

## [MS-RDPEPC]:

# Remote Desktop Protocol: Print 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](#) or the [Microsoft Community Promise](#). If you would prefer a written license, or if the technologies described in this documentation are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting [iplg@microsoft.com](mailto:iplg@microsoft.com).
- **License Programs.** To see all of the protocols in scope under a specific license program and the associated patents, visit the [Patent Map](#).
- **Trademarks.** The names of companies and products contained in this documentation might be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit [www.microsoft.com/trademarks](http://www.microsoft.com/trademarks).
- **Fictitious Names.** The example companies, organizations, products, domain names, email addresses, logos, people, places, and events that are depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

**Reservation of Rights.** All other rights are reserved, and this notice does not grant any rights other than as specifically described above, whether by implication, estoppel, or otherwise.

**Tools.** The Open Specifications documentation does not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments, you are free to take advantage of them. Certain Open Specifications documents are intended for use in conjunction with publicly available standards specifications and network programming art and, as such, assume that the reader either is familiar with the aforementioned material or has immediate access to it.

**Support.** For questions and support, please contact [dochelp@microsoft.com](mailto:dochelp@microsoft.com).

## Revision Summary

Date	Revision History	Revision Class	Comments
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	1.4	Minor	Clarified the meaning of the technical content.
8/29/2008	1.4.1	Editorial	Changed language and formatting in the technical content.
10/24/2008	1.5	Minor	Clarified the meaning of the technical content.
12/5/2008	1.6	Minor	Clarified the meaning of the technical content.
1/16/2009	1.6.1	Editorial	Changed language and formatting in the technical content.
2/27/2009	1.6.2	Editorial	Changed language and formatting in the technical content.
4/10/2009	1.6.3	Editorial	Changed language and formatting in the technical content.
5/22/2009	1.7	Minor	Clarified the meaning of the technical content.
7/2/2009	1.7.1	Editorial	Changed language and formatting in the technical content.
8/14/2009	1.7.2	Editorial	Changed language and formatting in the technical content.
9/25/2009	1.8	Minor	Clarified the meaning of the technical content.
11/6/2009	2.0	Major	Updated and revised the technical content.
12/18/2009	2.0.1	Editorial	Changed language and formatting in the technical content.
1/29/2010	2.1	Minor	Clarified the meaning of the technical content.
3/12/2010	2.1.1	Editorial	Changed language and formatting in the technical content.
4/23/2010	2.1.2	Editorial	Changed language and formatting in the technical content.
6/4/2010	2.1.3	Editorial	Changed language and formatting in the technical content.
7/16/2010	2.1.3	None	No changes to the meaning, language, or formatting of the technical content.
8/27/2010	3.0	Major	Updated and revised the technical content.
10/8/2010	4.0	Major	Updated and revised the technical content.
11/19/2010	4.0	None	No changes to the meaning, language, or formatting of the technical content.

<b>Date</b>	<b>Revision History</b>	<b>Revision Class</b>	<b>Comments</b>
1/7/2011	4.0	None	No changes to the meaning, language, or formatting of the technical content.
2/11/2011	4.0	None	No changes to the meaning, language, or formatting of the technical content.
3/25/2011	4.0	None	No changes to the meaning, language, or formatting of the technical content.
5/6/2011	4.0	None	No changes to the meaning, language, or formatting of the technical content.
6/17/2011	4.1	Minor	Clarified the meaning of the technical content.
9/23/2011	4.1	None	No changes to the meaning, language, or formatting of the technical content.
12/16/2011	5.0	Major	Updated and revised the technical content.
3/30/2012	5.0	None	No changes to the meaning, language, or formatting of the technical content.
7/12/2012	5.0	None	No changes to the meaning, language, or formatting of the technical content.
10/25/2012	5.0	None	No changes to the meaning, language, or formatting of the technical content.
1/31/2013	5.0	None	No changes to the meaning, language, or formatting of the technical content.
8/8/2013	6.0	Major	Updated and revised the technical content.
11/14/2013	6.0	None	No changes to the meaning, language, or formatting of the technical content.
2/13/2014	6.0	None	No changes to the meaning, language, or formatting of the technical content.
5/15/2014	6.0	None	No changes to the meaning, language, or formatting of the technical content.
6/30/2015	7.0	Major	Significantly changed the technical content.
10/16/2015	7.0	None	No changes to the meaning, language, or formatting of the technical content.
7/14/2016	7.0	None	No changes to the meaning, language, or formatting of the technical content.
6/1/2017	8.0	Major	Significantly changed the technical content.
9/15/2017	9.0	Major	Significantly changed the technical content.
12/1/2017	9.0	None	No changes to the meaning, language, or formatting of the technical content.
9/12/2018	10.0	Major	Significantly changed the technical content.
4/7/2021	11.0	Major	Significantly changed the technical content.
6/25/2021	12.0	Major	Significantly changed the technical content.

# Table of Contents

<b>1</b>	<b>Introduction</b>	<b>6</b>
1.1	Glossary	6
1.2	References	7
1.2.1	Normative References	7
1.2.2	Informative References	7
1.3	Overview	7
1.3.1	Purpose of Print Virtual Channel Extension	7
1.3.2	Protocol Initialization	7
1.4	Relationship to Other Protocols	8
1.5	Prerequisites/Preconditions	8
1.6	Applicability Statement	8
1.7	Versioning and Capability Negotiation	8
1.8	Vendor-Extensible Fields	8
1.9	Standards Assignments	8
<b>2</b>	<b>Messages</b>	<b>9</b>
2.1	Transport	9
2.2	Message Syntax	9
2.2.1	Common Data Types	9
2.2.1.1	Server Printer Cache Event (SERVER_PRINTER_CACHE_EVENT)	9
2.2.2	Printer Redirection Messages	10
2.2.2.1	Client Device List Announce Request (DR_PRN_DEVICE_ANNOUNCE)	10
2.2.2.2	Server Printer Set XPS Mode (DR_PRN_USING_XPS)	12
2.2.2.3	Add Printer Cachedata (DR_PRN_ADD_CACHEDATA)	12
2.2.2.4	Update Printer Cachedata (DR_PRN_UPDATE_CACHEDATA)	14
2.2.2.5	Delete Printer Cachedata (DR_PRN_DELETE_CACHEDATA)	14
2.2.2.6	Rename Printer Cachedata (DR_PRN_RENAME_CACHEDATA)	15
2.2.2.7	Server Create Request (DR_PRN_CREATE_REQ)	16
2.2.2.8	Server Printer Close Request (DR_PRN_CLOSE_REQ)	17
2.2.2.9	Server Printer Write Request (DR_PRN_WRITE_REQ)	17
2.2.2.10	Client Printer Create Response (DR_PRN_CREATE_RSP)	17
2.2.2.11	Client Printer Close Response (DR_PRN_CLOSE_RSP)	18
2.2.2.12	Client Printer Write Response (DR_PRN_WRITE_RSP)	18
<b>3</b>	<b>Protocol Details</b>	<b>19</b>
3.1	Common Details	19
3.1.1	Abstract Data Model	19
3.1.1.1	Cached Printer Config Data	19
3.1.1.2	XPS Mode	19
3.1.2	Timers	19
3.1.3	Initialization	19
3.1.4	Higher-Layer Triggered Events	19
3.1.5	Message Processing Events and Sequencing Rules	20
3.1.6	Timer Events	20
3.1.7	Other Local Events	20
3.2	Client Details	20
3.2.1	Abstract Data Model	20
3.2.2	Timers	20
3.2.3	Initialization	20
3.2.4	Higher-Layer Triggered Events	20
3.2.5	Message Processing Events and Sequencing Rules	20
3.2.5.1	Printer Redirection Messages	20
3.2.5.1.1	Sending a Client Device List Announce Request Message	20
3.2.5.1.2	Processing a Printer Set XPS Mode Message	21
3.2.5.1.3	Processing an Add Printer Cachedata Message	21

3.2.5.1.4	Processing an Update Printer Cachedata Message .....	21
3.2.5.1.5	Processing a Delete Printer Cachedata Message .....	21
3.2.5.1.6	Processing a Rename Printer Cachedata Message .....	21
3.2.5.1.7	Processing a Create Request Message .....	21
3.2.5.1.8	Processing a Close Request Message .....	21
3.2.5.1.9	Processing a Write Request Message .....	21
3.2.5.1.10	Sending a Printer Create Response Message .....	21
3.2.5.1.11	Sending a Printer Close Response Message .....	22
3.2.5.1.12	Sending a Printer Write Response Message .....	22
3.2.6	Timer Events .....	23
3.2.7	Other Local Events .....	23
3.3	Server Details .....	23
3.3.1	Abstract Data Model .....	23
3.3.2	Timers .....	23
3.3.3	Initialization .....	23
3.3.4	Higher-Layer Triggered Events .....	23
3.3.5	Message Processing Events and Sequencing Rules .....	23
3.3.5.1	Printer Redirection Messages .....	23
3.3.5.1.1	Processing a Client Device List Announce Request Message .....	23
3.3.5.1.2	Sending a Printer Set XPS Mode Message .....	24
3.3.5.1.3	Sending an Add Printer Cachedata Message .....	24
3.3.5.1.4	Sending an Update Printer Cachedata Message .....	24
3.3.5.1.5	Sending Delete Printer Cachedata .....	24
3.3.5.1.6	Sending a Rename Printer Cachedata Message .....	24
3.3.5.1.7	Sending a Printer Create Request Message .....	24
3.3.5.1.8	Sending a Printer Close Request Message .....	24
3.3.5.1.9	Sending a Printer Write Request Message .....	24
3.3.5.1.10	Processing a Printer Create Response Message .....	24
3.3.5.1.11	Processing a Printer Close Response Message .....	25
3.3.5.1.12	Processing a Printer Write Response Message .....	25
3.3.6	Timer Events .....	25
3.3.7	Other Local Events .....	25
<b>4</b>	<b>Protocol Examples .....</b>	<b>26</b>
4.1	Annotations for Printer Redirection .....	26
4.1.1	Client Printer Device (DR_PRN_DEVICE_ANNOUNCE) .....	26
4.1.2	Server Printer Set XPS Mode (DR_PRN_USING_XPS) .....	27
4.1.3	Add Printer Cachedata (DR_PRN_ADD_CACHEDATA) .....	27
4.1.4	Update Printer Cachedata (DR_PRN_UPDATE_CACHEDATA) .....	27
4.1.5	Delete Printer Cachedata (DR_PRN_DELETE_CACHEDATA) .....	28
4.1.6	Rename Printer Cachedata (DR_PRN_RENAME_CACHEDATA) .....	28
4.1.7	Server Printer Create Request (DR_PRN_CREATE_REQ) .....	29
4.1.8	Server Printer Close Request (DR_PRN_CLOSE_REQ) .....	29
4.1.9	Server Printer Write Request (DR_PRN_WRITE_REQ) .....	29
4.1.10	Client Printer Create Response (DR_PRN_CREATE_RSP) .....	30
4.1.11	Client Printer Close Response (DR_PRN_CLOSE_RSP) .....	30
4.1.12	Client Printer Write Response (DR_PRN_WRITE_RSP) .....	30
<b>5</b>	<b>Security .....</b>	<b>32</b>
5.1	Security Considerations for Implementers .....	32
5.2	Index of Security Parameters .....	32
<b>6</b>	<b>Appendix A: Product Behavior .....</b>	<b>33</b>
<b>7</b>	<b>Change Tracking .....</b>	<b>35</b>
<b>8</b>	<b>Index .....</b>	<b>36</b>

# 1 Introduction

This document specifies the Print Virtual Channel Extension to the Remote Desktop Protocol. This protocol is used to redirect printers from a **terminal client** to the **terminal server**. This allows the **server** access to printers physically connected to the **client** as if the 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.

