[MS-H26XPF]:
Real-Time Transport Protocol (RTP/RTCP): H.261 and H.263 Video Streams Extensions

---

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.
- **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
04/08/2008	0.1		Initial availability
05/16/2008	0.1.1	Editorial	Revised and edited the technical content.
06/20/2008	0.1.2	Editorial	Revised and edited the technical content.
07/25/2008	0.1.3	Editorial	Revised and edited the technical content.
08/29/2008	0.1.4	Editorial	Revised and edited the technical content.
10/24/2008	0.1.5	Editorial	Revised and edited the technical content.
12/05/2008	0.2	Minor	Updated the technical content.
01/16/2009	1.0	Major	Updated and revised the technical content.
02/27/2009	1.0.1	Editorial	Revised and edited the technical content.
04/10/2009	1.0.2	Editorial	Revised and edited the technical content.
05/22/2009	1.0.3	Editorial	Revised and edited the technical content.
07/02/2009	1.0.4	Editorial	Revised and edited the technical content.
08/14/2009	1.0.5	Editorial	Revised and edited the technical content.
09/25/2009	1.1	Minor	Updated the technical content.
11/06/2009	1.1.1	Editorial	Revised and edited the technical content.
12/18/2009	1.1.2	Editorial	Revised and edited the technical content.
01/29/2010	1.1.3	Editorial	Revised and edited the technical content.
03/12/2010	1.1.4	Editorial	Revised and edited the technical content.
04/23/2010	1.1.5	Editorial	Revised and edited the technical content.
06/04/2010	1.1.6	Editorial	Revised and edited the technical content.
07/16/2010	1.1.6	No change	No changes to the meaning, language, or formatting of the technical content.
08/27/2010	1.1.6	No change	No changes to the meaning, language, or formatting of the technical content.
10/08/2010	1.1.6	No change	No changes to the meaning, language, or formatting of the technical content.
11/19/2010	1.1.6	No change	No changes to the meaning, language, or formatting of the technical content.

Date	Revision History	Revision Class	Comments
01/07/2011	1.1.6	No change	No changes to the meaning, language, or formatting of the technical content.
02/11/2011	1.1.6	No change	No changes to the meaning, language, or formatting of the technical content.
03/25/2011	1.1.6	No change	No changes to the meaning, language, or formatting of the technical content.
05/06/2011	1.1.6	No change	No changes to the meaning, language, or formatting of the technical content.
06/17/2011	1.2	Minor	Clarified the meaning of the technical content.

Contents

1 Introduction

This is a specification of the Real-Time Transport Protocol (RTP/RTCP): H.261 and H.263 Video Streams Extensions (H26XPF).

H26XPF is an extension to the RTP payload format for H.261 video streams [RFC2032] and the RTP payload format for H.263 video streams [RFC2190]. It is used to transmit and receive H.261 or H.263 video streams in a two-party peer-to-peer call.

1.1 Glossary

The following terms are defined in [MS-GLOS]:

big-endian

The following terms are specific to this document:

bitstream: The transmission of binary digits as a simple, unstructured sequence of bits.

chrominance: The difference between a color and a chosen reference color of the same luminous intensity in color video streams.

Common Interface Format (CIF): For H.263, a picture consisting of 352x288 pixels for **luminance** and 176x144 pixels for **chrominance**.

draft mode: A mode that is specified by H26XPF video streams extensions for encapsulating H.263 video streams. Draft mode is used in conjunction with the H.323 [H323] application layer control protocol, and it supports an H.263 payload header format that is different from the format in RFC mode.

group of blocks (GOB): For H.263, k*16 lines, where k equals 1 for **QCIF**, and **CIF**.

group of blocks number (GOBN): GOB number in effect at the start of the packet.

interframe: A video frame that is intercoded, also called a P-Frame or P-picture. Refer to [H261] and [H263] for details concerning P-picture.

intraframe: A video frame that is intracoded, also called an I-Frame or I-picture. Refer to [H261] and [H263] for details concerning I-picture.

luminance: The luminous intensity of a surface in a given direction per unit of projected area.

