**[MS-RPRN]:**

**Print System Remote Protocol**

**Intellectual Property Rights Notice for Open Specifications Documentation**

* **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
* **Copyrights**. This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.
* **No Trade Secrets**. Microsoft does not claim any trade secret rights in this documentation.
* **Patents**. Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft [Open Specification Promise](http://go.microsoft.com/fwlink/?LinkId=214445) or the [Community Promise](http://go.microsoft.com/fwlink/?LinkId=214448). If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting [iplg@microsoft.com](mailto:iplg@microsoft.com).
* **Trademarks**. The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit [www.microsoft.com/trademarks](http://www.microsoft.com/trademarks).
* **Fictitious Names**. The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

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

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

**Revision Summary**

| Date | Revision History | Revision Class | Comments |
| --- | --- | --- | --- |
| 2/22/2007 | 0.01 |  | Version 0.01 release |
| 6/1/2007 | 1.0 | Major | Updated and revised the technical content. |
| 7/3/2007 | 2.0 | Major | Editorial changes, plus added a new section for a wsdmon monitor module for the WSD\_BACKUP\_PORT\_DATA packet structure. |
| 7/20/2007 | 2.0.1 | Editorial | Changed language and formatting in the technical content. |
| 8/10/2007 | 2.0.2 | Editorial | Changed language and formatting in the technical content. |
| 9/28/2007 | 2.1 | Minor | Clarified the meaning of the technical content. |
| 10/23/2007 | 2.2 | Minor | Clarified the meaning of the technical content. |
| 11/30/2007 | 2.2.1 | Editorial | Changed language and formatting in the technical content. |
| 1/25/2008 | 2.3 | Minor | Clarified the meaning of the technical content. |
| 3/14/2008 | 3.0 | Major | Updated and revised the technical content. |
| 5/16/2008 | 4.0 | Major | Updated and revised the technical content. |
| 6/20/2008 | 5.0 | Major | Updated and revised the technical content. |
| 7/25/2008 | 5.1 | Minor | Clarified the meaning of the technical content. |
| 8/29/2008 | 6.0 | Major | Updated and revised the technical content. |
| 10/24/2008 | 7.0 | Major | Updated and revised the technical content. |
| 12/5/2008 | 8.0 | Major | Updated and revised the technical content. |
| 1/16/2009 | 8.1 | Minor | Clarified the meaning of the technical content. |
| 2/27/2009 | 9.0 | Major | Updated and revised the technical content. |
| 4/10/2009 | 10.0 | Major | Updated and revised the technical content. |
| 5/22/2009 | 11.0 | Major | Updated and revised the technical content. |
| 7/2/2009 | 12.0 | Major | Updated and revised the technical content. |
| 8/14/2009 | 12.1 | Minor | Clarified the meaning of the technical content. |
| 9/25/2009 | 12.2 | Minor | Clarified the meaning of the technical content. |
| 11/6/2009 | 12.3 | Minor | Clarified the meaning of the technical content. |
| 12/18/2009 | 13.0 | Major | Updated and revised the technical content. |
| 1/29/2010 | 13.1 | Minor | Clarified the meaning of the technical content. |
| 3/12/2010 | 13.2 | Minor | Clarified the meaning of the technical content. |
| 4/23/2010 | 14.0 | Major | Updated and revised the technical content. |
| 6/4/2010 | 14.1 | Minor | Clarified the meaning of the technical content. |
| 7/16/2010 | 15.0 | Major | Updated and revised the technical content. |
| 8/27/2010 | 15.1 | Minor | Clarified the meaning of the technical content. |
| 10/8/2010 | 16.0 | Major | Updated and revised the technical content. |
| 11/19/2010 | 16.1 | Minor | Clarified the meaning of the technical content. |
| 1/7/2011 | 16.2 | Minor | Clarified the meaning of the technical content. |
| 2/11/2011 | 17.0 | Major | Updated and revised the technical content. |
| 3/25/2011 | 18.0 | Major | Updated and revised the technical content. |
| 5/6/2011 | 18.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 6/17/2011 | 18.1 | Minor | Clarified the meaning of the technical content. |
| 9/23/2011 | 19.0 | Major | Updated and revised the technical content. |
| 12/16/2011 | 20.0 | Major | Updated and revised the technical content. |
| 3/30/2012 | 21.0 | Major | Updated and revised the technical content. |
| 7/12/2012 | 22.0 | Major | Updated and revised the technical content. |
| 10/25/2012 | 23.0 | Major | Updated and revised the technical content. |
| 1/31/2013 | 24.0 | Major | Updated and revised the technical content. |
| 8/8/2013 | 25.0 | Major | Updated and revised the technical content. |
| 11/14/2013 | 26.0 | Major | Updated and revised the technical content. |
| 2/13/2014 | 26.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 5/15/2014 | 26.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 6/30/2015 | 27.0 | Major | Significantly changed the technical content. |

Table of Contents

[1 Introduction 11](#_Toc423366376)

[1.1 Glossary 11](#_Toc423366377)

[1.2 References 21](#_Toc423366378)

[1.2.1 Normative References 21](#_Toc423366379)

[1.2.2 Informative References 22](#_Toc423366380)

[1.3 Overview 23](#_Toc423366381)

[1.3.1 Management of the Print System 23](#_Toc423366382)

[1.3.2 Communication of Print Job Data 24](#_Toc423366383)

[1.3.3 Notification of Print System Changes 26](#_Toc423366384)

[1.4 Relationship to Other Protocols 27](#_Toc423366385)

[1.5 Prerequisites/Preconditions 28](#_Toc423366386)

[1.6 Applicability Statement 28](#_Toc423366387)

[1.7 Versioning and Capability Negotiation 28](#_Toc423366388)

[1.8 Vendor-Extensible Members 29](#_Toc423366389)

[1.9 Standards Assignments 29](#_Toc423366390)

[2 Messages 30](#_Toc423366391)

[2.1 Transport 30](#_Toc423366392)

[2.2 Common Data Types 30](#_Toc423366393)

[2.2.1 IDL Data Types 32](#_Toc423366394)

[2.2.1.1 Common IDL Data Types 32](#_Toc423366395)

[2.2.1.1.1 DEVMODE 32](#_Toc423366396)

[2.2.1.1.2 GDI\_HANDLE 33](#_Toc423366397)

[2.2.1.1.3 LANGID 33](#_Toc423366398)

[2.2.1.1.4 PRINTER\_HANDLE 33](#_Toc423366399)

[2.2.1.1.5 RECTL 34](#_Toc423366400)

[2.2.1.1.6 SIZE 34](#_Toc423366401)

[2.2.1.1.7 STRING\_HANDLE 34](#_Toc423366402)

[2.2.1.2 Containers 35](#_Toc423366403)

[2.2.1.2.1 DEVMODE\_CONTAINER 35](#_Toc423366404)

[2.2.1.2.2 DOC\_INFO\_CONTAINER 35](#_Toc423366405)

[2.2.1.2.3 DRIVER\_CONTAINER 35](#_Toc423366406)

[2.2.1.2.4 FORM\_CONTAINER 36](#_Toc423366407)

[2.2.1.2.5 JOB\_CONTAINER 36](#_Toc423366408)

[2.2.1.2.6 MONITOR\_CONTAINER 37](#_Toc423366409)

[2.2.1.2.7 PORT\_CONTAINER 38](#_Toc423366410)

[2.2.1.2.8 PORT\_VAR\_CONTAINER 38](#_Toc423366411)

[2.2.1.2.9 PRINTER\_CONTAINER 39](#_Toc423366412)

[2.2.1.2.10 RPC\_BIDI\_REQUEST\_CONTAINER 40](#_Toc423366413)

[2.2.1.2.11 RPC\_BIDI\_RESPONSE\_CONTAINER 40](#_Toc423366414)

[2.2.1.2.12 RPC\_BINARY\_CONTAINER 41](#_Toc423366415)

[2.2.1.2.13 SECURITY\_CONTAINER 41](#_Toc423366416)

[2.2.1.2.14 SPLCLIENT\_CONTAINER 41](#_Toc423366417)

[2.2.1.2.15 STRING\_CONTAINER 42](#_Toc423366418)

[2.2.1.2.16 SYSTEMTIME\_CONTAINER 42](#_Toc423366419)

[2.2.1.2.17 RPC\_BranchOfficeJobDataContainer 42](#_Toc423366420)

[2.2.1.3 Members in INFO Structures 42](#_Toc423366421)

[2.2.1.3.1 DRIVER\_INFO and RPC\_DRIVER\_INFO Members 43](#_Toc423366422)

[2.2.1.3.2 FORM\_INFO and RPC\_FORM\_INFO Members 44](#_Toc423366423)

[2.2.1.3.3 JOB\_INFO Members 45](#_Toc423366424)

[2.2.1.3.4 MONITOR\_INFO Members 46](#_Toc423366425)

[2.2.1.3.5 PORT\_INFO Members 46](#_Toc423366426)

[2.2.1.3.6 PRINTER\_INFO Members 46](#_Toc423366427)

[2.2.1.3.7 SPLCLIENT\_INFO Members 47](#_Toc423366428)

[2.2.1.4 DOC\_INFO\_1 47](#_Toc423366429)

[2.2.1.5 DRIVER\_INFO 47](#_Toc423366430)

[2.2.1.5.1 DRIVER\_INFO\_1 47](#_Toc423366431)

[2.2.1.5.2 DRIVER\_INFO\_2 48](#_Toc423366432)

[2.2.1.5.3 RPC\_DRIVER\_INFO\_3 48](#_Toc423366433)

[2.2.1.5.4 RPC\_DRIVER\_INFO\_4 48](#_Toc423366434)

[2.2.1.5.5 RPC\_DRIVER\_INFO\_6 49](#_Toc423366435)

[2.2.1.5.6 RPC\_DRIVER\_INFO\_8 49](#_Toc423366436)

[2.2.1.6 FORM\_INFO 51](#_Toc423366437)

[2.2.1.6.1 FORM\_INFO\_1 51](#_Toc423366438)

[2.2.1.6.2 RPC\_FORM\_INFO\_2 51](#_Toc423366439)

[2.2.1.7 JOB\_INFO 52](#_Toc423366440)

[2.2.1.7.1 JOB\_INFO\_1 52](#_Toc423366441)

[2.2.1.7.2 JOB\_INFO\_2 53](#_Toc423366442)

[2.2.1.7.3 JOB\_INFO\_3 53](#_Toc423366443)

[2.2.1.7.4 JOB\_INFO\_4 54](#_Toc423366444)

[2.2.1.8 MONITOR\_INFO 54](#_Toc423366445)

[2.2.1.8.1 MONITOR\_INFO\_1 54](#_Toc423366446)

[2.2.1.8.2 MONITOR\_INFO\_2 54](#_Toc423366447)

[2.2.1.9 PORT\_INFO 55](#_Toc423366448)

[2.2.1.9.1 PORT\_INFO\_1 55](#_Toc423366449)

[2.2.1.9.2 PORT\_INFO\_2 55](#_Toc423366450)

[2.2.1.9.3 PORT\_INFO\_3 56](#_Toc423366451)

[2.2.1.9.4 PORT\_INFO\_FF 57](#_Toc423366452)

[2.2.1.10 PRINTER\_INFO 57](#_Toc423366453)

[2.2.1.10.1 PRINTER\_INFO\_STRESS 57](#_Toc423366454)

[2.2.1.10.2 PRINTER\_INFO\_1 59](#_Toc423366455)

[2.2.1.10.3 PRINTER\_INFO\_2 59](#_Toc423366456)

[2.2.1.10.4 PRINTER\_INFO\_3 60](#_Toc423366457)

[2.2.1.10.5 PRINTER\_INFO\_4 60](#_Toc423366458)

[2.2.1.10.6 PRINTER\_INFO\_5 61](#_Toc423366459)

[2.2.1.10.7 PRINTER\_INFO\_6 61](#_Toc423366460)

[2.2.1.10.8 PRINTER\_INFO\_7 61](#_Toc423366461)

[2.2.1.10.9 PRINTER\_INFO\_8 62](#_Toc423366462)

[2.2.1.10.10 PRINTER\_INFO\_9 62](#_Toc423366463)

[2.2.1.11 SPLCLIENT\_INFO 63](#_Toc423366464)

[2.2.1.11.1 SPLCLIENT\_INFO\_1 63](#_Toc423366465)

[2.2.1.11.2 SPLCLIENT\_INFO\_2 63](#_Toc423366466)

[2.2.1.11.3 SPLCLIENT\_INFO\_3 63](#_Toc423366467)

[2.2.1.12 Bidirectional Communication Data 64](#_Toc423366468)

[2.2.1.12.1 RPC\_BIDI\_REQUEST\_DATA 64](#_Toc423366469)

[2.2.1.12.2 RPC\_BIDI\_RESPONSE\_DATA 64](#_Toc423366470)

[2.2.1.12.3 RPC\_BIDI\_DATA 64](#_Toc423366471)

[2.2.1.13 Printer Notification Data 65](#_Toc423366472)

[2.2.1.13.1 RPC\_V2\_NOTIFY\_OPTIONS 65](#_Toc423366473)

[2.2.1.13.2 RPC\_V2\_NOTIFY\_OPTIONS\_TYPE 66](#_Toc423366474)

[2.2.1.13.3 RPC\_V2\_NOTIFY\_INFO 66](#_Toc423366475)

[2.2.1.13.4 RPC\_V2\_NOTIFY\_INFO\_DATA 67](#_Toc423366476)

[2.2.1.13.5 RPC\_V2\_NOTIFY\_INFO\_DATA\_DATA 68](#_Toc423366477)

[2.2.1.13.6 RPC\_V2\_UREPLY\_PRINTER 68](#_Toc423366478)

[2.2.1.14 Job Named Properties 69](#_Toc423366479)

[2.2.1.14.1 RPC\_PrintPropertyValue 69](#_Toc423366480)

[2.2.1.14.2 RPC\_PrintNamedProperty 69](#_Toc423366481)

[2.2.1.14.3 RPC\_EPrintPropertyType 70](#_Toc423366482)

[2.2.1.14.4 SPLFILE\_CONTENT\_TYPE\_PROP\_NAME 70](#_Toc423366483)

[2.2.1.15 Branch Office Print Remote Logging Structures 70](#_Toc423366484)

[2.2.1.15.1 EBranchOfficeJobEventType 70](#_Toc423366485)

[2.2.1.15.2 RPC\_BranchOfficeJobData 71](#_Toc423366486)

[2.2.1.15.3 RPC\_BranchOfficeJobDataError 72](#_Toc423366487)

[2.2.1.15.4 RPC\_BranchOfficeJobDataPipelineFailed 73](#_Toc423366488)

[2.2.1.15.5 RPC\_BranchOfficeJobDataPrinted 73](#_Toc423366489)

[2.2.1.15.6 RPC\_BranchOfficeJobDataRendered 74](#_Toc423366490)

[2.2.1.15.7 RPC\_BranchOfficeLogOfflineFileFull 74](#_Toc423366491)

[2.2.2 Custom-Marshaled Data Types 75](#_Toc423366492)

[2.2.2.1 \_DEVMODE 77](#_Toc423366493)

[2.2.2.1.1 PostScript Driver Extra Data 90](#_Toc423366494)

[2.2.2.1.2 Generic Driver Extra Data 90](#_Toc423366495)

[2.2.2.1.3 OEM Driver Extra Data 90](#_Toc423366496)

[2.2.2.1.4 Print Ticket Driver Extra Data 90](#_Toc423366497)

[2.2.2.2 Members in Custom-Marshaled INFO structures 91](#_Toc423366498)

[2.2.2.3 DATATYPES\_INFO\_1 91](#_Toc423366499)

[2.2.2.4 \_DRIVER\_INFO 92](#_Toc423366500)

[2.2.2.4.1 \_DRIVER\_INFO\_1 92](#_Toc423366501)