## 1.1 Glossary

This document uses the following terms:

**ASCII:** The American Standard Code for Information Interchange (ASCII) is an 8-bit character-encoding scheme based on the English alphabet. ASCII codes represent text in computers, communications equipment, and other devices that work with text. ASCII refers to a single 8-bit ASCII character or an array of 8-bit ASCII characters with the high bit of each character set to zero.

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

**driver name:** A name assigned by the vendor of a printer driver, typically this name is the same as the name of a printer model (or family) that is serviced by a particular driver.

**manual printer redirection:** The **terminal server client** redirects printers that are not installed (that is, associated with a printer driver) on the **client**. For such redirection to work, the user must manually specify which printer driver to use.

**printer name:** The term **printer name** is used to represent a user-assigned, arbitrary name given to a printer queue. This protocol uses the term "**printer name**" to identify a given printer queue.

**printer queue:** A collection of printer drivers and printer ports that maintains a list of print jobs to the printer.

**PRN file:** A type of file that contains a sequence of raw bytes that are understood by the printer hardware. The printer driver produces these files and sends them to the printer port for actual printing. In the case of **terminal server** printer redirection, this stream is captured on the **terminal server** for the redirected printers and then sent to the client hardware for printing in raw mode. The contents of this stream are opaque to the Print Virtual Channel Extension.

**remote device:** A device that is attached to a remote (or **client**) 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** of a **terminal server**. A terminal client program that runs on the **client** machine.

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

**Unicode string:** A Unicode 8-bit string is an ordered sequence of 8-bit units, a Unicode 16-bit string is an ordered sequence of 16-bit code units, and a Unicode 32-bit string is an ordered sequence of 32-bit code units. In some cases, it could be acceptable not to terminate with a terminating null character. Unless otherwise specified, all **Unicode strings** follow the UTF-16LE encoding scheme with no Byte Order Mark (BOM).

**XML Paper Specification (XPS):** An XML-based document format. **XML Paper Specification (XPS)** specifies the set of conventions for the use of XML and other widely available

technologies to describe the content and appearance of paginated documents. For more information, see [\[MSFT-XMLPAPER\]](#).

**XPS file:** A type of file that contains **XPS** data. Such files contain all document elements as well as printer-specific rendering information.

**MAY, SHOULD, MUST, SHOULD NOT, MUST NOT:** These terms (in all caps) are used as defined in [\[RFC2119\]](#). All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

## 1.2 References

Links to a document in the Microsoft Open Specifications library point to the correct section in the most recently published version of the referenced document. However, because individual documents in the library are not updated at the same time, the section numbers in the documents may not match. You can confirm the correct section numbering by checking the [Errata](#).

### 1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact [dochelp@microsoft.com](mailto:dochelp@microsoft.com). We will assist you in finding the relevant information.

[MS-ERREF] Microsoft Corporation, "[Windows Error Codes](#)".

[MS-RDPEFS] Microsoft Corporation, "[Remote Desktop Protocol: File System Virtual Channel Extension](#)".

[MS-RDPESP] Microsoft Corporation, "[Remote Desktop Protocol: Serial and Parallel Port Virtual Channel Extension](#)".

[MS-RDPEXPS] Microsoft Corporation, "[Remote Desktop Protocol: XML Paper Specification \(XPS\) Print 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

None.

## 1.3 Overview

The Print Virtual Channel Extension specifies the communication used to enable the redirection of printers between a **terminal client** and a **terminal server**. By redirecting printers from the terminal client to the terminal server, applications running on a server machine can access the **remote devices** as if they were local printers.

### 1.3.1 Purpose of Print Virtual Channel Extension

The Print Virtual Channel Extension enables printer redirection. It allows applications running a **terminal server** to access the printers attached to the **terminal client**.

### 1.3.2 Protocol Initialization

This protocol relies for transportation of its packets on the Remote Desktop Protocol: File System Virtual Channel Extension as specified in [\[MS-RDPEFS\]](#). It describes the format of messages for printer

devices. This protocol is considered initialized after at least one printer device is advertised by the client, and after the "Server Device Announce Response" message that indicates successful initialization for that device is sent by the server as illustrated in the figure in section 1.3.1 of [MS-RDPEFS].

#### **1.4 Relationship to Other Protocols**

This protocol relies for transportation of its packets on the Remote Desktop Protocol: File System Virtual Channel Extension as specified in [\[MS-RDPEFS\]](#). This protocol extends the File System Virtual Channel Extension to enable printer redirection.

#### **1.5 Prerequisites/Preconditions**

The Print Virtual Channel Extension operates only after the Remote Desktop Protocol: File System Virtual Channel Extension [\[MS-RDPEFS\]](#) transport is fully established and a printer device is advertised by the client. This protocol describes the messages for printer devices only.

#### **1.6 Applicability Statement**

The Print Virtual Channel Extension is designed to be run within the context of a Remote Desktop Protocol virtual channel established between a client and server. This protocol extension is applicable when applications running on the **terminal server** need to access the printers physically connected to a client machine.

#### **1.7 Versioning and Capability Negotiation**

This protocol relies on the Print Virtual Channel to perform basic versioning and capability negotiation.

#### **1.8 Vendor-Extensible Fields**

This protocol contains no vendor-extensible fields.

#### **1.9 Standards Assignments**

This protocol does not use any standards assignments.



## 2 Messages

This protocol shares messages and common data types that are specified in [\[MS-RDPEFS\]](#) section 2 of Remote Desktop Protocol: File System Virtual Channel Extension. This section describes the messages and data types used by the Print Virtual Channel Extension.

### 2.1 Transport

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

The static virtual channel is considered terminated if the client and server are disconnected or if there are errors in processing the packet, as described in [\[MS-RDPEFS\]](#) section 3.1.5.2 "Processing Packet Errors".

### 2.2 Message Syntax

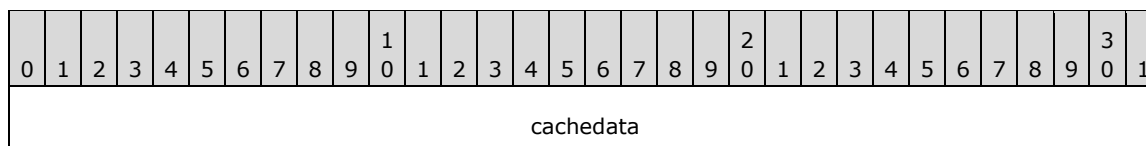
The following sections contain Print Virtual Channel Extension message syntax.

#### 2.2.1 Common Data Types

Printer redirection uses common data types defined and explained in the Remote Desktop Protocol: File System Virtual Channel Extension. The following are the data types that are specific to the Print Virtual Channel Extension.

##### 2.2.1.1 Server Printer Cache Event (SERVER\_PRINTER\_CACHE\_EVENT)

The Server Printer Cache Event is used to identify various **server** events associated with Printer Cached Config Data (section [3.1.1.1](#)) in the messages sent from the server to the **client**.



**cachedata (4 bytes):** A 32-bit unsigned integer value that MUST be set to one of the following values.

Value	Meaning
RDPDR_ADD_PRINTER_EVENT 0x00000001	Add printer cachedata event.
RDPDR_UPDATE_PRINTER_EVENT 0x00000002	Update printer cachedata event.
RDPDR_DELETE_PRINTER_EVENT 0x00000003	Delete printer cachedata event.
RDPDR_RENAME_PRINTER_EVENT 0x00000004	Rename printer cachedata event.

## 2.2.2 Printer Redirection Messages

### 2.2.2.1 Client Device List Announce Request (DR\_PRN\_DEVICE\_ANNOUNCE)

This message is specified in [MS-RDPEFS] section 2.2.2.9 (Client Device List Announce Request). For each redirected printer, a DEVICE\_ANNOUNCE header (as specified in [MS-RDPEFS] section 2.2.1.3) is generated by the client printer redirection extension followed by variable printer-specific data.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
DeviceType																															
DeviceId																															
PreferredDosName																															
...																															
DeviceDataLength																															
Flags																															
CodePage																															
PnPNameLen																															
DriverNameLen																															
PrintNameLen																															
CachedFieldsLen																															
PnPName (variable)																															
...																															
DriverName (variable)																															
...																															
PrinterName (variable)																															
...																															
CachedPrinterConfigData (variable)																															
...																															

**DeviceType (4 bytes):** This field is defined in [MS-RDPEFS] section 2.2.1.3. This field MUST be set to RDPDR\_DTYP\_PRINT.

**DeviceId (4 bytes):** This field is defined in [MS-RDPEFS] section 2.2.1.3. The **DeviceId** field MUST be set to the unique device ID to identify this printer device. This field is later used to refer to the same printer device by both the client and the server.

**PreferredDosName (8 bytes):** This field is defined in [MS-RDPEFS] section 2.2.1.3. The **PreferredDosName** field MUST be set to the port name on which the printer is installed. The first 3 characters MUST be "PRN" and the remainder MUST be digits.

