

[MS-RTPRAD]: Real-Time Transport Protocol (RTP/RTCP): Redundant Audio Data 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. 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 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	1.0	Major	Updated and revised the technical content.
07/25/2008	1.0.1	Editorial	Revised and edited the technical content.
08/29/2008	1.0.2	Editorial	Revised and edited the technical content.
10/24/2008	1.0.3	Editorial	Revised and edited the technical content.
12/05/2008	1.0.4	Editorial	Revised and edited the technical content.
01/16/2009	1.1	Minor	Updated the technical content.
02/27/2009	1.2	Minor	Updated the technical content.
04/10/2009	1.2.1	Editorial	Revised and edited the technical content.
05/22/2009	1.2.2	Editorial	Revised and edited the technical content.
07/02/2009	1.2.3	Editorial	Revised and edited the technical content.
08/14/2009	1.2.4	Editorial	Revised and edited the technical content.
09/25/2009	1.2.5	Editorial	Revised and edited the technical content.
11/06/2009	1.2.6	Editorial	Revised and edited the technical content.
12/18/2009	1.2.7	Editorial	Revised and edited the technical content.
01/29/2010	1.2.8	Editorial	Revised and edited the technical content.
03/12/2010	1.2.9	Editorial	Revised and edited the technical content.
04/23/2010	1.2.10	Editorial	Revised and edited the technical content.
06/04/2010	1.2.11	Editorial	Revised and edited the technical content.
07/16/2010	1.2.11	No change	No changes to the meaning, language, or formatting of the technical content.
08/27/2010	1.2.11	No change	No changes to the meaning, language, or formatting of the technical content.
10/08/2010	1.2.11	No change	No changes to the meaning, language, or formatting of the technical content.
11/19/2010	1.2.11	No change	No changes to the meaning, language, or formatting of the technical content.

Date	Revision History	Revision Class	Comments
01/07/2011	1.2.11	No change	No changes to the meaning, language, or formatting of the technical content.
02/11/2011	1.2.11	No change	No changes to the meaning, language, or formatting of the technical content.
03/25/2011	1.2.11	No change	No changes to the meaning, language, or formatting of the technical content.
05/06/2011	1.2.11	No change	No changes to the meaning, language, or formatting of the technical content.
06/17/2011	1.3	Minor	Clarified the meaning of the technical content.
09/23/2011	1.3	No change	No changes to the meaning, language, or formatting of the technical content.
12/16/2011	1.3	No change	No changes to the meaning, language, or formatting of the technical content.
03/30/2012	1.3	No change	No changes to the meaning, language, or formatting of the technical content.
07/12/2012	1.3	No change	No changes to the meaning, language, or formatting of the technical content.
10/25/2012	1.3	No change	No changes to the meaning, language, or formatting of the technical content.
01/31/2013	1.3	No change	No changes to the meaning, language, or formatting of the technical content.
08/08/2013	1.3	No change	No changes to the meaning, language, or formatting of the technical content.
11/14/2013	1.3	No change	No changes to the meaning, language, or formatting of the technical content.
02/13/2014	1.3	No change	No changes to the meaning, language, or formatting of the technical content.
05/15/2014	1.3	No change	No changes to the meaning, language, or formatting of the technical content.

Contents

1 Introduction	5
1.1 Glossary	5
1.2 References.....	5
1.2.1 Normative References.....	5
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
2 Messages	8
2.1 Transport.....	8
2.2 Message Syntax	8
2.2.1 Redundant Block	8
3 Protocol Details	9
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
4 Protocol Examples	11
5 Security	13
5.1 Security Considerations for Implementers.....	13
5.2 Index of Security Parameters	13
6 Appendix A: Product Behavior	14
7 Change Tracking	15
8 Index	16

1 Introduction

The Real-Time Transport Protocol (RTP/RTCP): Redundant Audio Data Extensions (RTPRAD) Protocol is a method for encoding redundant audio data for use with the Real-Time Transport Protocol (RTP) Extensions Protocol as specified in [\[MS-RTPME\]](#). RTPRAD is an extension of RTP Payload for Redundant Audio Data as specified in [\[RFC2198\]](#). [\[RFC2198\]](#) specifies a payload format for use with the Real-Time Transport Protocol (RTP) as specified in [\[RFC3550\]](#).

Sections 1.8, 2, and 3 of this specification are normative and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in RFC 2119. Sections 1.5 and 1.9 are also normative but cannot contain those terms. All other sections and examples in this specification are informative.

1.1 Glossary

The following terms are specific to this document:

Dual Tone Multiple Frequency (DTMF): The signaling system used in telephony systems, in which each digit is associated with two specific frequencies. Most commonly associated with telephone touch-tone keypads.

lossy network transport: Defined as a transport where data payload cannot be reliably delivered from source to destination.

Real-Time Transport Protocol (RTP): A network protocol that provides end-to-end network transport functions suitable for applications transmitting real-time data, such as audio and video.

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

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 Specifications documentation 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.

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

[MS-SDP] Microsoft Corporation, "[Session Description Protocol \(SDP\) 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.ietf.org/rfc/rfc2198.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>

1.2.2 Informative References

[RFC4733] Schulzrinne, H., "RTP Payload for DTMF Digits, Telephony Tones and Telephony Signals", RFC 4733, December 2006, <http://www.ietf.org/rfc/rfc4733.txt>

1.3 Overview

RTPRAD extends the RTP Payload for Redundant Audio Data as specified 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

RTPRAD relies on the Real-Time Transport Protocol (RTP/RTCP): Microsoft Extensions [MS-RTPME] as its transport.