[2.2.2.4.2 \_DRIVER\_INFO\_2 92](#_Toc423366502)

[2.2.2.4.3 \_DRIVER\_INFO\_3 94](#_Toc423366503)

[2.2.2.4.4 \_DRIVER\_INFO\_4 95](#_Toc423366504)

[2.2.2.4.5 \_DRIVER\_INFO\_5 97](#_Toc423366505)

[2.2.2.4.6 \_DRIVER\_INFO\_6 99](#_Toc423366506)

[2.2.2.4.7 \_DRIVER\_INFO\_7 101](#_Toc423366507)

[2.2.2.4.8 \_DRIVER\_INFO\_8 103](#_Toc423366508)

[2.2.2.4.9 \_DRIVER\_INFO\_101 106](#_Toc423366509)

[2.2.2.4.10 \_DRIVER\_FILE\_INFO 108](#_Toc423366510)

[2.2.2.5 \_FORM\_INFO 110](#_Toc423366511)

[2.2.2.5.1 \_FORM\_INFO\_1 110](#_Toc423366512)

[2.2.2.5.2 \_FORM\_INFO\_2 111](#_Toc423366513)

[2.2.2.6 \_JOB\_INFO 112](#_Toc423366514)

[2.2.2.6.1 \_JOB\_INFO\_1 112](#_Toc423366515)

[2.2.2.6.2 \_JOB\_INFO\_2 114](#_Toc423366516)

[2.2.2.6.3 \_JOB\_INFO\_3 116](#_Toc423366517)

[2.2.2.6.4 \_JOB\_INFO\_4 117](#_Toc423366518)

[2.2.2.7 \_MONITOR\_INFO 120](#_Toc423366519)

[2.2.2.7.1 \_MONITOR\_INFO\_1 120](#_Toc423366520)

[2.2.2.7.2 \_MONITOR\_INFO\_2 120](#_Toc423366521)

[2.2.2.8 \_PORT\_INFO 121](#_Toc423366522)

[2.2.2.8.1 \_PORT\_INFO\_1 121](#_Toc423366523)

[2.2.2.8.2 \_PORT\_INFO\_2 122](#_Toc423366524)

[2.2.2.9 \_PRINTER\_INFO 123](#_Toc423366525)

[2.2.2.9.1 \_PRINTER\_INFO\_STRESS 123](#_Toc423366526)

[2.2.2.9.2 \_PRINTER\_INFO\_1 125](#_Toc423366527)

[2.2.2.9.3 \_PRINTER\_INFO\_2 126](#_Toc423366528)

[2.2.2.9.4 \_PRINTER\_INFO\_3 128](#_Toc423366529)

[2.2.2.9.5 \_PRINTER\_INFO\_4 129](#_Toc423366530)

[2.2.2.9.6 \_PRINTER\_INFO\_5 130](#_Toc423366531)

[2.2.2.9.7 \_PRINTER\_INFO\_6 131](#_Toc423366532)

[2.2.2.9.8 \_PRINTER\_INFO\_7 131](#_Toc423366533)

[2.2.2.9.9 \_PRINTER\_INFO\_8 132](#_Toc423366534)

[2.2.2.10 PRINTPROCESSOR\_INFO\_1 133](#_Toc423366535)

[2.2.2.11 PRINTER\_ENUM\_VALUES 133](#_Toc423366536)

[2.2.2.12 UNIVERSAL\_FONT\_ID 135](#_Toc423366537)

[2.2.2.13 CORE\_PRINTER\_DRIVER 135](#_Toc423366538)

[2.2.2.14 TCPMON Structures 136](#_Toc423366539)

[2.2.2.14.1 CONFIG\_INFO\_DATA\_1 136](#_Toc423366540)

[2.2.2.14.2 DELETE\_PORT\_DATA\_1 136](#_Toc423366541)

[2.2.2.14.3 PORT\_DATA\_1 137](#_Toc423366542)

[2.2.2.14.4 PORT\_DATA\_2 139](#_Toc423366543)

[2.2.2.14.5 PORT\_DATA\_LIST\_1 141](#_Toc423366544)

[2.2.2.15 WSDMON Structures 141](#_Toc423366545)

[2.2.2.15.1 WSD\_DRIVER\_DATA 141](#_Toc423366546)

[2.2.2.15.2 WSD\_BACKUP\_PORT\_DATA 142](#_Toc423366547)

[2.2.2.15.3 WSD\_BACKUP\_PORT\_DATA\_EX 142](#_Toc423366548)

[2.2.3 Constants 144](#_Toc423366549)

[2.2.3.1 Access Values 144](#_Toc423366550)

[2.2.3.2 Change Notification Flags 146](#_Toc423366551)

[2.2.3.3 Job Notification Values 147](#_Toc423366552)

[2.2.3.4 Server Notification Values 149](#_Toc423366553)

[2.2.3.5 Notification Data Type Values 150](#_Toc423366554)

[2.2.3.6 Printer Change Flags 150](#_Toc423366555)

[2.2.3.6.1 Printer Change Flags for Use with a Printer Handle 150](#_Toc423366556)

[2.2.3.6.2 Printer Change Flags for Use with a Server Handle 151](#_Toc423366557)

[2.2.3.7 Printer Enumeration Flags 152](#_Toc423366558)

[2.2.3.8 Printer Notification Values 153](#_Toc423366559)

[2.2.3.9 Registry Type Values 156](#_Toc423366560)

[2.2.3.10 Server Handle Key Values 157](#_Toc423366561)

[2.2.3.10.1 OSVERSIONINFO 160](#_Toc423366562)

[2.2.3.10.2 OSVERSIONINFOEX 161](#_Toc423366563)

[2.2.3.10.3 OS\_TYPE Enumeration 161](#_Toc423366564)

[2.2.3.10.4 Event Log Flags 162](#_Toc423366565)

[2.2.3.10.5 Product Suite Flags 162](#_Toc423366566)

[2.2.3.10.6 Thread Priority Values 162](#_Toc423366567)

[2.2.3.11 Printer Data Values 163](#_Toc423366568)

[2.2.3.12 Status and Attribute Values 165](#_Toc423366569)

[2.2.3.13 BIDI\_TYPE Enumeration 168](#_Toc423366570)

[2.2.4 Rules for Members 169](#_Toc423366571)

[2.2.4.1 Access Values 169](#_Toc423366572)

[2.2.4.2 Datatype Names 169](#_Toc423366573)

[2.2.4.3 Driver Names 169](#_Toc423366574)

[2.2.4.4 Environment Names 169](#_Toc423366575)

[2.2.4.5 Form Names 169](#_Toc423366576)

[2.2.4.6 Job Control Values 169](#_Toc423366577)

[2.2.4.7 Key Names 170](#_Toc423366578)

[2.2.4.8 Monitor Names 170](#_Toc423366579)

[2.2.4.9 Path Names 170](#_Toc423366580)

[2.2.4.10 Port Names 170](#_Toc423366581)

[2.2.4.11 Print Processor Names 170](#_Toc423366582)

[2.2.4.12 Print Provider Names 170](#_Toc423366583)

[2.2.4.13 Printer Change Values 171](#_Toc423366584)

[2.2.4.14 Printer Names 171](#_Toc423366585)

[2.2.4.15 Registry Type Values 173](#_Toc423366586)

[2.2.4.16 Server Names 173](#_Toc423366587)

[2.2.4.17 User Names 173](#_Toc423366588)

[2.2.4.18 Value Names 173](#_Toc423366589)

[2.3 Directory Service Interaction 173](#_Toc423366590)

[2.3.1 Interaction Summary 173](#_Toc423366591)

[2.3.2 Directory Service Schema Elements 174](#_Toc423366592)

[2.3.3 Interaction Details 175](#_Toc423366593)

[2.3.3.1 Publishing a Print Queue to the Active Directory 176](#_Toc423366594)

[2.3.3.2 Modifying or Deleting a Print Queue in the Active Directory 177](#_Toc423366595)

[2.3.3.3 Searching for Print Queues in the Active Directory 178](#_Toc423366596)

[2.3.3.4 Initializing the Print Server for Active Directory 178](#_Toc423366597)

[3 Protocol Details 180](#_Toc423366598)

[3.1 Server Details 180](#_Toc423366599)

[3.1.1 Abstract Data Model 180](#_Toc423366600)

[3.1.2 Timers 184](#_Toc423366601)

[3.1.3 Initialization 184](#_Toc423366602)

[3.1.4 Message Processing Events and Sequencing Rules 184](#_Toc423366603)

[3.1.4.1 Commonly Used Parameters 194](#_Toc423366604)

[3.1.4.1.1 Datatype Name Parameters 194](#_Toc423366605)

[3.1.4.1.2 Dynamically Typed Query Parameters 194](#_Toc423366606)

[3.1.4.1.3 Environment Name Parameters 195](#_Toc423366607)

[3.1.4.1.4 Print Server Name Parameters 195](#_Toc423366608)

[3.1.4.1.5 Printer Name Parameters 196](#_Toc423366609)

[3.1.4.1.6 Standard Parameter Validation 197](#_Toc423366610)

[3.1.4.1.7 String Query Parameters 197](#_Toc423366611)

[3.1.4.1.8 CONTAINER Parameters 198](#_Toc423366612)

[3.1.4.1.8.1 DEVMODE\_CONTAINER Parameters 198](#_Toc423366613)

[3.1.4.1.8.2 DOC\_INFO\_CONTAINER Parameters 199](#_Toc423366614)

[3.1.4.1.8.3 DRIVER\_CONTAINER Parameters 199](#_Toc423366615)

[3.1.4.1.8.4 FORM\_CONTAINER Parameters 200](#_Toc423366616)

[3.1.4.1.8.5 PORT\_CONTAINER Parameters 200](#_Toc423366617)

[3.1.4.1.8.6 PRINTER\_CONTAINER Parameters 200](#_Toc423366618)

[3.1.4.1.8.7 SECURITY\_CONTAINER Parameters 201](#_Toc423366619)

[3.1.4.1.8.8 SPLCLIENT\_CONTAINER Parameters 202](#_Toc423366620)

[3.1.4.1.8.9 MONITOR\_CONTAINER Parameters 202](#_Toc423366621)

[3.1.4.1.9 INFO Structures Query Parameters 202](#_Toc423366622)

[3.1.4.1.10 PRINTER\_ENUM\_VALUES Structures Query Parameters 204](#_Toc423366623)

[3.1.4.1.11 PRINTER\_HANDLE Parameters 204](#_Toc423366624)

[3.1.4.2 Printer Management and Discovery Methods 205](#_Toc423366625)

[3.1.4.2.1 RpcEnumPrinters (Opnum 0) 207](#_Toc423366626)

[3.1.4.2.2 RpcOpenPrinter (Opnum 1) 208](#_Toc423366627)

[3.1.4.2.3 RpcAddPrinter (Opnum 5) 210](#_Toc423366628)

[3.1.4.2.4 RpcDeletePrinter (Opnum 6) 212](#_Toc423366629)

[3.1.4.2.5 RpcSetPrinter (Opnum 7) 212](#_Toc423366630)

[3.1.4.2.6 RpcGetPrinter (Opnum 8) 215](#_Toc423366631)

[3.1.4.2.7 RpcGetPrinterData (Opnum 26) 216](#_Toc423366632)

[3.1.4.2.8 RpcSetPrinterData (Opnum 27) 217](#_Toc423366633)

[3.1.4.2.9 RpcClosePrinter (Opnum 29) 218](#_Toc423366634)

[3.1.4.2.10 RpcCreatePrinterIC (Opnum 40) 219](#_Toc423366635)

[3.1.4.2.11 RpcPlayGdiScriptOnPrinterIC (Opnum 41) 220](#_Toc423366636)

[3.1.4.2.12 RpcDeletePrinterIC (Opnum 42) 221](#_Toc423366637)

[3.1.4.2.13 RpcResetPrinter (Opnum 52) 222](#_Toc423366638)

[3.1.4.2.14 RpcOpenPrinterEx (Opnum 69) 222](#_Toc423366639)

[3.1.4.2.15 RpcAddPrinterEx (Opnum 70) 224](#_Toc423366640)

[3.1.4.2.16 RpcEnumPrinterData (Opnum 72) 226](#_Toc423366641)

[3.1.4.2.17 RpcDeletePrinterData (Opnum 73) 227](#_Toc423366642)

[3.1.4.2.18 RpcSetPrinterDataEx (Opnum 77) 228](#_Toc423366643)

[3.1.4.2.19 RpcGetPrinterDataEx (Opnum 78) 229](#_Toc423366644)

[3.1.4.2.20 RpcEnumPrinterDataEx (Opnum 79) 231](#_Toc423366645)

[3.1.4.2.21 RpcEnumPrinterKey (Opnum 80) 232](#_Toc423366646)

[3.1.4.2.22 RpcDeletePrinterDataEx (Opnum 81) 233](#_Toc423366647)

[3.1.4.2.23 RpcDeletePrinterKey (Opnum 82) 234](#_Toc423366648)

[3.1.4.2.24 RpcAddPerMachineConnection (Opnum 85) 234](#_Toc423366649)

[3.1.4.2.25 RpcDeletePerMachineConnection (Opnum 86) 236](#_Toc423366650)

[3.1.4.2.26 RpcEnumPerMachineConnections (Opnum 87) 236](#_Toc423366651)

[3.1.4.2.27 RpcSendRecvBidiData (Opnum 97) 237](#_Toc423366652)

[3.1.4.3 Job Management Methods 239](#_Toc423366653)

[3.1.4.3.1 RpcSetJob (Opnum 2) 239](#_Toc423366654)

[3.1.4.3.2 RpcGetJob (Opnum 3) 242](#_Toc423366655)

[3.1.4.3.3 RpcEnumJobs (Opnum 4) 243](#_Toc423366656)

[3.1.4.3.4 RpcAddJob (Opnum 24) 244](#_Toc423366657)

[3.1.4.3.5 RpcScheduleJob (Opnum 25) 245](#_Toc423366658)

[3.1.4.4 Printer Driver Management Methods 245](#_Toc423366659)

[3.1.4.4.1 RpcAddPrinterDriver (Opnum 9) 246](#_Toc423366660)

[3.1.4.4.2 RpcEnumPrinterDrivers (Opnum 10) 247](#_Toc423366661)

[3.1.4.4.3 RpcGetPrinterDriver (Opnum 11) 249](#_Toc423366662)

[3.1.4.4.4 RpcGetPrinterDriverDirectory (Opnum 12) 250](#_Toc423366663)

[3.1.4.4.5 RpcDeletePrinterDriver (Opnum 13) 251](#_Toc423366664)

[3.1.4.4.6 RpcGetPrinterDriver2 (Opnum 53) 251](#_Toc423366665)

[3.1.4.4.7 RpcDeletePrinterDriverEx (Opnum 84) 253](#_Toc423366666)

[3.1.4.4.8 RpcAddPrinterDriverEx (Opnum 89) 255](#_Toc423366667)

[3.1.4.4.9 RpcGetCorePrinterDrivers (Opnum 102) 257](#_Toc423366668)

[3.1.4.4.10 RpcGetPrinterDriverPackagePath (Opnum 104) 258](#_Toc423366669)

[3.1.4.5 Form Management Methods 260](#_Toc423366670)

[3.1.4.5.1 RpcAddForm (Opnum 30) 260](#_Toc423366671)

[3.1.4.5.2 RpcDeleteForm (Opnum 31) 261](#_Toc423366672)

[3.1.4.5.3 RpcGetForm (Opnum 32) 261](#_Toc423366673)

[3.1.4.5.4 RpcSetForm (Opnum 33) 262](#_Toc423366674)

[3.1.4.5.5 RpcEnumForms (Opnum 34) 263](#_Toc423366675)

[3.1.4.6 Port Management Methods 264](#_Toc423366676)

[3.1.4.6.1 RpcEnumPorts (Opnum 35) 265](#_Toc423366677)

