such pictures.

The **ABNF** syntax is as follows:

```
idr-request-capability = "wfd_idr_request_capability:" SP idr-req-cap-val CRLF
idr-req-cap-val = "0" / "1"
```

If the **idr-req-cap-val** parameter has the value "1", it means that the Wi-Fi Display Sink is capable of sending RTSP M13 messages.

If the **idr-req-cap-val** parameter has the value "0", it means that the Wi-Fi Display Sink does not send any RTSP M13 messages.

2.6.2 Attributes

Not applicable.

2.6.3 Complex Types

Not applicable.

2.6.4 Simple Types

Not applicable.

2.7 Video Formats

Additional video formats, including 4K resolution formats and formats that use 3:2 picture aspect ratio, are supported via the Wi-Fi Display video formats protocol extension. This protocol extension allows a Wi-Fi Display Source to send additional video formats to a Wi-Fi Display Sink. [<7>](#)

The extension consists of two data structures to negotiate additional video formats. The data structures extend [\[WF-DTS1.1\]](#) section 6.1.3.

2.7.1 Elements

2.7.1.1 wfdx_video_formats

The **wfdx-video-formats** parameter replaces the **wfd-video-formats** parameter as specified in [\[WF-DTS1.1\]](#) section 6.1.3. [<8>](#) The **wfdx-video-formats** parameter specifies:

- Additional supported video resolutions that augment the video resolutions specified in [WF-DTS1.1] sections 5.1.5.1–5.1.5.3.
- A codec profile that augments the codec profile specified in [WF-DTS1.1] section 5.1.5.5.
- A level that augments the level specified in [WF-DTS1.1] section 5.1.5.5.
- Decoder latency.
- Minimum slice size.
- Slice encoding parameters and support for video frame rate control, including explicit frame rate change and implicit video frame skipping.

A Wi-Fi Display Sink can send both **wfd-video-formats** and **wfdx-video-formats** in an M3 response. If an M3 response contains both **wfd-video-formats** and **wfdx-video-formats**, the Wi-Fi Display Source MUST NOT use the **wfd-video-formats** parameter.

The M4 request MUST NOT contain both **wfd-video-formats** and **wfdx-video-formats**. If the M4 request contains both **wfd-video-formats** and **wfdx-video-formats**, the Wi-Fi Display Sink MUST NOT use the **wfd-video-formats** parameter.

The parameters definitions in the following **ABNF** syntax are different than those specified in [WF-DTS1.1] section 6.1.3: **wfd-video-formats**, **native**, **profile**, **level**, **CEA-Support**, and **VESA-Support**.

The ABNF syntax is as follows:

```
wfdx-video-formats = "wfdx_video_formats:" SP sink-video-list CRLF
sink-video-list = "none" / (native SP preferred-display-mode-supported SP H264-codec);
native = 4*4HEXDIG;
preferred-display-mode-supported = 2*2HEXDIG; 0-not supported, 1-supported, 2-255 reserved
H264-codec = profile SP level SP misc-params SP max-hres SP max-vres *(", " SP
              H264-codec)
profile = 4*4HEXDIG;
level = 4*4HEXDIG;
max-hres = "none" / (4*4HEXDIG);
max-vres = "none" / (4*4HEXDIG);
misc-params = CEA-Support SP VESA-Support SP HH-Support SP latency SP min-slice-size SP
slice-enc-params SP frame-rate-control-support
```

```

CEA-Support = 10*10HEXDIG;
VESA-Support = 10*10HEXDIG;
HH-Support = 8*8HEXDIG;
latency = 2*2HEXDIG;
min-slice-size = 4*4HEXDIG;
slice-enc-params = 4*4HEXDIG;
frame-rate-control-support = 2*2HEXDIG;

```

The codec is a list of one or more **profile**, **level**, **misc-params**, **max-hres**, or **max-vres** tuples for each codec profile, corresponding to the maximum level, miscellaneous parameters, maximum horizontal resolution, and maximum vertical resolution that are supported when included in an **RTSP** M3 response messages. The **level** parameter indicates the maximum level support for the specified profile. Tuples can appear in any order in an RTSP M3 response message.

