**[MS-RDPESP]:**

**Remote Desktop Protocol: Serial and Parallel Port Virtual Channel Extension**

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](http://go.microsoft.com/fwlink/?LinkId=214445) or the [Microsoft Community Promise](http://go.microsoft.com/fwlink/?LinkId=214448). 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](https://msdn.microsoft.com/en-us/openspecifications/dn750984).
* **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 |
| --- | --- | --- | --- |
| 7/20/2007 | 0.1 | Major | MCPP Milestone 5 Initial Availability |
| 9/28/2007 | 1.0 | Major | Updated and revised the technical content. |
| 10/23/2007 | 1.1 | Minor | Clarified the meaning of the technical content. |
| 11/30/2007 | 1.2 | Minor | Clarified the meaning of the technical content. |
| 1/25/2008 | 1.2.1 | Editorial | Changed language and formatting in the technical content. |
| 3/14/2008 | 1.2.2 | Editorial | Changed language and formatting in the technical content. |
| 5/16/2008 | 1.2.3 | Editorial | Changed language and formatting in the technical content. |
| 6/20/2008 | 1.3 | Minor | Clarified the meaning of the technical content. |
| 7/25/2008 | 1.3.1 | Editorial | Changed language and formatting in the technical content. |
| 8/29/2008 | 1.3.2 | Editorial | Changed language and formatting in the technical content. |
| 10/24/2008 | 1.3.3 | Editorial | Changed language and formatting in the technical content. |
| 12/5/2008 | 1.3.4 | Editorial | Editorial Update. |
| 1/16/2009 | 1.4 | Minor | Clarified the meaning of the technical content. |
| 2/27/2009 | 1.4.1 | Editorial | Changed language and formatting in the technical content. |
| 4/10/2009 | 1.4.2 | Editorial | Changed language and formatting in the technical content. |
| 5/22/2009 | 2.0 | Major | Updated and revised the technical content. |
| 7/2/2009 | 3.0 | Major | Updated and revised the technical content. |
| 8/14/2009 | 4.0 | Major | Updated and revised the technical content. |
| 9/25/2009 | 4.1 | Minor | Clarified the meaning of the technical content. |
| 11/6/2009 | 5.0 | Major | Updated and revised the technical content. |
| 12/18/2009 | 6.0 | Major | Updated and revised the technical content. |
| 1/29/2010 | 6.1 | Minor | Clarified the meaning of the technical content. |
| 3/12/2010 | 6.1.1 | Editorial | Changed language and formatting in the technical content. |
| 4/23/2010 | 6.1.2 | Editorial | Changed language and formatting in the technical content. |
| 6/4/2010 | 6.1.3 | Editorial | Changed language and formatting in the technical content. |
| 7/16/2010 | 6.1.3 | None | No changes to the meaning, language, or formatting of the technical content. |
| 8/27/2010 | 6.1.3 | None | No changes to the meaning, language, or formatting of the technical content. |
| 10/8/2010 | 6.1.3 | None | No changes to the meaning, language, or formatting of the technical content. |
| 11/19/2010 | 6.1.3 | None | No changes to the meaning, language, or formatting of the technical content. |
| 1/7/2011 | 6.1.3 | None | No changes to the meaning, language, or formatting of the technical content. |
| 2/11/2011 | 7.0 | Major | Updated and revised the technical content. |
| 3/25/2011 | 7.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 5/6/2011 | 7.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 6/17/2011 | 7.1 | Minor | Clarified the meaning of the technical content. |
| 9/23/2011 | 7.1 | None | No changes to the meaning, language, or formatting of the technical content. |
| 12/16/2011 | 8.0 | Major | Updated and revised the technical content. |
| 3/30/2012 | 8.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 7/12/2012 | 8.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 10/25/2012 | 8.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 1/31/2013 | 8.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 8/8/2013 | 8.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 11/14/2013 | 8.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 2/13/2014 | 8.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 5/15/2014 | 8.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 6/30/2015 | 9.0 | Major | Significantly changed the technical content. |
| 10/16/2015 | 9.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 7/14/2016 | 9.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 6/1/2017 | 9.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 9/15/2017 | 10.0 | Major | Significantly changed the technical content. |

Table of Contents

[1 Introduction 6](#_Toc492420565)

[1.1 Glossary 6](#_Toc492420566)

[1.2 References 6](#_Toc492420567)

[1.2.1 Normative References 6](#_Toc492420568)

[1.2.2 Informative References 7](#_Toc492420569)

[1.3 Overview 7](#_Toc492420570)

[1.3.1 Purpose of Device Redirection Extensions 7](#_Toc492420571)

[1.3.2 Protocol Initialization 7](#_Toc492420572)

[1.4 Relationship to Other Protocols 7](#_Toc492420573)

[1.5 Prerequisites/Preconditions 8](#_Toc492420574)

[1.6 Applicability Statement 8](#_Toc492420575)

[1.7 Versioning and Capability Negotiation 8](#_Toc492420576)

[1.8 Vendor-Extensible Fields 8](#_Toc492420577)

[1.9 Standards Assignments 8](#_Toc492420578)

[2 Messages 9](#_Toc492420579)

[2.1 Transport 9](#_Toc492420580)

[2.2 Message Syntax 9](#_Toc492420581)

[2.2.1 Common Data Types 9](#_Toc492420582)

[2.2.2 Port Redirection Messages 9](#_Toc492420583)

[2.2.2.1 Client Device List Announce Request 9](#_Toc492420584)

[2.2.2.2 Server Create Request (DR\_PORT\_CREATE\_REQ) 10](#_Toc492420585)

[2.2.2.3 Server Close Request (DR\_PORT\_CLOSE\_REQ) 10](#_Toc492420586)

[2.2.2.4 Server Read Request (DR\_PORT\_READ\_REQ) 10](#_Toc492420587)

[2.2.2.5 Server Write Request (DR\_PORT\_WRITE\_REQ) 10](#_Toc492420588)

[2.2.2.6 Server Device Control Request (DR\_PORT\_CONTROL\_REQ) 10](#_Toc492420589)

[2.2.2.7 Client Create Response (DR\_PORT\_CREATE\_RSP) 12](#_Toc492420590)

[2.2.2.8 Client Close Response (DR\_PORT\_CLOSE\_RSP) 12](#_Toc492420591)

[2.2.2.9 Client Read Response (DR\_PORT\_READ\_RSP) 12](#_Toc492420592)

[2.2.2.10 Client Write Response (DR\_PORT\_WRITE\_RSP) 12](#_Toc492420593)

[2.2.2.11 Client Device Control Response (DR\_PORT\_CONTROL\_RSP) 13](#_Toc492420594)

[3 Protocol Details 14](#_Toc492420595)

[3.1 Common Details 14](#_Toc492420596)

[3.1.1 Abstract Data Model 14](#_Toc492420597)

[3.1.2 Timers 14](#_Toc492420598)

[3.1.3 Initialization 14](#_Toc492420599)

[3.1.4 Higher-Layer Triggered Events 14](#_Toc492420600)

[3.1.5 Message Processing Events and Sequencing Rules 14](#_Toc492420601)

[3.1.6 Timer Events 14](#_Toc492420602)

[3.1.7 Other Local Events 14](#_Toc492420603)

[3.2 Client Details 14](#_Toc492420604)

[3.2.1 Abstract Data Model 14](#_Toc492420605)

[3.2.2 Timers 15](#_Toc492420606)

[3.2.3 Initialization 15](#_Toc492420607)

[3.2.4 Higher-Layer Triggered Events 15](#_Toc492420608)

[3.2.5 Message Processing Events and Sequencing Rules 15](#_Toc492420609)

[3.2.5.1 Port Redirection Messages 15](#_Toc492420610)

[3.2.5.1.1 Sending a Client Device List Announce Request Message 15](#_Toc492420611)

[3.2.5.1.2 Processing a Server Create Request Message 15](#_Toc492420612)

[3.2.5.1.3 Processing a Server Close Request Message 15](#_Toc492420613)

[3.2.5.1.4 Processing a Server Read Request Message 15](#_Toc492420614)

[3.2.5.1.5 Processing a Server Write Request Message 16](#_Toc492420615)

[3.2.5.1.6 Processing a Server Device Control Request Message 16](#_Toc492420616)

[3.2.5.1.7 Sending a Create Response Message 16](#_Toc492420617)

[3.2.5.1.8 Sending a Close Response Message 17](#_Toc492420618)

[3.2.5.1.9 Sending a Read Response Message 17](#_Toc492420619)

[3.2.5.1.10 Sending a Write Response Message 17](#_Toc492420620)

[3.2.5.1.11 Sending a Device Control Response Message 18](#_Toc492420621)

[3.2.6 Timer Events 18](#_Toc492420622)

[3.2.7 Other Local Events 18](#_Toc492420623)

[3.3 Server Details 19](#_Toc492420624)

[3.3.1 Abstract Data Model 19](#_Toc492420625)

[3.3.2 Timers 19](#_Toc492420626)

[3.3.3 Initialization 19](#_Toc492420627)

[3.3.4 Higher-Layer Triggered Events 19](#_Toc492420628)

[3.3.5 Message Processing Events and Sequencing Rules 19](#_Toc492420629)

[3.3.5.1 Port Redirection Messages 19](#_Toc492420630)

[3.3.5.1.1 Processing a Client Device List Announce Request Message 19](#_Toc492420631)

[3.3.5.1.2 Sending a Server Create Request Message 19](#_Toc492420632)

[3.3.5.1.3 Sending a Server Close Request Message 19](#_Toc492420633)

[3.3.5.1.4 Sending a Server Write Request Message 19](#_Toc492420634)

[3.3.5.1.5 Sending a Server Read Request Message 19](#_Toc492420635)

[3.3.5.1.6 Sending a Server Device Control Request Message 19](#_Toc492420636)

[3.3.5.1.7 Processing a Client Create Response Message 20](#_Toc492420637)

[3.3.5.1.8 Processing a Client Close Response Message 20](#_Toc492420638)

[3.3.5.1.9 Processing a Client Write Response Message 20](#_Toc492420639)

[3.3.5.1.10 Processing a Client Read Response Message 20](#_Toc492420640)

[3.3.5.1.11 Processing a Client Device Control Response Message 20](#_Toc492420641)

[3.3.6 Timer Events 20](#_Toc492420642)

[3.3.7 Other Local Events 20](#_Toc492420643)

[4 Protocol Examples 21](#_Toc492420644)

[4.1 Port Redirection Annotations 21](#_Toc492420645)

[4.2 Server Create Request Example 22](#_Toc492420646)

[4.3 Client Create Response Example 23](#_Toc492420647)

[4.4 IO Operations Examples 23](#_Toc492420648)

[5 Security 26](#_Toc492420649)

[5.1 Security Considerations for Implementers 26](#_Toc492420650)

[5.2 Index of Security Parameters 26](#_Toc492420651)

[6 Appendix A: Product Behavior 27](#_Toc492420652)

[7 Change Tracking 29](#_Toc492420653)

[8 Index 30](#_Toc492420654)

# Introduction

This document specifies the Remote Desktop Protocol: Serial and Parallel Port Virtual Channel Extension. This protocol is used to redirect serial and parallel ports from a [**terminal client**](#gt_0e260a32-2049-4eaa-bdec-bfdee19bad4b) to the [**terminal server**](#gt_b416f72e-cf04-4d80-bf93-f5753f3b0998). This allows the [**server**](#gt_434b0234-e970-4e8c-bdfa-e16a30d96703) to access [**client**](#gt_60e0e1fa-66fe-41e1-b5e3-ceab97e53506) ports as if the connected devices were local to the server.