[3.1.4.6.2 RpcDeletePort (Opnum 39) 266](#_Toc423366678)

[3.1.4.6.3 RpcAddPortEx (Opnum 61) 267](#_Toc423366679)

[3.1.4.6.4 RpcSetPort (Opnum 71) 268](#_Toc423366680)

[3.1.4.6.5 RpcXcvData (Opnum 88) 268](#_Toc423366681)

[3.1.4.7 Port Monitor Management Methods 270](#_Toc423366682)

[3.1.4.7.1 RpcEnumMonitors (Opnum 36) 271](#_Toc423366683)

[3.1.4.7.2 RpcAddMonitor (Opnum 46) 272](#_Toc423366684)

[3.1.4.7.3 RpcDeleteMonitor (Opnum 47) 272](#_Toc423366685)

[3.1.4.8 Print Processor Management Methods 273](#_Toc423366686)

[3.1.4.8.1 RpcAddPrintProcessor (Opnum 14) 274](#_Toc423366687)

[3.1.4.8.2 RpcEnumPrintProcessors (Opnum 15) 275](#_Toc423366688)

[3.1.4.8.3 RpcGetPrintProcessorDirectory (Opnum 16) 275](#_Toc423366689)

[3.1.4.8.4 RpcDeletePrintProcessor (Opnum 48) 276](#_Toc423366690)

[3.1.4.8.5 RpcEnumPrintProcessorDatatypes (Opnum 51) 277](#_Toc423366691)

[3.1.4.9 Document Printing Methods 278](#_Toc423366692)

[3.1.4.9.1 RpcStartDocPrinter (Opnum 17) 279](#_Toc423366693)

[3.1.4.9.2 RpcStartPagePrinter (Opnum 18) 280](#_Toc423366694)

[3.1.4.9.3 RpcWritePrinter (Opnum 19) 281](#_Toc423366695)

[3.1.4.9.4 RpcEndPagePrinter (Opnum 20) 282](#_Toc423366696)

[3.1.4.9.5 RpcAbortPrinter (Opnum 21) 282](#_Toc423366697)

[3.1.4.9.6 RpcReadPrinter (Opnum 22) 283](#_Toc423366698)

[3.1.4.9.7 RpcEndDocPrinter (Opnum 23) 284](#_Toc423366699)

[3.1.4.9.8 RpcFlushPrinter (Opnum 96) 285](#_Toc423366700)

[3.1.4.10 Notification Methods 286](#_Toc423366701)

[3.1.4.10.1 RpcWaitForPrinterChange (Opnum 28) 286](#_Toc423366702)

[3.1.4.10.2 RpcFindClosePrinterChangeNotification (Opnum 56) 287](#_Toc423366703)

[3.1.4.10.3 RpcRemoteFindFirstPrinterChangeNotification (Opnum 62) 288](#_Toc423366704)

[3.1.4.10.4 RpcRemoteFindFirstPrinterChangeNotificationEx (Opnum 65) 289](#_Toc423366705)

[3.1.4.10.5 RpcRouterRefreshPrinterChangeNotification (Opnum 67) 291](#_Toc423366706)

[3.1.4.11 Monitor Module Methods 292](#_Toc423366707)

[3.1.4.11.1 LOCALMON 293](#_Toc423366708)

[3.1.4.11.2 LPRMON 293](#_Toc423366709)

[3.1.4.11.3 TCPMON 294](#_Toc423366710)

[3.1.4.11.4 WSDMON 295](#_Toc423366711)

[3.1.4.12 Job Named Property Management Methods 297](#_Toc423366712)

[3.1.4.12.1 RpcGetJobNamedPropertyValue (Opnum 110) 297](#_Toc423366713)

[3.1.4.12.2 RpcSetJobNamedProperty (Opnum 111) 298](#_Toc423366714)

[3.1.4.12.3 RpcDeleteJobNamedProperty (Opnum 112) 299](#_Toc423366715)

[3.1.4.12.4 RpcEnumJobNamedProperties (Opnum 113) 300](#_Toc423366716)

[3.1.4.13 Branch Office Print Remote Logging Methods 301](#_Toc423366717)

[3.1.4.13.1 RpcLogJobInfoForBranchOffice (Opnum 116) 301](#_Toc423366718)

[3.1.5 Timer Events 302](#_Toc423366719)

[3.1.6 Other Local Events 302](#_Toc423366720)

[3.2 Client Details 302](#_Toc423366721)

[3.2.1 Abstract Data Model 302](#_Toc423366722)

[3.2.2 Timers 303](#_Toc423366723)

[3.2.3 Initialization 303](#_Toc423366724)

[3.2.4 Message Processing Events and Sequencing Rules 303](#_Toc423366725)

[3.2.4.1 Client-Side Notification Processing Methods 303](#_Toc423366726)

[3.2.4.1.1 RpcReplyOpenPrinter (Opnum 58) 304](#_Toc423366727)

[3.2.4.1.2 RpcRouterReplyPrinter (Opnum 59) 305](#_Toc423366728)

[3.2.4.1.3 RpcReplyClosePrinter (Opnum 60) 305](#_Toc423366729)

[3.2.4.1.4 RpcRouterReplyPrinterEx (Opnum 66) 306](#_Toc423366730)

[3.2.4.2 Client Interaction with the Print Server 307](#_Toc423366731)

[3.2.4.2.1 Printing a Document Using RpcStartDocPrinter 307](#_Toc423366732)

[3.2.4.2.2 Enumerating Printers on a Print Server 307](#_Toc423366733)

[3.2.4.2.3 Enumerating Jobs on a Printer 308](#_Toc423366734)

[3.2.4.2.4 Receiving Notifications from a Print Server 309](#_Toc423366735)

[3.2.4.2.5 Announcing Shared Printers to Print Servers 309](#_Toc423366736)

[3.2.4.2.6 Adding a Printer to a Print Server 309](#_Toc423366737)

[3.2.5 Timer Events 310](#_Toc423366738)

[3.2.6 Other Local Events 310](#_Toc423366739)

[4 Protocol Examples 311](#_Toc423366740)

[4.1 Adding a Printer to a Server 311](#_Toc423366741)

[4.2 Adding a Printer Driver to a Server 313](#_Toc423366742)

[4.3 Enumerating and Managing Printers 314](#_Toc423366743)

[4.4 Enumerating Jobs and Modifying Job Settings 317](#_Toc423366744)

[4.5 Receiving Notifications on Printing Events 319](#_Toc423366745)

[5 Security 322](#_Toc423366746)

[5.1 Security Considerations for Implementers 322](#_Toc423366747)

[5.2 Index of Security Parameters 322](#_Toc423366748)

[6 Appendix A: Full IDL 323](#_Toc423366749)

[7 Appendix B: Product Behavior 347](#_Toc423366750)

[8 Change Tracking 482](#_Toc423366751)

[9 Index 484](#_Toc423366752)

# Introduction

This is a specification of the Print System Remote Protocol. It is based on the [**Remote Procedure Call (RPC)**](#gt_8a7f6700-8311-45bc-af10-82e10accd331) protocol, as specific in [[C706]](http://go.microsoft.com/fwlink/?LinkId=89824) and [[MS-RPCE]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-RPCE%5d.pdf).

The Print System Remote Protocol supports synchronous printing and spooling operations between a client and server, including [**print job**](#gt_12a6e569-e97c-4761-92f0-e397f8d5125f) control and [**print system**](#gt_b49fcfa1-e15c-4dce-be33-d74d5bfda785) management. An enhanced replacement for this protocol is specified in [[MS-PAR]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-PAR%5d.pdf).

Sections 1.8, 2, and 3 of this specification are normative and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in [[RFC2119]](http://go.microsoft.com/fwlink/?LinkId=90317). Sections 1.5 and 1.9 are also normative but do not contain those terms. All other sections and examples in this specification are informative.

## Glossary

The following terms are specific to this document:

**3D printer**: A [**device**](#gt_4712ded2-d927-4930-9450-29499f98cef4) that constructs a physical, three-dimensional object from a digital model.

**access control entry (ACE)**: An entry in an access control list (ACL) that contains a set of user rights and a [**security identifier (SID)**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d) that identifies a principal for whom the rights are allowed, denied, or audited.

**access level**: The type of access that the client requests for an object, such as read access, write access, or administrative access.

**Active Directory**: A general-purpose network [**directory service**](#gt_c36db657-3138-4d9a-9289-ded5cbb8b40e). [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90) also refers to the Windows implementation of a [**directory service**](#gt_c36db657-3138-4d9a-9289-ded5cbb8b40e). [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90) stores information about a variety of objects in the network. Importantly, user accounts, computer accounts, groups, and all related credential information used by the Windows implementation of Kerberos are stored in [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90). [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90) is either deployed as Active Directory Domain Services (AD DS) or Active Directory Lightweight Directory Services (AD LDS). [[MS-ADTS]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ADTS%5d.pdf) describes both forms. For more information, see [[MS-AUTHSOD]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-AUTHSOD%5d.pdf) section 1.1.1.5.2, [**Lightweight Directory Access Protocol (LDAP)**](#gt_45643bfb-b4c4-432c-a10f-b98790063f8d) versions 2 and 3, Kerberos, and [**DNS**](#gt_604dcfcd-72f5-46e5-85c1-f3ce69956700).

**application server mode**: A mode in which [**Terminal Services**](#gt_ffff3f01-8c21-44d3-bbda-0062a1fbda4b) require a client access license (CAL) to allow remote access to sessions on a terminal server.

**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.

**authentication**: The ability of one entity to determine the identity of another entity.

**bidirectional**: The ability to move, transfer, or transmit in two directions.

**big-endian**: Multiple-byte values that are byte-ordered with the most significant byte stored in the memory location with the lowest address.

**branch office print mode**: An operating mode in which a [**print client**](#gt_3b2da3d1-c159-4399-a6dd-dfd5f76fa2f5) is able to perform branch office printing. Every shared printer on a [**print server**](#gt_59fb3ddc-63cf-45df-8a90-46a6af9e00cb) can be configured to operate in branch office print mode.

**branch office print remote logging**: An operating mode in which a [**print client**](#gt_3b2da3d1-c159-4399-a6dd-dfd5f76fa2f5) logs printing-related [**Windows Events**](#gt_b9456a1c-aee2-41aa-af78-a083db7a27c7) on the [**print server**](#gt_59fb3ddc-63cf-45df-8a90-46a6af9e00cb). Branch office print remote logging occurs only when the [**print client**](#gt_3b2da3d1-c159-4399-a6dd-dfd5f76fa2f5) is in [**branch office print mode**](#gt_88d553c4-8182-4c27-8e7f-9f66c480215e).

**checksum**: A value that is the summation of a byte stream. By comparing the checksums computed from a data item at two different times, one can quickly assess whether the data items are identical.

**class printer driver**: Any [**printer driver**](#gt_1a48eebd-e72c-494d-b8cb-84dfb7bc3b65) declared by its manufacturer to be one from which a [**derived printer driver**](#gt_3101a6be-4528-4314-afe4-ea9d26a106eb) can derive. A [**class printer driver**](#gt_f4048a04-2516-44fc-ba42-2b5bee2782d5) cannot itself be a [**derived printer driver**](#gt_3101a6be-4528-4314-afe4-ea9d26a106eb). Typically, [**class printer drivers**](#gt_f4048a04-2516-44fc-ba42-2b5bee2782d5) are generic and work with a variety of [**devices**](#gt_4712ded2-d927-4930-9450-29499f98cef4), while [**derived printer drivers**](#gt_3101a6be-4528-4314-afe4-ea9d26a106eb) work with a particular [**device**](#gt_4712ded2-d927-4930-9450-29499f98cef4) and support features specific to that [**device**](#gt_4712ded2-d927-4930-9450-29499f98cef4).

**color matching**: The conversion of a color, sent from its original color space, to its visually closest color in the destination color space. See also [**Image Color Management (ICM)**](#gt_fcd5ee55-57ff-45d6-b51e-2ccba47329e3).

**color profile**: A file that contains information about how to convert colors in the color space and the color gamut of a specific device into a device-independent color space. A device-specific color profile is called a "device profile". For more information on using color and device profiles, see [[MSDN-UDP]](http://go.microsoft.com/fwlink/?LinkId=90151).

**container**: An object in the directory that can serve as the parent for other objects. In the absence of schema constraints, all objects would be [**containers**](#gt_c3143e71-2ada-417e-83f4-3ef10eff2c56). The schema allows only objects of specific classes to be [**containers**](#gt_c3143e71-2ada-417e-83f4-3ef10eff2c56).

**core printer driver**: A [**printer driver**](#gt_1a48eebd-e72c-494d-b8cb-84dfb7bc3b65) that other printer drivers depend on. In Windows, this term includes the Unidrv and Pscript printer drivers. For more information, see [[MSDN-UNIDRV]](http://go.microsoft.com/fwlink/?LinkId=184749) and [[MSDN-PSCRIPT]](http://go.microsoft.com/fwlink/?LinkId=184750) respectively.

**cyclic redundancy check (CRC)**: An algorithm used to produce a [**checksum**](#gt_fa444149-ef93-4512-a278-2e756295630c) (a small, fixed number of bits) against a block of data, such as a packet of network traffic or a block of a computer file. The CRC is used to detect errors after transmission or storage. A CRC is designed to catch random errors, as opposed to intentional errors. If errors might be introduced by a motivated and intelligent adversary, a cryptographic hash function should be used instead.

**data type**: A string that specifies the format of data that a printing application sends to a printer in a [**print job**](#gt_12a6e569-e97c-4761-92f0-e397f8d5125f). Data types include [**enhanced metafile spool format (EMFSPOOL)**](#gt_3bd355a2-4faa-40ab-bc79-0d158eba43b6) and [**RAW format**](#gt_cea5f44d-0398-47cf-bb34-110ffddade11). For rules governing [**data type**](#gt_0d327bc0-a461-4723-8beb-d935fdab4895) names, see section 2.2.4.2.

**derived printer driver**: A [**printer driver**](#gt_1a48eebd-e72c-494d-b8cb-84dfb7bc3b65) declared by its manufacturer to depend on a particular [**class printer driver**](#gt_f4048a04-2516-44fc-ba42-2b5bee2782d5) by sharing modules with the [**class printer driver**](#gt_f4048a04-2516-44fc-ba42-2b5bee2782d5).

**device**: Any peripheral or part of a computer system that can send or receive data.

**device driver**: The software that the system uses to communicate with a device such as a display, printer, mouse, or communications adapter. An abstraction layer that restricts access of applications to various hardware devices on a given computer system. It is often referred to simply as a "driver".

**directed discovery**: A discovery method used by [**WSD**](#gt_734b2ac6-73db-47d5-ae14-213ddd6e958b) devices. Directed discovery is used to discover devices on a subnet that is not the local subnet.

**directory service (DS)**: A service that stores and organizes information about a computer network's users and network shares, and that allows network administrators to manage users' access to the shares. See also [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90).

**discretionary access control list (DACL)**: An access control list (ACL) that is controlled by the owner of an object and that specifies the access particular users or groups can have to the object.

**distinguished name (DN)**: A name that uniquely identifies an object by using the [**relative distinguished name (RDN)**](#gt_22198321-b40b-4c24-b8a2-29e44d9d92b9) for the object, and the names of container objects and domains that contain the object. The distinguished name (DN) identifies the object and its location in a tree.

**dithering**: A form of digital [**halftoning**](#gt_3934fcec-759d-45ac-8539-8cf2adeda698).

**domain**: A set of users and computers sharing a common namespace and management infrastructure. At least one computer member of the set must act as a [**domain controller (DC)**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd) and host a member list that identifies all members of the domain, as well as optionally hosting the [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90) service. The domain controller provides [**authentication**](#gt_8e961bf0-95ba-4f58-9034-b67ccb27f317) of members, creating a unit of trust for its members. Each domain has an identifier that is shared among its members. For more information, see [MS-AUTHSOD] section 1.1.1.5 and [MS-ADTS].

