

## [MS-RTPRADEX]:

# RTP Payload for Redundant Audio Data Extensions

---

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](mailto: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](http://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](mailto:dochelp@microsoft.com).

## Revision Summary

Date	Revision History	Revision Class	Comments
4/4/2008	0.1	New	Initial version
4/25/2008	0.2	Minor	Updated based on feedback
6/27/2008	1.0	Major	Updated based on feedback
8/15/2008	1.01	Minor	Updated based on feedback
12/12/2008	2.0	Major	Updated with latest template bug fixes (redlined)
2/13/2009	2.01	Minor	Updated with latest template bug fixes (redlined)
3/13/2009	2.02	Minor	Updated with latest template bug fixes (redlined)
7/13/2009	2.03	Major	Revised and edited the technical content
8/28/2009	2.04	Editorial	Revised and edited the technical content
11/6/2009	2.05	Editorial	Revised and edited the technical content
2/19/2010	2.06	Editorial	Revised and edited the technical content
3/31/2010	2.07	Major	Updated and revised the technical content
4/30/2010	2.08	Editorial	Revised and edited the technical content
6/7/2010	2.09	Editorial	Revised and edited the technical content
6/29/2010	2.10	Editorial	Changed language and formatting in the technical content.
7/23/2010	2.10	None	No changes to the meaning, language, or formatting of the technical content.
9/27/2010	3.0	Major	Significantly changed the technical content.
11/15/2010	3.0	None	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	3.0	None	No changes to the meaning, language, or formatting of the technical content.
3/18/2011	3.0	None	No changes to the meaning, language, or formatting of the technical content.
6/10/2011	3.0	None	No changes to the meaning, language, or formatting of the technical content.
1/20/2012	3.1	Minor	Clarified the meaning of the technical content.
4/11/2012	3.1	None	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	3.1	None	No changes to the meaning, language, or formatting of the technical content.
10/8/2012	3.1.1	Editorial	Changed language and formatting in the technical content.
2/11/2013	3.1.1	None	No changes to the meaning, language, or formatting of the technical content.

<b>Date</b>	<b>Revision History</b>	<b>Revision Class</b>	<b>Comments</b>
7/30/2013	3.1.1	None	No changes to the meaning, language, or formatting of the technical content.
11/18/2013	3.1.1	None	No changes to the meaning, language, or formatting of the technical content.
2/10/2014	3.1.1	None	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	3.1.1	None	No changes to the meaning, language, or formatting of the technical content.
7/31/2014	3.2	Minor	Clarified the meaning of the technical content.
10/30/2014	3.2	None	No changes to the meaning, language, or formatting of the technical content.
3/30/2015	4.0	Major	Significantly changed the technical content.
6/30/2015	4.0	None	No changes to the meaning, language, or formatting of the technical content.
9/4/2015	4.0	None	No changes to the meaning, language, or formatting of the technical content.
11/19/2015	5.0	Major	Significantly changed the technical content.
7/15/2016	5.0	None	No changes to the meaning, language, or formatting of the technical content.
9/14/2016	5.0	None	No changes to the meaning, language, or formatting of the technical content.
9/15/2017	6.0	Major	Significantly changed the technical content.
4/27/2018	7.0	Major	Significantly changed the technical content.
7/24/2018	8.0	Major	Significantly changed the technical content.
8/28/2018	9.0	Major	Significantly changed the technical content.

# Table of Contents

<b>1</b>	<b>Introduction</b>	<b>5</b>
1.1	Glossary	5
1.2	References	5
1.2.1	Normative References	6
1.2.2	Informative References	6
1.3	Overview	6
1.4	Relationship to Other Protocols	6
1.5	Prerequisites/Preconditions	6
1.6	Applicability Statement	6
1.7	Versioning and Capability Negotiation	6
1.8	Vendor-Extensible Fields	7
1.9	Standards Assignments	7
<b>2</b>	<b>Messages</b>	<b>8</b>
2.1	Transport	8
2.2	Message Syntax	8
2.2.1	Redundant Block	8
<b>3</b>	<b>Protocol Details</b>	<b>9</b>
3.1	Receiver Details	9
3.1.1	Abstract Data Model	9
3.1.2	Timers	9
3.1.3	Initialization	9
3.1.4	Higher-Layer Triggered Events	9
3.1.5	Message Processing Events and Sequencing Rules	9
3.1.6	Timer Events	9
3.1.7	Other Local Events	9
3.2	Sender Details	9
3.2.1	Abstract Data Model	10
3.2.2	Timers	10
3.2.3	Initialization	10
3.2.4	Higher-Layer Triggered Events	10
3.2.5	Message Processing Events and Sequencing Rules	10
3.2.6	Timer Events	10
3.2.7	Other Local Events	10
<b>4</b>	<b>Protocol Examples</b>	<b>11</b>
<b>5</b>	<b>Security</b>	<b>12</b>
5.1	Security Considerations for Implementers	12
5.2	Index of Security Parameters	12
<b>6</b>	<b>Appendix A: Product Behavior</b>	<b>13</b>
<b>7</b>	<b>Change Tracking</b>	<b>14</b>
<b>8</b>	<b>Index</b>	<b>15</b>

