[MS-WFDPE]:

Wi-Fi Display Protocol Extension

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
- **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 may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.
- **No Trade Secrets**. Microsoft does not claim any trade secret rights in this documentation.
- **Patents**. Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft Open Specification Promise or the Community Promise. If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **Trademarks**. The names of companies and products contained in this documentation may 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, e-mail addresses, logos, people, places, and events 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 specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications do 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 are intended for use in conjunction with publicly available standard specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

Revision Summary

Date	Revision History	Revision Class	Comments
6/30/2015	1.0	New	Released new document.

Table of Contents

1 Introduction

The Wi-Fi Display Protocol Extension extends the Wi-Fi Display Technical Specification v1.1 [WF-DTS1.1] 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 and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in [RFC2119]. All other sections and examples in this specification are informative.

1.1 Glossary

The following terms are specific to this document:

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].

context-adaptive binary arithmetic coding (CABAC): A form of entropy encoding used in H264.

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 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 H264 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].

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.

[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/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

[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

[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.ietf.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

[WF-DTS1.1] Wi-Fi Alliance, "Wi-Fi Display Technical Specification v1.1", April 2014, https://www.wi-fi.org/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.

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)
```

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.

Return value/code	Failure condition
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.

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"
```

If the **latency-management-cap** parameter has the value "supported", it means that the Wi-Fi Display Sink is capable of changing the latency mode. A value of "none" means that the Wi-Fi Display Sink is not capable of changing the latency mode.

When the value of the **latency-mode** parameter is "low", the Wi-Fi Display Sink is requested to keep latency under 50 milliseconds.

When the value of the **latency-mode** parameter is "normal", the Wi-Fi Display Sink is requested to keep latency under 100 milliseconds.

When the value of the **latency-mode** parameter is "high", the Wi-Fi Display Sink is requested to 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.

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 ScreenMaster 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
```

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 Technical Preview 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.

6 Change Tracking

No table of changes is available. The document is either new or has had no changes since its last release.

7 Index