**domain controller (DC)**: The service, running on a server, that implements [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90), or the server hosting this service. The service hosts the data store for objects and interoperates with other [**DCs**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd) to ensure that a local change to an object replicates correctly across all [**DCs**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd). When [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90) is operating as Active Directory Domain Services (AD DS), the [**DC**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd) contains full NC replicas of the configuration naming context (config NC), schema naming context (schema NC), and one of the domain NCs in its forest. If the AD DS [**DC**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd) is a global catalog server (GC server), it contains partial NC replicas of the remaining domain NCs in its forest. For more information, see [MS-AUTHSOD] section 1.1.1.5.2 and [MS-ADTS]. When [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90) is operating as Active Directory Lightweight Directory Services (AD LDS), several AD LDS [**DCs**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd) can run on one server. When [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90) is operating as AD DS, only one AD DS [**DC**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd) can run on one server. However, several AD LDS [**DCs**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd) can coexist with one AD DS [**DC**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd) on one server. The AD LDS [**DC**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd) contains full NC replicas of the config NC and the schema NC in its forest.

**domain name**: A [**domain name**](#gt_45a1c9f1-0263-49a8-97c7-7aca1a99308c) or a NetBIOS name that identifies a [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca).

**Domain Name System (DNS)**: A hierarchical, distributed database that contains mappings of [**domain names**](#gt_45a1c9f1-0263-49a8-97c7-7aca1a99308c) to various types of data, such as IP addresses. DNS enables the location of computers and services by user-friendly names, and it also enables the discovery of other information stored in the database.

**driver package**: A collection of the files needed to successfully load a driver. This includes the device information (.inf) file, the catalog file, and all of the binaries that are copied by the .inf file. Multiple drivers packaged together for deployment purposes.

**driver store**: A secure location on the local hard disk where the entire driver package is copied.

**endpoint**: A network-specific address of a remote procedure call (RPC) server process for remote procedure calls. The actual name and type of the endpoint depends on the [**RPC**](#gt_8a7f6700-8311-45bc-af10-82e10accd331) protocol sequence that is being used. For example, for RPC over TCP (RPC Protocol Sequence ncacn\_ip\_tcp), an endpoint might be TCP port 1025. For RPC over Server Message Block (RPC Protocol Sequence ncacn\_np), an endpoint might be the name of a named pipe. For more information, see [[C706]](http://go.microsoft.com/fwlink/?LinkId=89824).

**enhanced metafile format (EMF)**: A file format that supports the device-independent definitions of images.

**enhanced metafile spool format (EMFSPOOL)**: A format that specifies a structure of [**enhanced metafile format (EMF)**](#gt_d9d0bff9-d270-4528-9081-fe51db809c36) records used for defining application and device-independent printer [**spool files**](#gt_4c0e7da9-88a8-4430-abd7-27e7250b8180).

**event channel**: A collection of [**Windows Events**](#gt_b9456a1c-aee2-41aa-af78-a083db7a27c7) that is provided by the system. Also referred to as an event log. The name of an event channel is composed of an event provider name combined with a channel type string. Valid channel types are “Admin”, “Analytic”, “Debug”, and “Operational”. For more information, see [[MSDN-WINEV]](http://go.microsoft.com/fwlink/?LinkId=90161).

**event ID**: An identifier for the data represented by a [**Windows Event**](#gt_b9456a1c-aee2-41aa-af78-a083db7a27c7). Event IDs are unique with each event provider. For more information, see [MSDN-WINEV].

**fax printer**: A [**print queue**](#gt_569f1f1c-f426-46fa-91d2-3d1eb0b19aa1) that sends all print jobs to fax recipients as fax documents containing the printed data.

**file**: An entity of data in the file system that a user can access and manage. A [**file**](#gt_a04c146a-de3b-4e4b-829f-a9e772f3fe25) must have a unique name in its directory. It consists of one or more streams of bytes that hold a set of related data, plus a set of attributes (also called properties) that describe the [**file**](#gt_a04c146a-de3b-4e4b-829f-a9e772f3fe25) or the data within the [**file**](#gt_a04c146a-de3b-4e4b-829f-a9e772f3fe25). The creation time of a [**file**](#gt_a04c146a-de3b-4e4b-829f-a9e772f3fe25) is an example of a file attribute.

**file printer**: A [**print queue**](#gt_569f1f1c-f426-46fa-91d2-3d1eb0b19aa1) that does not represent a physical device, but instead converts all print jobs to files containing the printed data.

**fully qualified domain name (FQDN)**: An unambiguous [**domain name**](#gt_45a1c9f1-0263-49a8-97c7-7aca1a99308c) that gives an absolute location in the [**Domain Name System's (DNS)**](#gt_604dcfcd-72f5-46e5-85c1-f3ce69956700) hierarchy tree, as defined in [[RFC1035]](http://go.microsoft.com/fwlink/?LinkId=90264) section 3.1 and [[RFC2181]](http://go.microsoft.com/fwlink/?LinkId=127732) section 11.

**global catalog (GC)**: A unified partial view of multiple [**naming contexts (NCs)**](#gt_784c7cce-f782-48d8-9444-c9030ba86942) in a distributed partitioned directory. The [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90) directory service [**GC**](#gt_4f5d605a-7b3f-4db7-8c21-b146856d7169) is implemented by GC servers. The definition of [**global catalog**](#gt_4f5d605a-7b3f-4db7-8c21-b146856d7169) is specified in [MS-ADTS] section 3.1.1.1.8.

**globally unique identifier (GUID)**: A term used interchangeably with [**universally unique identifier (UUID)**](#gt_c4813fc3-b2e5-4aa3-bde7-421d950d68d3) in Microsoft protocol technical documents (TDs). Interchanging the usage of these terms does not imply or require a specific algorithm or mechanism to generate the value. Specifically, the use of this term does not imply or require that the algorithms described in [[RFC4122]](http://go.microsoft.com/fwlink/?LinkId=90460) or [C706] must be used for generating the [**GUID**](#gt_f49694cc-c350-462d-ab8e-816f0103c6c1). See also [**universally unique identifier (UUID)**](#gt_c4813fc3-b2e5-4aa3-bde7-421d950d68d3).

**Graphics Device Interface (GDI)**: A Windows API, supported on 16-bit and 32-bit versions of the operating system, that supports graphics operations and image manipulation on logical graphics objects.

**GUIDString**: A [**GUID**](#gt_f49694cc-c350-462d-ab8e-816f0103c6c1) in the form of an [**ASCII**](#gt_79fa85ca-ac61-467c-b819-e97dc1a7a599) or [**Unicode**](#gt_c305d0ab-8b94-461a-bd76-13b40cb8c4d8) string, consisting of one group of 8 hexadecimal digits, followed by three groups of 4 hexadecimal digits each, followed by one group of 12 hexadecimal digits. It is the standard representation of a GUID, as described in [RFC4122] section 3. For example, "6B29FC40-CA47-1067-B31D-00DD010662DA". Unlike a curly braced GUID string, a GUIDString is not enclosed in braces.

**halftoning**: The process of converting grayscale, or continuous-tone graphics or images, to a representation with a discrete number of gray (or tone) levels.

**Image Color Management (ICM)**: Technology that ensures that a color image, graphic, or text object is rendered as closely as possible to its original intent on any device despite differences in imaging technologies and color capabilities between devices.

**INF file**: A file providing Windows Setup with the information required to set up a device, such as a list of valid logical configurations for the device and the names of driver files associated with the device.

**information context**: A special-purpose printer object that can only be used to obtain information about fonts that are supported by a printer. For more information, see [[MSDN-FONTS]](http://go.microsoft.com/fwlink/?LinkId=90008).

**Interface Definition Language (IDL)**: The International Standards Organization (ISO) standard language for specifying the interface for remote procedure calls. For more information, see [C706] section 4.

**Internet Protocol version 4 (IPv4)**: An Internet protocol that has 32-bit source and destination addresses. IPv4 is the predecessor of IPv6.

**Internet Protocol version 6 (IPv6)**: A revised version of the Internet Protocol (IP) designed to address growth on the Internet. Improvements include a 128-bit IP address size, expanded routing capabilities, and support for [**authentication**](#gt_8e961bf0-95ba-4f58-9034-b67ccb27f317) and privacy.

**language monitor**: An executable object that provides a communications path between a [**print queue**](#gt_569f1f1c-f426-46fa-91d2-3d1eb0b19aa1) and a printer's [**port monitor**](#gt_1b5f368b-942b-40dd-84ea-fa9082916216). [**Language monitors**](#gt_28b48680-595a-405b-a857-7c2dc95ee7b4) add control information to the data stream, such as commands defined by a [**Page Description Language (PDL)**](#gt_a04e8872-d39e-4ae8-a870-b2e36b266302). They are optional, and are only associated with a particular type of printer if specified in the printer's [**INF file**](#gt_5f92d31c-3db8-4344-a7e0-a224983abe07).

**Lightweight Directory Access Protocol (LDAP)**: The primary access protocol for [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90). Lightweight Directory Access Protocol (LDAP) is an industry-standard protocol, established by the Internet Engineering Task Force (IETF), which allows users to query and update information in a [**directory service (DS)**](#gt_c36db657-3138-4d9a-9289-ded5cbb8b40e), as described in [MS-ADTS]. The Lightweight Directory Access Protocol can be either version 2 [RFC1777] or version 3 [RFC3377].

**little-endian**: Multiple-byte values that are byte-ordered with the least significant byte stored in the memory location with the lowest address.

**LOCALMON**: The [**port monitor**](#gt_1b5f368b-942b-40dd-84ea-fa9082916216) that manages local serial ("COM") and parallel ("LPT") [**ports**](#gt_ac027425-6043-4ad3-be27-56a381b75d38) on a Windows machine.

**LPRMON**: The [**port monitor module**](#gt_4c9689f8-854e-43e1-964e-5dd9a70b8aae) that allows Windows [**print servers**](#gt_59fb3ddc-63cf-45df-8a90-46a6af9e00cb) to send [**print jobs**](#gt_12a6e569-e97c-4761-92f0-e397f8d5125f) to machines that support UNIX [**print server**](#gt_59fb3ddc-63cf-45df-8a90-46a6af9e00cb) functions.

**marshal**: To encode one or more data structures into an octet stream using a specific [**remote procedure call (RPC)**](#gt_8a7f6700-8311-45bc-af10-82e10accd331) transfer syntax (for example, marshaling a 32-bit integer).

**marshaling**: The act of formatting COM parameters for transmission over a [**remote procedure call (RPC)**](#gt_8a7f6700-8311-45bc-af10-82e10accd331). For more information, see [[MS-DCOM]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-DCOM%5d.pdf).

**Microsoft-Windows-PrintService**: An event provider for printing services on Windows operating systems.

**monitor module**: An executable object that provides a communication path between the [**print system**](#gt_b49fcfa1-e15c-4dce-be33-d74d5bfda785) and the printers on a server.

**multicast discovery**: A discovery method used by [**WSD**](#gt_734b2ac6-73db-47d5-ae14-213ddd6e958b) devices. Multicast discovery is used to discover devices on the local subnet.

**multisz**: A data type that defines an array of null-terminated, 16-bit [**Unicode**](#gt_c305d0ab-8b94-461a-bd76-13b40cb8c4d8) [**UTF-16LE**](#gt_69864bcc-33be-41c9-9486-e18618ee3dd0)-encoded strings, with an additional null after the final string.

**naming context (NC)**: An [**NC**](#gt_784c7cce-f782-48d8-9444-c9030ba86942) is a set of objects organized as a tree. It is referenced by a DSName. The [**DN**](#gt_1175dd11-9368-41d5-98ed-d585f268ad4b) of the DSName is the distinguishedName attribute of the tree root. The [**GUID**](#gt_f49694cc-c350-462d-ab8e-816f0103c6c1) of the DSName is the objectGUID attribute of the tree root. The [**security identifier (SID)**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d) of the DSName, if present, is the objectSid attribute of the tree root; for Active Directory Domain Services (AD DS), the [**SID**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d) is present if and only if the [**NC**](#gt_784c7cce-f782-48d8-9444-c9030ba86942) is a domain naming context (domain NC). [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90) supports organizing several [**NCs**](#gt_784c7cce-f782-48d8-9444-c9030ba86942) into a tree structure.

**NetBIOS**: A particular network transport that is part of the LAN Manager protocol suite. [**NetBIOS**](#gt_b86c44e6-57df-4c48-8163-5e3fa7bdcff4) uses a broadcast communication style that was applicable to early segmented local area networks. The LAN Manager protocols were the default in Windows NT operating system environments prior to Windows 2000 operating system. A protocol family including name resolution, datagram, and connection services. For more information, see [[RFC1001]](http://go.microsoft.com/fwlink/?LinkId=90260) and [[RFC1002]](http://go.microsoft.com/fwlink/?LinkId=90261).

**Network Data Representation (NDR)**: A specification that defines a mapping from [**Interface Definition Language (IDL)**](#gt_73177eec-4092-420f-92c5-60b2478df824) data types onto octet streams. [**NDR**](#gt_9ebf9540-2c31-43bc-bc56-4a932faabf2d) also refers to the runtime environment that implements the mapping facilities (for example, data provided to [**NDR**](#gt_9ebf9540-2c31-43bc-bc56-4a932faabf2d)). For more information, see [[MS-RPCE]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-RPCE%5d.pdf) and [C706] section 14.

**n-up printing**: The act of arranging multiple logical pages on a physical sheet of paper.

**object identifier (OID)**: In the context of an object server, a 64-bit number that uniquely identifies an object.

**Open XML Paper Specification (OpenXPS)**: The [**XML Paper Specification (XPS)**](#gt_0b8be35a-bc52-4a6d-b711-3ee441b1bf60) document format based on the European Carton Makers Association (ECMA) standard ECMA-388 [[ECMA-388]](http://go.microsoft.com/fwlink/?LinkId=158846).

**opnum**: An operation number or numeric identifier that is used to identify a specific [**remote procedure call (RPC)**](#gt_8a7f6700-8311-45bc-af10-82e10accd331) method or a method in an interface. For more information, see [C706] section 12.5.2.12 or [MS-RPCE].

**page description language (PDL)**: The language for describing the layout and contents of a printed page. Common examples are PostScript and [**Printer Control Language (PCL)**](#gt_ffc25745-85c9-40bc-b2d3-f19ab7b48570).

**plug-in**: An executable module that can be loaded by the [**print server**](#gt_59fb3ddc-63cf-45df-8a90-46a6af9e00cb) to perform specific functions.

**port**: A logical name that represents a connection to a [**device**](#gt_4712ded2-d927-4930-9450-29499f98cef4). A [**port**](#gt_ac027425-6043-4ad3-be27-56a381b75d38) can represent a network address (for example, a TCP/IP address) or a local connection (for example, a [**USB**](#gt_2922ead8-cc31-4cea-a2aa-885ac2c9455d) [**port**](#gt_ac027425-6043-4ad3-be27-56a381b75d38)).

**port monitor**: A [**plug-in**](#gt_c731fa43-8a11-4f66-91f8-8519571ce1bf) that communicates with a [**device**](#gt_4712ded2-d927-4930-9450-29499f98cef4) that is connected to a port. A [**port monitor**](#gt_1b5f368b-942b-40dd-84ea-fa9082916216) may interact with the [**device**](#gt_4712ded2-d927-4930-9450-29499f98cef4) locally, remotely over a network, or through some other communication channel. The data that passes through a [**port monitor**](#gt_1b5f368b-942b-40dd-84ea-fa9082916216) is in a form that can be understood by the destination [**device**](#gt_4712ded2-d927-4930-9450-29499f98cef4), such as [**page description language (PDL)**](#gt_a04e8872-d39e-4ae8-a870-b2e36b266302).