A Wi-Fi Display Sink MUST set the Video Frame Rate Change Support bit in the **frame-rate-control-support** field to 1 in the RTSP M3 response message, if it supports this functionality. If the Wi-Fi Display Sink does not set this bit to 1, then the Wi-Fi Display Source MUST NOT set this bit to 1 in the RTSP M4 request message. If the Wi-Fi Display Source does not support the functionality, it MUST set this bit to 0.

A Wi-Fi Display Source MUST set the **min-slice-size** value to 0 in the RTSP M4 request message and MUST NOT transmit an encoded picture constructed by multiple slices to a Wi-Fi Display Sink that does not support decoding a picture constructed by multiple slices (the Wi-Fi Display Sink sets the **min-slice-size** value to 0 in the RTSP M3 response message).

2.7.1.1.1 Tables

CEA Resolutions and Refresh Rates with Extension

The following table specifies additional CEA (Consumer Electronics Association) resolutions and refresh rates, other than the CEA resolutions and refresh rates as specified in [\[WF-DTS1.1\]](#) section 5.1.5.1.

Bits	Index	Interpretation	Reference
16:0	-		As specified in [WF-DTS1.1] section 5.1.5.1
17	17	3840x2160 p30	As specified in [CEA-861-F] section 4
18	18	3840x2160 p60	As specified in [CEA-861-F] section 4
19	19	4096x2160 p30	As specified in [CEA-861-F] section 4
20	20	4096x2160 p60	As specified in [CEA-861-F] section 4
21	21	3840x2160 p25	As specified in [CEA-861-F] section 4
22	22	3840x2160 p50	As specified in [CEA-861-F] section 4
23	23	4096x2160 p25	As specified in [CEA-861-F] section 4
24	24	4096x2160 p50	As specified in [CEA-861-F] section 4
25	25	3840x2160 p24	As specified in [CEA-861-F] section 4
26	26	4096x2160 p24	As specified in [CEA-861-F] section 4
39:27	-	Reserved	

The Wi-Fi Display Sink SHOULD use the timings of the corresponding display device. For example, if a monitor is using CEA timings, then CEA-861-F timings apply as specified in [\[CEA-861-F\]](#) section 4. If a

monitor is using VESA timings, then, VESA CVT timings apply as specified in [\[VESA-CVT\]](#) section 4, even if the entry originated from the CEA table.

VESA Resolutions and Refresh Rates with Extension

The following table specifies additional VESA resolutions and refresh rates, other than the VESA resolutions and refresh rates as specified in [\[WF-DTS1.1\]](#) section 5.1.5.2.

Bits	Index	Interpretation	Reference
28:0	-		As specified in [WF-DTS1.1] section 5.1.5.2
29	29	2560x1440 p30 (WQHD)	As specified in [VESA-CVT] section 4
30	30	2560x1440 p60 (WQHD)	As specified in [VESA-CVT] section 4
31	31	2560x1600 p30 (WQXGA)	As specified in [VESA-CVT] section 4
32	32	2560x1600 p60 (WQXGA)	As specified in [VESA-CVT] section 4
39:33	-	Reserved	

Entries from Table 5-10.1 (CEA) are not duplicated in Table 5-11.1 (VESA). In order to use a CEA entry for a VESA monitor, VESA CVT timings are used instead of CEA timings.

All entries use DMT timings (if defined in VESA DMT); otherwise, the CVT-computed value for timings is used.

Display Native Resolution Refresh Rates with Extension

The following table specifies additional display native resolution refresh rates, other than the Display native resolution refresh rates as specified in [\[WF-DTS1.1\]](#) section 5.1.5.4.

Bits	Index	Interpretation
2:0	Table Selection bits	0b010 – 0b000: As specified in [WF-DTS1.1] section 5.1.5.4
8:3	-	Index into resolution/refresh rate table selected by [B2:B0] (twice the size specified in [WF-DTS1.1])
15:9	Reserved	Reserved

Profiles Bitmap with Extension

The following table specifies additional Profiles Bitmap, other than the Profiles Bitmap as specified in [\[WF-DTS1.1\]](#) section 5.1.5.5.