**DeviceDataLength (4 bytes):** This field is defined in [MS-RDPEFS] section 2.2.1.3. The **DeviceDataLength** field MUST be set to the length of data fields following the **DeviceDataLength** field.

**Flags (4 bytes):** A 32-bit unsigned integer that indicates the properties of the client **printer queue**. This bit field MUST be a valid combination of any of the following values.

Value	Meaning
RDPDR_PRINTER_ANNOUNCE_FLAG_ASCII 0x00000001	The <b>DriverName</b> field MUST be in <b>ASCII</b> characters. If not set, it MUST be in Unicode.
RDPDR_PRINTER_ANNOUNCE_FLAG_DEFAULTPRINTER 0x00000002	The printer is set as default. There MUST be only one printer with this flag set.
RDPDR_PRINTER_ANNOUNCE_FLAG_NETWORKPRINTER 0x00000004	This printer is from the network.
RDPDR_PRINTER_ANNOUNCE_FLAG_TSPRINTER 0x00000008	This flag is set when the printer to be redirected is not a local or network printer but is a <b>terminal server</b> client printer. This can happen in nested TS sessions; that is, this can happen when a TS connection is made from within a TS session.
RDPDR_PRINTER_ANNOUNCE_FLAG_XPSFORMAT 0x00000010	This client/printer supports <b>XML Paper Specification (XPS)</b> format (section <a href="#">3.1.1.2</a> ).

**CodePage (4 bytes):** A 32-bit unsigned integer. Reserved for future use. This field MUST be set to 0.

**PnPNameLen (4 bytes):** A 32-bit unsigned integer that specifies the number of bytes in the **PnPName** field, including its null terminator.

**DriverNameLen (4 bytes):** A 32-bit unsigned integer that specifies the number of bytes in the **DriverName** field, including its null terminator.

**PrintNameLen (4 bytes):** A 32-bit unsigned integer that specifies the number of bytes in the **PrintName** field, including its null terminator.

**CachedFieldsLen (4 bytes):** A 32-bit unsigned integer that specifies the number of bytes in the **CachedPrinterConfigData** field.

**PnPName (variable):** A null-terminated **Unicode string**. This field can be set to any valid Unicode string and MUST be ignored on receipt.

**DriverName (variable):** An array of characters. The type of characters is determined by the RDPDR\_PRINTER\_ANNOUNCE\_FLAG\_ASCII flag. If the flag is set, the **DriverName** field MUST be a null-terminated ASCII string; otherwise, it MUST be a null-terminated Unicode string. The **DriverName** field specifies the **driver name** used by the client for this printer. This name is used by the terminal server to determine the appropriate matching driver for the redirected printer, which is to be used on the server. [<1>](#)

**PrinterName (variable):** The **PrinterName** field is a null-terminated Unicode string. The client MUST specify the user-assigned **printer name** of the local printer in this field. This name is used by the server to generate the server-side redirected printer queue.

**CachedPrinterConfigData (variable):** A variable-length array of bytes. This field is a binary large object (BLOB) of data that describes the cached printer configuration (see section [3.1.1.1](#)).

### 2.2.2.2 Server Printer Set XPS Mode (DR\_PRN\_USING\_XPS)

This message is sent from server to client to set the device in **XPS** mode (see section [3.1.1.2](#)) and indicate to the client that future Printer Write Request (section [2.2.2.9](#)) messages will use the XPS format.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Header																															
PrinterId																															
Flags																															

**Header (4 bytes):** The common message header (as specified in [\[MS-RDPEFS\]](#) section 2.2.1.1). The **Component** field MUST be set to RDPDR\_CTYP\_PRN and the **PacketId** field MUST be set to PAKID\_PRN\_USING\_XPS.

**PrinterId (4 bytes):** A 32-bit unsigned integer. This message is handled by the Print Virtual Channel Extension only if the **PrinterId** field matches the previously established **DeviceId** field for a printer device, see [\[MS-RDPEFS\]](#) section 3.1.1.

**Flags (4 bytes):** A 32-bit unsigned integer. This field is unused. It can contain any value and MUST be ignored on receipt.

### 2.2.2.3 Add Printer Cachedata (DR\_PRN\_ADD\_CACHEDATA)

This message is sent from the server to the client when a **printer queue** is created manually on the redirected port (port redirection is explained in [\[MS-RDPESP\]](#)) on the server.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Header																															
EventId																															
PortDosName																															
...																															
PnPNameLen																															
DriverNameLen																															
PrintNameLen																															

CachedFieldsLen
PnPName (variable)
...
DriverName (variable)
...
PrinterName (variable)
...
CachedPrinterConfigData (variable)
...

**Header (4 bytes):** The common message header (as specified in [\[MS-RDPEFS\]](#) section 2.2.1.1). The **Component** field MUST be set to RDPDR\_CTYP\_PRN and the **PacketId** field MUST be set to PAKID\_PRN\_CACHE\_DATA.

**EventId (4 bytes):** A SERVER\_PRINTER\_CACHE\_EVENT (section [2.2.1.1](#)) packet. The **cachedata** field MUST be set to RDPDR\_ADD\_PRINTER\_EVENT (0x00000001).

**PortDosName (8 bytes):** A string of **ASCII** characters with a maximum length of 8 characters that represent the name of the device as it appears on the client. If this field is not be null-terminated, then the **PortDosName** string is equal to all 8 characters.

**PnPNameLen (4 bytes):** A 32-bit unsigned integer that specifies the number of bytes in the **PnPName** field, including its null terminator.

**DriverNameLen (4 bytes):** A 32-bit unsigned integer that specifies the number of bytes in the **DriverName** field, including its null terminator.

**PrintNameLen (4 bytes):** A 32-bit unsigned integer that specifies the number of bytes in the **PrintName** field, including its null terminator.

**CachedFieldsLen (4 bytes):** A 32-bit unsigned integer that specifies the number of bytes in the **CachedPrinterConfigData** field.

**PnPName (variable):** A null-terminated **Unicode string**. This field can be set to any valid Unicode string and MUST be ignored on receipt.

**DriverName (variable):** A null-terminated Unicode string. This string specifies the printer **driver name** that is associated with the printer configuration data.

**PrinterName (variable):** A null-terminated Unicode string that specifies the client printer, which was sent for this configuration data message.

**CachedPrinterConfigData (variable):** A variable-length array of bytes. This field is a BLOB of data that describes the cached printer configuration (see section [3.1.1.1](#)).

### 2.2.2.4 Update Printer Cachedata (DR\_PRN\_UPDATE\_CACHEDATA)

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Header																															
EventId																															
PrinterNameLen																															
ConfigDataLen																															
PrinterName (variable)																															
...																															
CachedPrinterConfigData (variable)																															
...																															

**Header (4 bytes):** The common message header (as specified in [\[MS-RDPEFS\]](#) section 2.2.1.1). The **Component** field MUST be set to RDPDR\_CTYP\_PRN and the **PacketId** field MUST be set to PAKID\_PRN\_CACHE\_DATA.

**EventId (4 bytes):** A SERVER\_PRINTER\_CACHE\_EVENT (section [2.2.1.1](#)) event. The **cachedata** field MUST be set to RDPDR\_UPDATE\_PRINTER\_EVENT (0x00000002).

**PrinterNameLen (4 bytes):** A 32-bit unsigned integer specifying the number of bytes in the **PrinterName** field, including its null terminator.

**ConfigDataLen (4 bytes):** A 32-bit unsigned integer specifying the number of bytes in the **CachedPrinterConfigData** field.

**PrinterName (variable):** A null-terminated **Unicode string** that specifies the printer, for which the updated printer configuration data is sent.

**CachedPrinterConfigData (variable):** A variable-length array of bytes. This field is a BLOB of data that describes the cached printer configuration (see section [3.1.1.1](#)).

### 2.2.2.5 Delete Printer Cachedata (DR\_PRN\_DELETE\_CACHEDATA)

This message is sent by the server when a manually created printer on the redirected port is deleted.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Header																															
EventId																															
PrinterNameLen																															
PrinterName (variable)																															

...

**Header (4 bytes):** The common message header (as specified in [\[MS-RDPEFS\]](#) section 2.2.1.1). The **Component** field MUST be set to RDPDR\_CTYP\_PRN and the **PacketId** field MUST be set to PAKID\_PRN\_CACHE\_DATA.

**EventId (4 bytes):** A SERVER\_PRINTER\_CACHE\_EVENT (section [2.2.1.1](#)) event. The **cachedata** field MUST be set to RDPDR\_DELETE\_PRINTER\_EVENT (0x00000003).

**PrinterNameLen (4 bytes):** A 32-bit unsigned integer that specifies the number of bytes in the **PrinterName** field, including its null terminator.

**PrinterName (variable):** A null-terminated **Unicode string** that specifies the printer that was deleted.

### 2.2.2.6 Rename Printer Cachedata (DR\_PRN\_RENAME\_CACHEDATA)

This message is sent by the server when the user renames a redirected printer.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Header																															
EventId																															
OldPrinterNameLen																															
NewPrinterNameLen																															
OldPrinterName (variable)																															
...																															
NewPrinterName (variable)																															
...																															

**Header (4 bytes):** The common message header (as specified in [\[MS-RDPEFS\]](#) section 2.2.1.1). The **Component** field MUST be set to RDPDR\_CTYP\_PRN and the **PacketId** field MUST be set to PAKID\_PRN\_CACHE\_DATA.

**EventId (4 bytes):** A SERVER\_PRINTER\_CACHE\_EVENT (section [2.2.1.1](#)) event. The **cachedata** field MUST be set to RDPDR\_RENAME\_PRINTER\_EVENT (0x00000004).

**OldPrinterNameLen (4 bytes):** A 32-bit unsigned integer that specifies the number of bytes in the **OldPrinterName** field, including its null terminator.

**NewPrinterNameLen (4 bytes):** A 32-bit unsigned integer that specifies the number of bytes in the **NewPrinterName** field, including its null terminator.

**OldPrinterName (variable):** A null-terminated **Unicode string** that specifies the **printer name** before the rename operation.

**NewPrinterName (variable):** A null-terminated Unicode string that specifies the printer name after the rename operation.