**port monitor module**: A [**monitor module**](#gt_971b7035-4384-4e85-9878-b88c49aef05e) for a [**port monitor**](#gt_1b5f368b-942b-40dd-84ea-fa9082916216).

**PostScript**: A [**page description language**](#gt_a04e8872-d39e-4ae8-a870-b2e36b266302) developed by Adobe Systems that is primarily used for printing documents on laser printers. It is the standard for desktop publishing.

**principal**: An authenticated entity that initiates a message or channel in a distributed system.

**print client**: The application or user that is trying to apply an operation on the print system either by printing a job or by managing the data structures or devices maintained by the print system.

**print job**: The rendered [**page description language (PDL)**](#gt_a04e8872-d39e-4ae8-a870-b2e36b266302) output data sent to a print device for a particular application or user request.

**Print Pipeline**: A service in the Windows implementation of the [**XPS**](#gt_0b8be35a-bc52-4a6d-b711-3ee441b1bf60) printing subsystem that applies a series of [**printer driver**](#gt_1a48eebd-e72c-494d-b8cb-84dfb7bc3b65)-defined filters to the data in an [**XPS**](#gt_0b8be35a-bc52-4a6d-b711-3ee441b1bf60) printer [**spool file**](#gt_4c0e7da9-88a8-4430-abd7-27e7250b8180).

**print processor**: A [**plug-in**](#gt_c731fa43-8a11-4f66-91f8-8519571ce1bf) that runs on the [**print server**](#gt_59fb3ddc-63cf-45df-8a90-46a6af9e00cb) and processes [**print job**](#gt_12a6e569-e97c-4761-92f0-e397f8d5125f) data before it is sent to a print [**device**](#gt_4712ded2-d927-4930-9450-29499f98cef4).

**print provider**: A [**plug-in**](#gt_c731fa43-8a11-4f66-91f8-8519571ce1bf) that runs on the [**print server**](#gt_59fb3ddc-63cf-45df-8a90-46a6af9e00cb) and routes [**print system**](#gt_b49fcfa1-e15c-4dce-be33-d74d5bfda785) requests. Print providers are a Windows implementation detail and are not required by this protocol.

**print queue**: The logical entity to which jobs may be submitted for a particular print device. Associated with a print queue is a print driver, a user's print configuration in the form of a DEVMODE structure, and a system print configuration stored in the system registry.

**print server**: A machine that hosts the print system and all its different components.

**print system**: A system component that is responsible for coordinating and controlling the operation of [**print queues**](#gt_569f1f1c-f426-46fa-91d2-3d1eb0b19aa1), [**printer drivers**](#gt_1a48eebd-e72c-494d-b8cb-84dfb7bc3b65), and [**print jobs**](#gt_12a6e569-e97c-4761-92f0-e397f8d5125f).

**print system remote protocol stress analysis**: An optional diagnostic procedure that is used to analyze [**print server**](#gt_59fb3ddc-63cf-45df-8a90-46a6af9e00cb) load, error counts, throughput, and other metrics.

**Printer Control Language (PCL)**: A [**page description language (PDL)**](#gt_a04e8872-d39e-4ae8-a870-b2e36b266302) developed by Hewlett Packard for its laser and ink-jet printers.

**printer driver**: The interface component between the operating system and the printer device. It is responsible for processing the application data into a [**page description language (PDL)**](#gt_a04e8872-d39e-4ae8-a870-b2e36b266302) that can be interpreted by the printer device.

**printer driver downgrade**: An upgrade operation where an older [**printer driver**](#gt_1a48eebd-e72c-494d-b8cb-84dfb7bc3b65) is installed, replacing a newer [**printer driver**](#gt_1a48eebd-e72c-494d-b8cb-84dfb7bc3b65).

**printer driver isolation**: An implementation technology by which a [**print server**](#gt_59fb3ddc-63cf-45df-8a90-46a6af9e00cb) segregates [**printer driver**](#gt_1a48eebd-e72c-494d-b8cb-84dfb7bc3b65) execution into one or more processes separate from the [**print server**](#gt_59fb3ddc-63cf-45df-8a90-46a6af9e00cb) to isolate the [**print server**](#gt_59fb3ddc-63cf-45df-8a90-46a6af9e00cb) and other [**printer drivers**](#gt_1a48eebd-e72c-494d-b8cb-84dfb7bc3b65) from the side effects of faulty drivers.

**printer driver manifest**: A file that is installed with a [**printer driver**](#gt_1a48eebd-e72c-494d-b8cb-84dfb7bc3b65) and lists attributes of the [**printer driver**](#gt_1a48eebd-e72c-494d-b8cb-84dfb7bc3b65). The formatting of [**printer driver manifests**](#gt_566759dc-6e10-4b36-b710-f4296c1fbc83) is specific to the print server implementation.

**printer driver upgrade**: An upgrade operation where a newer [**printer driver**](#gt_1a48eebd-e72c-494d-b8cb-84dfb7bc3b65) is installed, replacing an older [**printer driver**](#gt_1a48eebd-e72c-494d-b8cb-84dfb7bc3b65).

**printer form**: A preprinted blank paper form, or a print job's virtual representation of this form, that enables a printer to position form elements in their physical location on the page.

**printer key**: A string that uniquely identifies a path under the main [**registry**](#gt_2d67ea57-10e6-422f-985e-694834945a02) key where printer configuration data is kept. Rules for printer key names are specified in section 2.2.4.7.

**printer UI application**: An implementation-specific application optionally installed together with a printer driver. A printer UI application provides access to the user to discover available printer features, and monitor and modify printer configuration settings.

**RAW format**: A [**data type**](#gt_0d327bc0-a461-4723-8beb-d935fdab4895) consisting of [**PDL**](#gt_a04e8872-d39e-4ae8-a870-b2e36b266302) data that can be sent to a [**device**](#gt_4712ded2-d927-4930-9450-29499f98cef4) without further processing.

**registry**: A local system-defined database in which applications and system components store and retrieve configuration data. It is a hierarchical data store with lightly typed elements that are logically stored in tree format. Applications use the registry API to retrieve, modify, or delete registry data. The data stored in the registry varies according to the version of Windows.

**relative distinguished name (RDN)**: The name of an object relative to its parent. This is the leftmost attribute-value pair in the [**distinguished name (DN)**](#gt_1175dd11-9368-41d5-98ed-d585f268ad4b) of an object. For example, in the [**DN**](#gt_1175dd11-9368-41d5-98ed-d585f268ad4b) "cn=Peter Houston, ou=NTDEV, dc=microsoft, dc=com", the [**RDN**](#gt_22198321-b40b-4c24-b8a2-29e44d9d92b9) is "cn=Peter Houston". For more information, see [[RFC2251]](http://go.microsoft.com/fwlink/?LinkId=90325).

**Remote Administration Protocol (RAP)**: A synchronous request/response protocol, used prior to the development of the remote procedure call (RPC) protocol, for marshaling and unmarshaling procedure call input and output arguments into messages and for reliably transporting messages to and from clients and servers.

**remote procedure call (RPC)**: A context-dependent term commonly overloaded with three meanings. Note that much of the industry literature concerning RPC technologies uses this term interchangeably for any of the three meanings. Following are the three definitions: (\*) The runtime environment providing remote procedure call facilities. The preferred usage for this meaning is "RPC runtime". (\*) The pattern of request and response message exchange between two parties (typically, a client and a server). The preferred usage for this meaning is "RPC exchange". (\*) A single message from an exchange as defined in the previous definition. The preferred usage for this term is "RPC message". For more information about RPC, see [C706].

**RPC context handle**: A representation of state maintained between a remote procedure call (RPC) client and server. The state is maintained on the server on behalf of the client. An RPC context handle is created by the server and given to the client. The client passes the RPC context handle back to the server in method calls to assist in identifying the state. For more information, see [C706].

**RPC endpoint**: A network-specific address of a server process for remote procedure calls (RPCs). The actual name of the RPC endpoint depends on the RPC protocol sequence being used. For example, for the NCACN\_IP\_TCP RPC protocol sequence an RPC endpoint might be TCP port 1025. For more information, see [C706].

**RPC transfer syntax**: A method for encoding messages defined in an Interface Definition Language (IDL) file. Remote procedure call (RPC) can support different encoding methods or transfer syntaxes. For more information, see [C706].

**RPC transport**: The underlying network services used by the remote procedure call (RPC) runtime for communications between network nodes. For more information, see [C706] section 2.

**schema**: The set of attributes and object classes that govern the creation and update of objects.

**SD**: See [**security descriptor**](#gt_e5213722-75a9-44e7-b026-8e4833f0d350).

**security descriptor**: A data structure containing the security information associated with a securable object. A [**security descriptor**](#gt_e5213722-75a9-44e7-b026-8e4833f0d350) identifies an object's owner by its [**security identifier (SID)**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d). If access control is configured for the object, its [**security descriptor**](#gt_e5213722-75a9-44e7-b026-8e4833f0d350) contains a [**discretionary access control list (DACL)**](#gt_d727f612-7a45-48e4-9d87-71735d62b321) with [**SIDs**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d) for the security principals who are allowed or denied access. Applications use this structure to set and query an object's security status. The [**security descriptor**](#gt_e5213722-75a9-44e7-b026-8e4833f0d350) is used to guard access to an object as well as to control which type of auditing takes place when the object is accessed. The [**security descriptor**](#gt_e5213722-75a9-44e7-b026-8e4833f0d350) format is specified in [[MS-DTYP]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-DTYP%5d.pdf) section 2.4.6; a string representation of [**security descriptors**](#gt_e5213722-75a9-44e7-b026-8e4833f0d350), called SDDL, is specified in [MS-DTYP] section 2.5.1.

**security identifier (SID)**: An identifier for security principals in Windows that is used to identify an account or a group. Conceptually, the [**SID**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d) is composed of an account authority portion (typically a [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca)) and a smaller integer representing an identity relative to the account authority, termed the relative identifier (RID). The [**SID**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d) format is specified in [MS-DTYP] section 2.4.2; a string representation of [**SIDs**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d) is specified in [MS-DTYP] section 2.4.2 and [[MS-AZOD]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-AZOD%5d.pdf) section 1.1.1.2.

**security provider**: A pluggable security module that is specified by the protocol layer above the [**remote procedure call (RPC)**](#gt_8a7f6700-8311-45bc-af10-82e10accd331) layer, and will cause the [**RPC**](#gt_8a7f6700-8311-45bc-af10-82e10accd331) layer to use this module to secure messages in a communication session with the server. The security provider is sometimes referred to as an [**authentication**](#gt_8e961bf0-95ba-4f58-9034-b67ccb27f317) service. For more information, see [C706] and [MS-RPCE].

**server**: A computer on which the [**remote procedure call (RPC)**](#gt_8a7f6700-8311-45bc-af10-82e10accd331) server is executing.

**Server Message Block (SMB)**: A protocol that is used to request file and print services from server systems over a network. The SMB protocol extends the CIFS protocol with additional security, file, and disk management support. For more information, see [[CIFS]](http://go.microsoft.com/fwlink/?LinkId=89836) and [[MS-SMB]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-SMB%5d.pdf).

**server restart**: Any event that causes the [**print server**](#gt_59fb3ddc-63cf-45df-8a90-46a6af9e00cb) to stop and start again, including a service or process shutdown and restart, an operating system shutdown and restart, or an unscheduled event, such as a power failure.

**service printer**: A [**print queue**](#gt_569f1f1c-f426-46fa-91d2-3d1eb0b19aa1) that sends rendered print jobs to a destination external to the print server using an implementation-specific mechanism that is opaque to the print server.

**shared printer**: A [**print queue**](#gt_569f1f1c-f426-46fa-91d2-3d1eb0b19aa1) that is available to [**print clients**](#gt_3b2da3d1-c159-4399-a6dd-dfd5f76fa2f5) as a share.

**Simple and Protected GSS-API Negotiation Mechanism (SPNEGO)**: An [**authentication**](#gt_8e961bf0-95ba-4f58-9034-b67ccb27f317) mechanism that allows Generic Security Services (GSS) peers to determine whether their credentials support a common set of GSS-API security mechanisms, to negotiate different options within a given security mechanism or different options from several security mechanisms, to select a service, and to establish a security context among themselves using that service. [**SPNEGO**](#gt_bc2f6b5e-e5c0-408b-8f55-0350c24b9838) is specified in [[RFC4178]](http://go.microsoft.com/fwlink/?LinkId=90461).

**spool file**: A representation of application content data than can be processed by a [**printer driver**](#gt_1a48eebd-e72c-494d-b8cb-84dfb7bc3b65). Common examples are enhanced metafile format and [**XML Paper Specification (XPS)**](#gt_0b8be35a-bc52-4a6d-b711-3ee441b1bf60) [[MSDN-XMLP]](http://go.microsoft.com/fwlink/?LinkId=90172). For more information, see [[MSDN-META]](http://go.microsoft.com/fwlink/?LinkId=90040).

**string resource**: A string that is stored in a resource file and that can be retrieved with a key. A [**string resource**](#gt_3dbf1133-6132-4436-b218-b11d3da31e94) is localizable into multiple languages. It is up to an AsyncUI client implementation to determine which language string to retrieve for a given key.

**system access control list (SACL)**: An access control list (ACL) that controls the generation of audit messages for attempts to access a securable object. The ability to get or set an object's [**SACL**](#gt_c189801e-3752-4715-88f4-17804dad5782) is controlled by a privilege typically held only by system administrators.

**TCPMON**: The [**port monitor module**](#gt_4c9689f8-854e-43e1-964e-5dd9a70b8aae) that manages standard [**TCP**](#gt_b08d36f6-b5c6-4ce4-8d2d-6f2ab75ea4cb)/IP [**ports**](#gt_ac027425-6043-4ad3-be27-56a381b75d38) on a Windows machine. [**TCPMON**](#gt_7e4cced1-30e5-4baf-8848-8a050df1be4e) supports configuring a [**TCP**](#gt_b08d36f6-b5c6-4ce4-8d2d-6f2ab75ea4cb)/IP [**port**](#gt_ac027425-6043-4ad3-be27-56a381b75d38) and obtaining information about the [**port**](#gt_ac027425-6043-4ad3-be27-56a381b75d38) configuration.

**terminal services (TS)**: A service on a server computer that allows delivery of applications, or the desktop itself, to various computing devices. When a user runs an application on a terminal server, the application execution takes place on the server computer and only keyboard, mouse, and display information is transmitted over the network. Each user sees only his or her individual session, which is managed transparently by the server operating system and is independent of any other client session.

**Transmission Control Protocol (TCP)**: A protocol used with the Internet Protocol (IP) to send data in the form of message units between computers over the Internet. TCP handles keeping track of the individual units of data (called packets) that a message is divided into for efficient routing through the Internet.

**unicast**: A delivery method used by media servers for providing content to connected clients in which each client receives a discrete stream that no other client has access to.

**Unicode**: A character encoding standard developed by the Unicode Consortium that represents almost all of the written languages of the world. The [**Unicode**](#gt_c305d0ab-8b94-461a-bd76-13b40cb8c4d8) standard [[UNICODE5.0.0/2007]](http://go.microsoft.com/fwlink/?LinkId=154659) provides three forms (UTF-8, UTF-16, and UTF-32) and seven schemes (UTF-8, UTF-16, UTF-16 BE, UTF-16 LE, UTF-32, UTF-32 LE, and UTF-32 BE).

**Unicode string**: A [**Unicode**](#gt_c305d0ab-8b94-461a-bd76-13b40cb8c4d8) 8-bit string is an ordered sequence of 8-bit units, a [**Unicode**](#gt_c305d0ab-8b94-461a-bd76-13b40cb8c4d8) 16-bit string is an ordered sequence of 16-bit code units, and a [**Unicode**](#gt_c305d0ab-8b94-461a-bd76-13b40cb8c4d8) 32-bit string is an ordered sequence of 32-bit code units. In some cases, it may be acceptable not to terminate with a terminating null character. Unless otherwise specified, all [**Unicode strings**](#gt_b069acb4-e364-453e-ac83-42d469bb339e) follow the [**UTF-16LE**](#gt_69864bcc-33be-41c9-9486-e18618ee3dd0) encoding scheme with no Byte Order Mark (BOM).