macro block (MB): A macro block consists of four blocks of **luminance** and the spatially corresponding two blocks of **chrominance**. Each block is arranged in an 8x8 pixel configuration.

mode A: The H.263 Mode A payload header, which consists of four bytes, and is present before the actual compression of the H.263 video bitstream in a packet. It allows for fragmentation at **GOB** boundaries.

mode B: The H.263 Mode B payload header, which consists of eight bytes, and starts at the **luminance** boundaries without the **PB-frames** option.

mode C: The H.263 Mode C payload header, which consists of twelve bytes to support fragmentation at **macro block (MB)** boundaries for frames that are coded with the **PB-frames** option.

PB-Frame: A P frame and a B frame, which are coded into one bitstream with macro blocks from the two frames interleaved. In a packet, an **MB** from the P frame and an **MB** from the B frame must be treated together, because each **MB** for the B frame is coded based on the corresponding **MB** for the P frame. A means must be provided to ensure proper rendering of two frames in the right order. Additionally, if any part of this combined bitstream is lost, it will affect both frames, and possibly more.

Quarter Common Interface Format (QCIF): For H.263, a picture consisting of 176x144 pixels for **luminance** and 88x72 pixels for **chrominance**.

quantization: The process of approximating the continuous set of values in the image data with a finite set of values.

RFC Mode: A mode that is specified by H26XPF video streams extensions for encapsulating H.263 video streams. RFC mode is used in conjunction with the Session Initiation Protocol (SIP) [MS-SIP] application layer control protocol, and it supports an H.263 payload header format that is different from the format in **draft mode**.

Sub Quarter Common Interface Format (SQCIF): For H.263, a picture consisting of 128x96 pixels for **luminance** and 64x48 pixels for **chrominance**.

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

1.2 References

References to Microsoft Open Specification documents do not include a publishing year because links are to the latest version of the documents, which are updated frequently. References to other documents include a publishing year when one is available.

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. Please check the archive site, http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624, as an additional source.

[MS-RTPME] Microsoft Corporation, "Real-Time Transport Protocol (RTP/RTCP): Microsoft Extensions".

[RFC2032] Turletti, T., and Huitema, C., "RTP Payload Format for H.261 Video Streams", RFC 2032, Oct. 1996, http://www.ietf.org/rfc/rfc2032.txt

[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

[RFC2190] Zhu, C., "RTP Payload Format for H.263 Video Streams", RFC 2190, September 1997, http://www.ietf.org/rfc/rfc2190.txt

1.2.2 Informative References

[H261] ITU-T, "Video codec for audiovisual services at p x 64 kbit/s", Recommendation H.261, March 1993, http://www.itu.int/rec/T-REC-H.261/en

[H263] ITU-T, "Video coding for low bit rate communication", Recommendation H.263, January 2005, http://www.itu.int/rec/T-REC-H.263/en

[H323] ITU-T, "Packet-based multimedia communications systems", Recommendation H.323, June 2006, http://www.itu.int/rec/T-REC-H.323-200606-S/en

[MS-GLOS] Microsoft Corporation, "Windows Protocols Master Glossary".

[MS-SIP] Microsoft Corporation, "Session Initiation Protocol Extensions".

1.3 Overview

H26XPF specifies the payload format for encapsulating an H.261 [H261] **bitstream** and two payload formats for encapsulating an H.263 [H263] bitstream in the Real-Time Transport Protocol (RTP).

The payload format for H.261 video streams is an extension to the H.261 payload format [RFC2032].

The payload formats for H.263 video streams are an extension to the H.263 payload format [RFC2190]. H26XPF specifies two modes for encapsulating H.263 video streams: **RFC mode** and **draft mode**. RFC mode supports **mode A** and **mode B** of the H.263 video payload header [RFC2190] with some constraints. The payload format for H.263 video streams in draft mode differs from RFC mode in that it supports a different H.263 payload header format.

RFC mode of the H.263 payload format is used in conjunction with the Session Initiation Protocol (SIP) [MS-SIP] application layer control protocol. Draft mode of the H.263 payload format is used in conjunction with the H.323 [H323] application layer control protocol.