### 2.2.2.7 Server Create Request (DR\_PRN\_CREATE\_REQ)

This message is sent by the server to request the opening of the client **printer queue** and to prepare for receiving print data.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
DeviceIoRequest (24 bytes)																															
...																															
...																															
DesiredAccess																															
AllocationSize																															
...																															
FileAttributes																															
SharedAccess																															
Disposition																															
CreateOptions																															
PathLength																															

**DeviceIoRequest (24 bytes):** A DR\_DEVICE\_IOREQUEST header (as specified in [\[MS-RDPEFS\]](#) section 2.2.1.4).

The **MajorFunction** in this header MUST be set to IRP\_MJ\_CREATE.

This message is handled by the Print Virtual Channel Extension only if the **DeviceId** field matches the previously established **DeviceId** for a printer device, see [\[MS-RDPEFS\]](#) section 3.1.1.

**DesiredAccess (4 bytes):** A 32-bit unsigned integer. Unused. This field can contain any value and MUST be ignored on receipt.

**AllocationSize (8 bytes):** A 64-bit unsigned integer. Unused. This field can contain any value and MUST be ignored on receipt.

**FileAttributes (4 bytes):** A 32-bit unsigned integer. Unused. This field can contain any value and MUST be ignored on receipt.

**SharedAccess (4 bytes):** A 32-bit unsigned integer. Unused. This field can contain any value and MUST be ignored on receipt.

**Disposition (4 bytes):** A 32-bit unsigned integer. Unused. This field can contain any value and MUST be ignored on receipt.

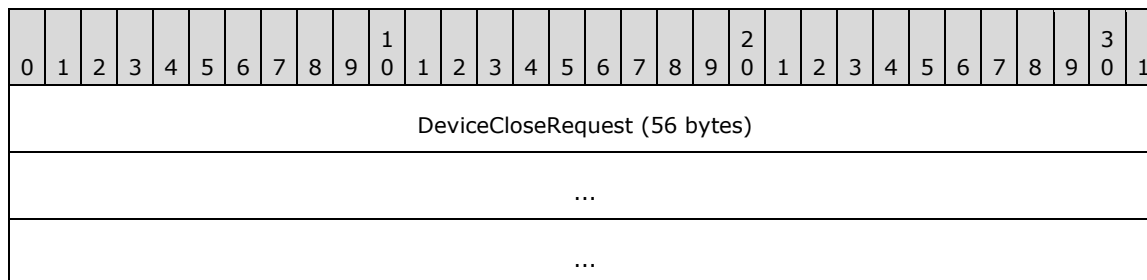


**CreateOptions (4 bytes):** A 32-bit unsigned integer. Unused. This field can contain any value and MUST be ignored on receipt.

**PathLength (4 bytes):** A 32-bit unsigned integer. This field MUST be set to 0 for the printer devices.

### 2.2.2.8 Server Printer Close Request (DR\_PRN\_CLOSE\_REQ)

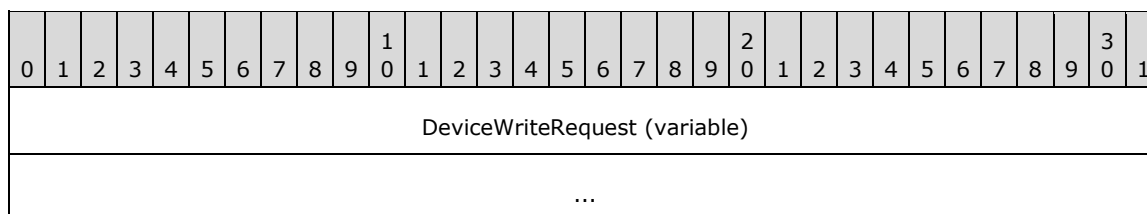
This message is sent by the server to request the closing of the **printer queue** that was previously created by a DR\_PRN\_CREATE\_REQ (section [2.2.2.7](#)).



**DeviceCloseRequest (56 bytes):** A DR\_CLOSE\_REQ message (as specified in [\[MS-RDPEFS\]](#) section 2.2.1.4.2). This request closes a printer opened by DR\_PRN\_CREATE\_REQ (section 2.2.2.7).

### 2.2.2.9 Server Printer Write Request (DR\_PRN\_WRITE\_REQ)

This message is sent from the server to send the print data to the client printer.

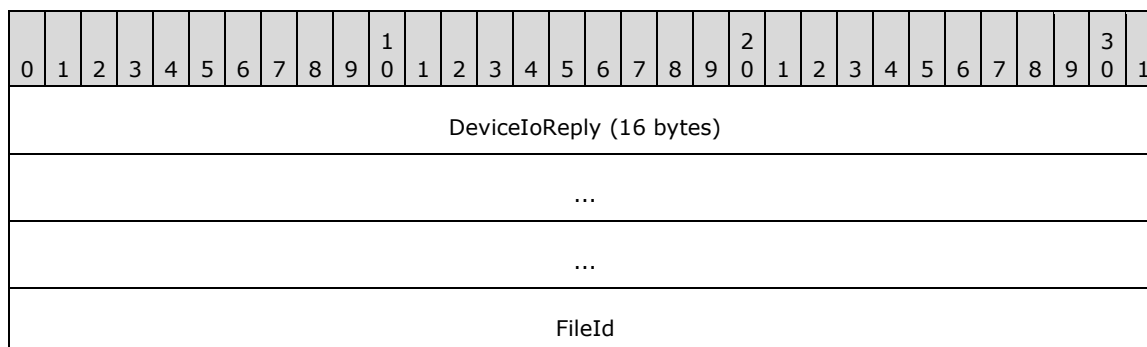


**DeviceWriteRequest (variable):** A DR\_WRITE\_REQ request (as specified in [\[MS-RDPEFS\]](#) section 2.2.1.4.4). The **Offset** field of this request is unused and reserved for future use. The **Offset** field can contain any value and MUST be ignored on receipt.

The **WriteData** field of the request could be either in **PRN file** or **XPS file** format.

### 2.2.2.10 Client Printer Create Response (DR\_PRN\_CREATE\_RSP)

The client responds to the [DR\\_PRN\\_CREATE\\_REQ](#) with this message.

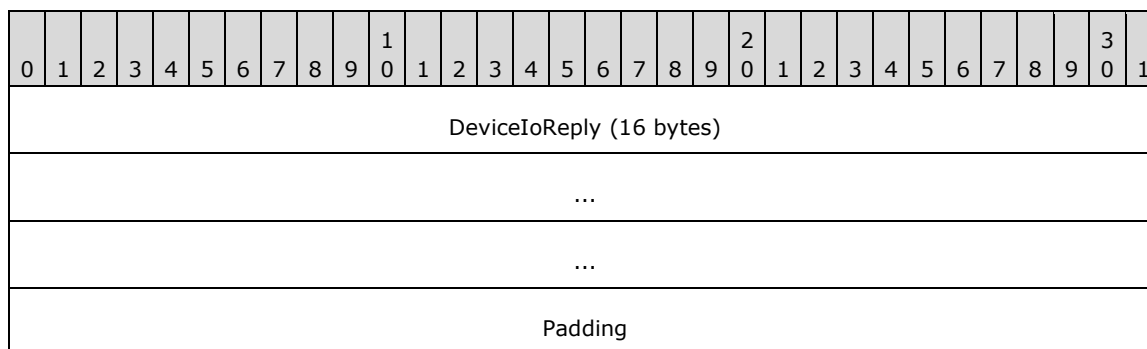


**DeviceIoReply (16 bytes):** A DR\_DEVICE\_IOCTLCOMPLETION header (as specified in [MS-RDPEFS] section 2.2.1.5). The **CompletionId** field of this header MUST match the **CompletionId** of a Device I/O Request message that had the **MajorFunction** field set to IRP\_MJ\_CREATE.

**FileId (4 bytes):** A 32-bit unsigned integer that specifies a unique ID that identifies the created file object. The ID MUST be reused after sending a Device Close Response (as specified in [MS-RDPEFS] section 2.2.1.5.2).

### 2.2.2.11 Client Printer Close Response (DR\_PRN\_CLOSE\_RSP)

The client responds to the [DR\\_PRN\\_CLOSE\\_REQ](#) with this message.

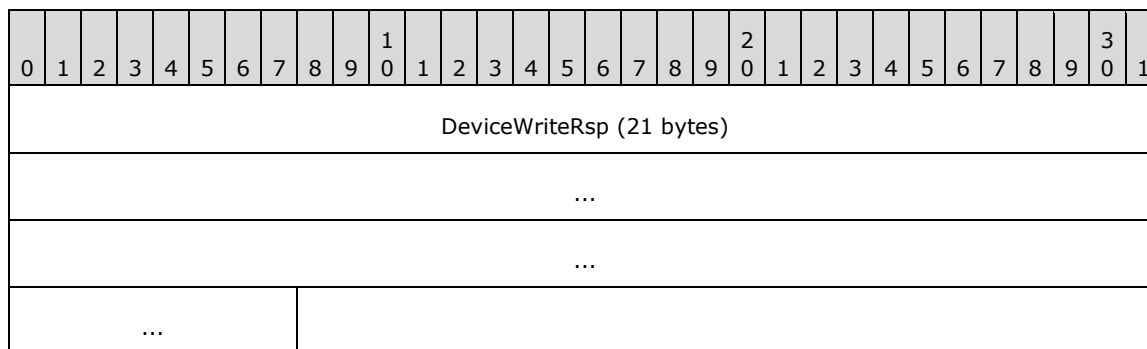


**DeviceIoReply (16 bytes):** A DR\_DEVICE\_IOCTLCOMPLETION header (as specified in [MS-RDPEFS] section 2.2.1.5). The **CompletionId** field of this header MUST match the **CompletionId** of a Device I/O Request message that had the **MajorFunction** field set to IRP\_MJ\_CLOSE.

**Padding (4 bytes):** An array of 4 bytes. This field is reserved for future use.

### 2.2.2.12 Client Printer Write Response (DR\_PRN\_WRITE\_RSP)

The client responds to the [DR\\_PRN\\_WRITE\\_REQ](#) with this message.



**DeviceWriteRsp (21 bytes):** A DR\_WRITE\_RSP request (as specified in [MS-RDPEFS] section 2.2.1.5.4). The **Length** field of this response is equal to the **Length** field of the Write Request (as specified in [MS-RDPEFS] section 2.2.3.3.4).