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.

## Glossary

This document uses the following terms:

**client**: A computer on which the remote procedure call (RPC) client is executing.

**device control**: Driver-specific operations that can be performed on various drivers. Each DeviceIOControl is associated with an operation code (called IoCode) and optionally input and output buffers. Device drivers depending on the IoCode take various actions on the input and output buffers

**pseudo device**: A virtual device object created by the server to represent a remote device attached to the remote (or client) machine. Applications and drivers on the server interact with this pseudo/virtual device and the server forwards requests to the remote device. Responses from the remote device are returned to the pseudo device, which then forwards them to the applications or drivers interacting with the device. Examples of pseudo devices include the pseudo port device, pseudo printer device, pseudo drive device, pseudo smartcard device, pseudo PnP device, and so on.

**remote device**: A device that is attached to a remote (or [**client**](#gt_60e0e1fa-66fe-41e1-b5e3-ceab97e53506)) machine, in contrast to a device physically attached to a machine.

**server**: A computer on which the remote procedure call (RPC) server is executing.

**terminal client**: A [**client**](#gt_60e0e1fa-66fe-41e1-b5e3-ceab97e53506) of a [**terminal server**](#gt_b416f72e-cf04-4d80-bf93-f5753f3b0998). A terminal client program that runs on the [**client**](#gt_60e0e1fa-66fe-41e1-b5e3-ceab97e53506) machine.

**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]](https://go.microsoft.com/fwlink/?LinkId=90317). All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

## 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](https://go.microsoft.com/fwlink/?linkid=850906).

### 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-ERREF] Microsoft Corporation, "[Windows Error Codes](%5bMS-ERREF%5d.pdf#Section_1bc92ddfb79e413cbbaa99a5281a6c90)".

[MS-RDPBCGR] Microsoft Corporation, "[Remote Desktop Protocol: Basic Connectivity and Graphics Remoting](%5bMS-RDPBCGR%5d.pdf#Section_5073f4ed1e9345e1b0396e30c385867c)".

[MS-RDPEFS] Microsoft Corporation, "[Remote Desktop Protocol: File System Virtual Channel Extension](%5bMS-RDPEFS%5d.pdf#Section_34d9de58b2b540b6b970f82d4603bdb5)".

[MS-SMB2] Microsoft Corporation, "[Server Message Block (SMB) Protocol Versions 2 and 3](%5bMS-SMB2%5d.pdf#Section_5606ad475ee0437a817e70c366052962)".

[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](https://go.microsoft.com/fwlink/?LinkId=90317)

### Informative References

[MSDN-DeviceTypes] Microsoft Corporation, "Specifying Device Types", [https://msdn.microsoft.com/en-us/library/ff563821.aspx](https://go.microsoft.com/fwlink/?LinkId=120804)

[MSDN-IoCtlCodes] Microsoft Corporation, "Defining I/O Control Codes", [http://msdn.microsoft.com/en-us/library/ff543023.aspx](https://go.microsoft.com/fwlink/?LinkId=120805)

[MSDN-Ntddpar] Microsoft Corporation, "Ntddpar.h", [http://msdn.microsoft.com/en-us/cc308431.aspx](https://go.microsoft.com/fwlink/?LinkId=114013)

[MSDN-Ntddser] Microsoft Corporation, "Ntddser.h", [http://msdn.microsoft.com/en-us/cc308432.aspx](https://go.microsoft.com/fwlink/?LinkId=114014)

[MSDN-PORTS] Microsoft Corporation, "Serial and Parallel ports", [http://msdn.microsoft.com/en-us/library/bb870477.aspx](https://go.microsoft.com/fwlink/?LinkId=102069)

[MSFT-WDDK] Microsoft Corporation, "Windows Driver Kit Version 7.1.0", [http://www.microsoft.com/en-us/download/details.aspx?displaylang=en&id=11800](https://go.microsoft.com/fwlink/?LinkID=507676)

## Overview

The Remote Desktop Protocol: Serial and Parallel Port Virtual Channel Extension specifies the communication used to enable the redirection of serial and parallel ports (ports for short) between a [**terminal client**](#gt_0e260a32-2049-4eaa-bdec-bfdee19bad4b) and a [**terminal server**](#gt_b416f72e-cf04-4d80-bf93-f5753f3b0998). By redirecting ports from the terminal client to the terminal server, applications running on a server machine can access the [**remote devices**](#gt_aa410c29-31c2-44ff-a018-07b96e5acf49) attached to those ports.

### Purpose of Device Redirection Extensions

This extension enables the redirection of serial and parallel port devices attached to the terminal client. With the redirection, such devices can then be accessed by the applications running on the server.

### Protocol Initialization

This extension can be considered as a subprotocol within the Remote Desktop Protocol: File System Virtual Channel Extension as specified in [[MS-RDPEFS]](%5bMS-RDPEFS%5d.pdf#Section_34d9de58b2b540b6b970f82d4603bdb5). It follows the initialization of the Remote Desktop Protocol: File System Virtual Channel Extension to enable port redirection.

## Relationship to Other Protocols

This extension can be considered as a subprotocol within Remote Desktop Protocol: File System Virtual Channel Extension as specified in [[MS-RDPEFS]](%5bMS-RDPEFS%5d.pdf#Section_34d9de58b2b540b6b970f82d4603bdb5). This extension extends the Remote Desktop Protocol: File System Virtual Channel Extension to enable port redirection.

## Prerequisites/Preconditions

The Remote Desktop Protocol: Serial and Parallel Port Virtual Channel Extension operates only after the Remote Desktop Protocol: File System Virtual Channel Extension transport, as specified in [[MS-RDPEFS]](%5bMS-RDPEFS%5d.pdf#Section_34d9de58b2b540b6b970f82d4603bdb5), is fully established.

## Applicability Statement

The Remote Desktop Protocol: Serial and Parallel Port Virtual Channel Extension is designed to be run within the context of a Remote Desktop Protocol virtual channel established between a [**client**](#gt_60e0e1fa-66fe-41e1-b5e3-ceab97e53506) and [**server**](#gt_434b0234-e970-4e8c-bdfa-e16a30d96703). This protocol extension is applicable when applications running on the [**terminal server**](#gt_b416f72e-cf04-4d80-bf93-f5753f3b0998) need to access the ports physically located on a client machine.

## Versioning and Capability Negotiation

This extension relies on the Remote Desktop Protocol: File System Virtual Channel Extension, as specified in [[MS-RDPEFS]](%5bMS-RDPEFS%5d.pdf#Section_34d9de58b2b540b6b970f82d4603bdb5), to perform basic versioning and capability negotiation.

## Vendor-Extensible Fields

This protocol uses NTSTATUS values, as defined in [[MS-ERREF]](%5bMS-ERREF%5d.pdf#Section_1bc92ddfb79e413cbbaa99a5281a6c90) section 2.3. Vendors are free to choose their own values for this field, as long as the C bit (0x20000000) is set, indicating it is a customer code.

## Standards Assignments

The Remote Desktop Protocol: Serial and Parallel Port Virtual Channel Extension contains no standards assignments.

# Messages

Because this is a subprotocol of Remote Desktop Protocol: File System Virtual Channel Extension, as specified in [[MS-RDPEFS]](%5bMS-RDPEFS%5d.pdf#Section_34d9de58b2b540b6b970f82d4603bdb5), this extension shares messages and common data types already specified in [MS-RDPEFS]. This section describes the messages and data types used by Remote Desktop Protocol: Serial and Parallel Port Virtual Channel Extension.

## Transport

All messages MUST be transported over an established Remote Desktop Protocol device extensions channel (as specified in [[MS-RDPEFS]](%5bMS-RDPEFS%5d.pdf#Section_34d9de58b2b540b6b970f82d4603bdb5) section 2.1).

## Message Syntax

The following sections contain Remote Desktop Protocol: Serial and Parallel Port Virtual Channel Extension message syntax.

### Common Data Types

Port redirection uses common data types specified in [[MS-RDPEFS]](%5bMS-RDPEFS%5d.pdf#Section_34d9de58b2b540b6b970f82d4603bdb5) section 2.

### Port Redirection Messages

This protocol does not define any specific messages. It uses a subset of the messages specified in [[MS-RDPEFS]](%5bMS-RDPEFS%5d.pdf#Section_34d9de58b2b540b6b970f82d4603bdb5) section 2. The messages in the following sections are used by this protocol.

#### Client Device List Announce Request

This message is described in [[MS-RDPEFS]](%5bMS-RDPEFS%5d.pdf#Section_34d9de58b2b540b6b970f82d4603bdb5) section 2.2.2.9. The port redirection [**client**](#gt_60e0e1fa-66fe-41e1-b5e3-ceab97e53506) generates the elements of type DEVICE\_ANNOUNCE (as specified in [MS-RDPEFS] section 2.2.1.3) for the port devices it wants to redirect.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| DeviceAnnounceHeader | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**DeviceAnnounceHeader (4 bytes):** For each redirected port a DEVICE\_ANNOUNCE header (as specified in [MS-RDPEFS] section 2.2.1.3) is generated by the client port redirection extension.

The header fields are initialized as follows:

* **DeviceType**: Identifies the device. This value MUST be set to RDPDR\_DTYP\_PARALLEL for parallel ports and RDPDR\_DTYP\_SERIAL for serial ports.
* **DeviceId**: A unique ID that identifies the announced device. The **DeviceId** field follows the semantics described in [MS-RDPEFS] section 2.2.1.3.
* **PreferredDosName**: This field follows the semantic described in [MS-RDPEFS] section 2.2.1.3. It describes the name of the port device as it appears on the client. This protocol does not enforce any semantic limitations on port naming. Client and server implementations determine the port naming convention.[<1>](#Appendix_A_1" \o "Product behavior note 1)
* **DeviceDataLength**: Number of bytes in the **DeviceData** field. For port devices, this value is set to 0.