Bits	Name	Interpretation
0	H.264 Constrained Baseline Profile	As specified in [WF-DTS1.1] section 5.1.5.5

Bits	Name	Interpretation
1	H.264 Constrained High Profile	As specified in [WF-DTS1.1] section 5.1.5.5
2	H.265 Version 1 Main Profile (8-bit 4:2:0)	0=unsupported, 1=supported
3	H.265 Version 1 Main 10 Profile (10bit 4:2:0)	0=unsupported, 1=supported
15:4	Reserved	Reserved

All entries in the previous table MUST NOT use the B slice tool and MUST NOT use the CABAC entropy coding tool, as specified in [WF-DTS1.1] section 3.4.2.

Levels Bitmap with Extension

The following table specifies additional Levels Bitmap, other than the Levels Bitmap as specified in [WF-DTS1.1] section 5.1.5.6.

Bits	Name	Interpretation
0	Level 3.1 bit	0b0: Level 3.1 not supported 0b1: Level 3.1 supported
1	Level 3.2 bit	0b0: Level 3.2 not supported 0b1: Level 3.2 supported
2	Level 4 bit	0b0: Level 4 not supported 0b1: Level 4 supported
3	Level 4.1 bit	0b0: Level 4.1 not supported 0b1: Level 4.1 supported
4	Level 4.2 bit	0b0: Level 4.2 not supported 0b1: Level 4.2 supported
5	Level 5 bit	0b0: Level 5 not supported 0b1: Level 5 supported
6	Level 5.1 bit	0b0: Level 5.1 not supported 0b1: Level 5.1 supported
7	Level 5.2 bit	0b0: Level 5.2 not supported 0b1: Level 5.2 supported
15:8	Reserved	Reserved

2.7.1.2 microsoft_video_formats

The **microsoft-video-formats** parameter specifies additional supported video resolutions that augment the video resolutions specified in [\[WF-DTS1.1\]](#) sections 5.1.5.1–5.1.5.3. [<9>](#)

The video resolutions specified by the **microsoft-video-formats** parameter apply to each **H264-codec** parameter specified in the **wfd-video-formats parameter** (section [2.7.1.1](#)), as long as the resolution fits within the constraints of the profile and level specified in the **H264-codec** parameter.

The ABNF syntax is as follows:

```
microsoft-video-formats = "microsoft_video_formats:" SP microsoft-resolutions CRLF
microsoft-resolutions = 12HEXDIG
```

The possible values for the **microsoft-video-formats** parameter are specified in the table in section [2.7.1.2.1](#).

2.7.1.2.1 Tables

The following table specifies the resolutions and refresh rates that are specified by the **microsoft-video-formats** parameter (section [2.7.1.2](#)).

Bits	Index	Interpretation
0	0	1920x1280 p30
1	1	1920x1280 p60
2	2	1920x1280 p24
3	3	2160x1440 p30
4	4	2160x1440 p60
5	5	2160x1440 p24
6	6	2256x1504 p30
7	7	2256x1504 p60
8	8	2256x1504 p24
9	9	2736x1824 p30
10	10	2736x1824 p60
11	11	2736x1824 p24
12	12	3000x2000 p30
13	13	3000x2000 p60
14	14	3000x2000 p24
15	15	3240x2160 p30
16	16	3240x2160 p60
17	17	3240x2160 p24

Bits	Index	Interpretation
18	18	4500x3000 p30
19	19	4500x3000 p60
20	20	4500x3000 p24

The Wi-Fi Display Sink SHOULD use the timings of the corresponding display device. For example, if a monitor is using CEA timings, then CEA-861-F timings apply as specified in [\[CEA-861-F\]](#) section 4. If a monitor is using VESA timings, then, VESA CVT timings apply as specified in [\[VESA-CVT\]](#) section 4.

2.7.2 Attributes

Not applicable.

2.7.3 Complex Types

Not applicable.

2.7.4 Simple Types

Not applicable.

2.8 RTCP

The RTCP extension enables the Wi-Fi Display Source and Wi-Fi Display Sink to communicate information regarding the quality of the network connection between the two devices. [<10>](#) A Wi-Fi Display Source can use this information to adjust the encoding **bit rate** in accordance with network conditions.

2.8.1 Elements

2.8.1.1 microsoft_rtcp_capability

The **microsoft_rtcp_capability** parameter specifies whether a Wi-Fi Display Sink supports **Real-Time Transport Control Protocol (RTCP)** in accordance with [\[RFC3550\]](#). The **ABNF** syntax is as follows:

```
microsoft-rtcp-capability = "microsoft_rtcp_capability:" SP rtcp-caps CRLF
rtcp-caps = "supported" / "none"
```

This parameter is included by the Wi-Fi Display Source in the **RTSP** M3 request message to specify support for RTCP and by the Wi-Fi Display Sink in the RTSP M3 response message to specify support for RTCP.