## 3 Protocol Details

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

### 3.1 Common Details

#### 3.1.1 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: Print Virtual Channel Extension follows the abstract data model specified in [\[MS-RDPEFS\]](#) section 3.1.1. The following abstract entities are specific to this protocol.

##### 3.1.1.1 Cached Printer Config Data

Printer drivers store default printing preferences for each printer. The format of these preferences is driver-specific. For the redirected printers, when a user modifies printing preferences on the **server** side, these modified printing preferences are sent back to the **client** and are cached there. When such a printer is redirected again, this cached data is sent by the client to the server along with other printer information.

The cached configuration information SHOULD be treated as an opaque data BLOB by the client. The client SHOULD treat the printer name and data BLOB pairs as records in a database. The server sends packets to the client to add, rename, delete or update the **printer name** or the data BLOB of an existing record.

##### 3.1.1.2 XPS Mode

When a client redirects printers, it adds a special flag (RDPDR\_PRINTER\_ANNOUNCE\_FLAG\_XPSFORMAT) if the client can handle the XPS format for the given printer. For these printers, the server MAY [<2>](#) choose to use either the **XPS** format or the printer driver-specific format. The server MUST notify the client with the [DR\\_PRN\\_USING\\_XPS \(section 2.2.2.2\)](#) message as described in section [3.3.5.1.2](#) if it chooses to use the XPS format. When this type of message is received, the client marks the printer in XPS mode. Redirection of XPS data using the Remote Desktop Protocol: XML Paper Specification (XPS) Print Virtual Channel Extension is described in [\[MS-RDPEXPS\]](#).

### 3.1.2 Timers

No common timers are used.

### 3.1.3 Initialization

The Print Virtual Channel Extension protocol MUST be initialized before the printer can be redirected.

### 3.1.4 Higher-Layer Triggered Events

No higher-layer triggered events are used.

### 3.1.5 Message Processing Events and Sequencing Rules

The common message processing events and rules that are described in [\[MS-RDPEFS\]](#) section 3.1.5 apply to this protocol. For client-specific and server-specific message processing, see sections [3.2.5](#) and [3.3.5](#).

### 3.1.6 Timer Events

No common timer events are used.

### 3.1.7 Other Local Events

There are no common local events.

## 3.2 Client Details

### 3.2.1 Abstract Data Model

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

### 3.2.2 Timers

No timers are used.

### 3.2.3 Initialization

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

### 3.2.4 Higher-Layer Triggered Events

No client higher-layer triggered events are used.

### 3.2.5 Message Processing Events and Sequencing Rules

#### 3.2.5.1 Printer Redirection Messages

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

After the Remote Desktop Protocol: File System Virtual Channel Extension completes its initialization, the client MUST send a DR\_CORE\_DEVICELIST\_ANNOUNCE\_REQ message to the server with information for various devices. This message is specified in [\[MS-RDPEFS\]](#) section 2.2.2.9. The Print Virtual Channel Extension prepares the printer device information that goes into this packet. The printer-specific structure is specified in section [2.2.2.1](#).

The printer redirection extension enumerates the **printer queues** and manually configured printers (**manual printer redirection**) and determines the devices to be redirected. For each printer, the client collects the following information and prepares DR\_PRN\_DEVICE\_ANNOUNCE (section 2.2.2.1) packet. This packet contains the following:

- The name of the printer queue assigned by the user.
- The name of the printer driver assigned by the driver manufacturer.
- Various flags associated with the printer.

- Printer cache data, if this data exists. See section [3.1.1.1](#).

### 3.2.5.1.2 Processing a Printer Set XPS Mode Message

This message MUST be received only for the printers that were marked with RDPDR\_PRINTER\_ANNOUNCE\_FLAG\_XPSFORMAT. When a server chooses to use an **XPS** driver for the redirected printer, it notifies the client using this message. The client MUST then mark the printer to be in XPS mode (see section [3.1.1.2](#)). When printer data arrives for these printers, it MUST be treated as an **XPS file** and not a **PRN file**. For other printers, the I/O data MUST be treated as a PRN file.

### 3.2.5.1.3 Processing an Add Printer Cachedata Message

This message is received when a **printer queue** is manually installed on the server on the client's redirected port. The name of the port is stored in the **PortDosName** field. The client SHOULD store this manual printer configuration in its persistent storage, and it MUST use this configuration to redirect the printer the next time the client connects to the same server.

### 3.2.5.1.4 Processing an Update Printer Cachedata Message

This message is received when the configuration information of a redirected **printer queue** is updated on the server. The client MUST store the updated **CachedPrinterConfigData** field (see section [3.1.1.1](#)), and it MUST use it when redirecting the printer next time.

### 3.2.5.1.5 Processing a Delete Printer Cachedata Message

This message is received when a redirected **printer queue** is deleted by the server. The client MUST delete the cached configuration data previously stored for this printer.

### 3.2.5.1.6 Processing a Rename Printer Cachedata Message

This message is received when the user renames the redirected printer on the server. The client MUST modify its local cached data for this printer to reflect this change, and the client MUST rename the local **printer queue**.

### 3.2.5.1.7 Processing a Create Request Message

When the server starts a print job on a redirected printer, before sending the actual printing data it sends a create request. The client MUST prepare the printer specified by the **DeviceId** field to receive the printer output.

### 3.2.5.1.8 Processing a Close Request Message

When the server finishes sending a print job to a redirected printer, it sends a close request. The client MUST end the printing process initiated when the create request was made.

### 3.2.5.1.9 Processing a Write Request Message

Multiple write requests are received along with the printing data. These write requests are sent between create and close requests. The client MUST send the data associated with the write requests to the corresponding **printer queue**. The format of this data could be either printer driver-specific (non-XPS mode) or **XPS** format (XPS mode).

### 3.2.5.1.10 Sending a Printer Create Response Message

The client MUST fill out the various members of the DR\_DEVICE\_IOCOMPLETION message (as specified in [\[MS-RDPEFS\]](#) section 2.2.1.5) as shown here.

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 header:

- The **CompletionId** field MUST be set to match the **CompletionId** field from the corresponding I/O request (section [3.2.5.1.7](#)).
- The **DeviceId** field MUST be set to match the corresponding **DeviceId** field from the I/O request.
- The **IoStatus** field MUST be set to the NTSTATUS value indicating the result of the operation. NTSTATUS codes are specified in [\[MS-ERREF\]](#) section 2.3.

### 3.2.5.1.11 Sending a Printer Close Response Message

The client MUST fill out the various members of the DR\_DEVICE\_IOCOMPLETION message (as specified in [\[MS-RDPEFS\]](#) section 2.2.1.5) as shown here.

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 header:

- The **CompletionId** field MUST be set to match the **CompletionId** field from the corresponding I/O request (section [3.2.5.1.8](#)).
- The **DeviceId** MUST be set to match the corresponding **DeviceId** from the I/O request.
- The **IoStatus** field MUST be set to the NTSTATUS value indicating the result of the operation. NTSTATUS codes are specified in [\[MS-ERREF\]](#) section 2.3.

### 3.2.5.1.12 Sending a Printer Write Response Message

The client MUST fill out the various members of the DR\_DEVICE\_IOCOMPLETION message (as specified in [\[MS-RDPEFS\]](#) section 2.2.1.5) as shown here.

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 header:

- The **CompletionId** field MUST be set to match the **CompletionId** field from the corresponding I/O request (section [3.2.5.1.9](#)).
- The **DeviceId** field MUST be set to match the corresponding **DeviceId** field from the I/O request.
- The **IoStatus** field MUST be set to the NTSTATUS value indicating the result of the operation.

The client MUST set the **Length** field of DR\_WRITE\_RSP (see [\[MS-RDPEFS\]](#) section 2.2.1.5.4) to the bytes written successfully to the printer device. NTSTATUS codes are specified in [\[MS-ERREF\]](#) section 2.3.

## 3.2.6 Timer Events

No timer events are used.

## 3.2.7 Other Local Events

There are no other local events.

## 3.3 Server Details

### 3.3.1 Abstract Data Model

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

### 3.3.2 Timers

No timers are used.

### 3.3.3 Initialization

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

### 3.3.4 Higher-Layer Triggered Events

The **server** sends various messages to the **client** in response to the following application and user-driven events:

- When a printer is installed manually for the redirected port, the server sends an add printer cachedata message to the client (see section [3.3.5.1.3](#)).
- When the user modifies the redirected printer queue configuration, the server sends an update printer cachedata message to the client (see section [3.3.5.1.4](#)).
- When the user deletes the redirected printer queue, the server sends a delete printer cachedata message to the client (see section [3.3.5.1.5](#)).
- When the user renames the redirected printer queue, the server sends a rename printer cachedata message to the client (see section [3.3.5.1.6](#)).
- When an application running on the server opens the redirected printer queue for printing, the server sends a printer create request to the client (see section [3.3.5.1.7](#)).
- When an application running on the server closes the redirected printer queue, the server sends a printer close request to the client (see section [3.3.5.1.8](#)).
- When an application running on the server sends data to the redirected printer queue for printing, the server sends printer write requests to the client (see section [3.3.5.1.9](#)).

### 3.3.5 Message Processing Events and Sequencing Rules

#### 3.3.5.1 Printer Redirection Messages

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

After receiving a [Client Device List Announce Request \(section 2.2.2.1\)](#), the server creates a printer queue to represent each client-side printer device. The server MUST use the appropriate driver for this redirected printer queue. The server MAY use a pseudo driver that produces XPS-format output if the RDPDR\_PRINTER\_ANNOUNCE\_FLAG\_XPSFORMAT flag is supplied for the device. If this flag is not supplied, the server MAY use the **DriverName** field specified in this message to choose the appropriate printer driver on the server. If this message contains a **CachedPrinterConfigData** field, the server MUST use the information contained to configure the instance of the printer queue. The server MAY use the **PrinterName** field specified in the message to generate the name for the redirected printer queue.

#### **3.3.5.1.2 Sending a Printer Set XPS Mode Message**

If the client printer supports **XPS** format, it indicates this support using the RDPDR\_PRINTER\_ANNOUNCE\_FLAG\_XPSFORMAT flag when redirecting this printer (see section [2.2.2.1](#)). For these redirected printers, the server can choose to send print job output in XPS format. **<3>** If the server chooses to send print data in XPS format, the server MUST send this message to the client prior to sending any data in the Printer Write Request (section [2.2.2.9](#)) message.