This specification only addresses the redundancy (and thereby loss and error tolerance) of audio data streams.

1.5 Prerequisites/Preconditions

Because the Real-Time Transport Protocol (RTP/RTCP): Microsoft Extensions (RTPME) act as a transport for this protocol, a valid RTP session must be established.

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-SDP].

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 in-band **Dual Tone Multiple Frequency (DTMF)** tones.

1.7 Versioning and Capability Negotiation

- Supported Transports: The RTP/RTCP: Redundant Audio Data Extensions are implemented on top of [MS-RTPME] as the transport mechanism.
- Protocol Versions: The RTP/RTCP: Redundant Audio Data Extensions, as a payload format of **RTP**, do 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 specification does not describe any security or authentication methods. Security and authentication are dependent on the security method, authentication method, or both as used by [MS-RTPME]

- Localization: None.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

Because RTPRAD uses RTP as its transport [\[MS-RTPME\]](#), 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 [\[MS-SDP\]](#).

2.2 Message Syntax

The structure and syntax of RTPRAD are defined within RTP Payload for Redundant Audio Data [\[RFC2198\]](#). 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 in-band DTMF tones. For more information about in-band DTMF tones, see [\[RFC4733\]](#).

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

- Section 2 of [\[RFC2198\]](#) provides for one or more redundant audio blocks for each RTP payload. However, 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.
- Section 2 of [\[RFC2198\]](#) describes the mechanism for including the redundancy information in the RTP packet header. However, RTPRAD does not support redundant information in the RTP header. The RTP header MUST NOT contain redundant information. RTPRAD MUST be made part of a dynamic RTP payload type and negotiate as such during SDP negotiation.
- While section 2 of [\[RFC2198\]](#) allows for static typing of payload types, systems interoperating with an implementation of this protocol MUST negotiate for dynamic redundancy payload type using SDP or redundancy is not enabled.

2.2.1 Redundant Block

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

3 Protocol Details

RTPRAD uses a Sender and Receiver model.

The following sections detail the behavioral difference between the protocol specified by [\[RFC2198\]](#) and this protocol specification.

3.1 Receiver Details

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

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. 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

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

The redundancy data block MUST NOT have a distance greater than 3.

Distance is defined as the number of RTP packets succeeding the main block for which the redundancy block applies. For example, if RTP packet X contains main block A, and RTP packet X + n contains the redundancy block for main block A, that redundancy block has a distance of n. The maximum value of n MUST NOT exceed 3.

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

All redundant audio data from the Sender MUST be the same encoding (same codec) as the main audio block. This requirement deviates from [\[RFC2198\]](#) where secondary, tertiary, or other successive codecs are supported.

This means that the main 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

A typical example of the SDP negotiation under Microsoft Office Communicator would appear similar to the following.

```
m=audio 51712 RTP/AVP 114 111 112 115 116 4 8 0 97 101]
...
a=rtpmap:114 x-msrta/16000
a=fmtp:114 bitrate=29000
a=rtpmap:111 SIREN/16000
a=fmtp:111 bitrate=16000
a=rtpmap:112 G7221/16000
a=fmtp:112 bitrate=24000
a=rtpmap:115 x-msrta/8000
a=fmtp:115 bitrate=11800
a=rtpmap:116 AAL2-G726-32/8000
a=rtpmap:4 G723/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:0 PCMU/8000
a=rtpmap:97 RED/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
```

The RED/8000 line uses the default redundant payload type mapping for Microsoft Office Communicator (PT=97). Given this if the negotiated encoding is RT-Audio 16Khz, the payload containing the main block + redundant block would appear as follows.

```
0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|V=2|P|X| CC=0 |M|          PT          | sequence number of primary |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     timestamp of primary encoding |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               synchronization source (SSRC) identifier |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|1| block PT=97 | timestamp offset          | block length |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|0| block PT=114|
+---+---+---+---+---+
|
+
|           RT-Audio encoded redundant data (PT=97)
|           (38 bytes)
+
|
+
|
+
|
+-----+
|
+
|           RT-Audio encoded primary data (PT=114)
+           (38 bytes)
|
```



5 Security

5.1 Security Considerations for Implementers

There are no additional protocol considerations beyond those described in [\[RFC2198\]](#) section 6.

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:

- Windows 2000 Server 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

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

8 Index

A

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

C

[Capability negotiation](#) 6
[Change tracking](#) 15

D

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

E

[Examples - overview](#) 11

F

[Fields - vendor-extensible](#) 7

G

[Glossary](#) 5

H

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

I

[Implementer - security considerations](#) 13
[Index of security parameters](#) 13
[Informative references](#) 6
Initialization
[receiver](#) 9
[sender](#) 10
[Introduction](#) 5

L

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

M

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

[overview](#) 8
[redundant block layout](#) 8
[transport](#) 8

N

[Normative references](#) 5

O

[Overview](#) 6

P

[Parameters - security index](#) 13
[Preconditions](#) 6
[Prerequisites](#) 6
[Product behavior](#) 14

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 layout](#) 8
References
[informative](#) 6
[normative](#) 5
[Relationship to other protocols](#) 6

S

Security
[implementer considerations](#) 13
[parameter index](#) 13
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
Syntax
[overview](#) 8
[redundant block layout](#) 8

T

Timer events

[receiver](#) 9

[sender](#) 10

Timers

[receiver](#) 9

[sender](#) 10

[Tracking changes](#) 15

[Transport](#) 8

Triggered events - higher-layer

[receiver](#) 9

[sender](#) 10

V

[Vendor-extensible fields](#) 7

[Versioning](#) 6