If the Wi-Fi Display Sink specifies "supported" as the value of the **rtcp-caps** field in **microsoft_rtcp_capability** in the RTSP M3 response message, then the Wi-Fi Display Source MUST include the second port number on the **server_port** field on the Transport header of the RTSP SETUP response message, as specified in [\[RFC2326\]](#) section 12.39. The Wi-Fi Display Sink SHOULD [<11>](#) transmit RTCP packets ([RFC3550] section 6) to the **UDP** port that is specified as the second port number on the **server_port** field on the Transport header of the RTSP SETUP response message.

If the Wi-Fi Display Sink specifies "none" as the value of the **rtcp-caps** field in **microsoft_rtcp_capability**, or does not include **microsoft_rtcp_capability** in the RTSP M3

response message, then the Wi-Fi Display Source MUST specify exactly one port number in the **server_port** field on the Transport header of the RTSP SETUP response message.

2.8.2 Attributes

Not applicable.

2.8.3 Complex Types

Not applicable.

2.8.4 Simple Types

Not applicable.

2.9 High-Fidelity Color Space Conversion

The high-fidelity color space conversion extension enables the Wi-Fi Display Source to encode additional color information in repeat frames.[<12>](#<12>) In general, the Wi-Fi Display Source has to downsample (lower the sampling rate) the color information when converting from a **4:4:4 color space** to a **4:2:0 color space** for H.264 encoding. The extension uses repeat frames (a **video frame** that is identical to the previous frame) to include the color information that was lost during the conversion, enabling the Wi-Fi Display Sink to connect color information from four frames and generate one complete frame with full color fidelity.

2.9.1 Elements

2.9.1.1 microsoft_color_space_conversion

The **microsoft_color_space_conversion** parameter specifies whether a Wi-Fi Display Sink supports the [high-fidelity color space conversion \(section 2.9\)](#<13>) scheme.[<13>](#<13>)

The **ABNF** syntax is as follows:

```
microsoft-color-space-conversion = "microsoft_color_space_conversion:" SP csc-caps CRLF
csc-caps = "supported" / "none"
```

The **microsoft_color_space_conversion** parameter is included by the Wi-Fi Display Source in the **RTSP** M3 request message ([\[WF-DTS1.1\]](#<14>) section 6.4.3) and by the Wi-Fi Display Sink in the RTSP M3 response message ([\[WF-DTS1.1\]](#<14>) section 6.4.3) to specify support for the color space conversion scheme.

When the Wi-Fi Display Source receives the **microsoft_color_space_conversion** parameter in the M3 response message with the value "supported", the source performs color space conversion for repeat frames. A repeat frame is a **video frame** that is identical to the previous frame, except for small changes such as a blinking mouse cursor.

The Wi-Fi Display Source performs the color space conversion scheme only on video frames that don't include motion (where the image is still with respect to the previous frame).

When generating a full color fidelity frame, the Wi-Fi Display Source keeps the frame counter state variable. The frame counter is an integer between 0 and 3, inclusive, that tracks the current frame in the four-frame sequence used to reproduce a full color frame. The frame counter increments for each repeat frame converted according to the scheme and resets back to 0 when it increments past 3.

When performing color space conversion from a **4:4:4 color space** to a **4:2:0 color space**, the color space conversion component checks the frame counter variable to determine how to down sample the color information from a square of four pixels to one pixel in the output frame. The following table describes how the color space converter copies a pixel based on the current value of the frame counter variable.

Value	Meaning
Frame counter = 0	Specifies that the color space converter copies the top-left pixel to the output pixel.
Frame counter = 1	Specifies that the color space converter copies the top-right pixel to the output pixel.
Frame counter = 2	Specifies that the color space converter copies the bottom-left pixel to the output pixel.
Frame counter = 3	Specifies that the color space converter copies the bottom-right pixel to the output pixel.