**Uniform Resource Identifier (URI)**: A string that identifies a resource. The URI is an addressing mechanism defined in Internet Engineering Task Force (IETF) Uniform Resource Identifier (URI): Generic Syntax [[RFC3986]](http://go.microsoft.com/fwlink/?LinkId=90453).

**Uniform Resource Locator (URL)**: A string of characters in a standardized format that identifies a document or resource on the World Wide Web. The format is as specified in [[RFC1738]](http://go.microsoft.com/fwlink/?LinkId=90287).

**Universal Naming Convention (UNC)**: A string format that specifies the location of a resource. For more information, see [MS-DTYP] section 2.2.57.

**Universal Plug and Play (UPnP)**: A set of computer network protocols, published by the UPnP Forum [[UPnP]](http://go.microsoft.com/fwlink/?LinkId=90553), that allow devices to connect seamlessly and that simplify the implementation of networks in home (data sharing, communications, and entertainment) and corporate environments. UPnP achieves this by defining and publishing UPnP device control protocols built upon open, Internet-based communication standards.

**universal serial bus (USB)**: An external bus that supports Plug and Play installation. It allows devices to be connected and disconnected without shutting down or restarting the computer.

**universally unique identifier (UUID)**: A 128-bit value. UUIDs can be used for multiple purposes, from tagging objects with an extremely short lifetime, to reliably identifying very persistent objects in cross-process communication such as client and server interfaces, manager entry-point vectors, and [**RPC**](#gt_8a7f6700-8311-45bc-af10-82e10accd331) objects. UUIDs are highly likely to be unique. UUIDs are also known as [**globally unique identifiers (GUIDs)**](#gt_f49694cc-c350-462d-ab8e-816f0103c6c1) and these terms are used interchangeably in the Microsoft protocol technical documents (TDs). Interchanging the usage of these terms does not imply or require a specific algorithm or mechanism to generate the UUID. Specifically, the use of this term does not imply or require that the algorithms described in [RFC4122] or [C706] must be used for generating the UUID.

**UNIX**: A multiuser, multitasking operating system developed at Bell Laboratories in the 1970s. In this document, the term "[**UNIX**](#gt_6084f060-d2f4-4ebb-8679-2d2ffa43b3ed)" is used to refer to any derivatives of this operating system.

**USBMON**: The port monitor that manages local USB ports on a Windows machine.

**UTF-16LE**: The Unicode Transformation Format - 16-bit, Little Endian encoding scheme. It is used to encode [**Unicode**](#gt_c305d0ab-8b94-461a-bd76-13b40cb8c4d8) characters as a sequence of 16-bit codes, each encoded as two 8-bit bytes with the least-significant byte first.

**virtual printer**: A [**print queue**](#gt_569f1f1c-f426-46fa-91d2-3d1eb0b19aa1) that does not produce physical printed output, and is not a [**fax printer**](#gt_ae9f38f1-e339-4eba-aacb-c0593a9ac177), [**file printer**](#gt_c55516ca-bd87-4ced-ad49-5cd31099832b), or [**service printer**](#gt_6ebe6443-5eaa-4a1d-b586-bbb6ae38c5bf).

**Web Services for Devices (WSD)**: A technology and associated API that expands on Microsoft's Web Services Dynamic Discovery Protocol [[WS-Discovery]](http://go.microsoft.com/fwlink/?LinkId=90576) to allow a client to discover and access remote devices and associated services across a network. WSD supports device discovery, description, control, and eventing.

**Web Services on Devices (WSD)**: A function-discovery protocol used to discover and communicate certain data structures in a HomeGroup network environment. Implementation details are specified in [[DPWS]](http://go.microsoft.com/fwlink/?LinkId=127841).

**well-known endpoint**: A preassigned, network-specific, stable address for a particular client/server instance. For more information, see [C706].

**white point**: The color value used as the reference to which the user adapts.

**Windows Event**: A technology and associated API that is typically used for troubleshooting application and driver software on a computer running Windows. A Windows Event contains an identifier and associated data. Events are published by an event provider to an [**event channel**](#gt_43907602-849d-4d1c-9ff8-a2c3228b3735) for consumption, and the identifiers are unique to the event provider. For more information, see [MSDN-WINEV].

**writability**: The abstract feature capability representing the ability of a [**domain controller (DC)**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd) to accept modifications and issue originating updates, with respect to a given [**naming context (NC)**](#gt_784c7cce-f782-48d8-9444-c9030ba86942) replica.

**WSDMON**: The [**port monitor**](#gt_1b5f368b-942b-40dd-84ea-fa9082916216) that supports printing to network printers that comply with [**WSD**](#gt_734b2ac6-73db-47d5-ae14-213ddd6e958b) technology.

**WS-Print**: The [**schema**](#gt_fd49ea36-576c-4417-93bd-d1ac63e71093) for [**WSD**](#gt_734b2ac6-73db-47d5-ae14-213ddd6e958b) printing. For more information, see [[MSDN-WSPRINT]](http://go.microsoft.com/fwlink/?LinkId=299386).

**XML Paper Specification (XPS)**: A Microsoft XML-based document format introduced in Windows Vista operating system. [**XML Paper Specification (XPS)**](#gt_0b8be35a-bc52-4a6d-b711-3ee441b1bf60) 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]](http://go.microsoft.com/fwlink/?LinkId=90214).

**MAY, SHOULD, MUST, SHOULD NOT, MUST NOT:** These terms (in all caps) are used as defined in [[RFC2119]](http://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](http://msdn.microsoft.com/en-us/library/dn781092.aspx).

### 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.

[C706] The Open Group, "DCE 1.1: Remote Procedure Call", C706, August 1997, [https://www2.opengroup.org/ogsys/catalog/c706](http://go.microsoft.com/fwlink/?LinkId=89824)

[IEEE1284] Institute of Electrical and Electronics Engineers, "IEEE Standard Signaling Method for a Bidirectional Parallel Peripheral Interface for Personal Computers - Description", IEEE Std 1284, 1994, [http://standards.ieee.org/reading/ieee/std\_public/description/busarch/1284-1994\_desc.html](http://go.microsoft.com/fwlink/?LinkId=89898)

[MS-ADA3] Microsoft Corporation, "[Active Directory Schema Attributes N-Z](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ADA3%5d.pdf)".

[MS-ADSC] Microsoft Corporation, "[Active Directory Schema Classes](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ADSC%5d.pdf)".

[MS-ADTS] Microsoft Corporation, "[Active Directory Technical Specification](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ADTS%5d.pdf)".

[MS-DRSR] Microsoft Corporation, "[Directory Replication Service (DRS) Remote Protocol](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-DRSR%5d.pdf)".

[MS-DTYP] Microsoft Corporation, "[Windows Data Types](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-DTYP%5d.pdf)".

[MS-ERREF] Microsoft Corporation, "[Windows Error Codes](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf)".

[MS-LCID] Microsoft Corporation, "[Windows Language Code Identifier (LCID) Reference](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-LCID%5d.pdf)".

[MS-PAR] Microsoft Corporation, "[Print System Asynchronous Remote Protocol](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-PAR%5d.pdf)".

[MS-RPCE] Microsoft Corporation, "[Remote Procedure Call Protocol Extensions](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-RPCE%5d.pdf)".

[MS-RRP] Microsoft Corporation, "[Windows Remote Registry Protocol](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-RRP%5d.pdf)".

[MS-SMB2] Microsoft Corporation, "[Server Message Block (SMB) Protocol Versions 2 and 3](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-SMB2%5d.pdf)".

[MS-SMB] Microsoft Corporation, "[Server Message Block (SMB) Protocol](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-SMB%5d.pdf)".

[RFC1001] Network Working Group, "Protocol Standard for a NetBIOS Service on a TCP/UDP Transport: Concepts and Methods", RFC 1001, March 1987, [http://www.ietf.org/rfc/rfc1001.txt](http://go.microsoft.com/fwlink/?LinkId=90260)

[RFC1157] Case, J., Fedor, M., Schoffstall, M., and Davin, J., "A Simple Network Management Protocol (SNMP)", RFC 1157, May 1990, [http://www.ietf.org/rfc/rfc1157.txt](http://go.microsoft.com/fwlink/?LinkId=121934)

[RFC1179] McLaughlin III, L., "Line Printer Daemon Protocol", RFC 1179, August 1990, [http://www.ietf.org/rfc/rfc1179.txt](http://go.microsoft.com/fwlink/?LinkId=90269)

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

[RFC2251] Wahl, M., Howes, T., and Kille, S., "Lightweight Directory Access Protocol (v3)", RFC 2251, December 1997, [http://www.ietf.org/rfc/rfc2251.txt](http://go.microsoft.com/fwlink/?LinkId=90325)

[RFC2254] Howes, T., "The String Representation of LDAP Search Filters", RFC 2254, December 1997, [http://www.ietf.org/rfc/rfc2254.txt](http://go.microsoft.com/fwlink/?LinkId=90328)

[RFC2616] Fielding, R., Gettys, J., Mogul, J., et al., "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999, [http://www.rfc-editor.org/rfc/rfc2616.txt](http://go.microsoft.com/fwlink/?LinkId=90372)

[RFC2781] Hoffman, P., and Yergeau, F., "UTF-16, an encoding of ISO 10646", RFC 2781, February 2000, [http://www.rfc-editor.org/rfc/rfc2781.txt](http://go.microsoft.com/fwlink/?LinkId=90380)

[RFC3986] Berners-Lee, T., Fielding, R., and Masinter, L., "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005, [http://www.ietf.org/rfc/rfc3986.txt](http://go.microsoft.com/fwlink/?LinkId=90453)

[RFC4122] Leach, P., Mealling, M., and Salz, R., "A Universally Unique Identifier (UUID) URN Namespace", RFC 4122, July 2005, [http://www.ietf.org/rfc/rfc4122.txt](http://go.microsoft.com/fwlink/?LinkId=90460)

### Informative References

[DEVMODE] Microsoft Corporation, "DEVMODE structure", [http://msdn.microsoft.com/en-us/library/dd183565(VS.85).aspx](http://go.microsoft.com/fwlink/?LinkId=89844)

[ECMA-388] ECMA International, "Open XML Paper Specification", ECMA-388, June 2009, [http://www.ecma-international.org/publications/standards/Ecma-388.htm](http://go.microsoft.com/fwlink/?LinkId=158846)

[IEEE802.3-2008] Institute of Electrical and Electronics Engineers, "Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications - Description", IEEE Std 802.3, 2008, [http://standards.ieee.org/getieee802/802.3.html](http://go.microsoft.com/fwlink/?LinkId=161001)

[MS-ADLS] Microsoft Corporation, "[Active Directory Lightweight Directory Services Schema](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ADLS%5d.pdf)".

[MS-ADOD] Microsoft Corporation, "[Active Directory Protocols Overview](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ADOD%5d.pdf)".

[MS-AZOD] Microsoft Corporation, "[Authorization Protocols Overview](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-AZOD%5d.pdf)".

[MS-EMFSPOOL] Microsoft Corporation, "[Enhanced Metafile Spool Format](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-EMFSPOOL%5d.pdf)".

[MS-EMF] Microsoft Corporation, "[Enhanced Metafile Format](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-EMF%5d.pdf)".

[MS-PAN] Microsoft Corporation, "[Print System Asynchronous Notification Protocol](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-PAN%5d.pdf)".

[MS-RAP] Microsoft Corporation, "[Remote Administration Protocol](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-RAP%5d.pdf)".

[MSDN-ADOVRVW] Microsoft Corporation, "Active Directory Schema Terminology", [http://msdn.microsoft.com/en-us/library/ms675087.aspx](http://go.microsoft.com/fwlink/?LinkId=89951)

[MSDN-BIDI] Microsoft Corporation, "Bidirectional Communication", [http://msdn.microsoft.com/en-us/library/aa907381.aspx](http://go.microsoft.com/fwlink/?LinkId=122185)

[MSDN-CAB] Microsoft Corporation, "Microsoft Cabinet Format", March 1997, [http://msdn.microsoft.com/en-us/library/bb417343.aspx](http://go.microsoft.com/fwlink/?LinkId=226293)

[MSDN-FIELD] Microsoft Corporation, "Field Attributes", [http://msdn.microsoft.com/en-us/library/aa373864(VS.85).aspx](http://go.microsoft.com/fwlink/?LinkId=184577)

[MSDN-GPDFiles] Microsoft Corporation, "Introduction to GPD Files", [http://msdn.microsoft.com/en-us/library/ff551750(VS.85).aspx](http://go.microsoft.com/fwlink/?LinkId=221046)

[MSDN-MPD] Microsoft Corporation, "Microsoft Print Drivers", [http://msdn.microsoft.com/en-us/library/ff556565.aspx](http://go.microsoft.com/fwlink/?LinkId=91840)

[MSDN-MUI] Microsoft Corporation, "Language Identifier Constants and Strings", [https://msdn.microsoft.com/en-us/library/windows/desktop/dd318693(v=vs.85).aspx](http://go.microsoft.com/fwlink/?LinkId=90048)

[MSDN-SPOOL] Microsoft Corporation, "Print Spooler Components", [http://msdn.microsoft.com/en-us/library/ff561109.aspx](http://go.microsoft.com/fwlink/?LinkId=90134)

[MSDN-UINF] Microsoft Corporation, "Using INF Files", [http://msdn.microsoft.com/en-us/library/Aa741213.aspx](http://go.microsoft.com/fwlink/?LinkId=90152)

[MSDN-XMLP] Microsoft Corporation, "A First Look at APIs For Creating XML Paper Specification Documents", [http://msdn.microsoft.com/en-us/magazine/cc163664.aspx](http://go.microsoft.com/fwlink/?LinkId=90172)

[MSFT-XMLPAPER] Microsoft Corporation, "XML Paper Specification", [http://www.microsoft.com/whdc/xps/default.mspx](http://go.microsoft.com/fwlink/?LinkId=90214)

[PS-PPD4.3] Adobe Systems Incorporated, "PostScript Printer Description File Format Specification", version 4.3, February 1996, [http://partners.adobe.com/public/developer/en/ps/5003.PPD\_Spec\_v4.3.pdf](http://go.microsoft.com/fwlink/?LinkId=221049)

[RFC819] Su, Z.S. and Postel, J., "The Domain Naming Convention for Internet User Applications", RFC 819, August 1982, [http://www.ietf.org/rfc/rfc0819.txt](http://go.microsoft.com/fwlink/?LinkId=90495)

[USBPRINT] USB Implementers Forum, "Universal Serial Bus Device Class Definition for Printing Devices", version 1.1, January 2000, [http://www.usb.org/developers/docs/devclass\_docs/usbprint11a021811.pdf](http://go.microsoft.com/fwlink/?LinkId=240056)

## Overview

The Print System Remote Protocol provides the following functions:

* Management of the print system of a [**print server**](#gt_59fb3ddc-63cf-45df-8a90-46a6af9e00cb) from a client.
* Communication of print job data from a client to a print server.
* Notifications to the client of changes in the print server's print system.

Server processing instructions are specified by the parameters that are used in the protocol methods. These parameters include:

* [**Printer driver**](#gt_1a48eebd-e72c-494d-b8cb-84dfb7bc3b65) configuration information.
* The [**spool file**](#gt_4c0e7da9-88a8-4430-abd7-27e7250b8180) format for the print data that is sent by the client.
* The [**access level**](#gt_54c8a9d7-fa94-444b-b8d8-f6485bad223f) of the connection.
* The target [**print queue**](#gt_569f1f1c-f426-46fa-91d2-3d1eb0b19aa1) name for name-based methods.
* A handle to the target print queue for handle-based methods.