**Note:** The Client Drive Device List Remove message is not supported by the MS-RDPESP protocol.

#### Server Create Request (DR\_PORT\_CREATE\_REQ)

This message is sent by the [**server**](#gt_434b0234-e970-4e8c-bdfa-e16a30d96703) to open an instance of the port device. The packet for this message is specified in [[MS-RDPEFS]](%5bMS-RDPEFS%5d.pdf#Section_34d9de58b2b540b6b970f82d4603bdb5) section 2.2.1.4.1 (DR\_CREATE\_REQ). The **DeviceId** of the **DeviceIoRequest** field in the DR\_CREATE\_REQ packet MUST match the **DeviceId** value that is sent in the [Client Device List Announce Request](#Section_535df5533d5c4666b33332a0d3d3cab4) packet.

The **PathLength** field of the DR\_CREATE\_REQ packet MUST be set to 0x00000000. This automatically results in setting the packet **Path** field to empty.

Some of the parameters that are passed with this request (**DesiredAccess**, **AllocationSize**, **FileAttributes**, **SharedAccess**, **Disposition** and **CreateOptions**) are treated as opaque by this protocol. The interpretation of these parameters is determined by the client-side driver. The various possible values are specified in [[MS-SMB2]](%5bMS-SMB2%5d.pdf#Section_5606ad475ee0437a817e70c366052962) section 2.2.13.

#### Server Close Request (DR\_PORT\_CLOSE\_REQ)

This message is sent from the [**server**](#gt_434b0234-e970-4e8c-bdfa-e16a30d96703) to close the previously-opened device instance. The packet is specified in [[MS-RDPEFS]](%5bMS-RDPEFS%5d.pdf#Section_34d9de58b2b540b6b970f82d4603bdb5) section 2.2.1.4.2 (DR\_CLOSE\_REQ).

#### Server Read Request (DR\_PORT\_READ\_REQ)

This message is sent from the [**server**](#gt_434b0234-e970-4e8c-bdfa-e16a30d96703) to read data from the port device instance. The packet is specified in [[MS-RDPEFS]](%5bMS-RDPEFS%5d.pdf#Section_34d9de58b2b540b6b970f82d4603bdb5) section 2.2.1.4.3 (DR\_READ\_REQ).

The **Offset** field in this request MUST be set to 0.

Zero-length request semantics: The protocol allows the client and server to request or to complete read/write operations with the **Length** field set to zero. The behavior of these requests and their interpretation is determined by the server application and the client driver.

#### Server Write Request (DR\_PORT\_WRITE\_REQ)

This message is sent from the server to write data to the port device instance. The packet is specified in [[MS-RDPEFS]](%5bMS-RDPEFS%5d.pdf#Section_34d9de58b2b540b6b970f82d4603bdb5) section 2.2.1.4.4 (DR\_WRITE\_REQ).

The **Offset** field in this request MUST be set to 0.

Zero-length request semantics: The protocol allows the [**client**](#gt_60e0e1fa-66fe-41e1-b5e3-ceab97e53506) and [**server**](#gt_434b0234-e970-4e8c-bdfa-e16a30d96703) to request or to complete read/write operations with the **Length** field set to zero. The behavior of these requests and their interpretation is determined by the server application and the client driver.

#### Server Device Control Request (DR\_PORT\_CONTROL\_REQ)

This message is sent by the server to request a [**device control**](#gt_2db1698b-feac-437f-becc-236343578c8c) operation. The packet is specified in [[MS-RDPEFS]](%5bMS-RDPEFS%5d.pdf#Section_34d9de58b2b540b6b970f82d4603bdb5) section 2.2.1.4.5 (DR\_CONTROL\_REQ).

The possible values for the **IoControlCode** member and the corresponding Input and Output buffers applicable to parallel and serial ports are as defined in [[MSFT-WDDK]](https://go.microsoft.com/fwlink/?LinkID=507676), and in [[MSDN-PORTS]](https://go.microsoft.com/fwlink/?LinkId=102069).

Serial and Parallel IOCTL handles applicable to this protocol:

| Name | Value |
| --- | --- |
| IOCTL\_SERIAL\_SET\_BAUD\_RATE | 0x001B0004 |
| IOCTL\_SERIAL\_GET\_BAUD\_RATE | 0x001B0050 |
| IOCTL\_SERIAL\_SET\_LINE\_CONTROL | 0x001B000C |
| IOCTL\_SERIAL\_GET\_LINE\_CONTROL | 0x001B0054 |
| IOCTL\_SERIAL\_SET\_TIMEOUTS | 0x001B001C |
| IOCTL\_SERIAL\_GET\_TIMEOUTS | 0x001B0020 |
| IOCTL\_SERIAL\_SET\_CHARS | 0x001B0058 |
| IOCTL\_SERIAL\_GET\_CHARS | 0x001B005C |
| IOCTL\_SERIAL\_SET\_DTR | 0x001B0024 |
| IOCTL\_SERIAL\_CLR\_DTR | 0x001B0028 |
| IOCTL\_SERIAL\_RESET\_DEVICE | 0x001B002C |
| IOCTL\_SERIAL\_SET\_RTS | 0x001B0030 |
| IOCTL\_SERIAL\_CLR\_RTS | 0x001B0034 |
| IOCTL\_SERIAL\_SET\_XOFF | 0x001B0038 |
| IOCTL\_SERIAL\_SET\_XON | 0x001B003C |
| IOCTL\_SERIAL\_SET\_BREAK\_ON | 0x001B0010 |
| IOCTL\_SERIAL\_SET\_BREAK\_OFF | 0x001B0014 |
| IOCTL\_SERIAL\_SET\_QUEUE\_SIZE | 0x001B0008 |
| IOCTL\_SERIAL\_GET\_WAIT\_MASK | 0x001B0040 |
| IOCTL\_SERIAL\_SET\_WAIT\_MASK | 0x001B0044 |
| IOCTL\_SERIAL\_WAIT\_ON\_MASK | 0x001B0048 |
| IOCTL\_SERIAL\_IMMEDIATE\_CHAR | 0x001B0018 |
| IOCTL\_SERIAL\_PURGE | 0x001B004C |
| IOCTL\_SERIAL\_GET\_HANDFLOW | 0x001B0060 |
| IOCTL\_SERIAL\_SET\_HANDFLOW | 0x001B0064 |
| IOCTL\_SERIAL\_GET\_MODEMSTATUS | 0x001B0068 |
| IOCTL\_SERIAL\_GET\_DTRRTS | 0x001B0078 |
| IOCTL\_SERIAL\_GET\_COMMSTATUS | 0x001B006C |
| IOCTL\_SERIAL\_GET\_PROPERTIES | 0x001B0074 |
| IOCTL\_SERIAL\_XOFF\_COUNTER | 0x001B0070 |
| IOCTL\_SERIAL\_LSRMST\_INSERT | 0x001B007C |
| IOCTL\_SERIAL\_CONFIG\_SIZE | 0x001B0080 |
| IOCTL\_SERIAL\_GET\_STATS | 0x001B008C |
| IOCTL\_SERIAL\_CLEAR\_STATS | 0x001B0090 |
| IOCTL\_SERIAL\_GET\_MODEM\_CONTROL | 0x001B0094 |
| IOCTL\_SERIAL\_SET\_MODEM\_CONTROL | 0x001B0098 |
| IOCTL\_SERIAL\_SET\_FIFO\_CONTROL | 0x001B009C |
| IOCTL\_PAR\_QUERY\_INFORMATION | 0x00160004 |
| IOCTL\_PAR\_SET\_INFORMATION | 0x00160008 |
| IOCTL\_PAR\_QUERY\_DEVICE\_ID | 0x0016000C |
| IOCTL\_PAR\_QUERY\_DEVICE\_ID\_SIZE | 0x00160010 |
| IOCTL\_IEEE1284\_GET\_MODE | 0x00160014 |
| IOCTL\_IEEE1284\_NEGOTIATE | 0x00160018 |
| IOCTL\_PAR\_SET\_WRITE\_ADDRESS | 0x0016001C |
| IOCTL\_PAR\_SET\_READ\_ADDRESS | 0x00160020 |
| IOCTL\_PAR\_GET\_DEVICE\_CAPS | 0x00160024 |
| IOCTL\_PAR\_GET\_DEFAULT\_MODES | 0x00160028 |
| IOCTL\_PAR\_QUERY\_RAW\_DEVICE\_ID | 0x00160030 |
| IOCTL\_PAR\_IS\_PORT\_FREE | 0x00160054 |

#### Client Create Response (DR\_PORT\_CREATE\_RSP)

The client responds with this message to notify the [**server**](#gt_434b0234-e970-4e8c-bdfa-e16a30d96703) about the result of the server create request (section [2.2.2.2](#Section_fc7ca06b7e234dd48605b6f75c75efb2)). This message is specified in [[MS-RDPEFS]](%5bMS-RDPEFS%5d.pdf#Section_34d9de58b2b540b6b970f82d4603bdb5) section 2.2.1.5.1 (DR\_CREATE\_RSP).

#### Client Close Response (DR\_PORT\_CLOSE\_RSP)

The [**client**](#gt_60e0e1fa-66fe-41e1-b5e3-ceab97e53506) responds with this message to notify the [**server**](#gt_434b0234-e970-4e8c-bdfa-e16a30d96703) about the result of the server close request (section [2.2.2.3](#Section_561b2a288c8a4dd9a8102822480ca835)). This message is specified in [[MS-RDPEFS]](%5bMS-RDPEFS%5d.pdf#Section_34d9de58b2b540b6b970f82d4603bdb5) section 2.2.1.5.2 (DR\_CLOSE\_RSP).

#### Client Read Response (DR\_PORT\_READ\_RSP)

The client responds with this message to notify the server about the result of server read request (section [2.2.2.4](#Section_fa44247afe61480a8465d14a41370bf9)). This message is specified in [[MS-RDPEFS]](%5bMS-RDPEFS%5d.pdf#Section_34d9de58b2b540b6b970f82d4603bdb5) section 2.2.1.5.3 (DR\_READ\_RSP).

#### Client Write Response (DR\_PORT\_WRITE\_RSP)

The client responds with this message to notify the server about the result of the server write request (section [2.2.2.5](#Section_38bf440fe055416c8e4c3ac2d59cb110)). This message is specified in [[MS-RDPEFS]](%5bMS-RDPEFS%5d.pdf#Section_34d9de58b2b540b6b970f82d4603bdb5) section 2.2.1.5.4 (DR\_WRITE\_RSP).

#### Client Device Control Response (DR\_PORT\_CONTROL\_RSP)

The client responds with this message to notify the server about the result of the server device control request ([2.2.2.6](#Section_824cc217febd4a2ea95f5b5bdc32848f)). This message is specified in [[MS-RDPEFS]](%5bMS-RDPEFS%5d.pdf#Section_34d9de58b2b540b6b970f82d4603bdb5) section 2.2.1.5.5 (DR\_CONTROL\_RSP).

# Protocol Details

The following sections specify protocol details, including abstract data models and message processing rules.

