

[MS-RDPEECO]:

Remote Desktop Protocol: Virtual Channel Echo 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
12/16/2011	1.0	New	Released new document.
3/30/2012	1.0	None	No changes to the meaning, language, or formatting of the technical content.
7/12/2012	1.0	None	No changes to the meaning, language, or formatting of the technical content.
10/25/2012	1.0	None	No changes to the meaning, language, or formatting of the technical content.
1/31/2013	1.0	None	No changes to the meaning, language, or formatting of the technical content.
8/8/2013	1.0	None	No changes to the meaning, language, or formatting of the technical content.
11/14/2013	1.0	None	No changes to the meaning, language, or formatting of the technical content.
2/13/2014	1.0	None	No changes to the meaning, language, or formatting of the technical content.
5/15/2014	1.0	None	No changes to the meaning, language, or formatting of the technical content.
6/30/2015	2.0	Major	Significantly changed the technical content.

Table of Contents

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

1 Introduction

This document specifies the Remote Desktop Protocol: Virtual Channel Echo Extension to Remote Desktop Protocol: Basic Connectivity and Graphics Remoting, as specified in [\[MS-RDPBCGR\]](#). The echo messages defined in section [2.2](#) are used to bounce a payload sent by a **terminal server** off of a connected terminal-server client, thereby providing a simple mechanism to determine network characteristics such as **round-trip time (RTT)**.

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 [\[RFC2119\]](#). Sections 1.5 and 1.9 are also normative but do not contain those terms. All other sections and examples in this specification are informative.

1.1 Glossary

The following terms are specific to this document:

ANSI character: An 8-bit Windows-1252 character set unit.

protocol data unit (PDU): Information that is delivered as a unit among peer entities of a network and that may contain control information, address information, or data. For more information on remote procedure call (RPC)-specific PDUs, see [\[C706\]](#) section 12.

round-trip time (RTT): The time that it takes a packet to be sent to a remote partner and for that partner's acknowledgment to arrive at the original sender. This is a measurement of latency between partners.

terminal server: A computer on which terminal services is running.

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.

[MS-DTYP] Microsoft Corporation, "[Windows Data Types](#)".

[MS-RDPEDYC] Microsoft Corporation, "[Remote Desktop Protocol: Dynamic Channel Virtual Channel Extension](#)".

[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>

1.2.2 Informative References

[MS-RDPBCGR] Microsoft Corporation, "[Remote Desktop Protocol: Basic Connectivity and Graphics Remoting](#)".

1.3 Overview

The sequence of messages exchanged by the Remote Desktop Protocol: Virtual Channel Echo Extension is described in the following figure. The messages exchanged in this diagram are strictly sequential.

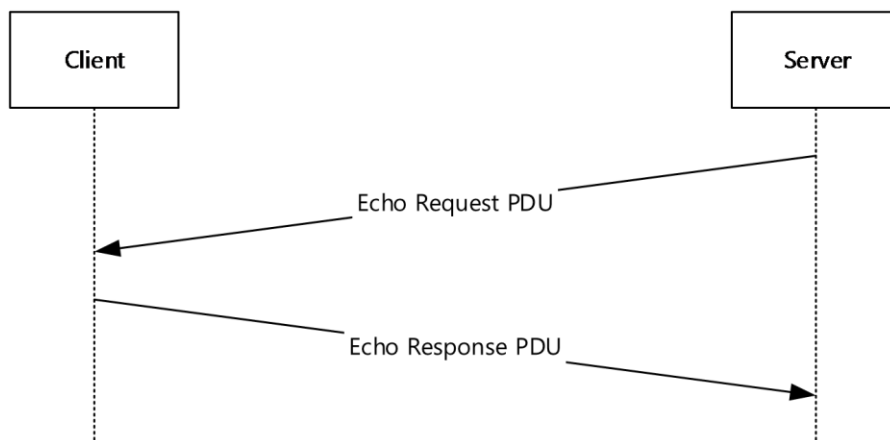


Figure 1: The echo message sequence

The terminal server originates all Echo Request **protocol data units (PDUs)** (section [2.2.1](#)). Each Echo Request PDU sent to a terminal-server client contains a sequence of bytes ([\[MS-DTYP\]](#) section 2.1.2) that must be sent back to the server in an Echo Response PDU (section [2.2.2](#)).

1.4 Relationship to Other Protocols

The Remote Desktop Protocol: Virtual Channel Echo Extension is embedded in a dynamic virtual channel transport, as specified in [\[MS-RDPEDYC\]](#) sections 1 to 3.

1.5 Prerequisites/Preconditions

The Remote Desktop Protocol: Virtual Channel Echo Extension operates only after the dynamic virtual channel transport, as specified in [\[MS-RDPEDYC\]](#) sections 1 to 3, is fully established. If the dynamic virtual channel transport is terminated, the Remote Desktop Protocol: Virtual Channel Echo Extension is also terminated. The protocol is terminated by closing the underlying dynamic virtual channel. For details about closing the dynamic virtual channel, refer to [\[MS-RDPEDYC\]](#) section 3.2.5.2.