1.4 Relationship to Other Protocols

H26XPF extends the base protocol for the H.261 payload format [RFC2032] and the base protocol for the H.263 payload format [RFC2190]. It carries a payload consisting of an H.261 bitstream or an H.263 bitstream in the formats specified in [H261] or [H263] and, in turn, it is carried as a payload of the RTP extensions specified in [MS-RTPME].

1.5 Prerequisites/Preconditions

H26XPF specifies only the payload formats for H.261 or H.263 video streams. It requires the establishment of an RTP stream, a mechanism for obtaining H.261 or H.263 video frames for it to convert to packets, and a mechanism for rendering H.261 or H.263 video frames that are converted to packets.

H26XPF requires an upper layer to select only one of the three payload formats explicitly.

1.6 Applicability Statement

H26XPF can only be used to transform H.261 or H.263 video frames into packets.

1.7 Versioning and Capability Negotiation

H26XPF has no versioning or capability negotiation constraints beyond those specified in [RFC2032] and [RFC2190].

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

H26XPF has no standards assignments beyond those specified in [RFC2032] and [RFC2190].

2 Messages

2.1 Transport

H26XPF is carried as a payload in RTP [MS-RTPME] and therefore relies on RTP for providing the means to transport its payload over the network.

2.2 Message Syntax

2.2.1 H.261 Payload Header

The H.261 payload header is specified in [RFC2032] section 4.1.

2.2.2 H.263 Payload Header, RFC Mode

The H.263 payload header including mode A, mode B, and **mode C** are specified in [RFC2190] section 5. H26XPF imposes the following constraints on values in the H.263 payload header in RFC mode:

The **TR** field MUST be ignored.

The **SRC** field MUST be 1 (**Sub Quarter Common Interface Format (SQCIF)**), 2 (**Quarter Common Interface Format (QCIF)**), or 3 (**Common Interface Format (CIF)**).

The **U** field MUST be 0.

The **S** field MUST be 0.

The **A** field MUST be 0.

In addition, the **I** field has a different meaning than that specified in [RFC2190]. The value 0 MUST be used for an **interframe**. The value 1 MUST be used for an **intraframe**.

H26XPF does not support **PB-frames**. As a result, the value of the **P** field in the payload MUST be 0. The sender MUST NOT send the mode C payload header or the mode A payload header with the P field set to 1.

2.2.3 H.263 Payload Header, Draft Mode

The fields defined in the H.263 payload header in draft mode differ from the payload header in RFC mode in the following ways:

- The orders of the following fields are rearranged: I, A, S, R, HMV1, VMV1, HMV2, and VMV2.
- The sizes of the following fields are different: MBA, HMV1, VMV1, HMV2, VMV2, and R.
- The H.263 payload header in draft mode does not specify a U field.

Details of these differences are specified in the following sections.

2.2.3.1 Mode A

The H.263 mode A payload header, which consists of 4 bytes, and is present before the actual compression of the H.263 video bitstream in a packet. It allows for fragmentation at **group of blocks (GOB)** boundaries.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
F	P	SBIT			EBIT			SRC			R					I	A	S	DBQ		TRB			TR							

F (1 bit): A flag that indicates the optional PB-frames mode, as defined by H.263. For a mode A packet this value MUST be zero.

P (1 bit): A flag that indicates the optional PB-frames mode, as defined by H.263. This value MUST be zero.

SBIT (3 bits): The start bit position, which specifies the number of bits to be ignored in the first data byte, starting with the most significant.

EBIT (3 bits): The end bit position, which specifies the number of bits to be ignored in the last data byte, starting with the least significant.

SRC (3 bits): The source format specifies the resolution of the current picture.

Value	Meaning
1	SQCIF
2	QCIF
3	CIF

R (5 bits): This value MUST be zero.

I (1 bit): Picture coding type.

Value	Meaning
0	Intercoded.
1	Intracoded.

A (1 bit): This value MUST be zero.