# 1 Introduction

The RTP Payload for Redundant Audio Data Extensions Protocol is a set of extensions for encoding redundant audio data for use with the Real-Time Transport Protocol (RTP) Extensions Protocol, as described in [\[MS-RTP\]](#). This protocol is a proprietary extension of RTP Payload for Redundant Audio Data, as described in [\[RFC2198\]](#). [RFC2198] describes a payload format for use with the Real-Time Transport Protocol (RTP).

Sections 1.5, 1.8, 1.9, 2, and 3 of this specification are normative. All other sections and examples in this specification are informative.

## 1.1 Glossary

This document uses the following terms:

**codec:** An algorithm that is used to convert media between digital formats, especially between raw media data and a format that is more suitable for a specific purpose. Encoding converts the raw data to a digital format. Decoding reverses the process.

**dual-tone multi-frequency (DTMF):** In telephony systems, a signaling system in which each digit is associated with two specific frequencies. This system typically is associated with touch-tone keypads for telephones.

**lossy network transports:** A transport that cannot deliver a data payload reliably from a source to a destination.

**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 [\[RFC3550\]](#).

**RTP packet:** A data packet consisting of the fixed RTP header, a possibly empty list of contributing sources, and the payload data. Some underlying protocols may require an encapsulation of the RTP packet to be defined. Typically one packet of the underlying protocol contains a single RTP packet, but several RTP packets can be contained if permitted by the encapsulation method. See [\[RFC3550\]](#) section 3.

**RTP payload:** The data transported by **RTP** in a packet, for example audio samples or compressed video data. For more information, see [\[RFC3550\]](#) section 3.

**RTP session:** An association among a set of participants who are communicating by using the **Real-Time Transport Protocol (RTP)**, as described in [\[RFC3550\]](#). Each RTP session maintains a full, separate space of Synchronization Source (SSRC) identifiers.

**Session Description Protocol (SDP):** A protocol that is used for session announcement, session invitation, and other forms of multimedia session initiation. For more information see [\[MS-SDP\]](#) and [\[RFC3264\]](#).

**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](mailto:dochelp@microsoft.com). We will assist you in finding the relevant information.

[MS-SDPEXT] Microsoft Corporation, "[Session Description Protocol \(SDP\) Version 2.0 Extensions](#)".

[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>

[RFC2198] Perkins, C., "RTP Payload for Redundant Audio Data", RFC 2198, September 1997, <http://www.rfc-editor.org/rfc/rfc2198.txt>

### 1.2.2 Informative References

[MS-RTP] Microsoft Corporation, "[Real-time Transport Protocol \(RTP\) Extensions](#)".

## 1.3 Overview

This protocol extends the **Real-Time Transport Protocol (RTP)** Payload for Redundant Audio Data protocol, as described in [\[RFC2198\]](#), by restricting an RTP audio payload to one block of redundant audio data. The redundant block of audio data is implemented in the **RTP payload** along with the primary block of audio data.

## 1.4 Relationship to Other Protocols

This protocol relies on the Real-Time Transport Protocol (RTP) Extension protocol, as described in [\[MS-RTP\]](#), as its transport.

This document only addresses the redundancy and thereby loss and error tolerance of audio data streams. Non-audio data redundancy is beyond the scope of this document.

## 1.5 Prerequisites/Preconditions

Because the Real-Time Transport Protocol (RTP) Extensions Protocol acts as a transport for this protocol, a valid **RTP session** is required to be established. Refer to [\[MS-RTP\]](#) for details.

It is further assumed that a valid **Session Description Protocol (SDP)** negotiation has been completed to bind the dynamic payload information for the redundancy data. For information about SDP, see [\[MS-SDPEXT\]](#).