#### **3.3.5.1.3 Sending an Add Printer Cachedata Message**

When a user manually installs a printer driver for the redirected port, the server collects the configuration data for such a **printer queue** and sends it to the client using this message.

#### **3.3.5.1.4 Sending an Update Printer Cachedata Message**

When a user modifies the configuration data, such as the printing preferences for the redirected **printer queue**, the server sends this message along with the configuration data to inform the client.

#### **3.3.5.1.5 Sending Delete Printer Cachedata**

When a user deletes the **printer queue**, the server sends this message to the client.

#### **3.3.5.1.6 Sending a Rename Printer Cachedata Message**

When a user manually renames the redirected **printer queue**, the server sends this message to the client specifying old and new **printer names**.

#### **3.3.5.1.7 Sending a Printer Create Request Message**

When a server application starts printing to the redirected **printer queue**, the server MUST send this message to the client before sending any printer data so that the client can prepare for receiving printer data.

#### **3.3.5.1.8 Sending a Printer Close Request Message**

When the server application is done with a printing job, the server MUST send this message to the client.

#### **3.3.5.1.9 Sending a Printer Write Request Message**

As a server application prints to the redirected **printer queue**, the server MUST send the printer job data to the client. A print job can be distributed over multiple write requests. This message MUST be sent only after the [DR\\_PRN\\_CREATE\\_RSP \(section 2.2.2.10\)](#) message is received.

#### **3.3.5.1.10 Processing a Printer Create Response Message**



The server MUST inspect the result of the printer create operation and abort/fail the printer operation if the create request failed.

#### **3.3.5.1.11 Processing a Printer Close Response Message**

The server MUST ignore this message after validating it according to the rules specified in [\[MS-RDPEFS\]](#) section 3.1.5.2.

#### **3.3.5.1.12 Processing a Printer Write Response Message**

The server MUST inspect the amount of data written and send the next packet of data if there are any packets remaining.

### **3.3.6 Timer Events**

There are no timer events used.

### **3.3.7 Other Local Events**

There are no other local events.

## 4 Protocol Examples

### 4.1 Annotations for Printer Redirection

#### 4.1.1 Client Printer Device (DR\_PRN\_DEVICE\_ANNOUNCE)

The **client** sends this message to the **server** to indicate the list of printers that it wants to redirect.

```
Channel Name = RDPDR, 264, Client to server
00000000 72 44 41 44 03 00 00 00 04 00 00 00 04 00 00 00
00000010 50 52 4e 34 00 00 00 00 50 00 00 00 10 00 00 00
00000020 00 00 00 00 00 00 00 00 1c 00 00 00 1c 00 00 00
00000030 00 00 00 00 41 00 70 00 6f 00 6c 00 6c 00 6f 00
00000040 20 00 50 00 2d 00 31 00 32 00 30 00 30 00 00 00
00000050 41 00 70 00 6f 00 6c 00 6c 00 6f 00 20 00 50 00
00000060 2d 00 31 00 32 00 30 00 30 00 00 00 04 00 00 00
00000070 03 00 00 00 50 52 4e 33 00 00 00 00 74 00 00 00
00000080 12 00 00 00 00 00 00 00 00 00 00 00 2e 00 00 00
00000090 2e 00 00 00 00 00 00 00 43 00 61 00 6e 00 6f 00
000000a0 6e 00 20 00 42 00 75 00 62 00 62 00 6c 00 65 00
000000b0 2d 00 4a 00 65 00 74 00 20 00 42 00 4a 00 2d 00
000000c0 33 00 30 00 00 00 43 00 61 00 6e 00 6f 00 6e 00
000000d0 20 00 42 00 75 00 62 00 62 00 6c 00 65 00 2d 00
000000e0 4a 00 65 00 74 00 20 00 42 00 4a 00 2d 00 33 00
000000f0 30 00 00 00 02 00 00 00 02 00 00 00 4c 50 54 31
00000100 00 00 00 00 00 00 00 00

72 44 -> RDPDR_CTYP_CORE = 0x4472
41 44 -> PAKID_CORE_DEVICELIST_ANNOUNCE = 0x4441
03 00 00 00 -> DeviceCount = 0x00000003
04 00 00 00 -> DeviceType (RDPDR_DTYP_PRINT) = 0x00000004
04 00 00 00 -> DeviceId = 0x00000004
50 52 4e 34 00 00 00 00 -> PreferredDosName (8 characters)
50 00 00 00 -> DeviceDataLength = 0x00000050
10 00 00 00 -> Flags (RDPDR_PRINTER_ANNOUNCE_FLAG_XPSFORMAT)
= 0x00000010
00 00 00 00 -> CodePage = 0x00000000
00 00 00 00 -> PnpNameLen = 0x00000000
1c 00 00 00 -> DriverNameLen = 0x0000001c
1c 00 00 00 -> PrinterNameLen = 0x0000001c
00 00 00 00 -> CachedFieldsLen = 0x00000000
41 00 70 00 6f 00 6c 00 6c 00 6f 00 20 00 50 00 2d 00 31 00 32 00 30 00 30 00 00 00
-> DriverName (28 bytes)
41 00 70 00 6f 00 6c 00 6c 00 6f 00 20 00 50 00 2d 00 31 00 32 00 30 00 30 00 00 00
-> PrinterName (28 bytes)

04 00 00 00 -> DeviceType (RDPDR_DTYP_PRINT) = 0x00000004
03 00 00 00 -> DeviceId = 0x00000003
50 52 4e 33 00 00 00 00 -> PreferredDosName (8 characters)
74 00 00 00 -> DeviceDataLength = 0x00000074
12 00 00 00 -> Flags (RDPDR_PRINTER_ANNOUNCE_FLAG_DEFAULTPRINTER |
RDPDR_PRINTER_ANNOUNCE_FLAG_XPSFORMAT) = 0x00000012
00 00 00 00 -> CodePage = 0x00000000
00 00 00 00 -> PnpNameLen = 0x00000000
2e 00 00 00 -> DriverNameLen = 0x0000002e
2e 00 00 00 -> PrinterNameLen = 0x0000002e
00 00 00 00 -> CachedFieldsLen = 0x00000000
43 00 61 00 6e 00 6f 00 6e 00 20 00 42 00 75 00 62 00 62 00 6c 00 65 00 2d 00
4a 00 65 00 74 00 20 00 42 00 4a 00 2d 00 33 00 30 00 00 00
-> DriverName (46 bytes)
43 00 61 00 6e 00 6f 00 6e 00 20 00 42 00 75 00 62 00 62 00 6c 00 65 00 2d 00
4a 00 65 00 74 00 20 00 42 00 4a 00 2d 00 33 00 30 00 00 00
-> PrinterName (46 bytes)

02 00 00 00 -> DeviceType (RDPDR_DTYP_PARALLEL) = 0x00000002
```

```

02 00 00 00          -> DeviceId                = 0x00000002
4c 50 54 31 00 00 00 00 -> PreferredDosName (8 characters)
00 00 00 00          -> DeviceDataLength        = 0x00000000

```

#### 4.1.2 Server Printer Set XPS Mode (DR\_PRN\_USING\_XPS)

When the server decides to use the **XPS** format for the output, it sends this message to prepare the client.

```

Channel Name = RDPDR, 12, Server to client
00000000 52 50 43 55 01 00 00 00 f8 5b fa 7f

52 50          -> RDPDR CTYP PRN          = 0x5052
43 55          -> PAKID_PRN_USING_XPS      = 0x5543
01 00 00 00    -> PrinterId                = 0x00000001
f8 5b fa 7f    -> Flags                    = 0x7ffa5bf8

```

#### 4.1.3 Add Printer Cachedata (DR\_PRN\_ADD\_CACHEDATA)

When a printer is manually installed on the redirected port, the server sends this message to allow the client an opportunity to cache the configuration.

```

Channel Name = RDPDR, 116, Server to client
00000000 52 50 43 50 01 00 00 00 43 4f 4d 32 00 00 3a 00
00000010 00 00 00 00 2a 00 00 00 2a 00 00 00 00 00 00 00
00000020 42 00 72 00 6f 00 74 00 68 00 65 00 72 00 20 00
00000030 44 00 43 00 50 00 2d 00 31 00 30 00 30 00 30 00
00000040 20 00 55 00 53 00 42 00 00 00 42 00 72 00 6f 00
00000050 74 00 68 00 65 00 72 00 20 00 44 00 43 00 50 00
00000060 2d 00 31 00 30 00 30 00 30 00 20 00 55 00 53 00
00000070 42 00 00 00

52 50          -> RDPDR CTYP PRN          = 0x5052
43 50          -> PAKID_PRN_CACHE_DATA     = 0x5043
01 00 00 00    -> RDPDR ADD PRINTER EVENT  = 0x00000001
43 4f 4d 32 00 00 3a 00 -> PortDosName (64 bits)
00 00 00 00    -> PnPNameLen              = 0x00000000
2a 00 00 00    -> DriverNameLen           = 0x0000002a
2a 00 00 00    -> PrintNameLen           = 0x0000002a
00 00 00 00    -> CachedFieldsLen        = 0x00000000
42 00 72 00 6f 00 74 00 68 00 65 00 72 00 20 00 44 00 43 00 50
00 2d 00 31 00 30 00 30 00 30 00 20 00 55 00 53 00 42 00 00 00
-> DriverName (variable - 42 bytes)
    = "Brother DCP-1000 USB"
42 00 72 00 6f 00 74 00 68 00 65 00 72 00 20 00 44 00 43 00 50
00 2d 00 31 00 30 00 30 00 30 00 20 00 55 00 53 00 42 00 00 00
-> PrinterName (variable - 42 bytes)
    = "Brother DCP-1000 USB"

```

#### 4.1.4 Update Printer Cachedata (DR\_PRN\_UPDATE\_CACHEDATA)

When printing preferences or other configuration settings are modified on the server for a redirected printer, the server notifies the client with the configuration data.

```

Channel Name = RDPDR, 16330, Server to client
00000000 52 50 43 50 02 00 00 00 2a 00 00 00 90 3f 00 00
00000010 42 00 72 00 6f 00 74 00 68 00 65 00 72 00 20 00