S (1 bit): This value MUST be zero.

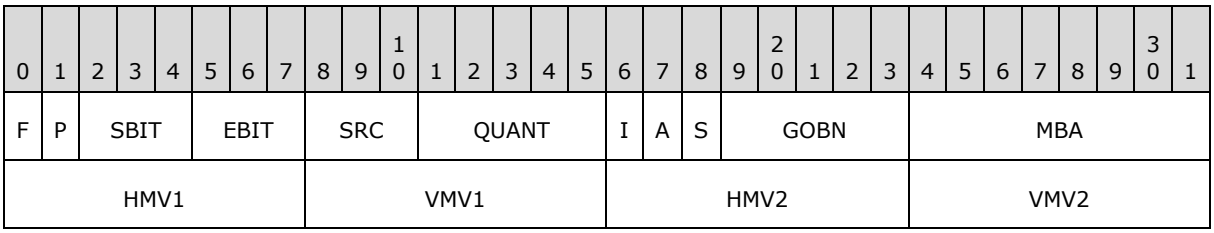
DBQ (2 bits): Differential **quantization** parameter used to calculate the quantizer for the B frame based on the quantizer for the P frame, when PB-Frames option is used. The PB-Frames option is not supported in H26XPF. This value MUST be zero.

TRB (3 bits): Temporal Reference for the B frame as defined by [H263]. The PB-Frames option is not supported in H26XPF. This value MUST be zero.

TR (1 byte): Temporal Reference for the P frame as defined by [H263]. The PB-Frames option is not supported in H26XPF. This value MUST be ignored.

2.2.3.2 Mode B

The H.263 mode B payload header, which consists of 8 bytes and starts at the **luminance** boundaries without the PB-frames option.



F (1 bit): The mode of the payload header. This value MUST be one.

P (1 bit): Optional PB-frames mode as defined by the H.263. This value MUST be zero.

SBIT (3 bits): The start bit position, which specifies the number of bits to be ignored in the first data byte, starting with the most significant.

EBIT (3 bits): The end bit position, which specifies the number of bits to be ignored in the last data byte, starting with the least significant.

SRC (3 bits): The source format specifies the resolution of the current picture.

Value	Meaning
1	SQCIF
2	QCIF
3	CIF

QUANT (5 bits): Quantization value for the first **macro block (MB)** coded at the start of the packet. Set to zero if the packet begins with a GOB header.

I (1 bit): Picture coding type.

Value	Meaning
0	Intercoded.
1	Intracoded.

A (1 bit): This value MUST be zero.

S (1 bit): This value MUST be zero.

GOBN (5 bits): The **group of blocks number (GOBN)** value is defined by the inherent resolution.

MBA (1 byte): The address within the GOB of the first MB in the packet, counting form zero in scan order.

HMV1 (1 byte): Horizontal vector predictor for the first MB in this packet.

VMV1 (1 byte): Vertical vector predictor for the first MB in this packet.

HMV2 (1 byte): Horizontal motion vector predictors for the third MB in this packet.

VMV2 (1 byte): Vertical motion vector predictors for the third MB in this packet.

3 Protocol Details

3.1 Client and Server Role Details

H26XPF does not have any role-specific behavior. The behavior specified in this section applies to both client and server roles.

3.1.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with those described.

3.1.1.1 H.261 Payload Format

An H.261 video frame is fragmented and converted to packets using the mechanism specified in [RFC2032] and in this document.

An H.261 video frame is constructed by concatenating H.261 video payload data of all RTP packets for the video frame.

H.261 video packets are considered to be complete if they satisfy the following condition:

- An H.261 I-Frame SHOULD be sent every 15 seconds.

Note The above conceptual data can be implemented using a variety of techniques. An implementation is at liberty to implement such data in any way it pleases.

3.1.1.2 H.263 Payload Formats

The description in this section applies to the H.263 payload headers in both RFC mode and draft mode.

An H.263 video frame is fragmented and converted to packets using the mechanism specified in [RFC2190] and in this document.

An H.263 video frame is constructed by concatenating all H.263 video payload data of all RTP packets for the video frame.