## 1.6 Applicability Statement

This protocol is applicable for a real-time audio communication scenario where redundant data exchange is needed to mitigate **lossy network transports**.

This protocol does not cover all audio data redundancy. It is limited to in-band audio communication data. This protocol does not apply to redundancy for audio data such as out-of-band **dual-tone multi-frequency (DTMF)** tones. Out-of-band DTMF tones are defined as exchange of DTMF information in a separate band from the media stream.

## 1.7 Versioning and Capability Negotiation

**Supported Transports:** This protocol is implemented on top of the Real-Time Transport Protocol (RTP) Extension protocol as the transport mechanism.

**Protocol Versions:** This protocol, as a payload format of **RTP**, does not provide for versioning information within the scope of the protocol itself. However, as a part of the **RTP payload**, any versioning information on the RTP level applies.

**Security and Authentication Methods:** This document does not describe any security or authentication methods. Security and authentication is dependent on the security method, authentication method, or both methods used by the Real-Time Transport Protocol (RTP) Extensions protocol.

**Localization:** None.

## **1.8 Vendor-Extensible Fields**

None.

## **1.9 Standards Assignments**

None.

## 2 Messages

### 2.1 Transport

Because this protocol uses the Real-Time Transport Protocol (RTP) Extensions protocol as its transport, a successful **RTP session** MUST be established with valid redundancy payload information negotiated.

This MUST be done with the Session Description Protocol, as specified in [\[MS-SDPEXT\]](#).

### 2.2 Message Syntax

The structure and syntax of this protocol is defined within the RFC for RTP Payload for Redundant Audio Data, as specified in [\[RFC2198\]](#) section 3. This protocol does not cover all audio data redundancy. It is limited to in-band audio communication data. This protocol MUST NOT be used to carry audio data redundancy for audio data such as out-of-band **DTMF** tones.

The deviation from [\[RFC2198\]](#) is as follows:

[\[RFC2198\]](#) section 2 provides for one or more redundant audio blocks for each **RTP payload**. This protocol description allows for only one redundant block for every RTP payload. Therefore, each RTP payload MUST NOT contain more than two blocks total: one redundancy block and one primary block.

[\[RFC2198\]](#) section 2 describes the mechanism for including the redundancy information in the **RTP packet** header. This protocol does not support redundant information in the **RTP** header. The RTP header MUST NOT contain redundant information. It MUST be made part of a dynamic RTP payload type and negotiate as such during **SDP** negotiation.

While [\[RFC2198\]](#) section 2 allows for static typing of payload types, systems interoperating with implementation of this protocol MUST negotiate for dynamic redundancy payload type using SDP to enable redundancy as specified in [\[MS-SDPEXT\]](#) section [3.1.5.3](#).

#### 2.2.1 Redundant Block

See [\[RFC2198\]](#) section 3 for a detailed description of the redundant block layout.

## 3 Protocol Details

### 3.1 Receiver Details

This protocol can be described using a Sender and Receiver model. This section details the behavioral difference between the protocol specified by [\[RFC2198\]](#) and this protocol implementation.

The Receiver side of this protocol MUST negotiate using **SDP** for a dynamic payload type binding for the redundancy data. The payload type binding MUST be symmetrical. This means the receive payload type and send payload type MUST be the same. Asymmetrical payload type information MUST NOT be used.

#### 3.1.1 Abstract Data Model

None.

#### 3.1.2 Timers

None.

#### 3.1.3 Initialization

Receivers MUST negotiate a dynamic payload type for the redundancy data as specified in [\[MS-SDPEXT\]](#) section 3.1.5.3. Receivers MUST NOT expect redundancy data to be part of the **RTP** extended header structure.

#### 3.1.4 Higher-Layer Triggered Events

None.

#### 3.1.5 Message Processing Events and Sequencing Rules

None.

#### 3.1.6 Timer Events

None.

#### 3.1.7 Other Local Events

None.

### 3.2 Sender Details

This protocol can be described using a Sender and Receiver model.

This section details the behavioral difference between the protocol specified by [\[RFC2198\]](#) and this protocol implementation.

The Sender side of this protocol MUST negotiate using **SDP** for a dynamic payload type binding for the redundancy data.

Distance is defined as the number of **RTP packets** succeeding the primary block for which the redundancy block applies. For example, if RTP packet X contains primary block A, and RTP packet X +

n contains the redundancy block for primary block A, that redundancy block has a distance of n. The redundancy data block MUST NOT have a distance greater than 3.