## Common Details

### Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The organization is provided to explain 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.

The Remote Desktop Protocol: Serial and Parallel Port Virtual Channel Extension follows the abstract data model specified in [[MS-RDPEFS]](%5bMS-RDPEFS%5d.pdf#Section_34d9de58b2b540b6b970f82d4603bdb5) section 3.1.1.

### Timers

There are no common timers.

### Initialization

The Remote Desktop Protocol: File System Virtual Channel Extension MUST be initialized before the ports can be redirected. This initialization is complete when the client sends the Client Core Capability Response, as specified in [[MS-RDPEFS]](%5bMS-RDPEFS%5d.pdf#Section_34d9de58b2b540b6b970f82d4603bdb5) section 2.2.2.8.

### Higher-Layer Triggered Events

IO requests are generated in response to IO calls that the server applications make on the redirected device. Otherwise, no higher-layer triggered events are used.

### Message Processing Events and Sequencing Rules

The common message processing events and rules that are described in [[MS-RDPEFS]](%5bMS-RDPEFS%5d.pdf#Section_34d9de58b2b540b6b970f82d4603bdb5) section 3.1.5 apply to this protocol. For client-specific and server-specific message processing, see sections [3.2.5](#Section_6841899ee7364b7fb2f91729e1ecf07d) and [3.3.5](#Section_064a26a3c2b142f4aa54525ad80034b6).

### Timer Events

No common timer events are used.

### Other Local Events

There are no common local events.

## Client Details

### Abstract Data Model

The abstract data model is specified in section [3.1.1](#Section_e2c667da24294df29630250a4bef4f67).

### Timers

No timers are used.

### Initialization

Initialization is specified in section [3.1.3](#Section_d861abb109594a08b611f5e336ab4b1f).

### Higher-Layer Triggered Events

No [**client**](#gt_60e0e1fa-66fe-41e1-b5e3-ceab97e53506) higher-layer triggered events are used.

### Message Processing Events and Sequencing Rules

#### Port Redirection Messages

##### Sending a Client Device List Announce Request Message

After Remote Desktop Protocol: File System Virtual Channel Extension finishes initialization by sending a Client Core Capability Response message, as specified in [[MS-RDPEFS]](%5bMS-RDPEFS%5d.pdf#Section_34d9de58b2b540b6b970f82d4603bdb5) section 2.2.2.8, the [**client**](#gt_60e0e1fa-66fe-41e1-b5e3-ceab97e53506) sends a DR\_CORE\_DEVICELIST\_ANNOUNCE\_REQ message to the [**server**](#gt_434b0234-e970-4e8c-bdfa-e16a30d96703) along with information for various devices, as specified in [MS-RDPEFS] section 2.2.2.9. The device type is set to RDPDR\_TYP\_SERIAL for serial port type devices and RDPDR\_TYP\_PARALLEL for parallel ports. The port extension prepares the port devices information that goes into this packet. The port-specific structure is specified in section [2.2.2.1](#Section_535df5533d5c4666b33332a0d3d3cab4).

The port redirection extension enumerates the local serial and parallel ports that need to be redirected. It MUST set the appropriate **PreferredDosName** fields, generate unique IDs for the devices, set the appropriate device types, and let the Remote Desktop Protocol: File System Virtual Channel Extension send the information over to the server.

The **DeviceId** field generated in this message is used to refer to this port in subsequent messages.

##### Processing a Server Create Request Message

After receiving the create request, the [**client**](#gt_60e0e1fa-66fe-41e1-b5e3-ceab97e53506) SHOULD open and prepare the port for IO operations. The opened instance of the port is maintained by the client as a **FileId** field of the DR\_DEVICE\_IOREQUEST message, as specified in [[MS-RDPEFS]](%5bMS-RDPEFS%5d.pdf#Section_34d9de58b2b540b6b970f82d4603bdb5) section 2.2.1.4. This ID is used to refer to subsequent IO operations on the port instance. After the create request processing is complete, the client responds with a create response message (section [3.2.5.1.7](#Section_bd93a9e3f14f483796bb95982c52b98d)).

##### Processing a Server Close Request Message

The **DeviceId** and **FileId** fields of DR\_DEVICE\_IOREQUEST (as specified in [[MS-RDPEFS]](%5bMS-RDPEFS%5d.pdf#Section_34d9de58b2b540b6b970f82d4603bdb5) section 2.2.1.4) identify the previously-opened instance of the port to operate on.

After receiving the close request, the [**client**](#gt_60e0e1fa-66fe-41e1-b5e3-ceab97e53506) MUST close the previously opened port instance. It MUST also cancel pending IO operations, if any, on the port. The client MUST respond with close response message (section [3.2.5.1.8](#Section_37dbfa6cd54e43389dd9167cbb5ec653)).

##### Processing a Server Read Request Message

The **DeviceId** and **FileId** fields of DR\_DEVICE\_IOREQUEST (as specified in [[MS-RDPEFS]](%5bMS-RDPEFS%5d.pdf#Section_34d9de58b2b540b6b970f82d4603bdb5) section 2.2.1.4) identify the previously opened instance of the port to operate on.

The **Length** and **Offset** fields of DR\_READ\_REQ (as specified in [MS-RDPEFS] section 2.2.1.4.3) identify the length and offset values requested by the [**server**](#gt_434b0234-e970-4e8c-bdfa-e16a30d96703) for this operation.

If the **Offset** field is not set to 0, the value MUST be ignored.

The client MUST perform the read operation on the instance of the port accordingly. After the read operation is complete, the client MUST send a read response message (section [3.2.5.1.9](#Section_4936ec4f01884424aa69b1dcdcad09b6)) to let the server know about the result of the operation.

The semantics of read requests are determined by the client-side driver. The protocol allows partial read results. The result of the read operations, including the data read, is passed to the server and is considered opaque to the protocol.

##### Processing a Server Write Request Message

The **DeviceId** and **FileId** fields of DR\_DEVICE\_IOREQUEST (as specified in [[MS-RDPEFS]](%5bMS-RDPEFS%5d.pdf#Section_34d9de58b2b540b6b970f82d4603bdb5) section 2.2.1.4) identify the previously opened instance of the port to operate on.

The **Length** and **WriteData** fields of DR\_WRITE\_REQ (as specified in [MS-RDPEFS] section 2.2.1.4.4) identify the parameters for the write operation. The **Offset** field is ignored.

The [**client**](#gt_60e0e1fa-66fe-41e1-b5e3-ceab97e53506) MUST perform the write operation on the instance of the port accordingly. After the operation is complete, the client MUST send response message DR\_WRITE\_RSP (as specified in [MS-RDPEFS] section 2.2.1.5.4) to let the [**server**](#gt_434b0234-e970-4e8c-bdfa-e16a30d96703) know about the result of the operation.

##### Processing a Server Device Control Request Message

The **DeviceId** and **FileId** fields of DR\_DEVICE\_IOREQUEST (as specified in [[MS-RDPEFS]](%5bMS-RDPEFS%5d.pdf#Section_34d9de58b2b540b6b970f82d4603bdb5) section 2.2.1.4) identify the previously-opened instance of the port to operate on.

The **OutputBufferLength**, **InputBufferLength**, **IoControlCode** and **InputBuffer** fields of DR\_CONTROL\_REQ (as specified in [MS-RDPEFS] section 2.2.1.4.5) identify the parameters for the [**device control**](#gt_2db1698b-feac-437f-becc-236343578c8c) operation.

The client MUST perform the device control operation specified by the **IoControlCode** field on the instance of the port accordingly. After the operation is complete, the client MUST send a Device Control Response message (section [3.2.5.1.11](#Section_22c62065fe954729aa0378a69d6a0f7f)) to notify the server about the result of the operation.[<2>](#Appendix_A_2" \o "Product behavior note 2)

##### Sending a Create Response Message

This message is sent in response to the server create request (section [3.2.5.1.2](#Section_777a5a06cb5849b1b899745e994b2e75)).

The client MUST fill out various fields of DR\_CREATE\_RSP (as specified in [[MS-RDPEFS]](%5bMS-RDPEFS%5d.pdf#Section_34d9de58b2b540b6b970f82d4603bdb5) section 2.2.1.5.1) as follows:

For the RDPDR\_HEADER header:

* The **Component** field MUST be set to RDPDR\_CTYP\_CORE.
* The **PacketId** field MUST be set to PAKID\_CORE\_DEVICE\_IOCOMPLETION.

For the DR\_DEVICE\_IOCOMPLETION header:

* The **DeviceId** field MUST be set to match the corresponding **DeviceId** field from the IO request.
* The **CompletionId** field MUST be set to match the **CompletionId** field from the corresponding IO request (section 3.2.5.1.2).
* The **IoStatus** field MUST be set to the NTSTATUS indicating the result of the operation. NTSTATUS codes are specified in [[MS-ERREF]](%5bMS-ERREF%5d.pdf#Section_1bc92ddfb79e413cbbaa99a5281a6c90) section 2.3.

After completing the create request, the client MUST set the **FileId** field to a unique **FileId** value to identify the instance of the port. This **FileId** field is used in subsequent IO operations to refer to the port instance.

The **Information** field MUST be set to 0.

##### Sending a Close Response Message

This message is sent in response to the server close request ([3.2.5.1.3](#Section_ec816dc7bc434947a00fbe66248ef269)).

The [**client**](#gt_60e0e1fa-66fe-41e1-b5e3-ceab97e53506) MUST fill out the various members of DR\_CLOSE\_RSP (as specified in [[MS-RDPEFS]](%5bMS-RDPEFS%5d.pdf#Section_34d9de58b2b540b6b970f82d4603bdb5) section 2.2.1.5.2) as follows:

For the RDPDR\_HEADER:

* The **Component** field MUST be set to RDPDR\_CTYP\_CORE.
* The **PacketId** field MUST be set to PAKID\_CORE\_DEVICE\_IOCOMPLETION.

For the DR\_DEVICE\_IOCOMPLETION:

* The **DeviceId** field MUST be set to match the corresponding **DeviceId** field from the IO request.
* The **CompletionId** field MUST be set to match the **CompletionId** field from the corresponding IO request (section 3.2.5.1.3).
* The **IoStatus** field MUST be set to the NTSTATUS value indicating the result of the operation.

##### Sending a Read Response Message

This message is sent in response to the server read request ([3.2.5.1.4](#Section_74a13a3a98344fe191786ed7ce775098)).

The [**client**](#gt_60e0e1fa-66fe-41e1-b5e3-ceab97e53506) MUST fill out the various members of DR\_READ\_RSP (as specified in [[MS-RDPEFS]](%5bMS-RDPEFS%5d.pdf#Section_34d9de58b2b540b6b970f82d4603bdb5) section 2.2.1.5.3) as follows:

For the RDPDR\_HEADER:

* The **Component** field MUST be set to RDPDR\_CTYP\_CORE.
* The **PacketId** field MUST be set to PAKID\_CORE\_DEVICE\_IOCOMPLETION.