When the Wi-Fi Display Source creates the MPEG-2 TS PES packets as specified in [\[ISO/IEC-13818-1\]](#) sections 2.4 and 2.4.3.6, for a video frame, it checks the frame counter. The Wi-Fi Display Source adds "frame counter + 1" stuffing bytes to the PES packet header; For example, 1 stuffing byte would be included for a frame counter of 0, and 4 stuffing bytes would be included for a frame counter of 3.

When the Wi-Fi Display Sink parses the MPEG-2 TS PES packets for a video frame, it checks the number of stuffing bytes in the PES header. If there are no stuffing bytes or more than 4 stuffing bytes, the video frame is not assigned a frame counter value. If there are 1 to 4 stuffing bytes, the video frame is assigned a frame counter of "stuffing bytes – 1"; For example, a frame counter of 0 for 1 stuffing byte and a frame counter of 3 for 4 stuffing bytes.

When the Wi-Fi Display Sink performs color space conversion from the 4:2:0 color space output of the H.264 decoder to a 4:4:4 color space format, it checks the frame counter. When converting from 4:2:0 to 4:4:4, a color space conversion scheme upsamples (adds pixels) the color information from a pixel to a square of four pixels in the output frame. The following table describes how the color space converter copies a pixel based on the current value of the frame counter variable.

Value	Meaning
Frame counter = 0	Specifies that the color space converter creates a new output buffer and copies the input pixel from the source image into all four output pixels.
Frame counter = 1	Specifies that the color space converter copies the input pixel to the top-right output pixel in the output buffer, leaving the other pixels unchanged.
Frame counter = 2	Specifies that the color space converter copies the input pixel to the bottom-left output pixel in the output buffer, leaving the other pixels unchanged.
Frame counter = 3	Specifies that the color space converter copies the input pixel to the bottom-right output pixel in the output buffer, leaving the other pixels unchanged.

After the video frame with a frame counter of 3 is received, the output frame has full color fidelity.

2.9.2 Attributes

Not applicable.

2.9.3 Complex Types

Not applicable.

2.9.4 Simple Types

Not applicable.

2.10 Maximum Supported Bitrate

The maximum supported bitrate conversion extension enables Wi-Fi Display Sinks to specify the maximum supported bitrate, which is the dataflow transmitted over the network in bits per second. [<14>](#) Video encoders are configured to encode at a particular bitrate, with higher bitrates supporting greater data quality but requiring a larger network bandwidth.

2.10.1 Elements

2.10.1.1 microsoft_max_bitrate

The **microsoft_max_bitrate** parameter specifies the maximum video bitrate, which is the bits per second transmitted over the network, supported by a Wi-Fi Display Sink. [<15>](#)

The **ABNF** syntax is as follows:

```
microsoft-max-bitrate = "microsoft_max_bitrate:" SP max-bitrate CRLF
max-bitrate = 1*10DIGIT
```

The **microsoft_max_bitrate** is included by the Wi-Fi Display Source in the **RTSP** M3 request message ([\[WF-DTS1.1\]](#) section 6.4.3), and by the Wi-Fi Display Sink in the RTSP M3 response message ([\[WF-DTS1.1\]](#) section 6.4.3) to specify the maximum supported bitrate.

The **max-bitrate** value specifies the maximum video bitrate supported by the Wi-Fi Display Sink, in terms of bits per second. The Wi-Fi Display Source **MUST** encode the video stream at or below this bitrate.

2.10.2 Attributes

Not applicable.

2.10.3 Complex Types

Not applicable.

2.10.4 Simple Types

Not applicable.

3 Structure Examples

The following is an example of an **M3** request for device metadata (section [2.1](#)).

```
GET_PARAMETER rtsp://localhost/wfd1.0 RTSP/1.0
CSeq: 2
Content-Type: text/parameters
Content-Length: 142

wfd video formats
wfd audio codecs
intel_friendly_name
intel_sink_manufacturer_name
intel_sink_model_name
intel_sink_device_URL
intel_sink_version
```

The following is an example of an **M3** response for device metadata (section 2.1).

```
RTSP/1.0 200 OK
CSeq: 2
Content-Length: 402
Content-Type: text/parameters

wfd_video_formats: 00 00 01 01 00000001 00000000 00000000 00 0000 0000 00 none none
wfd_audio_codecs: LPCM 00000003 00
intel_friendly_name: Contoso Scr 2000
intel_sink_manufacturer_name: Contoso Inc.
intel_sink_model_name: ScreenMaster 2000
intel_sink_device URL: http://www.example.com/screenmaster/
intel_sink_version: product_ID=G4716-2000 hw_version=1.1.5.1345 sw_version=1.2.4.2451
```