```

```

00000020 44 00 43 00 50 00 2d 00 31 00 30 00 30 00 30 00
00000030 20 00 55 00 53 00 42 00 00 00 48 00 00 00 00 00
00000040 00 00 94 20 00 00 00 00 00 00 37 00 00 00 00 00

52 50          -> RDPDR_CTYP_PRN          = 0x5052
43 50          -> PAKID_PRN_CACHE_DATA     = 0x5043
02 00 00 00    -> RDPDR_UPDATE_PRINTER_EVENT = 0x00000002
2a 00 00 00    -> PrinterNameLen          = 0x0000002a
90 3f 00 00    -> ConfigDataLen            = 0x000039e4
42 00 72 00 6f 00 74 00 68 00 65 00 72 00 20 00 44 00 43 00 50
00 2d 00 31 00 30 00 30 00 30 00 20 00 55 00 53 00 42 00 00 00
                                -> PrinterName (42 bytes)
                                    = "Brother DCP-1000 USB"
48 00 00 00 00 00 ... -> ConfigData (variable)

```

#### 4.1.5 Delete Printer Cachedata (DR\_PRN\_DELETE\_CACHEDATA)

The redirected printer was deleted manually.

```

Channel Name = RDPDR, 54, Server to client
00000000 52 50 43 50 03 00 00 00 2a 00 00 00 42 00 72 00
00000010 6f 00 74 00 68 00 65 00 72 00 20 00 44 00 43 00
00000020 50 00 2d 00 31 00 30 00 30 00 30 00 20 00 55 00
00000030 53 00 42 00 00 00

52 50          -> RDPDR_CTYP_PRN          = 0x5052
43 50          -> PAKID_PRN_CACHE_DATA     = 0x5043
03 00 00 00    -> RDPDR_DELETE_PRINTER_EVENT
                                                = 0x00000003
2a 00 00 00    -> PrinterNameLen          = 0x0000002a
42 00 72 00 6f 00 74 00 68 00 65 00 72 00 20 00 44 00 43 00 50
00 2d 00 31 00 30 00 30 00 30 00 20 00 55 00 53 00 42 00 00 00
                                -> PrinterName (42 bytes)
                                    = "Brother DCP-1000 USB"

```

#### 4.1.6 Rename Printer Cachedata (DR\_PRN\_RENAME\_CACHEDATA)

The redirected printer was renamed.

```

Channel Name = RDPDR, 120, Server to client
00000000 52 50 43 50 04 00 00 00 2a 00 00 00 3e 00 00 00
00000010 42 00 72 00 6f 00 74 00 68 00 65 00 72 00 20 00
00000020 44 00 43 00 50 00 2d 00 31 00 30 00 30 00 30 00
00000030 20 00 55 00 53 00 42 00 00 00 42 00 72 00 6f 00
00000040 74 00 68 00 65 00 72 00 20 00 44 00 43 00 50 00
00000050 2d 00 31 00 30 00 30 00 30 00 20 00 55 00 53 00
00000060 42 00 20 00 28 00 72 00 65 00 6e 00 61 00 6d 00
00000070 65 00 64 00 29 00 00 00

52 50          -> RDPDR_CTYP_PRN          = 0x5052
43 50          -> PAKID_PRN_CACHE_DATA     = 0x5043
04 00 00 00    -> RDPDR_RENAME_PRINTER_EVENT
                                                = 0x00000004
2a 00 00 00    -> OldPrinterNameLen          = 0x0000002a
3e 00 00 00    -> NewPrinterNameLen          = 0x0000003e
42 00 72 00 6f 00 74 00 68 00 65 00 72 00 20 00 44 00 43 00 50
00 2d 00 31 00 30 00 30 00 30 00 20 00 55 00 53 00 42 00 00 00
                                -> OldPrinterName (42 bytes)
                                    = "Brother DCP-1000 USB"
42 00 72 00 6f 00 74 00 68 00 65 00 72 00 20 00 44 00 43 00 50
00 2d 00 31 00 30 00 30 00 30 00 20 00 55 00 53 00 42 00 20 00

```

```

28 00 72 00 65 00 6e 00 61 00 6d 00 65 00 64 00 29 00 00 00
-> NewPrinterName (62 bytes)
    = "Brother DCP-1000 USB (renamed)"

```

#### 4.1.7 Server Printer Create Request (DR\_PRN\_CREATE\_REQ)

The server sends a create request before starting a print job.

```

Channel Name = RDPDR, 56, Server to client
00000000 72 44 52 49 02 00 00 00 00 00 00 00 00 00 00 00
00000010 00 00 00 00 00 00 00 00 9f 01 12 00 00 00 00 00
00000020 00 00 00 00 00 00 00 00 00 00 03 00 00 00 01 00 00 00
00000030 40 00 00 00 00 00 00 00

72 44          -> RDPDR_CTYP_CORE           = 0x4472
52 49          -> PAKID_CORE_DEVICE_IOREQUEST    = 0x4952
02 00 00 00    -> DeviceId                = 0x00000002
00 00 00 00    -> FileId                  = 0x00000000
00 00 00 00    -> CompletionId           = 0x00000000
00 00 00 00    -> MajorFunction (IRP MJ CREATE) = 0x00000000
00 00 00 00    -> MinorFunction           = 0x00000000
9f 01 12 00    -> DesiredAccess           = 0x0012019f
00 00 00 00 00 00 00 00 -> AllocationSize (8 bytes)
00 00 00 00    -> FileAttributes          = 0x00000000
03 00 00 00    -> SharedAccess            = 0x00000003
01 00 00 00    -> Disposition             = 0x00000001
40 00 00 00    -> CreateOptions           = 0x00000004
00 00 00 00    -> PathLength              = 0x00000000

```

#### 4.1.8 Server Printer Close Request (DR\_PRN\_CLOSE\_REQ)

The server sends a close message when it finishes sending a print job.

```

Channel Name = RDPDR, 56, Server to client
00000000 72 44 52 49 02 00 00 00 00 00 00 00 00 00 00 00
00000010 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000030 00 00 00 00 00 00 00 00

72 44          -> RDPDR_CTYP_CORE           = 0x4472
52 49          -> PAKID_CORE_DEVICE_IOREQUEST    = 0x4952
02 00 00 00    -> DeviceId                = 0x00000002
00 00 00 00    -> FileId                  = 0x00000000
00 00 00 00    -> CompletionId           = 0x00000000
02 00 00 00    -> MajorFunction (IRP MJ CLOSE)   = 0x00000002
00 00 00 00    -> MinorFunction           = 0x00000000
00 00 .....    -> Padding (32 bytes)

```

#### 4.1.9 Server Printer Write Request (DR\_PRN\_WRITE\_REQ)

The server sends multiple write requests to send the print job to the client.

```

Channel Name = RDPDR, 65592, Server to client
00000000 72 44 52 49 02 00 00 00 00 00 00 00 00 00 00 00
00000010 04 00 00 00 00 00 00 00 00 00 01 00 00 00 00 00
00000020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000030 00 00 00 00 00 00 00 00 .....

```

```

72 44          -> RDPDR_CTYP_CORE           = 0x4472
52 49          -> PAKID_CORE_DEVICE_IOREQUEST    = 0x4952
02 00 00 00    -> DeviceId           = 0x00000002
00 00 00 00    -> FileId             = 0x00000000
00 00 00 00    -> CompletionId      = 0x00000000
04 00 00 00    -> MajorFunction (IRP_MJ_WRITE) = 0x00000004
00 00 00 00    -> MinorFunction     = 0x00000000
00 00 01 00    -> Length             = 0x00010000
00 00 00 00 00 00 00 00 -> Offset (8 bytes)
00 00 ....     -> Padding (20 bytes)

```

(skipped)

#### 4.1.10 Client Printer Create Response (DR\_PRN\_CREATE\_RSP)

The client responds to the create request from the server.

```

Channel Name = RDPDR, 20, Client to server
00000000 72 44 43 49 02 00 00 00 00 00 00 00 00 00 00 00
00000010 00 00 00 00

```

```

72 44          -> RDPDR_CTYP_CORE           = 0x4472
43 49          -> PAKID_CORE_DEVICE_IOCOMPLETION = 0x4943
02 00 00 00    -> DeviceId           = 0x00000002
00 00 00 00    -> CompletionId      = 0x00000000
00 00 00 00    -> IoStatus           = 0x00000000
00 00 00 00    -> FileId             = 0x00000000

```

#### 4.1.11 Client Printer Close Response (DR\_PRN\_CLOSE\_RSP)

The client responds to the close request.

```

Channel Name = RDPDR, 20, Client to server
00000000 72 44 43 49 02 00 00 00 00 00 00 00 00 00 00 00
00000010 00 00 00 00

```

```

72 44          -> RDPDR_CTYP_CORE           = 0x4472
43 49          -> PAKID_CORE_DEVICE_IOCOMPLETION = 0x4943
02 00 00 00    -> DeviceId           = 0x00000002
00 00 00 00    -> CompletionId      = 0x00000000
00 00 00 00    -> IoStatus           = 0x00000000
00 00 00 00    -> Padding             = 0x00000000

```

#### 4.1.12 Client Printer Write Response (DR\_PRN\_WRITE\_RSP)

The client responds to the write request.

```

Channel Name = RDPDR, 21, Client to server
00000000 72 44 43 49 02 00 00 00 00 00 00 00 00 00 00 00
00000010 00 00 01 00 00

```

```

72 44          -> RDPDR_CTYP_CORE           = 0x4472
43 49          -> PAKID_CORE_DEVICE_IOCOMPLETION = 0x4943
02 00 00 00    -> DeviceId           = 0x00000002
00 00 00 00    -> CompletionId      = 0x00000000
00 00 00 00    -> IoStatus           = 0x00000000

```

00 00 01 00

-> Length

= 0x00010000

## **5 Security**

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

There are no security considerations for Print Virtual Channel Extension.

### **5.2 Index of Security Parameters**

There are no security parameters in the Remote Desktop Protocol: Print Virtual Channel Extension.



## 6 Appendix A: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include updates to those products.

- 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
- Windows Server 2019 operating system
- Windows Server 2022 operating system
- Windows 11 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](#): Applicable Windows Server releases look at the **driver name** specified by the client and perform an exact string comparison to find the matching printer driver on the server. The matching printer driver might not be available on the server for various reasons. If the matching printer driver is not available, the server cannot reliably load an appropriate printer driver, and it fails to redirect the printer. This is an early design limitation, which was fixed for Windows Server 2008.