For the DR\_DEVICE\_IOCOMPLETION:

* The **DeviceId** field MUST be set to match the corresponding **DeviceId** field from the IO request.
* The **CompletionId** field MUST be set to match the **CompletionId** from the corresponding IO request (section 3.2.5.1.4).
* The **IoStatus** field MUST be set to the NTSTATUS value indicating the result of the operation.

The client prepares a reply message with the result of the read operation. The client populates the **Length** field with the number of bytes read. The actual data read follows the **Length** field. The **Length** field MAY be less than the requested length; however, the **Length** field MUST NOT be greater than requested length. These partial read requests are supported by the server.

##### Sending a Write Response Message

This message is sent in response to the server write request (section [3.2.5.1.5](#Section_b0ee05af912e4ceb99c9ae82a1747cf6)).

The client MUST fill out the various members of DR\_WRITE\_RSP (as specified in [[MS-RDPEFS]](%5bMS-RDPEFS%5d.pdf#Section_34d9de58b2b540b6b970f82d4603bdb5) section 2.2.1.5.4) as follows:

For the RDPDR\_HEADER:

* The **Component** field MUST be set to RDPDR\_CTYP\_CORE.
* The **PacketId** field MUST be set to PAKID\_CORE\_DEVICE\_IOCOMPLETION.

For the DR\_DEVICE\_IOCOMPLETION:

* The **DeviceId** field MUST be set to match the corresponding **DeviceId** field from the IO request.
* The **CompletionId** field MUST be set to match the **CompletionId** field from the corresponding IO request (section 3.2.5.1.5).
* The **IoStatus** field MUST be set to the NTSTATUS indicating the result of the operation.
* The **Padding** field is unused and MUST be ignored by the server.

The client prepares a reply message with the result of the write operation. The client populates the **Length** field with the number of bytes written.

##### Sending a Device Control Response Message

This message is sent in response to the server device control request (section [3.2.5.1.6](#Section_cfdc5c7626574e3fb6986e35ff6a83d6)).

The [**client**](#gt_60e0e1fa-66fe-41e1-b5e3-ceab97e53506) MUST fill out the various members of DR\_CONTROL\_RSP (as specified in [[MS-RDPEFS]](%5bMS-RDPEFS%5d.pdf#Section_34d9de58b2b540b6b970f82d4603bdb5) section 2.2.1.5.5) as follows:

For the RDPDR\_HEADER:

* The **Component** field MUST be set to RDPDR\_CTYP\_CORE.
* The **PacketId** field MUST be set to PAKID\_CORE\_DEVICE\_IOCOMPLETION.

For the DR\_DEVICE\_IOCOMPLETION:

* The **DeviceId** field MUST be set to match the corresponding **DeviceId** field from the IO request.
* The **CompletionId** field MUST be set to match the **CompletionId** field from the corresponding IO request (section 3.2.5.1.6).
* The **IoStatus** field MUST be set to the NTSTATUS indicating the result of the operation.

The client prepares a reply message with the result of the [**device control**](#gt_2db1698b-feac-437f-becc-236343578c8c) operation. The client populates the **OutputBufferLength** field with the number of bytes returned by the device control operation. The actual data returned, if any, follows the packet.

### Timer Events

There are no timer events.

### Other Local Events

There are no other local events.

## Server Details

### Abstract Data Model

The abstract data model is specified in section [3.1.1](#Section_e2c667da24294df29630250a4bef4f67).

### Timers

There are no timers used.

### Initialization

Initialization is specified in section [3.1.3](#Section_d861abb109594a08b611f5e336ab4b1f).

### Higher-Layer Triggered Events

All IO requests are originated by the applications making IO calls on the redirected device. Otherwise, no other higher-layer triggered events are used.

### Message Processing Events and Sequencing Rules

#### Port Redirection Messages

##### Processing a Client Device List Announce Request Message

After receiving the Client Device List Announce Request message (section [2.2.2.1](#Section_535df5533d5c4666b33332a0d3d3cab4)) for each device on the announce list, the [**server**](#gt_434b0234-e970-4e8c-bdfa-e16a30d96703) MUST create a [**pseudo port device**](#gt_081a3797-49a5-4760-bb9c-730309c18822) that emulates the [**client**](#gt_60e0e1fa-66fe-41e1-b5e3-ceab97e53506) device. The server MUST maintain the association of the **DeviceId** value obtained from the client with such a pseudo device.

##### Sending a Server Create Request Message

The [**server**](#gt_434b0234-e970-4e8c-bdfa-e16a30d96703) sends this message when any server application opens the [**pseudo port device**](#gt_081a3797-49a5-4760-bb9c-730309c18822). The server passes all the create parameters obtained from the application request over to the client for actual processing.

##### Sending a Server Close Request Message

The server sends this message to the client in response to the server application requesting the close operation on the [**pseudo port device**](#gt_081a3797-49a5-4760-bb9c-730309c18822) for actual processing.

##### Sending a Server Write Request Message

The server sends this message to the client in response to the server application requesting the write operation on the [**pseudo port device**](#gt_081a3797-49a5-4760-bb9c-730309c18822) for actual processing.

##### Sending a Server Read Request Message

The server sends this message to the client in response to the server application requesting the read operation on the [**pseudo port device**](#gt_081a3797-49a5-4760-bb9c-730309c18822) for actual processing.

##### Sending a Server Device Control Request Message

The server sends this message to the client in response to the server application requesting the [**device control**](#gt_2db1698b-feac-437f-becc-236343578c8c) operation on the [**pseudo port device**](#gt_081a3797-49a5-4760-bb9c-730309c18822) for actual processing.

##### Processing a Client Create Response Message

After receiving the create response, the [**server**](#gt_434b0234-e970-4e8c-bdfa-e16a30d96703) responds to the application that initiated the create operation (section [3.3.5.1.1](#Section_38ba503d728646ae8421b45fc5a35506)). The server MUST maintain an association between the **FileId** returned by the [**client**](#gt_60e0e1fa-66fe-41e1-b5e3-ceab97e53506) and the file handle returned to the application. For any subsequent IO operations on the file handle, the server SHOULD send the IO to the client for completion using the same **FileId** field.

##### Processing a Client Close Response Message

The [**server**](#gt_434b0234-e970-4e8c-bdfa-e16a30d96703) responds to the application with the result of the close response received from the [**client**](#gt_60e0e1fa-66fe-41e1-b5e3-ceab97e53506).

##### Processing a Client Write Response Message

The server forwards the result of the write response to the application that requested the write operation.

##### Processing a Client Read Response Message

The server forwards the result of the read response to the application that initiated the read operation.

##### Processing a Client Device Control Response Message

The server forwards the result of the device control response to the application that initiated the operation.

### Timer Events

There are no timer events.

### Other Local Events

There are no other local events.

# Protocol Examples

## Port Redirection Annotations

After the Remote Desktop Protocol: Serial and Parallel Port Virtual Channel Extension finishes its initialization, the [**client**](#gt_60e0e1fa-66fe-41e1-b5e3-ceab97e53506) sends the client device list announce message to the [**server**](#gt_434b0234-e970-4e8c-bdfa-e16a30d96703). Information about the ports to be redirected is contained within the message.

The following sequence shows a COM2 port being sent to the server for redirection.

1. Channel Name = RDPDR, 28, Client to server
2. 00000000 72 44 41 44 01 00 00 00 01 00 00 00 01 00 00 00
3. 00000010 43 4f 4d 32 00 00 00 00 00 00 00 00
4. 72 44 -> RDPDR\_CTYP\_CORE = 0x4472
5. 41 44 -> PAKID\_CORE\_DEVICELIST\_ANNOUNCE = 0x4441
6. 01 00 00 00 -> DeviceCount = 0x00000001
7. 01 00 00 00 -> DeviceType (RDPDR\_DTYP\_SERIAL) = 0x00000001
8. 01 00 00 00 -> DeviceId = 0x00000001
9. 43 4f 4d 32 00 00 00 00 -> PreferredDosName(8 characters) = "COM2"
10. 00 00 00 00 -> DeviceDataLength = 0x00000000

The following sequence shows an LPT1 port being sent for redirection. In this example, the Device Announce packet contains three devices. This example only annotates the header and LPT1 Device portion from this packet.