The following is an example of an **M3** request for enhanced diagnostics (section [2.2](#)).

```
GET_PARAMETER rtsp://localhost/wfd1.0 RTSP/1.0
CSeq: 2
Content-Type: text/parameters
Content-Length: 69

wfd video formats
wfd_audio_codecs
microsoft_diagnostics_capability
```

The following is an example of an **M3** response for enhanced diagnostics (section 2.2).

```
RTSP/1.0 200 OK
CSeq: 2
Content-Length: 388
Content-Type: text/parameters

wfd_video_formats: 00 00 01 01 00000001 00000000 00000000 00 0000 0000 00 none none
wfd_audio_codecs: LPCM 00000003 00
microsoft_diagnostics_capability: supported
```

The following is an example of a **TEARDOWN** request for enhanced diagnostics.

```
TEARDOWN rtsp://192.168.173.1/wfd1.0/streamid=0 RTSP/1.0
CSeq: 329
```

```
Session: 12345678
Content-Type: text/parameters
Content-Length: 74
```

```
microsoft_teardown_reason: C00D4278 No RTP data was provided for 2 minutes
```

The following is an example of an **M3** request for dynamic resolution and refresh rate (section [2.3](#)).

```
GET_PARAMETER rtsp://localhost/wfd1.0 RTSP/1.0
CSeq: 2
Content-Type: text/parameters
Content-Length: 69

wfd_video_formats
wfd_audio_codecs
microsoft_format_change_capability
```

The following is an example of an **M3** response for dynamic resolution and refresh rate (section 2.3).

```
RTSP/1.0 200 OK
CSeq: 2
Content-Length: 166
Content-Type: text/parameters

wfd_video_formats: 00 00 01 01 00000001 00000000 00000000 00 0000 0000 00 none none
wfd_audio_codecs: LPCM 00000003 00
microsoft_format_change_capability: supported
```

The following is an example of an **M3** request for latency management (section [2.4](#)).

```
GET_PARAMETER rtsp://localhost/wfd1.0 RTSP/1.0
CSeq: 2
Content-Type: text/parameters
Content-Length: 78

wfd_video_formats
wfd_audio_codecs
microsoft_latency_management_capability
```

The following is an example of an **M3** response for latency management (section 2.4).

```
RTSP/1.0 200 OK
CSeq: 2
Content-Length: 173
Content-Type: text/parameters

wfd_video_formats: 00 00 01 01 00000001 00000000 00000000 00 0000 0000 00 none none
wfd_audio_codecs: LPCM 00000003 00
microsoft_latency_management_capability: supported
```

The following is an example of a **SET_PARAMETER** request for latency mode changes.

```
SET_PARAMETER rtsp://localhost/wfd1.0 RTSP/1.0
CSeq: 7
Content-Type: text/parameters
Content-Length: 46

microsoft_latency_management_capability: low
```

The following is an example of an **M2** request and response where the response contains Display Source Identification (section [2.5](#)).

```
OPTIONS * RTSP/1.0
CSeq: 2
Require: org.wfa.wfd1.0

RTSP/1.0 200 OK
CSeq: 2
Date: Sun, Aug 21 2011 04:20:53 GMT
Public: org.wfa.wfd1.0, SETUP, TEARDOWN, PLAY, PAUSE, GET_PARAMETER, SET_PARAMETER
Server: MSMiracastSource/10.00.10011.0000 guid/be113d06-9e40-43e4-98e6-540a325e9ced
```

The following is an example of an **M3** request for device capabilities (section [2.6](#)).

```
GET_PARAMETER rtsp://localhost/wfd1.0 RTSP/1.0
CSeq: 2
Content-Type: text/parameters
Content-Length: 63

wfd_video_formats
wfd_audio_codecs
wfd_idr_request_capability
```

The following is an example of an **M3** response for device capabilities (section 2.6).

```
RTSP/1.0 200 OK
CSeq: 2
Content-Length: 374
Content-Type: text/parameters

wfd video formats: 00 00 01 01 00000001 00000000 00000000 00 0000 0000 00 none none
wfd audio codecs: LPCM 00000003 00
wfd_idr_request_capability: 1
```