There MUST NOT be more than one redundancy block per RTP packet. At most two blocks are allowed per RTP packet: one primary block and one redundancy block.

All redundant audio data from the Sender MUST be the same encoding, or **codec**, as the primary audio block. This requirement deviates from [RFC2198] where secondary, tertiary, and other codecs are supported.

The primary audio block and redundant audio block MUST use the same codec.

### **3.2.1 Abstract Data Model**

None

### **3.2.2 Timers**

None.

### **3.2.3 Initialization**

The Sender MUST negotiate a dynamic payload type for the redundancy data.

### **3.2.4 Higher-Layer Triggered Events**

None.

### **3.2.5 Message Processing Events and Sequencing Rules**

None.

### **3.2.6 Timer Events**

None.

### **3.2.7 Other Local Events**

None.

## 4 Protocol Examples

Refer to [\[RFC2198\]](#) section 7 for examples of this protocol structure.

## **5 Security**

### **5.1 Security Considerations for Implementers**

There are no additional protocol security considerations beyond what is described in [\[RFC2198\]](#).

### **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 updates to those products.

- Microsoft Office Communications Server 2007
- Microsoft Office Communicator 2007
- Microsoft Office Communications Server 2007 R2
- Microsoft Office Communicator 2007 R2
- Microsoft Lync Server 2010
- Microsoft Lync 2010
- Microsoft Lync Server 2013
- Microsoft Lync Client 2013/Skype for Business
- Microsoft Skype for Business 2016
- Microsoft Skype for Business Server 2015
- Windows 10 v1511 operating system
- Windows Server 2016 operating system
- Windows Server operating system
- Microsoft Skype for Business 2019
- Microsoft Skype for Business Server 2019

Exceptions, if any, are noted in this section. If an update version, service pack or Knowledge Base (KB) number appears with a product name, the behavior changed in that update. The new behavior also applies to subsequent updates 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 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](mailto:dochelp@microsoft.com).

Section	Description	Revision class
<a href="#">6</a> Appendix A: Product Behavior	Updated list of supported products.	major

## 8 Index

### A

Abstract data model  
[receiver](#) 9  
[sender](#) 10  
[Applicability](#) 6

### C

[Capability negotiation](#) 6  
[Change tracking](#) 14

### D

Data model - abstract  
[receiver](#) 9  
[sender](#) 10

### E

[Examples](#) 11

### F

[Fields - vendor-extensible](#) 7

### G

[Glossary](#) 5

### H

Higher-layer triggered events  
[receiver](#) 9  
[sender](#) 10

### I

[Implementer - security considerations](#) 12  
[Index of security parameters](#) 12  
[Informative references](#) 6  
Initialization  
[receiver](#) 9  
[sender](#) 10  
[Introduction](#) 5

### L

Local events  
[receiver](#) 9  
[sender](#) 10

### M

Message processing  
[receiver](#) 9  
[sender](#) 10  
[Message syntax](#) 8  
Messages  
[Redundant Block](#) 8

[syntax](#) 8  
[transport](#) 8

### N

[Normative references](#) 6

### O

[Overview \(synopsis\)](#) 6

### P

[Parameters - security index](#) 12  
[Preconditions](#) 6  
[Prerequisites](#) 6  
[Product behavior](#) 13

### R

Receiver  
[abstract data model](#) 9  
[higher-layer triggered events](#) 9  
[initialization](#) 9  
[local events](#) 9  
[message processing](#) 9  
[overview](#) 9  
[sequencing rules](#) 9  
[timer events](#) 9  
[timers](#) 9  
[Redundant Block message](#) 8  
[References](#) 5  
[informative](#) 6  
[normative](#) 6  
[Relationship to other protocols](#) 6

### S

Security  
[implementer considerations](#) 12  
[parameter index](#) 12  
Sender  
[abstract data model](#) 10  
[higher-layer triggered events](#) 10  
[initialization](#) 10  
[local events](#) 10  
[message processing](#) 10  
[overview](#) 9  
[sequencing rules](#) 10  
[timer events](#) 10  
[timers](#) 10  
Sequencing rules  
[receiver](#) 9  
[sender](#) 10  
[Standards assignments](#) 7

### T

Timer events  
[receiver](#) 9  
[sender](#) 10

Timers

[receiver](#) 9

[sender](#) 10

[Tracking changes](#) 14

[Transport](#) 8

Triggered events

[receiver](#) 9

[sender](#) 10

**V**

[Vendor-extensible fields](#) 7

[Versioning](#) 6