1. Channel Name = RDPDR, 264, Client to server
2. 00000000 72 44 41 44 03 00 00 00 04 00 00 00 04 00 00 00
3. 00000010 50 52 4e 34 00 00 00 00 50 00 00 00 10 00 00 00
4. 00000020 00 00 00 00 00 00 00 00 1c 00 00 00 1c 00 00 00
5. 00000030 00 00 00 00 41 00 70 00 6f 00 6c 00 6c 00 6f 00
6. 00000040 20 00 50 00 2d 00 31 00 32 00 30 00 30 00 00 00
7. 00000050 41 00 70 00 6f 00 6c 00 6c 00 6f 00 20 00 50 00
8. 00000060 2d 00 31 00 32 00 30 00 30 00 00 00 04 00 00 00
9. 00000070 03 00 00 00 50 52 4e 33 00 00 00 00 74 00 00 00
10. 00000080 12 00 00 00 00 00 00 00 00 00 00 00 2e 00 00 00
11. 00000090 2e 00 00 00 00 00 00 00 43 00 61 00 6e 00 6f 00
12. 000000a0 6e 00 20 00 42 00 75 00 62 00 62 00 6c 00 65 00
13. 000000b0 2d 00 4a 00 65 00 74 00 20 00 42 00 4a 00 2d 00
14. 000000c0 33 00 30 00 00 00 43 00 61 00 6e 00 6f 00 6e 00
15. 000000d0 20 00 42 00 75 00 62 00 62 00 6c 00 65 00 2d 00
16. 000000e0 4a 00 65 00 74 00 20 00 42 00 4a 00 2d 00 33 00
17. 000000f0 30 00 00 00 02 00 00 00 02 00 00 00 4c 50 54 31
18. 00000100 00 00 00 00 00 00 00 00
19. 72 44 -> RDPDR\_CTYP\_CORE = 0x4472
20. 41 44 -> PAKID\_CORE\_DEVICELIST\_ANNOUNCE = 0x4441
21. 03 00 00 00 -> DeviceCount = 0x00000003
22. 04 00 00 -> DeviceType (RDPDR\_DTYP\_PRINT) = 0x00000004
23. 04 00 00 00 -> DeviceId = 0x00000004
24. 50 52 4e 34 00 00 00 00 -> PreferredDosName (8 characters) = "PRN4"
25. 50 00 00 00 -> DeviceDataLength = 0x00000050
26. 10 00 00 00 -> Flags( RDPDR\_PRINTER\_ANNOUNCE\_FLAG\_XPSFORMAT)
27. = 0x00000010
28. 00 00 00 00 -> CodePage = 0x00000000
29. 00 00 00 00 -> PnPNameLen = 0x00000000
30. 1c 00 00 00 -> DriverNameLen = 0x0000001c
31. 1c 00 00 00 -> PrintNameLen (28 bytes) = 0x0000001c
32. 00 00 00 00 -> CachedFieldsLen = 0x00000000
33. 41 00 70 00 6f 00 6c 00
34. 6c 00 6f 00 20 00 50 00
35. 2d 00 31 00 32 00 30 00
36. 30 00 00 00 -> DriverName (28 bytes) = "Apollo P-1200"
37. 41 00 70 00 6f 00 6c 00 6c 00 6f 00 20 00 50 00
38. 2d 00 31 00 32 00 30 00 30 00 00 00
39. -> PrintName (28 bytes) = "Apollo P-1200"
40. 04 00 00 00 -> DeviceType (RDPDR\_DTYP\_PRINT) = 0x00000004
41. 03 00 00 00 -> DeviceId = 0x00000003
42. 50 52 4e 33 00 00 00 00 -> PreferredDosName (8 characters) = "PRN3"
43. 74 00 00 00 -> DeviceDataLength = 0x00000074
44. 12 00 00 00 -> Flags(RDPDR\_PRINTER\_ANNOUNCE\_FLAG\_XPSFORMAT |
45. RDPDR\_PRINTER\_ANNOUNCE\_FLAG\_DEFAULTPRINTER)
46. = 0x00000012
47. 00 00 00 0 -> CodePage = 0x00000000
48. 00 00 00 00 -> PnPNameLen = 0x00000000
49. 2e 00 00 00 -> DriverNameLen = 0x0000002e
50. 2e 00 00 00 -> PrintNameLen = 0x0000002e
51. 00 00 00 00 -> CachedFieldsLen = 0x00000000
52. 43 00 61 00 6e 00 6f 00
53. 6e 00 20 00 42 00 75 00
54. 62 00 62 00 6c 00 65 00
55. 2d 00 4a 00 65 00 74 00
56. 20 00 42 00 4a 00 2d 00
57. 33 00 30 00 00 00 -> DriverName (46 bytes) = "Canon Bubble-Jet BJ-30"
58. 43 00 61 00 6e 00 6f 00
59. 6e 00 20 00 42 00 75 00
60. 62 00 62 00 6c 00 65 00
61. 2d 00 4a 00 65 00 74 00
62. 20 00 42 00 4a 00 2d 00
63. 33 00 30 00 00 00 -> PrintName (46 bytes) = "Canon Bubble-Jet BJ-30"
64. 02 00 00 00 -> DeviceType (RDPDR\_DTYP\_PARALLEL) = 0x00000002
65. 02 00 00 00 -> DeviceId = 0x00000002
66. 4c 50 54 31 00 00 00 00 -> PreferredDosName (8 characters) = "LPT1"
67. 00 00 00 00 -> DeviceDataLength = 0x00000000

## Server Create Request Example

The server sends a request to create an instance of the port. The following sequence captures such a request.

1. RDPDR, 56, Server to client
2. 00000000 72 44 52 49 01 00 00 00 00 00 00 00 01 00 00 00
3. 00000010 00 00 00 00 00 00 00 00 80 00 10 00 00 00 00 00
4. 00000020 00 00 00 00 00 00 00 00 07 00 00 00 01 00 00 00
5. 00000030 60 00 00 00 00 00 00 00
6. 72 44 -> RDPDR\_CTYP\_CORE = 0x4472
7. 52 49 -> PAKID\_CORE\_DEVICE\_IOREQUEST = 0x4952
8. 01 00 00 00 -> DeviceId = 0x00000001
9. 00 00 00 00 -> FileId = 0x00000000
10. 01 00 00 00 -> CompletionId = 0x00000001
11. 00 00 00 00 -> MajorFunction(IRP\_MJ\_CREATE) = 0x00000000
12. 00 00 00 00 -> MinorFunction = 0x00000000
13. 80 00 10 00 -> DesiredAccess = 0x00100080
14. 00 00 00 00 00 00 00 00 -> AllocationSize(64 bits) = 0x0
15. 00 00 00 00 -> FileAttributes = 0x00000000
16. 07 00 00 00 -> SharedAccess = 0x00000007
17. 01 00 00 00 -> CreateDisposition = 0x00000001
18. 60 00 00 00 -> CreateOptions = 0x00000060
19. 00 00 00 00 -> PathLength = 0x0000000

## Client Create Response Example

The client responds with the following response. This establishes a **FileId** value that is used subsequently for the IO operations.

1. RDPDR, 21, Client to server
2. 00000000 72 44 43 49 02 00 00 00 00 00 00 00 00 00 00 00
3. 00000010 01 00 00 00 00
4. 72 44 -> RDPDR\_CTYP\_CORE = 0x4472
5. 43 49 -> PAKID\_CORE\_DEVICE\_IOCOMPLETION = 0x4943
6. 02 00 00 00 -> DeviceId = 0x00000001
7. 00 00 00 00 -> CompletionId = 0x00000001
8. 00 00 00 00 -> NTSTATUS = 0x00000000
9. 01 00 00 00 -> FileId = 0x00000001
10. 00 -> Information = 0x00

## IO Operations Examples

The server on behalf of the application sends IO operations on the **FieldId**. The following message sequences illustrate the packets for read, write, and [**device control**](#gt_2db1698b-feac-437f-becc-236343578c8c) operations.

Server read request (DR\_PORT\_READ\_REQ)

1. RDPDR, 56, Server to client
2. 00000000 72 44 52 49 02 00 00 00 02 00 00 00 00 00 00 00
3. 00000010 03 00 00 00 00 00 00 00 08 02 00 00 00 00 00 00
4. 00000020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
5. 00000030 00 00 00 00 00 00 00 00
6. 72 44 -> RDPDR\_CTYP\_CORE = 0x4472
7. 52 49 -> PAKID\_CORE\_DEVICE\_IOREQUEST = 0x4952
8. 02 00 00 00 -> DeviceId = 0x00000002
9. 02 00 00 00 -> FileId = 0x00000002
10. 00 00 00 00 -> CompletionId = 0x00000000
11. 03 00 00 00 -> MajorFunction (IRP\_MJ\_READ) = 0x00000003
12. 00 00 00 00 -> MinorFunction = 0x00000000
13. 08 02 00 00 -> Length = 0x00000208
14. 00 00 00 00 00 00 00 00
15. -> Offset
16. 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
17. -> Padding (20 bytes)

Server write request (DR\_PORT\_WRITE\_REQ)

1. RDPDR, 576, Server to client
2. 00000000 72 44 52 49 01 00 00 00 02 00 00 00 00 00 00 00
3. 00000010 04 00 00 00 00 00 00 00 08 02 00 00 00 00 00 00
4. 00000020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
5. 00000030 00 00 00 00 00 00 00 00 29 00 00 00 c4 00 7a 00 ...
6. 72 44 -> RDPDR\_CTYP\_CORE = 0x4472
7. 52 49 -> PAKID\_CORE\_DEVICE\_IOREQUEST = 0x4952
8. 01 00 00 00 -> DeviceId = 0x00000001
9. 02 00 00 00 -> FileId = 0x00000002
10. 00 00 00 00 -> CompletionId = 0x00000000
11. 04 00 00 00 -> MajorFunction (IRP\_MJ\_WRITE) = 0x00000004
12. 00 00 00 00 -> MinorFunction = 0x00000000
13. 08 02 00 00 -> Length (520 bytes written) = 0x00000208
14. 00 00 00 00 00 00 00 00
15. -> Offset
16. 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
17. -> Padding (20 bytes)
18. 00 00 ... -> WriteData (520 bytes)

Server device control request (DR\_PORT\_CONTROL\_REQ)

1. RDPDR, 56, Server to client
2. 00000000 72 44 52 49 02 00 00 00 02 00 00 00 00 00 00 00
3. 00000010 0E 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00
4. 00000020 50 00 1b 00 00 00 00 00 00 00 00 00 00 00 00 00
5. 00000030 00 00 00 00 00 00 00 00
6. 72 44 -> RDPDR\_CTYP\_CORE = 0x4472
7. 52 49 -> PAKID\_CORE\_DEVICE\_IOREQUEST = 0x4952
8. 02 00 00 00 -> DeviceId = 0x00000002
9. 02 00 00 00 -> FileId = 0x00000002
10. 00 00 00 00 -> CompletionId = 0x00000000
11. 0E 00 00 00 -> MajorFunction (IRP\_MJ\_DEVICE\_CONTROL) = 0x0000000E
12. 00 00 00 00 -> MinorFunction = 0x00000000
13. 04 00 00 00 -> OutputBufferLength = 0x00000004
14. 00 00 00 00 -> InputBufferLength = 0x00000000
15. 50 00 1b 00 -> IoControlCode = 0x001B0050
16. 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
17. -> Padding (20 bytes)

The following packets show the typical responses from the client for these requests.

Client read response (DR\_PORT\_READ\_RSP)

1. RDPDR, 540, Client to server
2. 00000000 72 44 43 49 02 00 00 00 00 00 00 00 00 00 00 00
3. 00000010 08 02 00 00 29 00 00 00 c4 00 7a 00 4c 4f 43 41 ...
4. 72 44 -> RDPDR\_CTYP\_CORE = 0x4472
5. 43 49 -> PAKID\_CORE\_DEVICE\_IOCOMPLETION = 0x4943
6. 02 00 00 00 -> DeviceId = 0x00000001
7. 00 00 00 00 -> CompletionId = 0x00000001
8. 00 00 00 00 -> NTSTATUS = 0x00000000
9. 08 02 00 00 -> Length = 0x00000208
10. 29 00 ... -> ReadData (520 bytes)

Client write response (DR\_PORT\_WRITE\_RSP)

1. RDPDR,21,send
2. 00000000 72 44 43 49 01 00 00 00 00 00 00 00 00 00 00 00
3. 00000010 08 02 00 00 00
4. 72 44 -> RDPDR\_CTYP\_CORE = 0x4472
5. 43 49 -> PAKID\_CORE\_DEVICE\_IOCOMPLETION = 0x4943
6. 01 00 00 00 -> DeviceId = 0x00000001
7. 00 00 00 00 -> CompletionId = 0x00000001
8. 00 00 00 00 -> NTSTATUS = 0x00000000
9. 08 02 00 00 -> Length
10. 00 -> Padding

Client device control response (DR\_PORT\_CONTROL\_RSP)

1. RDPDR,24,send
2. 00000000 72 44 43 49 02 00 00 00 00 00 00 00 00 00 00 00
3. 00000010 04 00 00 00 80 25 00 00
4. 72 44 -> RDPDR\_CTYP\_CORE = 0x4472
5. 43 49 -> PAKID\_CORE\_DEVICE\_IOCOMPLETION = 0x4943
6. 02 00 00 00 -> DeviceId = 0x00000002
7. 00 00 00 00 -> CompletionId = 0x00000000
8. 00 00 00 00 -> IoStatus = 0x00000000
9. 04 00 00 00 -> OutputBufferLength = 0x00000004
10. 80 25 00 00 -> OutputBuffer = 0x00002580

Finally, the server calls a close request to close the port instance.