The following is an example of an **M3** request for extended video formats using the **wfdx-video-formats** parameter (section [2.7.1.1](#)).

```
GET_PARAMETER rtsp://localhost/wfd1.0 RTSP/1.0
CSeq: 2
Content-Type: text/parameters
Content-Length: 63

wfdx_video_formats
wfd_audio_codecs
wfd_client_rtp_ports
```

The following is an example of an **M3** response for extended video formats using the **wfdx-video-formats** parameter (section 2.7.1.1).

```
RTSP/1.0 200 OK
CSeq: 2
Content-Type: text/parameters
Content-Length: 228

wfd audio codecs: LPCM 00000003 00, AAC 00000001 00, AC3 00000000 00
wfdx_video_formats: 0040 00 0001 0001 0000500001 0010000000 00000000 00 0000 0000 11 none
none
```

```
wfd_client_rtp_ports: RTP/AVP/UDP;unicast 19000 0 mode=play
```

The following is an example of an **M3** request for extended video formats using the **microsoft-video-formats** parameter (section [2.7.1.2](#)).

```
GET_PARAMETER rtsp://localhost/wfd1.0 RTSP/1.0
CSeq: 2
Content-Type: text/parameters
Content-Length: 88

wfd_video_formats
wfd_audio_codecs
wfd_client_rtp_ports
microsoft_video_formats
```

The following is an example of an **M3** response for extended video formats using the **microsoft-video-formats** parameter (section [2.7.1.2](#)).

```
RTSP/1.0 200 OK
CSeq: 2
Content-Type: text/parameters
Content-Length: 259

wfd_video_formats: 00 00 01 01 00000001 00000000 00000000 00 0000 0000 00 none none
wfd_audio_codecs: LPCM 00000003 00, AAC 00000001 00, AC3 00000000 00
wfd_client_rtp_ports: RTP/AVP/UDP;unicast 19000 0 mode=play
microsoft_video_formats: 0000001ffffff
```

The following is an example of an **M3** request for RTCP support (section [2.8](#)).

```
GET_PARAMETER rtsp://localhost/wfd1.0 RTSP/1.0
CSeq: 2
Content-Type: text/parameters
Content-Length: 60

wfd_video_formats
wfd_audio_codecs
microsoft_rtcp_capability
```

The following is an example of an **M3** response for RTCP support (section [2.8](#)).

```
RTSP/1.0 200 OK
Content-Type: text/parameters
Content-Length: 157

wfd_video_formats: 00 00 01 01 00000001 00000000 00000000 00 0000 0000 00 none none
wfd_audio_codecs: LPCM 00000003 00
microsoft_rtcp_capability: supported
```

The following is an example of an **M3** request for high-fidelity color space conversion (section [2.9](#)).

```
GET_PARAMETER rtsp://localhost/wfd1.0 RTSP/1.0
CSeq: 2
Content-Type: text/parameters
Content-Length: 67

wfd_video_formats
wfd_audio_codecs
```

microsoft_color_space_conversion

The following is an example of an **M3** response for high-fidelity color space conversion (section 2.9).

```
RTSP/1.0 200 OK
Content-Type: text/parameters
Content-Length: 164

wfd_video_formats: 00 00 01 01 00000001 00000000 00000000 00 0000 0000 00 none none
wfd_audio_codecs: LPCM 00000003 00
microsoft_color_space_conversion: supported
```

The following is an example of an **M3** request for the maximum supported bitrate (section [2.10](#)).

```
GET PARAMETER rtsp://localhost/wfd1.0 RTSP/1.0
CSeq: 2
Content-Type: text/parameters
Content-Length: 56

wfd video formats
wfd_audio_codecs
microsoft_max_bitrate
```

The following is an example of an **M3** response for the maximum supported bitrate (section 2.10).

```
RTSP/1.0 200 OK
Content-Type: text/parameters
Content-Length: 152

wfd_video_formats: 00 00 01 01 00000001 00000000 00000000 00 0000 0000 00 none none
wfd_audio_codecs: LPCM 00000003 00
microsoft_max_bitrate: 25000000
```

4 Security

4.1 Security Considerations for Implementers

None.