1.6 Applicability Statement

The Remote Desktop Protocol: Virtual Channel Echo Extension is applicable in scenarios where a simple mechanism to determine network characteristics (such as round-trip time (RTT)) between a terminal server and a terminal server client is required.

1.7 Versioning and Capability Negotiation

None.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

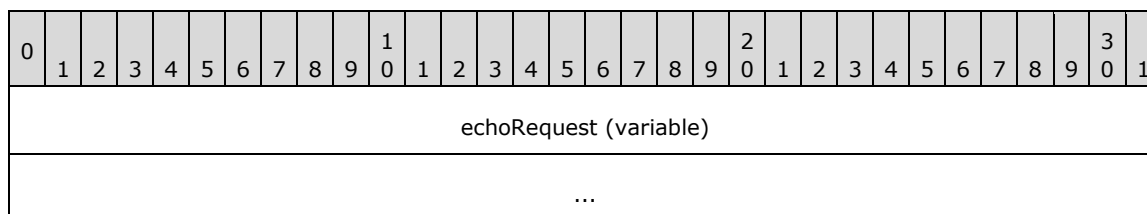
The Remote Desktop Protocol: Virtual Channel Echo Extension is designed to operate over a dynamic virtual channel, as specified in [MS-RDPEDYC] sections 1 to 3. The dynamic virtual channel name is the null-terminated **ANSI character** string "ECHO". The channel is opened by the server and accepted by client<1> as described in [MS-RDPEDYC] section 3.3.3.2. The usage of channel names in the context of opening a dynamic virtual channel is specified in [MS-RDPEDYC] section 2.2.2.1.

2.2 Message Syntax

The following sections specify the Remote Desktop Protocol: Virtual Channel Echo Extension message syntax.

2.2.1 ECHO_REQUEST_PDU

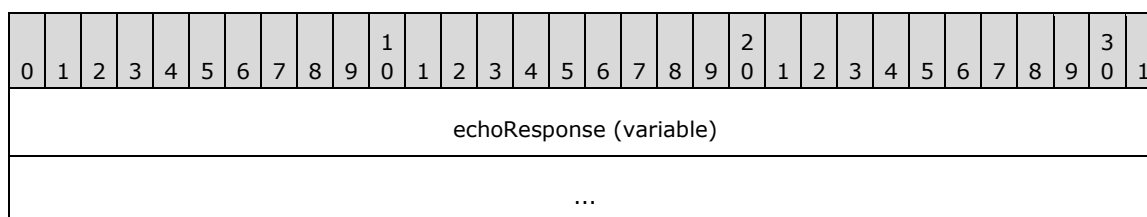
The ECHO_REQUEST_PDU message is a server-to-client PDU that is used to transmit a sequence of bytes that MUST be echoed back to the server using an ECHO_RESPONSE_PDU message (section 2.2.2).



echoRequest (variable): A variable-length array of bytes containing a message that MUST be replayed by the client using an ECHO_RESPONSE_PDU message (section 2.2.2).

2.2.2 ECHO_RESPONSE_PDU

The ECHO_RESPONSE_PDU message is a client-to-server PDU that is used to echo the sequence of bytes transmitted in an ECHO_REQUEST_PDU message (section 2.2.1). The ECHO_RESPONSE_PDU message MUST be sent only in response to an ECHO_REQUEST_PDU message.



echoResponse (variable): A variable-length array of bytes containing the message that was transmitted in the **echoRequest** field of an ECHO_REQUEST_PDU message (section 2.2.1).

3 Protocol Details

3.1 Server Details

3.1.1 Abstract Data Model

None.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Higher-Layer Triggered Events

None.

3.1.5 Message Processing Events and Sequencing Rules

3.1.5.1 Sending ECHO_REQUEST_PDU

The structure and fields of the ECHO_REQUEST_PDU message are specified in section [2.2.1](#). The **echoRequest** field of the ECHO_REQUEST_PDU message MUST be populated with a byte stream of at least one byte in size.

3.1.5.2 Processing ECHO_RESPONSE_PDU

The structure and fields of the ECHO_RESPONSE_PDU message are specified in section [2.2.2](#). Upon receiving the ECHO_RESPONSE_PDU message, the server MAY inspect the data in the **echoResponse** field.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

3.2 Client Details

3.2.1 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 that described in this document.

Note It is possible to implement the following conceptual data by using a variety of techniques as long as the implementation produces external behavior that is consistent with that described in this document.

3.2.1.1 Echo Byte Stream

The **Echo Byte Stream** store contains the stream of bytes that was embedded in the **echoRequest** field of the most recently received ECHO_REQUEST_PDU message (section [2.2.1](#)).

3.2.2 Timers

None.

3.2.3 Initialization

None.

3.2.4 Higher-Layer Triggered Events

None.