Server close request (DR\_PORT\_CLOSE\_REQ)

1. RDPDR, 56, Server to client
2. 00000000 72 44 52 49 01 00 00 00 01 00 00 00 01 00 00 00
3. 00000010 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
4. 00000020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
5. 00000030 00 00 00 00 00 00 00 00
6. 72 44 -> RDPDR\_CTYP\_CORE = 0x4472
7. 52 49 -> PAKID\_CORE\_DEVICE\_IOREQUEST = 0x4952
8. 01 00 00 00 -> DeviceId = 0x00000001
9. 01 00 00 00 -> FileId = 0x00000001
10. 01 00 00 00 -> CompletionId = 0x00000001
11. 02 00 00 00 -> MajorFunction (IRP\_MJ\_CLOSE) = 0x00000002
12. 00 00 00 00 -> MinorFunction = 0x00000000
13. 00 00 ... -> Padding (32 bytes)

After closing the local port instance, the client responds with this message.

Client close response (DR\_PORT\_CLOSE\_RSP)

1. RDPDR, 20, Client to server
2. 00000000 72 44 43 49 02 00 00 00 00 00 00 00 00 00 00 00
3. 00000010 00 00 00 00
4. 72 44 -> RDPDR\_CTYP\_CORE = 0x4472
5. 43 49 -> PAKID\_CORE\_DEVICE\_IOCOMPLETION = 0x4943
6. 02 00 00 00 -> DeviceId = 0x00000001
7. 00 00 00 00 -> CompletionId = 0x00000001
8. 00 00 00 00 -> NTSTATUS = 0x00000000
9. 00 00 00 00 -> Padding

# Security

## Security Considerations for Implementers

There are no security considerations for Remote Desktop Protocol: Serial and Parallel Port Virtual Channel Extension messages because all static virtual channel traffic is secured by the underlying Remote Desktop Protocol core protocol. The implemented security-related mechanisms are specified in [[MS-RDPBCGR]](%5bMS-RDPBCGR%5d.pdf#Section_5073f4ed1e9345e1b0396e30c385867c) section 5.

## Index of Security Parameters

There are no security parameters in Remote Desktop Protocol: Serial and Parallel Port Virtual Channel Extension.

# 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.

* Windows 2000 operating system
* Windows XP operating system
* Windows Server 2003 operating system
* 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 operating system
* Windows Server operating system

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.

[<1> Section 2.2.2.1](#Appendix_A_Target_1): Windows uses symbolic link names with standard naming conventions in its server implementations, in order to ensure compatibility with most Windows applications and drivers that access ports.

For serial ports, Windows assigns a symbolic link name that uses the standard naming convention "COM<n>", where <n> is the COM port number.

For parallel ports, Windows assigns a symbolic link name that uses the standard naming convention "LPT<n>", where <n> is the LPT port number.

For more information on port naming conventions, see [[MSDN-PORTS]](https://go.microsoft.com/fwlink/?LinkId=102069).

[<2> Section 3.2.5.1.6](#Appendix_A_Target_2): Windows Implementations use IOCTL constants for **IoControlCode** values. The content and values of the IOCTLs are opaque to the protocol. On the server side, the data contained in an IOCTL is simply packaged and sent to the client side. For maximum compatibility between the different versions of the Windows operating system, the client implementation only singles out critical IOCTLs and invokes the applicable Win32 port API. The other IOCTLS are passed directly to the client-side driver, and the processing of this value depends on the drivers installed on the client side. The values and parameters for these IOCTLS can be found in [[MSFT-WDDK]](https://go.microsoft.com/fwlink/?LinkID=507676) and in [MSDN-PORTS].

The parameters of the device request/response are mapped to the IOCTL as described in [MSDN-PORTS] as follows:

* If an IOCTL requires input, then the AssociatedIrp.SystemBuffer is set to the content of **InputBuffer** field and the **InputBufferSize** field is assigned to Parameters.DeviceIoControl.InputBufferLength.
* If an IOCTL requires output, then the Parameters.DeviceIoControl.OutputBufferLength is filled with the value of **OutputBufferLength** of the request. The **OutputBuffer** field in the response contains the data from AssociatedIrp.SystemBuffer, and the **OutputBufferLength** in the response is set to the returned value in the **Information** field of the I/O Status Block.
* The **IoStatus** field of the response contains the returned NTSTATUS code from the IOCTL.

For more information about parallel and serial device codes and the related IO structures, see [[MSDN-Ntddpar]](https://go.microsoft.com/fwlink/?LinkId=114013) and [[MSDN-Ntddser]](https://go.microsoft.com/fwlink/?LinkId=114014).

For more information about device type values, see [[MSDN-DeviceTypes]](https://go.microsoft.com/fwlink/?LinkId=120804).

For more information about I/O control codes and the CTL\_CODE macro, see [[MSDN-IoCtlCodes]](https://go.microsoft.com/fwlink/?LinkId=120805).

# 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 |
| --- | --- | --- |
| [6](#Section_5eb5045ec0e14546b16e5c5cca1be523) Appendix A: Product Behavior | Added Windows Server operating system to the list of applicable products. | Major |

# Index

A

Abstract data model

client ([section 3.1.1](#section_e2c667da24294df29630250a4bef4f67) 14, [section 3.2.1](#section_8949ecee1f804450800348c5baee4e78) 14)

server ([section 3.1.1](#section_e2c667da24294df29630250a4bef4f67) 14, [section 3.3.1](#section_92fc004ec874437f9d9748f3541e7b34) 19)

[Applicability](#section_e9b8d5fc0c914003a58c2c53ea658f37) 8

C

[Capability negotiation](#section_86345ed6dbff415ebbbdd4edddedcd28) 8

[Change tracking](#section_687d2cff37564b57aae8dff6007047a9) 29

Client

abstract data model ([section 3.1.1](#section_e2c667da24294df29630250a4bef4f67) 14, [section 3.2.1](#section_8949ecee1f804450800348c5baee4e78) 14)

[close response DR\_PORT\_CLOSE\_RSP](#section_a8ced545f76345a0a3b4d3ae0f8eb9ca) 12

[create response DR\_PORT\_CREATE\_RSP](#section_a4068ee068434b0d967511f1d84b7a4e) 12

[create response example](#section_2a6786b0f07a4becbbeef45c83e1da6d) 23

[device control response DR\_PORT\_CONTROL\_RSP](#section_fbad3f73259547ffa33576161d58203a) 13

[device list announce request](#section_e5d6d7af5f6c4a6aa00b5e2fcc5d70aa) 15

higher-layer triggered events ([section 3.1.4](#section_0055f26184144d4abb004a7740db8b94) 14, [section 3.2.4](#section_d61ae4ef49ee4f08909e1127402f0f18) 15)

initialization ([section 3.1.3](#section_d861abb109594a08b611f5e336ab4b1f) 14, [section 3.2.3](#section_aa571827550d4bb4bd8ea41182350f3a) 15)

local events ([section 3.1.7](#section_d90125d876ea46999f3ec1b929622216) 14, [section 3.2.7](#section_b92377f765fb4f7bb84031e1932df3f5) 18)

message processing ([section 3.1.5](#section_5717449279fb490d9202d5800d11a642) 14, [section 3.2.5](#section_6841899ee7364b7fb2f91729e1ecf07d) 15)

[other local events](#section_b92377f765fb4f7bb84031e1932df3f5) 18

[read response DR\_PORT\_READ\_RSP](#section_34faabe78ccd44139ee0545ccc6e1420) 12

sequencing rules ([section 3.1.5](#section_5717449279fb490d9202d5800d11a642) 14, [section 3.2.5](#section_6841899ee7364b7fb2f91729e1ecf07d) 15)

timer events ([section 3.1.6](#section_f99f57ac741d473681c442bccc57de9b) 14, [section 3.2.6](#section_85fda02a2a644dec98a5877bfc2a6198) 18)

timers ([section 3.1.2](#section_14bddd0d42754c168cb548a4904dcdee) 14, [section 3.2.2](#section_3789d08da9164b4bb004811b3467bc9e) 15)

[write response DR\_PORT\_WRITE\_RSP](#section_b07fa05262c44379a498dce673f94168) 12

[Client\_Device\_List\_Announce\_Request packet](#section_535df5533d5c4666b33332a0d3d3cab4) 9

[Close response](#section_37dbfa6cd54e43389dd9167cbb5ec653) 17

[Common data types](#section_b7ad7a2e5ba74ca9a43f8eb0972372af) 9

[Common Data Types message](#section_b7ad7a2e5ba74ca9a43f8eb0972372af) 9

[Create response](#section_bd93a9e3f14f483796bb95982c52b98d) 16

D

Data model - abstract

client ([section 3.1.1](#section_e2c667da24294df29630250a4bef4f67) 14, [section 3.2.1](#section_8949ecee1f804450800348c5baee4e78) 14)

server ([section 3.1.1](#section_e2c667da24294df29630250a4bef4f67) 14, [section 3.3.1](#section_92fc004ec874437f9d9748f3541e7b34) 19)

[Device control response](#section_22c62065fe954729aa0378a69d6a0f7f) 18

[Device redirection extensions - purpose](#section_0c173c66bf2043b08d58fcf57468ca92) 7

[DR\_PORT\_CLOSE\_REQ](#section_561b2a288c8a4dd9a8102822480ca835) 10

[DR\_PORT\_CLOSE\_RSP](#section_a8ced545f76345a0a3b4d3ae0f8eb9ca) 12

[DR\_PORT\_CONTROL\_REQ](#section_824cc217febd4a2ea95f5b5bdc32848f) 10

[DR\_PORT\_CONTROL\_RSP](#section_fbad3f73259547ffa33576161d58203a) 13

[DR\_PORT\_CREATE\_REQ](#section_fc7ca06b7e234dd48605b6f75c75efb2) 10

[DR\_PORT\_CREATE\_RSP](#section_a4068ee068434b0d967511f1d84b7a4e) 12

[DR\_PORT\_READ\_REQ](#section_fa44247afe61480a8465d14a41370bf9) 10

[DR\_PORT\_READ\_RSP](#section_34faabe78ccd44139ee0545ccc6e1420) 12

[DR\_PORT\_WRITE\_REQ](#section_38bf440fe055416c8e4c3ac2d59cb110) 10

[DR\_PORT\_WRITE\_RSP](#section_b07fa05262c44379a498dce673f94168) 12

E

Examples

[client - create response](#section_2a6786b0f07a4becbbeef45c83e1da6d) 23

[IO operations](#section_8ce1a120d469430db2e2dd8151aa150a) 23

[port redirection annotations](#section_fabd29bc57e24f3b95c8671bdebd25b1) 21

[server - create request](#section_7ce7f7a0a64a435eb1335dc553a0050b) 22

F

[Fields - vendor-extensible](#section_8f10a5fc4cf945748db751b7024d9d5e) 8

G

[Glossary](#section_0fce97ee34b54edfbad50e73189fdbee) 6

H

Higher-layer triggered events

client ([section 3.1.4](#section_0055f26184144d4abb004a7740db8b94) 14, [section 3.2.4](#section_d61ae4ef49ee4f08909e1127402f0f18) 15)

server ([section 3.1.4](#section_0055f26184144d4abb004a7740db8b94) 14, [section 3.3.4](#section_f049f156735e4c70bfc340169700af6d) 19)