4.2 Index Of Security Fields

None.

5 Appendix A: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include released service packs.

- Windows 10 operating system
- Windows Server 2016 operating system

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms "SHOULD" or "SHOULD NOT" implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term "MAY" implies that the product does not follow the prescription.

[<1> Section 1.3](#): The video formats protocol extension is not supported in Windows 10 v1507 operating system.

[<2> Section 1.3](#): The RTCP protocol extension is not supported in Windows 10 v1507 and Windows 10 v1511 operating system.

[<3> Section 1.3](#): The high-fidelity color space conversion extension is not supported in Windows 10 v1507, Windows 10 v1511, Windows 10 v1607 operating system, and Windows Server 2016.

[<4> Section 1.3](#): The maximum supported bitrate conversion extension is not supported in Windows 10 v1507, Windows 10 v1511, Windows 10 v1607, and Windows Server 2016.

[<5> Section 2.4.1.1](#): On Windows 10 v1507 the latency of the Wi-Fi Display Sink exceeds 100 milliseconds even when the value of the **latency-mode** parameter is "low" or "normal".

[<6> Section 2.4.1.1](#): On Windows 10 v1507 the latency of the Wi-Fi Display Sink exceeds 100 milliseconds even when the value of the **latency-mode** parameter is "low" or "normal".

[<7> Section 2.7](#): The video formats protocol extension is not supported in Windows 10 v1507.

[<8> Section 2.7.1.1](#): The **wfdx-video-formats** parameter is supported only in Windows 10 v1511.

[<9> Section 2.7.1.2](#): The **microsoft-video-formats** parameter is not supported in Windows 10 v1507 and Windows 10 v1511.

[<10> Section 2.8](#): The RTCP protocol extension is not supported in Windows 10 v1507 and Windows 10 v1511.

[<11> Section 2.8.1.1](#): The RTCP protocol extension is not supported in Windows 10 v1507 and Windows 10 v1511.

[<12> Section 2.9](#): The high-fidelity color space conversion extension is not supported in Windows 10 v1507, Windows 10 v1511, Windows 10 v1607, and Windows Server 2016.

[<13> Section 2.9.1.1](#): The high-fidelity color space conversion extension is not supported in Windows 10 v1507, Windows 10 v1511, Windows 10 v1607, and Windows Server 2016.

[<14> Section 2.10](#): The maximum supported bitrate conversion extension is not supported in Windows 10 v1507, Windows 10 v1511, Windows 10 v1607, and Windows Server 2016.

[<15> Section 2.10.1.1](#): The maximum supported bitrate conversion extension is not supported in Windows 10 v1507, Windows 10 v1511, Windows 10 v1607, and Windows Server 2016.

6 Change Tracking

This section identifies changes that were made to this document since the last release. Changes are classified as Major, Minor, or None.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements.
- A document revision that captures changes to protocol functionality.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **None** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the relevant technical content is identical to the last released version.

The changes made to this document are listed in the following table. For more information, please contact dochelp@microsoft.com.

Section	Description	Revision class
2.1.1.3 intel_sink_manufacturer_logo	Updated the ABNF syntax and removed the underscore character.	Minor
2.8.1.1 microsoft_rtcp_capability	Updated the ABNF syntax and removed the underscore character.	Minor
3 Structure Examples	7387 : Updated the intel_friendly_name in the example for M3 response for device metadata.	Minor

7 Index

A

[Applicability](#) 8

C

[Change tracking](#) 33

[Common data types and fields](#) 9

D

[Data types and fields - common](#) 9

Details

[common data types and fields](#) 9

E

[Examples](#) 26

F

[Fields - vendor-extensible](#) 8

G

[Glossary](#) 5

I

[Implementer - security considerations](#) 31

[Informative references](#) 7

[Introduction](#) 5

L

[Localization](#) 8

N

[Normative references](#) 6

O

[Overview \(synopsis\)](#) 7

P

[Product behavior](#) 32

R

[References](#) 6

[informative](#) 7

[normative](#) 6

[Relationship to protocols and other structures](#) 8

S

Security

[implementer considerations](#) 31

Structures

[overview](#) 9

T

[Tracking changes](#) 33

V

[Vendor-extensible fields](#) 8

[Versioning](#) 8