3.2.5 Processing Events and Sequencing Rules

3.2.5.1 Processing ECHO_REQUEST_PDU

The structure and fields of the ECHO_REQUEST_PDU message are specified in section [2.2.1](#). Upon receiving the ECHO_REQUEST_PDU message, the client MUST save the byte stream that is embedded in the **echoRequest** field to the **Echo Byte Stream** store (section [3.2.1.1](#)). The client then uses this saved data to construct an ECHO_RESPONSE_PDU message (section [2.2.2](#)), which it transmits to the server (section [3.2.5.2](#)).

3.2.5.2 Sending ECHO_RESPONSE_PDU

The structure and fields of the ECHO_RESPONSE_PDU message are specified in section [2.2.2](#). The **echoResponse** field of the ECHO_RESPONSE_PDU message MUST be populated with the data that was stored in the **Echo Byte Stream** store (section [3.2.1.1](#)) during the processing of the ECHO_REQUEST_PDU message (section [3.2.5.1](#)).

3.2.6 Timer Events

None.

3.2.7 Other Local Events

None.

4 Protocol Examples

4.1 ECHO_REQUEST_PDU

The following is an annotated dump of an ECHO_REQUEST_PDU message (section [2.2.1](#)).

```
00000000 48 65 6c 6c 6f 20 77 6f 72 6c 64 21          Hello world!
48 65 6c 6c 6f 20 77 6f 72 6c 64 21 -> ECHO_REQUEST_PDU::echoRequest
```

4.2 ECHO_RESPONSE_PDU

The following is an annotated dump of the ECHO_RESPONSE_PDU message (section [2.2.2](#)) sent in response to the PDU in section [4.1](#).

```
00000000 48 65 6c 6c 6f 20 77 6f 72 6c 64 21          Hello world!
48 65 6c 6c 6f 20 77 6f 72 6c 64 21 -> ECHO_RESPONSE_PDU::echoResponse
```

5 Security

5.1 Security Considerations for Implementers

None.

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 Vista operating system
- Windows Server 2008 operating system
- Windows 7 operating system
- Windows Server 2008 R2 operating system
- Windows 8 operating system
- Windows Server 2012 operating system
- Windows 8.1 operating system
- Windows Server 2012 R2 operating system
- 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.

[<1> Section 2.1](#): All Windows clients accept connections on the ECHO channel. The ECHO channel is not opened by Windows Vista, Windows Server 2008, Windows 7, and Windows Server 2008 R2.

7 Change Tracking

This section identifies changes that were made to this document since the last release. 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.
- The removal of a document from the documentation set.

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 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 changes were introduced. Minor editorial and formatting changes may have been made, but the technical content of the document is identical to the last released version.

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.
- 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 dochelp@microsoft.com.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
6 Appendix A: Product Behavior	Added Windows 10 to applicability list.	Y	Content update.

8 Index

A

Abstract data model

[client](#) 8

[server](#) 8

[Applicability](#) 5

C

[Capability negotiation](#) 5

[Change tracking](#) 13

Client

[abstract data model](#) 8

[higher-layer triggered events](#) 9

[initialization](#) 9

[message processing](#) 9

[other local events](#) 9

[sequencing rules](#) 9

[timer events](#) 9

[timers](#) 9

D

Data model - abstract

[client](#) 8

[server](#) 8

E

[ECHO_REQUEST_PDU example](#) 10

[ECHO_REQUEST_PDU message](#) 7

[ECHO_RESPONSE_PDU example](#) 10

[ECHO_RESPONSE_PDU message](#) 7

Examples

[ECHO_REQUEST_PDU](#) 10

[ECHO_RESPONSE_PDU](#) 10

F

[Fields - vendor-extensible](#) 6

G

[Glossary](#) 4

H

Higher-layer triggered events

[client](#) 9

[server](#) 8

I

[Implementer - security considerations](#) 11

[Index of security parameters](#) 11

[Informative references](#) 4

Initialization

[client](#) 9

[server](#) 8

[Introduction](#) 4

M

Message processing

[client](#) 9

Messages

[ECHO_REQUEST_PDU](#) 7

[ECHO_REQUEST_PDU message](#) 7

[ECHO_RESPONSE_PDU](#) 7

[ECHO_RESPONSE_PDU message](#) 7

[transport](#) 7

N

[Normative references](#) 4

O

Other local events

[client](#) 9

[server](#) 8

[Overview \(synopsis\)](#) 5

P

[Parameters - security index](#) 11

[Preconditions](#) 5

[Prerequisites](#) 5

[Product behavior](#) 12

R

[References](#) 4

[informative](#) 4

[normative](#) 4

[Relationship to other protocols](#) 5

S

Security

[implementer considerations](#) 11
[parameter index](#) 11

Sequencing rules

[client](#) 9

Server

[abstract data model](#) 8
[higher-layer triggered events](#) 8
[initialization](#) 8
[other local events](#) 8
[timer events](#) 8
[timers](#) 8
[Standards assignments](#) 6

T

Timer events

[client](#) 9
[server](#) 8

Timers

[client](#) 9
[server](#) 8
[Tracking changes](#) 13
[Transport](#) 7

Triggered events - higher-layer

[client](#) 9
[server](#) 8

V

[Vendor-extensible fields](#) 6
[Versioning](#) 5