For printer redirection in applicable Windows Server releases (this is not supported in Windows Server 2003), **XPS**-based drivers are recommended. When a client supports the XPS format (that is, the RDPDR\_PRINTER\_ANNOUNCE\_FLAG\_XPSFORMAT flag is set in the [Client Device List Announce Request \(section 2.2.2.1\)](#) message), Windows implementations use the terminal server-specific XPS-capable printer driver and ignore the **DriverName** field.

[<2> Section 3.1.1.2](#): XPS Mode is not supported for Windows 2000, Windows XP, and Windows Server 2003.

[<3> Section 3.3.5.1.2](#): Windows has the option of using XPS format or not, based on group policy that instructs the implementation to use the XPS format or a matching printer driver.

## 7 Change Tracking

This section identifies changes that were made to this document since the last release. Changes are classified as Major, Minor, or None.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements.
- A document revision that captures changes to protocol functionality.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **None** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the relevant technical content is identical to the last released version.

The changes made to this document are listed in the following table. For more information, please contact [dochelp@microsoft.com](mailto:dochelp@microsoft.com).

Section	Description	Revision class
<a href="#">6</a> Appendix A: Product Behavior	Updated for this version of Windows Client.	Major

## 8 Index

### A

Abstract data model  
client ([section 3.1.1](#) 19, [section 3.2.1](#) 20)  
server ([section 3.1.1](#) 19, [section 3.3.1](#) 23)  
[Add printer cachedata \(DR\\_PRN\\_ADD\\_CACHEDATA\)](#)  
27  
[Annotations - printer redirection](#) 26  
[Applicability](#) 8

### C

[Cached printer config data](#) 19  
[Capability negotiation](#) 8  
[Change tracking](#) 35  
Client  
abstract data model ([section 3.1.1](#) 19, [section 3.2.1](#) 20)  
higher-layer triggered events ([section 3.1.4](#) 19, [section 3.2.4](#) 20)  
initialization ([section 3.1.3](#) 19, [section 3.2.3](#) 20)  
local events ([section 3.1.7](#) 20, [section 3.2.7](#) 23)  
message processing ([section 3.1.5](#) 20, [section 3.2.5](#) 20)  
[other local events](#) 23  
[printer redirection messages](#) 20  
sequencing rules ([section 3.1.5](#) 20, [section 3.2.5](#) 20)  
timer events ([section 3.1.6](#) 20, [section 3.2.6](#) 23)  
timers ([section 3.1.2](#) 19, [section 3.2.2](#) 20)  
Client printer  
[close response \(DR\\_PRN\\_CLOSE\\_RSP\)](#) 30  
[device \(DR\\_PRN\\_DEVICE\\_ANNOUNCE\)](#) 26  
[write response \(DR\\_PRN\\_WRITE\\_RSP\)](#) 30  
[Client printer create response \(DR\\_PRN\\_CREATE\\_RSP\)](#) 30  
[Common data types](#) 9  
[Common Data Types message](#) 9

### D

Data model - abstract  
client ([section 3.1.1](#) 19, [section 3.2.1](#) 20)  
server ([section 3.1.1](#) 19, [section 3.3.1](#) 23)  
[Data types](#) 9  
[Delete printer cachedata \(DR\\_PRN\\_DELETE\\_CACHEDATA\)](#) 28  
[DR\\_PRN\\_ADD\\_CACHEDATA](#) 27  
[DR\\_PRN\\_ADD\\_CACHEDATA packet](#) 12  
[DR\\_PRN\\_CLOSE\\_REQ](#) 29  
[DR\\_PRN\\_CLOSE\\_REQ packet](#) 17  
[DR\\_PRN\\_CLOSE\\_RSP](#) 30  
[DR\\_PRN\\_CLOSE\\_RSP packet](#) 18  
[DR\\_PRN\\_CREATE\\_REQ](#) 29  
[DR\\_PRN\\_CREATE\\_REQ packet](#) 16  
[DR\\_PRN\\_CREATE\\_RSP](#) 30  
[DR\\_PRN\\_CREATE\\_RSP packet](#) 17  
[DR\\_PRN\\_DELETE\\_CACHEDATA](#) 28  
[DR\\_PRN\\_DELETE\\_CACHEDATA packet](#) 14  
[DR\\_PRN\\_DEVICE\\_ANNOUNCE](#) 26  
[DR\\_PRN\\_DEVICE\\_ANNOUNCE packet](#) 10

[DR\\_PRN\\_RENAME\\_CACHEDATA](#) 28  
[DR\\_PRN\\_RENAME\\_CACHEDATA packet](#) 15  
[DR\\_PRN\\_UPDATE\\_CACHEDATA](#) 27  
[DR\\_PRN\\_UPDATE\\_CACHEDATA packet](#) 14  
[DR\\_PRN\\_USING\\_XPS](#) 27  
[DR\\_PRN\\_USING\\_XPS packet](#) 12  
[DR\\_PRN\\_WRITE\\_REQ](#) 29  
[DR\\_PRN\\_WRITE\\_REQ packet](#) 17  
[DR\\_PRN\\_WRITE\\_RSP](#) 30  
[DR\\_PRN\\_WRITE\\_RSP packet](#) 18

### E

[Examples](#) 26

### F

[Fields - vendor-extensible](#) 8

### G

[Glossary](#) 6

### H

Higher-layer triggered events  
client ([section 3.1.4](#) 19, [section 3.2.4](#) 20)  
server ([section 3.1.4](#) 19, [section 3.3.4](#) 23)

### I

[Implementer - security considerations](#) 32  
[Index of security parameters](#) 32  
[Informative references](#) 7  
Initialization  
client ([section 3.1.3](#) 19, [section 3.2.3](#) 20)  
server ([section 3.1.3](#) 19, [section 3.3.3](#) 23)  
[Introduction](#) 6

### L

Local events  
client ([section 3.1.7](#) 20, [section 3.2.7](#) 23)  
server ([section 3.1.7](#) 20, [section 3.3.7](#) 25)

### M

Message processing  
client ([section 3.1.5](#) 20, [section 3.2.5](#) 20)  
server ([section 3.1.5](#) 20, [section 3.3.5](#) 23)  
Messages  
[Common Data Types](#) 9  
[data types](#) 9  
[overview](#) 9  
[printer redirection messages](#) 10  
[syntax](#) 9  
[transport](#) 9

### N

[Normative references](#) 7

## O

Other local events  
[client](#) 23  
[server](#) 25  
[Overview \(synopsis\)](#) 7

## P

[Parameters - security index](#) 32  
[Preconditions](#) 8  
[Prerequisites](#) 8  
[Print Virtual Channel Extension](#) 7  
[Printer redirection- annotations](#) 26  
[Printer redirection messages](#) 10  
[Printer redirection messages - client](#) 20  
[Printer redirection messages - server](#) 23  
Processing message  
[add printer cachedata](#) 21  
[client device list announce request](#) 23  
[close request](#) 21  
[create request](#) 21  
[delete printer cachedata](#) 21  
[printer close response](#) 25  
[printer create response](#) 24  
[printer set XPS mode](#) 21  
[printer write response](#) 25  
[rename printer cachedata](#) 21  
[update printer cachedata](#) 21  
[write request](#) 21  
[Product behavior](#) 33  
Protocol Details  
[overview](#) 19

## R

[RDPEFS - subprotocol within](#) 7  
[References](#) 7  
[informative](#) 7  
[normative](#) 7  
[Relationship to other protocols](#) 8  
[Rename printer cachedata  
\(DR\\_PRN\\_RENAME\\_CACHEDATA\)](#) 28

## S

Security  
[implementer considerations](#) 32  
[parameter index](#) 32  
Sending message  
[add printer cachedata](#) 24  
[add printer create request](#) 24  
[client device list announce request](#) 20  
[delete printer cachedata](#) 24  
[printer close request](#) 24  
[printer close response](#) 22  
[printer create response](#) 21  
[printer set XPS mode](#) 24  
[printer write response](#) 22  
[rename printer cachedata](#) 24  
[sending printer write request](#) 24  
[update printer cachedata](#) 24  
Sequencing rules  
client ([section 3.1.5](#) 20, [section 3.2.5](#) 20)

server ([section 3.1.5](#) 20, [section 3.3.5](#) 23)  
Server  
abstract data model ([section 3.1.1](#) 19, [section 3.3.1](#) 23)  
higher-layer triggered events ([section 3.1.4](#) 19, [section 3.3.4](#) 23)  
initialization ([section 3.1.3](#) 19, [section 3.3.3](#) 23)  
local events ([section 3.1.7](#) 20, [section 3.3.7](#) 25)  
message processing ([section 3.1.5](#) 20, [section 3.3.5](#) 23)  
[other local events](#) 25  
[printer redirection messages](#) 23  
sequencing rules ([section 3.1.5](#) 20, [section 3.3.5](#) 23)  
timer events ([section 3.1.6](#) 20, [section 3.3.6](#) 25)  
timers ([section 3.1.2](#) 19, [section 3.3.2](#) 23)  
Server printer  
[close request \(DR\\_PRN\\_CLOSE\\_REQ\)](#) 29  
[create Request \(DR\\_PRN\\_CREATE\\_REQ\)](#) 29  
[set XPS Mode \(DR\\_PRN\\_USING\\_XPS\)](#) 27  
[Server printer write request \(DR\\_PRN\\_WRITE\\_REQ\)](#) 29  
[SERVER\\_PRINTER\\_CACHE\\_EVENT packet](#) 9  
[Standards assignments](#) 8  
[Syntax](#) 9

## T

Timer events  
client ([section 3.1.6](#) 20, [section 3.2.6](#) 23)  
server ([section 3.1.6](#) 20, [section 3.3.6](#) 25)  
Timers  
client ([section 3.1.2](#) 19, [section 3.2.2](#) 20)  
server ([section 3.1.2](#) 19, [section 3.3.2](#) 23)  
[Tracking changes](#) 35  
[Transport](#) 9  
Triggered events - higher-layer  
client ([section 3.1.4](#) 19, [section 3.2.4](#) 20)  
server ([section 3.1.4](#) 19, [section 3.3.4](#) 23)

## U

[Update printer cachedata  
\(DR\\_PRN\\_UPDATE\\_CACHEDATA\)](#) 27

## V

[Vendor-extensible fields](#) 8  
[Versioning](#) 8

## X

[XPS mode](#) 19