[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, 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
12/16/2011	1.0	New	Released new document.
03/30/2012	1.0	No change	No changes to the meaning, language, or formatting of the technical content.
07/12/2012	1.0	No change	No changes to the meaning, language, or formatting of the technical content.
10/25/2012	1.0	No change	No changes to the meaning, language, or formatting of the technical content.
01/31/2013	1.0	No change	No changes to the meaning, language, or formatting of the technical content.
08/08/2013	1.0	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 7
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 ECHO_REQUEST_PDU 8
2.2.2 ECHO_RESPONSE_PDU 8

3 Protocol Details 9
3.1 Server 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
3.1.5.1 Sending ECHO_REQUEST_PDU 9
3.1.5.2 Processing ECHO_RESPONSE_PDU 9
3.1.6 Timer Events 9
3.1.7 Other Local Events 9
3.2 Client Details 9
3.2.1 Abstract Data Model 9
3.2.1.1 Echo Byte Stream 10
3.2.2 Timers 10
3.2.3 Initialization 10
3.2.4 Higher-Layer Triggered Events 10
3.2.5 Processing Events and Sequencing Rules 10
3.2.5.1 Processing ECHO_REQUEST_PDU 10
3.2.5.2 Sending ECHO_RESPONSE_PDU 10
3.2.6 Timer Events 10
3.2.7 Other Local Events 10

4 Protocol Examples 11
4.1 ECHO_REQUEST_PDU 11
4.2 ECHO_RESPONSE_PDU 11

5 Security 12
5.1 Security Considerations for Implementers 12
5.2 Index of Security Parameters 12

6 Appendix A: Product Behavior 13

7 Change Tracking.....14

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 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 defined in [MS-GLOS]:

PDU
protocol data unit (PDU)
round-trip time (RTT)

The following terms are specific to this document:

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

terminal server: The server to which the client initiated the remote desktop connection.

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.

A reference marked "(Archived)" means that the reference document was either retired and is no longer being maintained or was replaced with a new document that provides current implementation details. We archive our documents online [Windows Protocol].

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

[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.

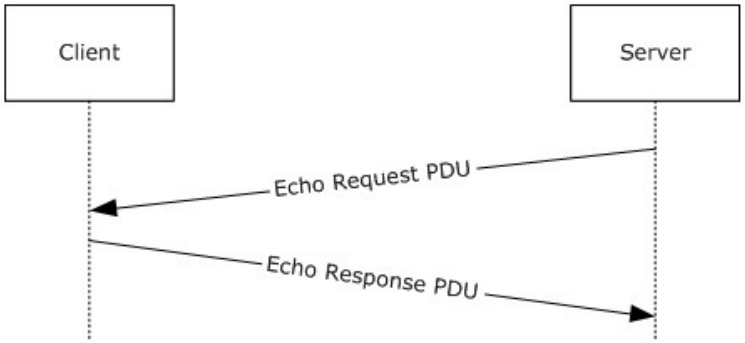


Figure 1: The echo message sequence

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

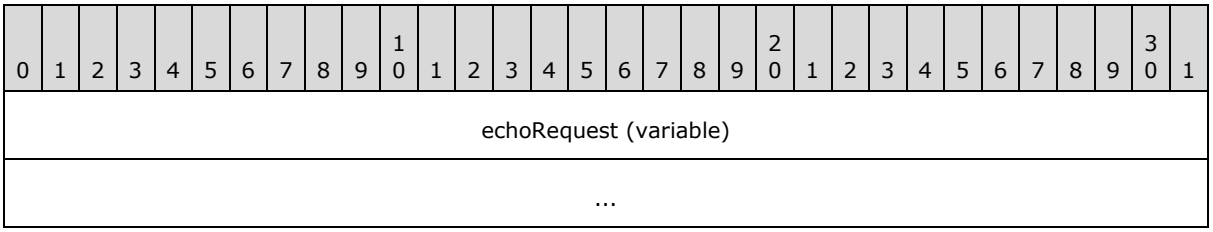
The Remote Desktop Protocol: Virtual Channel Echo Extension is designed to operate over a dynamic virtual channel, as specified in [MS-RDPEDYC] sections 1 to 3. The dynamic virtual channel name is the null-terminated **ANSI character** string "ECHO". The channel is opened by the server and accepted by 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

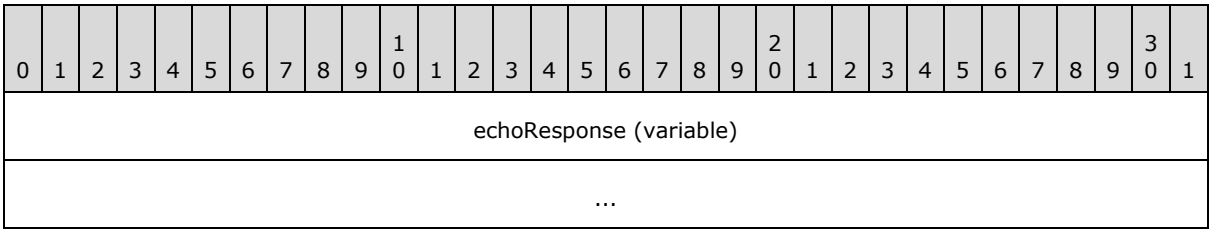
The ECHO_REQUEST_PDU message is a server-to-client PDU that is used to transmit a sequence of bytes that MUST be echoed back to the server using an ECHO_RESPONSE_PDU message (section 2.2.2).



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

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: The ECHO channel is only opened by Windows 8, Windows Server 2012, Windows 8.1, and Windows Server 2012 R2 servers. All Windows clients accept connections on the ECHO channel.

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
  client 9
  server 9
Applicability 6

C

Capability negotiation 7
Change tracking 14
Client
  abstract data model 9
  higher-layer triggered events 10
  initialization 10
  message processing 10
  other local events 10
  sequencing rules 10
  timer events 10
  timers 10

D

Data model - abstract
  client 9
  server 9

E

ECHO_REQUEST_PDU example 11
ECHO_REQUEST_PDU message 8
ECHO_RESPONSE_PDU example 11
ECHO_RESPONSE_PDU message 8
Examples
  ECHO_REQUEST_PDU 11
  ECHO_RESPONSE_PDU 11

F

Fields - vendor-extensible 7

G

Glossary 5

H

Higher-layer triggered events
  client 10
  server 9

I

Implementer - security considerations 12
Index of security parameters 12
Informative references 6
Initialization
  client 10
  server 9
Introduction 5

M

Message processing
  client 10
Messages
  ECHO_REQUEST_PDU message 8
  ECHO_RESPONSE_PDU message 8
  transport 8

N

Normative references 5

O

Other local events
  client 10
  server 9
Overview (synopsis) 6

P

Parameters - security index 12
Preconditions 6
Prerequisites 6
Product behavior 13

R

References
  informative 6
  normative 5
Relationship to other protocols 6

S

Security
  implementer considerations 12
  parameter index 12
Sequencing rules
  client 10
Server
  abstract data model 9
  higher-layer triggered events 9
  initialization 9
  other local events 9
  timer events 9
  timers 9
Standards assignments 7

T

Timer events
  client 10
  server 9
Timers
  client 10

server 9
Tracking changes 14
Transport 8
Triggered events - higher-layer
client 10
server 9

V

Vendor-extensible fields 7
Versioning 7