I

[Implementer - security considerations](#section_031010c84b2f4c759bf40e445a776471) 26

[Index of security parameters](#section_61bc0c7930d34440a87108bb0c8e2bbc) 26

[Informative references](#section_e547961f10ac40a4861c581cd33789ab) 7

Initialization

client ([section 3.1.3](#section_d861abb109594a08b611f5e336ab4b1f) 14, [section 3.2.3](#section_aa571827550d4bb4bd8ea41182350f3a) 15)

[overview](#section_cf49ea87d9fc4c2fb6f3e47fddd0cf19) 7

server ([section 3.1.3](#section_d861abb109594a08b611f5e336ab4b1f) 14, [section 3.3.3](#section_e589429d81a642cdac3db3b0f2152115) 19)

[Introduction](#section_56ea9f51c65c4bcd85f601152b011ab9) 6

[IO operations examples](#section_8ce1a120d469430db2e2dd8151aa150a) 23

L

Local events

client ([section 3.1.7](#section_d90125d876ea46999f3ec1b929622216) 14, [section 3.2.7](#section_b92377f765fb4f7bb84031e1932df3f5) 18)

server ([section 3.1.7](#section_d90125d876ea46999f3ec1b929622216) 14, [section 3.3.7](#section_52a8464840bd4793accb17496d123f08) 20)

M

Message processing

client ([section 3.1.5](#section_5717449279fb490d9202d5800d11a642) 14, [section 3.2.5](#section_6841899ee7364b7fb2f91729e1ecf07d) 15)

server ([section 3.1.5](#section_5717449279fb490d9202d5800d11a642) 14, [section 3.3.5](#section_064a26a3c2b142f4aa54525ad80034b6) 19)

Messages

[Common Data Types](#section_b7ad7a2e5ba74ca9a43f8eb0972372af) 9

[data types](#section_b7ad7a2e5ba74ca9a43f8eb0972372af) 9

[overview](#section_01c0c8d5822846c189b94644d35d8c99) 9

[Port Redirection Messages](#section_73cc7f19717e44f19c09102a8c8e8b7a) 9

[syntax](#section_330064150c61464a954bb54f8fd8360e) 9

[transport](#section_dbdaf5d15fd545b6a497e48aa5fd956a) 9

N

[Normative references](#section_f319521596d4466cb4be2e1fafb0697e) 6

O

Other local events

[client](#section_b92377f765fb4f7bb84031e1932df3f5) 18

[server](#section_52a8464840bd4793accb17496d123f08) 20

[Overview - messages](#section_01c0c8d5822846c189b94644d35d8c99) 9

[Overview (synopsis)](#section_e1eea3c9970a4791b6abf445966e15a4) 7

P

[Parameters - security index](#section_61bc0c7930d34440a87108bb0c8e2bbc) 26

[Port close response](#section_c64a4dd67f88468e9834e1651eb4b40e) 20

[Port create response](#section_3480e48d24724e4bb4a0cf7d89df65a9) 20

[Port device announce](#section_38ba503d728646ae8421b45fc5a35506) 19

[Port device control response](#section_c5b6ec2330cd433d8a7828e3e23bd946) 20

[Port instance close request](#section_33d862070737499cbb28626f2d7fd0bb) 19

[Port instance create request](#section_bde0d09429cd4708b1670eb8fcbbed3e) 19

[Port instance device control request](#section_e56773c01ef742989ab2ba96b07c6cae) 19

[Port instance read request](#section_3645effa5f9c45e7a408b7911845db0d) 19

[Port instance write request](#section_d84ed7b6fea1461db2f68cc16df1568a) 19

[Port read response](#section_3948e680aff44e2b8610680beb577460) 20

[Port redirection annotations](#section_fabd29bc57e24f3b95c8671bdebd25b1) 21

Port redirection messages ([section 2.2.2](#section_73cc7f19717e44f19c09102a8c8e8b7a) 9, [section 3.2.5.1](#section_afe0c111814f4ca3b9b87603d309d0b5) 15, [section 3.3.5.1](#section_a12244352d0847039d878c918a9ff591) 19)

[Port Redirection Messages message](#section_73cc7f19717e44f19c09102a8c8e8b7a) 9

[Port write response](#section_df1d6493592441d4a53c0faf3c333573) 20

[Preconditions](#section_d53d45eab6eb4b35941cef5bc441270e) 8

[Prerequisites](#section_d53d45eab6eb4b35941cef5bc441270e) 8

[Product behavior](#section_5eb5045ec0e14546b16e5c5cca1be523) 27

Protocol Details

[overview](#section_fef40c261f9945c583bb46083c7621da) 14

R

[Read response](#section_4936ec4f01884424aa69b1dcdcad09b6) 17

[References](#section_358fd851a34747418df277880460c80d) 6

[informative](#section_e547961f10ac40a4861c581cd33789ab) 7

[normative](#section_f319521596d4466cb4be2e1fafb0697e) 6

[Relationship to other protocols](#section_2b7f34da20d34d37b456e3daf9190702) 7

S

Security

[implementer considerations](#section_031010c84b2f4c759bf40e445a776471) 26

[parameter index](#section_61bc0c7930d34440a87108bb0c8e2bbc) 26

Sequencing rules

client ([section 3.1.5](#section_5717449279fb490d9202d5800d11a642) 14, [section 3.2.5](#section_6841899ee7364b7fb2f91729e1ecf07d) 15)

server ([section 3.1.5](#section_5717449279fb490d9202d5800d11a642) 14, [section 3.3.5](#section_064a26a3c2b142f4aa54525ad80034b6) 19)

Server

abstract data model ([section 3.1.1](#section_e2c667da24294df29630250a4bef4f67) 14, [section 3.3.1](#section_92fc004ec874437f9d9748f3541e7b34) 19)

[close request](#section_ec816dc7bc434947a00fbe66248ef269) 15

[close request DR\_PORT\_CLOSE\_REQ](#section_561b2a288c8a4dd9a8102822480ca835) 10

[create request](#section_777a5a06cb5849b1b899745e994b2e75) 15

[create request DR\_PORT\_CREATE\_REQ](#section_fc7ca06b7e234dd48605b6f75c75efb2) 10

[create request example](#section_7ce7f7a0a64a435eb1335dc553a0050b) 22

[device control request](#section_cfdc5c7626574e3fb6986e35ff6a83d6) 16

[device control request DR\_PORT\_CONTROL\_REQ](#section_824cc217febd4a2ea95f5b5bdc32848f) 10

higher-layer triggered events ([section 3.1.4](#section_0055f26184144d4abb004a7740db8b94) 14, [section 3.3.4](#section_f049f156735e4c70bfc340169700af6d) 19)

initialization ([section 3.1.3](#section_d861abb109594a08b611f5e336ab4b1f) 14, [section 3.3.3](#section_e589429d81a642cdac3db3b0f2152115) 19)

local events ([section 3.1.7](#section_d90125d876ea46999f3ec1b929622216) 14, [section 3.3.7](#section_52a8464840bd4793accb17496d123f08) 20)

message processing ([section 3.1.5](#section_5717449279fb490d9202d5800d11a642) 14, [section 3.3.5](#section_064a26a3c2b142f4aa54525ad80034b6) 19)

[other local events](#section_52a8464840bd4793accb17496d123f08) 20

[read request](#section_74a13a3a98344fe191786ed7ce775098) 15

[read request DR\_PORT\_READ\_REQ](#section_fa44247afe61480a8465d14a41370bf9) 10

sequencing rules ([section 3.1.5](#section_5717449279fb490d9202d5800d11a642) 14, [section 3.3.5](#section_064a26a3c2b142f4aa54525ad80034b6) 19)

timer events ([section 3.1.6](#section_f99f57ac741d473681c442bccc57de9b) 14, [section 3.3.6](#section_785d27c938494154b147644b9b0b4dc0) 20)

timers ([section 3.1.2](#section_14bddd0d42754c168cb548a4904dcdee) 14, [section 3.3.2](#section_256b304074f64533a47e25473a90e576) 19)

[write request](#section_b0ee05af912e4ceb99c9ae82a1747cf6) 16

[write request DR\_PORT\_WRITE\_REQ](#section_38bf440fe055416c8e4c3ac2d59cb110) 10

[Standards assignments](#section_87497b77c42f427390bff8fb8041ebba) 8

[Syntax](#section_330064150c61464a954bb54f8fd8360e) 9

T

Timer events

client ([section 3.1.6](#section_f99f57ac741d473681c442bccc57de9b) 14, [section 3.2.6](#section_85fda02a2a644dec98a5877bfc2a6198) 18)

server ([section 3.1.6](#section_f99f57ac741d473681c442bccc57de9b) 14, [section 3.3.6](#section_785d27c938494154b147644b9b0b4dc0) 20)

Timers

client ([section 3.1.2](#section_14bddd0d42754c168cb548a4904dcdee) 14, [section 3.2.2](#section_3789d08da9164b4bb004811b3467bc9e) 15)

server ([section 3.1.2](#section_14bddd0d42754c168cb548a4904dcdee) 14, [section 3.3.2](#section_256b304074f64533a47e25473a90e576) 19)

[Tracking changes](#section_687d2cff37564b57aae8dff6007047a9) 29

[Transport](#section_dbdaf5d15fd545b6a497e48aa5fd956a) 9

Triggered events - higher-layer

client ([section 3.1.4](#section_0055f26184144d4abb004a7740db8b94) 14, [section 3.2.4](#section_d61ae4ef49ee4f08909e1127402f0f18) 15)

server ([section 3.1.4](#section_0055f26184144d4abb004a7740db8b94) 14, [section 3.3.4](#section_f049f156735e4c70bfc340169700af6d) 19)

V

[Vendor-extensible fields](#section_8f10a5fc4cf945748db751b7024d9d5e) 8

[Versioning](#section_86345ed6dbff415ebbbdd4edddedcd28) 8

W

[Write response](#section_454c8469b014461fa1d00abe0fcd8a72) 17