Status information is communicated back to the client in the return codes from calls that are made to the print server.

The following sections give an overview of these functions.

### Management of the Print System

A client can use this protocol to perform remote management operations on a print server. With server access credentials, client applications can manipulate the print server state and print server components, such as printer driver configuration and print queue configuration, or add printer drivers and printers; they can monitor the print queue status; and they can perform general print server administration.

These operations are supported in the protocol by a set of [container](#Section_8ca20b730d864e57a009f797ef1b2de8) structures that are used by different print system components, specifically: [DRIVER\_CONTAINER](#Section_3a3f9cf78ec44921b1f686cf8d139bc2), [FORM\_CONTAINER](#Section_49508bc38fcb49f5b92d0724666d76a4), [JOB\_CONTAINER](#Section_0cd5f47b54214b199ed05a29d63f3c2a), [PORT\_CONTAINER](#Section_e05adf6ca04e4e398b34e7c118ce2bc8), [SECURITY\_CONTAINER](#Section_6cf2b4ff1b2b453c87f985243823cea8), and [PRINTER\_CONTAINER](#Section_ad7d4d4e0feb45e093edb3b2b9a313d1). These print system components are supported as specified in section [2.2.1](#Section_df6af0abee9745aaae87186f83ab2265).

To produce printed output that is the same, regardless of the configuration, the printer driver that is installed on the client computer must be identical to or compatible with the printer driver that is installed on the print server. This protocol provides the methods that the client can use after it connects to a printer on a print server to obtain the information about the printer driver that is associated with the printer. If necessary, the client computer can use this information to download the printer driver from the print server. For more information about printer drivers, see [[MSDN-MPD]](http://go.microsoft.com/fwlink/?LinkId=91840).

The client can also use this protocol to obtain detailed information about the settings of the printer and the printer driver that are installed on the server. The client application can use this information to perform layout and to make device-specific choices about paper formats, resolution, and color handling. After the client connects to a printer, this protocol provides the methods that the client can use to query these settings.

The following figure illustrates this interaction, using the scenario of adding a new printer as an example.

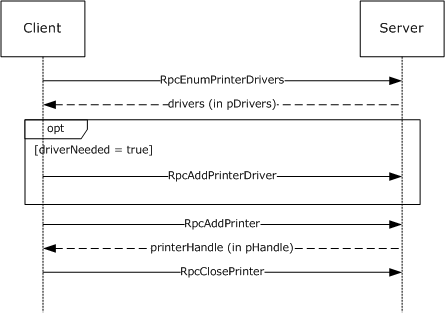


Figure : Adding a new printer

### Communication of Print Job Data

Communication of print job data enables a client to print to devices that are hosted by print servers.

In one configuration, a client uses a printer driver that is installed on the client computer to convert a graphical representation of application content and layout into [**device**](#gt_4712ded2-d927-4930-9450-29499f98cef4)-specific [**page description language (PDL)**](#gt_a04e8872-d39e-4ae8-a870-b2e36b266302) data. It then sends the data, also called [**RAW**](#gt_cea5f44d-0398-47cf-bb34-110ffddade11) data, to the print server using methods this protocol provides. The print server can temporarily store the RAW data from the client in a spool file, or it can print it immediately. As the print server sends the data to the target printer, the [**print processor**](#gt_062f3e76-3149-468c-bf64-1535651725d1) on the print server that is associated with the target printer can post-process the RAW data in an implementation-specific way.

In another configuration, a client sends data to the print server in an intermediate format that contains graphics primitives and layout information in addition to processing instructions for the print server. The print server can temporarily store this intermediate data in a spool file, or it can print it immediately. As the data is sent to the printer, the print processor on the print server that is associated with the printer converts the data from the intermediate spool file to device-specific PDL data, typically by using the printer driver that is installed on the print server.

The following diagram illustrates this interaction.

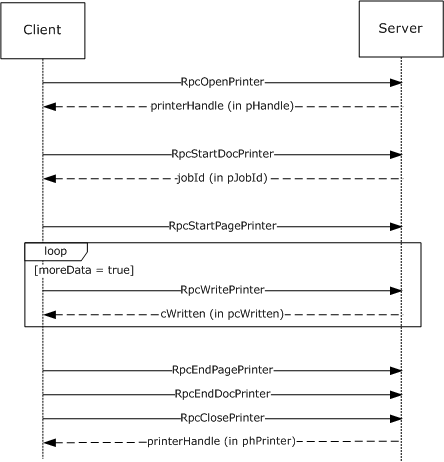


Figure : Communication of print job data

### Notification of Print System Changes

This protocol also provides the methods that a [**print client**](#gt_3b2da3d1-c159-4399-a6dd-dfd5f76fa2f5) can use to register for incremental change notifications. These notifications enable the client application to maintain an accurate local view of the printer and printer driver settings by enabling the client application to synchronize the local view with the actual settings of those components on the print server, without having to repeatedly query the server for its complete configuration information.

For status updates, a print client registers for notifications of state changes when it connects to a print server. The server creates a new remote procedure call (RPC) connection in the reverse direction, back to the client, which is subsequently used to send notifications to the client. When the status of a server resource changes—such as a print queue goes online, goes offline, or enters an error state—the server sends a notification to the registered client.

Notifications include status changes of print server resources; for example, when a print queue goes online, goes offline, or enters an error state.

The following diagram illustrates this interaction. For more information, see section [3.2.4.2.4](#Section_42a5b690e5cf4e37a3106dd70071af57).

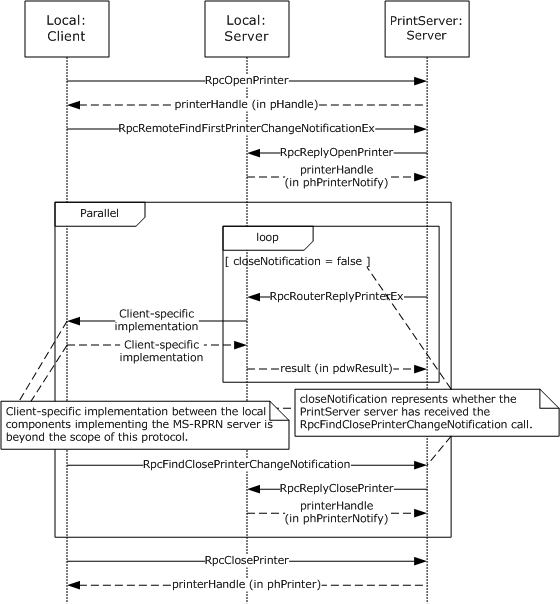


Figure : Notification of print system changes

The box labeled 'Local Server' in the above diagram represents an entity running on the client system. This entity is either implemented in the end-user application or in a separate process. Since the entity implements a Print System Remote Protocol [**endpoint**](#gt_b91c1e27-e8e0-499b-8c65-738006af72ee), which can only be registered once per system, there can be at most one of these entities running on the client system at any time. If an end-user application directly implements this entity, this end-user application would not be able to run on a system that also runs a print system implementation, and only one such application could.

In addition to composing and returning the notifications, the print server maintains a change identifier that it changes whenever the server-side printing configuration changes; for example, changes to user-configurable settings, print queue items, print job status, or to the printer driver would cause this identifier to change. The print client can query this change identifier by using the [RpcGetPrinterData (section 3.1.4.2.7)](#Section_9c04a0f2adc443938df47c36a14183f5) method that is defined in this protocol and calling it with the *pValueName* parameter pointing to the string "ChangeID".

When a disconnected print client reconnects to the print server, it can query the change identifier again, and if the change identifier is different from the one returned when it queried before it was disconnected, the client should retrieve the complete configuration information and update its view of the server configuration. The client retrieves the complete configuration using the functions for [Management of the Print System (section 1.3.1)](#Section_da76c8862f7749799a76c2255e6de8bc).

## Relationship to Other Protocols

The Print System Remote Protocol is dependent on the RPC protocol specified in [[MS-RPCE]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-RPCE%5d.pdf).

The Print System Remote Protocol does not specify methods for file transfer between client and server; therefore, the [**Server Message Block (SMB)**](#gt_09dbec39-5e75-4d9a-babf-1c9f1d499625) Version 2.0 Protocol, specified in [[MS-SMB2]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-SMB2%5d.pdf), is the preferred protocol for all file transfer operations, including printer driver downloads.

These protocol relationships are shown in the following figure:

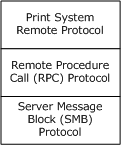


Figure : Protocol Relationships

The Print System Remote Protocol is related to other protocols as follows:

* The Print System Asynchronous Notification Protocol, specified in [[MS-PAN]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-PAN%5d.pdf), is dependent on the Print System Remote Protocol.
* The Print System Asynchronous Remote Protocol, specified in [[MS-PAR]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-PAR%5d.pdf), uses many data structures and parameter definitions that are also specified in sections [2.2.1](#Section_df6af0abee9745aaae87186f83ab2265) and [2.2.2](#Section_99fdf2cf79bb43bf997c88b33538a5e1) of this specification.

**Note**  The implementation of Print System Remote is required for all print servers, but a print server can additionally implement Print System Asynchronous Remote.

* Limited enumeration of printer configuration information can be done using the [**Remote Administration Protocol (RAP)**](#gt_6442eac9-3264-4d95-a435-fdc08e71603f), specified in [[MS-RAP]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-RAP%5d.pdf); however, the Print System Asynchronous Remote Protocol provides richer semantics than RAP. Because RAP is not supported over SMB Version 2.0, Print System Remote is the preferred access protocol for printer configuration information.
* The print client and print server implementations of this protocol use the Active Directory Technical Specification Protocol specified in [[MS-ADTS]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ADTS%5d.pdf), to locate domain controllers and use [**LDAP**](#gt_45643bfb-b4c4-432c-a10f-b98790063f8d), as specified in [[RFC2251]](http://go.microsoft.com/fwlink/?LinkId=90325), to access data in the [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90), when available. This protocol does not require availability of Active Directory. For more details, see [Directory Service Interaction (section 2.3)](#Section_5baf16d5919d4ed68086fd0e63775d5e).[<1>](#Appendix_A_1)

## Prerequisites/Preconditions

This protocol is an RPC interface and therefore has the prerequisites specified in [[MS-RPCE]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-RPCE%5d.pdf) section 1.5 as being common to RPC interfaces.

It is assumed that a client of this protocol has obtained the name of a print server that supports this protocol before it is invoked. There are various ways a client can accomplish that; for information see [[MS-ADLS]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ADLS%5d.pdf), [[MS-ADSC]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ADSC%5d.pdf), [[MS-RAP]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-RAP%5d.pdf), and [[MS-SMB2]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-SMB2%5d.pdf).

## Applicability Statement

The Print System Remote Protocol is applicable only for printing operations between a system functioning as a client and a system functioning as a print server. This protocol scales from home use, in which a single printer is connected to a single computer; to office use, in which print-devices are shared between computers; to enterprise use, in which multiple print servers are employed in a cluster configuration, and the client configuration is managed by a directory access protocol, such as Active Directory as specified in [[MS-ADTS]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ADTS%5d.pdf).

## Versioning and Capability Negotiation

This specification covers versioning issues in the following areas:

* **Supported Transports**: The Print System Remote Protocol requires use of RPC over named pipes only.
* **Protocol Versions**: The protocol version specified in the [**Interface Definition Language (IDL)**](#gt_73177eec-4092-420f-92c5-60b2478df824) file is 1.0.

Versioning of data structures defined by the protocol is controlled using a Level value in all [Containers (section 2.2.1.2)](#Section_8ca20b730d864e57a009f797ef1b2de8); the usage pattern of Level values is defined in section [2.2](#Section_9b03c5103a0e48219aa5fc79dbf8a188). Levels can be sequential, and data structures identified by a later version level, if extending an earlier level, are a superset of the data structure identified by the earlier level. The Level value is also a parameter to some RPC methods.

* **Security and Authentication Methods**: Versioning of security is handled by the underlying [**RPC transport**](#gt_c2eeb200-3cd0-4916-966e-d7d6bff1737a); see [[MS-RPCE]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-RPCE%5d.pdf) section 3.3.3.3 for more information.
* **Localization**: This protocol specifies languages and localizable [**string resources**](#gt_3dbf1133-6132-4436-b218-b11d3da31e94) for [**printer forms**](#gt_8dfb0042-b0de-4108-ac52-b148427603db). See [LANGID (section 2.2.1.1.3)](#Section_5fc1886e9a6848c3a696997e569517f7) and [RPC\_FORM\_INFO\_2 (section 2.2.1.6.2)](#Section_dca49b4f3eb146569e972f0b0c569bf6) for details.
* **Return Values**: The methods that make up this RPC interface require a return value of zero to indicate successful completion and nonzero values to indicate failure, except where specified otherwise. A server-side implementation of this protocol would preferably use any nonzero Win32 error value to signify an error condition, as described in section [1.8](#Section_39112098aa6d440ab661a5061dfd100d). Unless otherwise specified, the client side of the Print System Remote Protocol is not allowed to interpret returned error codes. Unless otherwise specified, the client side of the Print System Remote Protocol is required to simply return error codes to the invoking application without taking any protocol action.
* **Capability Negotiation**: Functional negotiation is supported through the use of Container levels; see section 2.2.1.2. On connection to a server, the client requests a level. If the information level is a level supported by the server, the server is required to process the request. Otherwise, the server has to return an error to the client, and the client would preferably repeat the request with a lower level.

Furthermore, to avoid unnecessary network calls, the client determines the server's capabilities by comparing the value returned by the server in the **dwBuildNumber** member of [OSVERSIONINFO (section 2.2.3.10.1)](#Section_5942648eb54f4e22a0e22d000e084b23) with well-known version-specific **dwBuildNumber** values.[<2>](#Appendix_A_2)

## Vendor-Extensible Members

The methods defined in the Print System Remote Protocol specify either the DWORD or HRESULT [**data type**](#gt_0d327bc0-a461-4723-8beb-d935fdab4895) for return values. DWORD return values are Win32 error codes taken from the Windows error number space specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf) section 2.2. Implementers reuse those values with their indicated meanings. Choosing any other value runs the risk of collisions.

HRESULT method return values are used as defined in [MS-ERREF] section 2.1. Vendors can choose their own HRESULT values, but the C bit (0x20000000) is set, indicating that it is a customer code.

Print server implementations MUST generate [**GUID**](#gt_f49694cc-c350-462d-ab8e-816f0103c6c1), as defined in [[MS-DTYP]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-DTYP%5d.pdf) sections 2.3.4, 2.3.4.2, and 2.3.4.3, strings for the purpose of identifying specific printers. The 128-bit value encoded by the GUID string SHOULD conform to the specification in [[RFC4122]](http://go.microsoft.com/fwlink/?LinkId=90460) section 4.

## Standards Assignments

The Print System Remote Protocol requires use of the following private assignments:

| Parameter | Value | Reference |
| --- | --- | --- |
| RPC [**UUID**](#gt_c4813fc3-b2e5-4aa3-bde7-421d950d68d3) | 12345678-1234-ABCD-EF00-0123456789AB | [Message Processing Events and Sequencing Rules (section 3.1.4)](#Section_700f007d72b74b44951756569704bb8d) |
| RPC [**well-known endpoint**](#gt_a7498f8e-e85d-473b-9fc9-d2fffdf71c8a) | **\pipe\spoolss** | [Transport (section 2.1)](#Section_fdf1138af6b94b00ada61fbb6097d683) |

# Messages

## Transport

The Print System Remote Protocol uses RPC over named pipes (see [[MS-RPCE]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-RPCE%5d.pdf) section 2.1.1.2) for RPC sequences. The well-known endpoint **\pipe\spoolss** is used for RPC calls made from the print client to the print server. The client MUST use no [**authentication**](#gt_8e961bf0-95ba-4f58-9034-b67ccb27f317), and the server MUST accept connections without authentication.