H.263 video packets are considered to be complete if they satisfy the following condition:

- An H.263 I-Frame SHOULD be sent every 15 seconds.

Note The above conceptual data can be implemented using a variety of techniques. An implementation is at liberty to implement such data in any way it pleases.

3.1.2 Timers

H26XPF has no additional timers beyond those specified in [RFC2032] and [RFC2190].

3.1.3 Initialization

H26XPF has no additional initialization requirements beyond those specified in [RFC2032] and [RFC2190].

3.1.4 Higher-Layer Triggered Events

H26XPF has no additional higher-layer triggered events beyond those specified in [RFC2032] and [RFC2190].

3.1.5 Message Processing Events and Sequencing Rules

H26XPF has no additional message processing events or sequencing rules beyond those specified in [RFC2032] and [RFC2190].

3.1.6 Timer Events

H26XPF has no additional timer events beyond those specified in [RFC2032] and [RFC2190].

3.1.7 Other Local Events

H26XPF has no additional local events beyond those specified in [RFC2032] and [RFC2190].

4 Protocol Examples

The field names in the payload headers used in this section are defined in [RFC2032], [RFC2190], or in this document.

4.1 H.261 Payload Header, Intraframe

Consider the following payload header in **big-endian** byte order.

```
0x9B, 0x00, 0x00, 0x00
```

This corresponds to payload header fields with the following values.

Field	Value
SBIT	4
EBIT	6
I	1 (intraframe)
V	1
GOBN	0
MBAP	0
QUANT	0
HMVD	0
VMVD	0

4.2 H.263 Payload Header in Draft Mode, Mode B, Interframe

Consider the following payload header in big-endian byte order.

```
0x9C, 0x66, 0x80, 0x06, 0x00, 0x00, 0x00, 0x00
```

This corresponds to payload header fields with the following values.

Field	Value
F	1
P	0
SBIT	3
EBIT	4
SRC	3 (CIF)
QUANT	6

Field	Value
I	0 (interframe)
A	0
S	0
GOBN	0
MBA	6
HMV1	0
VMV1	0
HMV2	0
VMV2	0

4.3 H.261 Payload Header, Interframe

Consider the following payload header in big-endian byte order.

```
0xB1, 0x00, 0x00, 0x00
```

This corresponds to payload header fields with the following values.

Field	Value
SBIT	5
EBIT	4
I	0 (interframe)
V	1
GOBN	0
MBAP	0
QUANT	0
HMVD	0
VMVD	0

4.4 H.263 Payload Header in RFC Mode, Mode A, Intraframe

Consider the following payload header in big-endian byte order.

```
0x05, 0x70, 0x00, 0x01
```

This corresponds to payload header fields with the following values.

Field	Value
F	0
P	0
SBIT	0
EBIT	5
SRC	3 (CIF)
I	1 (intraframe)
U	0
S	0
A	0
R	0
DBQ	0
TRB	0
TR	1

4.5 H.263 Payload Header in RFC Mode, Mode A, Interframe

Consider the following payload header in big-endian byte order.

```
0x02, 0x60, 0x00, 0x02
```

This corresponds to payload header fields with the following values.

Field	Value
F	0
P	0
SBIT	0
EBIT	2
SRC	3 (CIF)
I	0 (interframe)
U	0
S	0
A	0
R	0

Field	Value
DBQ	0
TRB	0
TR	2

4.6 H.263 Payload Header in RFC Mode, Mode B, Intraframe

Consider the following payload header in big-endian byte order.

```
0xBD, 0x67, 0x00, 0x14, 0x80, 0x00, 0x00, 0x00
```

This corresponds to payload header fields with the following values.

Field	Value
F	1
P	0
SBIT	7
EBIT	5
SRC	3 (CIF)
QUANT	7
GOBN	0
MBA	5
R	0
I	1 (intraframe)
U	0
S	0
A	0
HMV1	0
VMV1	0
HMV2	0
VMV2	0

4.7 H.263 Payload Header in RFC Mode, Mode B, Interframe

Consider the following payload header in big-endian byte order.

```
0xA1, 0x67, 0x00, 0x18, 0x0F, 0x00, 0x80, 0x00
```

This corresponds to payload header fields with the following values.

Field	Value
F	1
P	0
SBIT	4
EBIT	1
SRC	3 (CIF)
QUANT	7
GOBN	0
MBA	6
R	0
I	0 (interframe)
U	0
S	0
A	0
HMV1	120
VMV1	2
HMV2	0
VMV2	0

4.8 H.263 Payload Header in Draft Mode, Mode A, Intraframe

Consider the following payload header in big-endian byte order.

```
0x00, 0x40, 0x80, 0x00
```

This corresponds to payload header fields with the following values.

Field	Value
F	0
P	0
SBIT	0

Field	Value
EBIT	0
SRC	2 (QCIF)
R	0
I	1 (intraframe)
A	0
S	0
DBQ	0
TRB	0
TR	0

4.9 H.263 Payload Header in Draft Mode, Mode A, Interframe

Consider the following payload header in big-endian byte order.

```
0x00, 0x40, 0x00, 0x05
```

This corresponds to payload header fields with the following values.

Field	Value
F	0
P	0
SBIT	0
EBIT	0
SRC	2 (QCIF)
R	0
I	0 (interframe)
A	0
S	0
DBQ	0
TRB	0
TR	5

4.10 H.263 Payload Header in Draft Mode, Mode B, Intraframe

Consider the following payload header in big-endian byte order.

```
0xBD, 0x67, 0x80, 0x05, 0x00, 0x00, 0x00, 0x00
```

This corresponds to payload header fields with the following values.

Field	Value
F	1
P	0
SBIT	7
EBIT	5
SRC	3 (CIF)
QUANT	7
I	1 (intraframe)
A	0
S	0
GOBN	0
MBA	5
HMV1	0
VMV1	0
HMV2	0
VMV2	0

5 Security

5.1 Security Considerations for Implementers

H26XPF has no additional security considerations beyond those specified in [RFC2032] and [RFC2190].

5.2 Index of Security Parameters

None.

6 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:

- Microsoft Windows® 2000 operating system
- Windows® XP operating system
- Windows Server® 2003 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.

7 Change Tracking

This section identifies changes that were made to the [MS-H26XPF] protocol document between the May 2011 and June 2011 releases. Changes are classified as New, Major, Minor, Editorial, or No change.

The revision class **New** means that a new document is being released.

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 or functionality.
- An extensive rewrite, addition, or deletion of major portions of content.
- The removal of a document from the documentation set.
- Changes made for template compliance.

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 **Editorial** means that the language and formatting in the technical content was changed. Editorial changes apply to grammatical, formatting, and style issues.

The revision class **No change** means that no new technical or language changes were introduced. The technical content of the document is identical to the last released version, but minor editorial and formatting changes, as well as updates to the header and footer information, and to the revision summary, may have been made.

Major and minor changes can be described further using the following change types:

- New content added.
- Content updated.
- Content removed.
- New product behavior note added.
- Product behavior note updated.
- Product behavior note removed.
- New protocol syntax added.
- Protocol syntax updated.
- Protocol syntax removed.
- New content added due to protocol revision.
- Content updated due to protocol revision.
- Content removed due to protocol revision.
- New protocol syntax added due to protocol revision.

- Protocol syntax updated due to protocol revision.
- Protocol syntax removed due to protocol revision.
- New content added for template compliance.
- Content updated for template compliance.
- Content removed for template compliance.
- Obsolete document removed.

Editorial changes are always classified with the change type **Editorially updated.**

Some important terms used in the change type descriptions are defined as follows:

- **Protocol syntax** refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.
- **Protocol revision** refers to changes made to a protocol that affect the bits that are sent over the wire.

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

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
1.1 Glossary	Removed the term "M bit" from the list of document-specific terms.	N	Content updated.
1.2 References	Added explanatory statement regarding the removal of the publishing year from Microsoft Open Specification document references.	N	Content updated.

8 Index