An endpoint with the same name MUST also be used for RPC calls made from the server to send printer change notifications back to the client; those calls are [RpcReplyOpenPrinter (section 3.2.4.1.1)](#Section_7fcd3036d45a4ec7b081f2b860e66676), [RpcRouterReplyPrinter (section 3.2.4.1.2)](#Section_38bf5d2ce4a744b2b16f91ff8226f836), [RpcReplyClosePrinter (section 3.2.4.1.3)](#Section_d6c73ecbc02b4150a01bad73a642990a), and [RpcRouterReplyPrinterEx (section 3.2.4.1.4)](#Section_1b44cb8295b5485c9ec90360c3a3ecb0). The client MUST accept connections without authentication from the server for these methods. If the client provides authentication information (as specified in [MS-RPCE] 2.2.1.1.8), the server MAY impersonate the client (as specified in [MS-RPCE] 2.2.1.1.9), while processing a method.[<3>](#Appendix_A_3)

## Common Data Types

The Print System Remote Protocol MUST indicate to the RPC runtime that it is to support both the [**Network Data Representation (NDR)**](#gt_9ebf9540-2c31-43bc-bc56-4a932faabf2d) and **NDR64** [**RPC transfer syntaxes**](#gt_01216ea7-ac8a-4cc8-9d19-b901bc424c09) and provide a negotiation mechanism for determining which transfer syntax will be used, as specified in [[MS-RPCE]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-RPCE%5d.pdf) section 3.

This protocol MUST enable the **ms\_union** extension as specified in [MS-RPCE] section 2.2.4.

The Print System Remote Protocol employs a combination of the following data representations:

* IDL data structures used with RPC methods, including structures used as containers for custom-[**marshaled**](#gt_37e89d91-2800-49d1-9d17-45ebe167ad87), custom C data, are specified in section [2.2.1](#Section_df6af0abee9745aaae87186f83ab2265).
* Custom C data structures and their wire formats used within custom marshaled data streams are specified in section [2.2.2](#Section_99fdf2cf79bb43bf997c88b33538a5e1).

Unless noted otherwise, the following statements apply to this specification:

* All strings defined in this protocol MUST consist of characters encoded in [**Unicode**](#gt_c305d0ab-8b94-461a-bd76-13b40cb8c4d8) [**UTF-16LE**](#gt_69864bcc-33be-41c9-9486-e18618ee3dd0) and MUST be null-terminated. Each UTF-16 codepoint in a string, including terminating null characters, MUST occupy 16 bits as specified in [[RFC2781]](http://go.microsoft.com/fwlink/?LinkId=90380), section 2.1.
* A list of strings is referred to as a [**multisz**](#gt_70f2be54-4d89-416a-965a-c7055873d2b1) in this protocol. In a multisz, the characters making up the string N+1 MUST directly follow the terminating null character of string N. The last string in a multisz MUST be terminated by two null characters.
* All parameters or members specifying the number of characters in a string or multisz specify the number of characters including terminating null characters.
* All constraints specifying the maximum number of characters in a string or multisz specify the number of characters including terminating null characters.
* All parameters or members specifying the number of bytes in buffers containing a string or multisz specify the number of bytes including terminating null characters.
* Custom-marshaled data that consists of more than a single byte is specified in [**little-endian**](#gt_079478cb-f4c5-4ce5-b72b-2144da5d2ce7) byte order.
* The term NULL means a NULL pointer, and zero means the number 0.
* All parameters or members specifying the size of a buffer pointed to by another parameter or member MUST be zero if the pointer parameter or member is NULL.
* The term "empty string" means a string containing only the terminating null character.
* The term "optional pointer" means that providing a pointer value in the parameter or member is optional. If the pointer value is not provided, the value of the parameter or member MUST be NULL.
* This protocol specification uses curly-braced string GUIDs as specified in [[MS-DTYP]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-DTYP%5d.pdf) section 2.3.4.3.

This protocol introduces a variety of data types that bundle information about printers, printer drivers, print jobs, and [**ports**](#gt_ac027425-6043-4ad3-be27-56a381b75d38). These data types are collectively referred to as [INFO](#Section_f29e9057df2940d49c51c615c0da62a1) data types, and include [DRIVER\_INFO\_1 (section 2.2.1.5.1)](#Section_df308bbf8ff74b0f930bcb9b90080b90), [PRINTER\_INFO\_1 (section 2.2.1.10.2)](#Section_0d1a296b7027441c83938aa5c3764b27), [JOB\_INFO\_1 (section 2.2.1.7.1)](#Section_794c1ba9ba0a4498ba9c2635563460dd), and [PORT\_INFO\_1 (section 2.2.1.9.1)](#Section_9bab3fb5b36948f28f083082ca13e739). As data types were refined in the evolution of this protocol, new INFO data type versions have been introduced to represent extended or different bundles of information. The term "level" is used to differentiate between the different type versions, and the number of the level is reflected in the name of the data type, for example, JOB\_INFO\_1, [JOB\_INFO\_2](#Section_5a48c6c9fc5f42ffa5a7f111fee58d1a), and [JOB\_INFO\_3](#Section_1385aef9d3234e8d9d285f2f011bd39c).

To simplify method parameter lists and to increase robustness of RPC [**marshaling**](#gt_e0245240-9927-4991-85af-f5babcd95bfd), the protocol introduces CONTAINER data types, which consolidate the input parameters used by RPC methods. Some CONTAINER data types hold a Level value along with a union of pointer values pointing to different INFO data type versions; the specific Level values available for each CONTAINER data type are documented in section [2.2.1.2](#Section_8ca20b730d864e57a009f797ef1b2de8). For example, a [JOB\_CONTAINER (section 2.2.1.2.5)](#Section_0cd5f47b54214b199ed05a29d63f3c2a) contains a level value and a union of pointers to the different [JOB\_INFO](#Section_87ea3e7722fb4914b803ef3b2619195a) data type versions, which are selected by the Level value. Other CONTAINER data types hold a pointer value that points to a structure, along with a numerical value representing the size of the structure. For example, a [DEVMODE\_CONTAINER (section 2.2.1.2.1)](#Section_c2b14fe764794460a7ba633a845dd61a) contains a size value and a pointer to a custom-marshaled structure. Finally, several CONTAINER data types hold a version value, a value representing a set of flags, an array of structures, and a value representing the number of elements in the array. The [RPC\_BIDI\_REQUEST\_CONTAINER (section 2.2.1.2.10)](#Section_4ea856c44c374485868ae7ac8e331e5e) is an example of a CONTAINER data type in this category.

Most of the INFO data types have an IDL form and a custom-marshaled form. IDL forms can be used in conjunction with CONTAINER data types, as input parameters to methods that set values, such as [RpcSetPrinter (section 3.1.4.2.5)](#Section_382997b5b02a4b6a8c6849a64b8c45f8), while custom-marshaled forms can be used as output parameters to methods that get values, such as [RpcGetPrinter (section 3.1.4.2.6)](#Section_ccbf288a0768416dbede0ef1987fc286). The layout and order of members of IDL forms are in most cases the same as those of corresponding custom-marshaled forms, with the distinction that IDL forms use the type "[string] wchar\_t \*" to point to strings, while custom-marshaled forms use an offset relative to the start of the structure.

As an exception to the preceding rule, the layout of IDL-marshaled structures that contain pointers to multisz data **differs** from the layout of custom-marshaled forms, in that IDL-marshaled structures need to define a **length** member for each IDL-marshaled member of type **pointer to** multisz; the names of such IDL-marshaled members start with RPC\_.

To increase clarity, an underscore has been prepended to the names of all custom-marshaled structures, for which an IDL-marshaled form exists. For example, [\_DRIVER\_INFO\_1](#Section_59eab35482fd4299884e9a1226073a4a) is the name of the custom-marshaled structure that corresponds to DRIVER\_INFO\_1, the IDL-marshaled form.

When IDL-marshaled structures that contain pointer types to variable-length data without field IDL attributes (see [[MSDN-FIELD]](http://go.microsoft.com/fwlink/?LinkId=184577)), such as [string] or [size\_is(...)], are used as input arguments to methods, either directly or in CONTAINER structures, the pointers and variables to which they point cannot be marshaled by RPC, because RPC does not know the length of the data that is pointed to. Examples are the **pSecurityDescriptor** and **pDevMode** members of the [PRINTER\_INFO\_2](#Section_d40734e1ac9b4b2a862883a1bc7bbeed) structure.

To address this problem, methods that specify such input arguments accept separate CONTAINER structures that pass in custom-marshaled or self-relative forms of the pointers and the variables that they reference. Examples of such methods are RpcSetPrinter and [RpcAddPrinterEx (section 3.1.4.2.15)](#Section_3af45717308f4b4f8428430d12856110). Individual method sections specify how affected pointer members and CONTAINER structures MUST be treated.

### IDL Data Types

In addition to the RPC base types and definitions specified in [[C706]](http://go.microsoft.com/fwlink/?LinkId=89824) and [[MS-DTYP]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-DTYP%5d.pdf), the Print System Remote Protocol defines data types in the following sections:

* [Common IDL Data Types (section 2.2.1.1)](#Section_8c5b1f140514438da966e327c1dad594)
* [Containers (section 2.2.1.2)](#Section_8ca20b730d864e57a009f797ef1b2de8)
* [Members in INFO Structures (section 2.2.1.3)](#Section_964b517f67304b4a99edee9648e7f1f7)
* [DOC\_INFO\_1 (section 2.2.1.4)](#Section_20f7b510e6a04f6c97a05630d1e70879)
* [DRIVER\_INFO (section 2.2.1.5)](#Section_17943579a5144d0b8ded422265b23f45)
* [FORM\_INFO (section 2.2.1.6)](#Section_2fef1fa4518f4e958572c6491442cf2a)
* [JOB\_INFO (section 2.2.1.7)](#Section_87ea3e7722fb4914b803ef3b2619195a)
* [MONITOR\_INFO (section 2.2.1.8)](#Section_822a6b4d37814a579eefd572dcf4d86f)
* [PORT\_INFO (section 2.2.1.9)](#Section_05c1a8d20bcb4292bc65c8cc2e7f90cf)
* [PRINTER\_INFO (section 2.2.1.10)](#Section_125fb29b3c5141fca333b1e0e8794a91)
* [SPLCLIENT\_INFO (section 2.2.1.11)](#Section_d23fee1a3f624cbb8a06d01180fe2ff9)
* [Bidirectional Communication Data (section 2.2.1.12)](#Section_38fb51d87f9e4b16ac406f48244214ed)
* [Printer Notification Data (section 2.2.1.13)](#Section_c7e020b1c21948d29130a4e96ac03c3f)
* [Job Named Properties (section 2.2.1.14)](#Section_aa85aaa9888e4f9eab77f75c1cd9ae08)
* [Branch Office Print Remote Logging Structures (section 2.2.1.15)](#Section_b331caea43e24a00b5031969e7a0ab92)

#### Common IDL Data Types

##### DEVMODE

The DEVMODE structure is a truncated form of the variable-length, custom-marshaled [\_DEVMODE](#Section_ccc2a501794e4d2bb312f69c75131c2e) structure (section 2.2.2.1), which is version-specific and implementation-specific and cannot be expressed using IDL attributes.

All members of this structure are specified in section 2.2.2.1.

1. typedef struct \_devicemode {
2. wchar\_t dmDeviceName[32];
3. unsigned short dmSpecVersion;
4. unsigned short dmDriverVersion;
5. unsigned short dmSize;
6. unsigned short dmDriverExtra;
7. DWORD dmFields;
8. short dmOrientation;
9. short dmPaperSize;
10. short dmPaperLength;
11. short dmPaperWidth;
12. short dmScale;
13. short dmCopies;
14. short dmDefaultSource;
15. short dmPrintQuality;
16. short dmColor;
17. short dmDuplex;
18. short dmYResolution;
19. short dmTTOption;
20. short dmCollate;
21. wchar\_t dmFormName[32];
22. unsigned short reserved0;
23. DWORD reserved1;
24. DWORD reserved2;
25. DWORD reserved3;
26. DWORD dmNup;
27. DWORD reserved4;
28. DWORD dmICMMethod;
29. DWORD dmICMIntent;
30. DWORD dmMediaType;
31. DWORD dmDitherType;
32. DWORD reserved5;
33. DWORD reserved6;
34. DWORD reserved7;
35. DWORD reserved8;
36. } DEVMODE;

##### GDI\_HANDLE

The GDI\_HANDLE serves as an [**RPC context handle**](#gt_9dfe84e3-a772-4ad1-9a25-1255c1dc0f84) for methods that specify a printer [**information context**](#gt_e6193b0f-37ad-4f30-978b-0e1e1291f674) handle parameter. RPC context handles are specified in [[C706]](http://go.microsoft.com/fwlink/?LinkId=89824) sections 2 and 6.

This type is declared as follows:

1. typedef [context\_handle] void\* GDI\_HANDLE;

The GDI\_HANDLE context handle is returned by [RpcCreatePrinterIC](#Section_b0686ba23a6b4a6c943fb26600dbd88c).

##### LANGID

The LANGID data type identifies the human language used for the user interface for printing. Details are specified in [[MS-LCID]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-LCID%5d.pdf).

This type is declared as follows:

1. typedef unsigned short LANGID;

##### PRINTER\_HANDLE

The PRINTER\_HANDLE serves as an RPC context handle for methods that specify a printer object handle parameter. RPC context handles are specified in [[C706]](http://go.microsoft.com/fwlink/?LinkId=89824) sections 2 and 6.

This type is declared as follows:

1. typedef [context\_handle] void\* PRINTER\_HANDLE;

The PRINTER\_HANDLE context handle is returned by [RpcAddPrinter](#Section_bd0bb8d526fd470581f5614d32b9f38f), [RpcAddPrinterEx](#Section_3af45717308f4b4f8428430d12856110), [RpcOpenPrinter](#Section_989357e2446e4872bb381dce21e1313f), and [RpcOpenPrinterEx](#Section_0e81ce1872b146c38584a205393b04ff).

##### RECTL

The RECTL structure defines a rectangle on a form, with two (x,y) coordinates in 1/1000 millimeter units.

1. typedef struct {
2. long left;
3. long top;
4. long right;
5. long bottom;
6. } RECTL;

**left:**  The value of this member MUST specify the x-coordinate of the upper-left corner of the rectangle relative to the left edge of the form. This value MUST be an integer greater than or equal to 0 and it MUST be smaller than or equal to the 'right'.

**top:**  The value of this member MUST specify the y-coordinate of the upper-left corner of the rectangle relative to the top edge of the form. This value MUST be an integer greater than or equal to 0 and it MUST be smaller than or equal to the 'bottom'.

**right:**  The value of this member MUST specify the x-coordinate of the lower-right corner of the rectangle relative to the left edge of the form. This value MUST be greater than or equal to 'left'.

**bottom:**  The value of this member MUST specify the y-coordinate of the lower-right corner of the rectangle relative to the top edge of the form. This value MUST be greater than or equal to 'top'.

##### SIZE

The SIZE structure defines the area of a form, with a width and height in thousandth-of-a-millimeter units.

1. typedef struct {
2. long cx;
3. long cy;
4. } SIZE;

**cx:**  The value of this member MUST specify the width, and it MUST be an integer greater than or equal to 0.

**cy:**  The value of this member MUST specify the height, and it MUST be an integer greater than or equal to 0.

##### STRING\_HANDLE

The STRING\_HANDLE serves as an RPC binding handle for methods that do not specify a [PRINTER\_HANDLE](#Section_10bfa8c2d4564ad0912de674766a36c8) parameter. RPC binding handles are specified in [[C706]](http://go.microsoft.com/fwlink/?LinkId=89824).

This type is declared as follows:

1. typedef [handle] wchar\_t\* STRING\_HANDLE;

To build the binding handle for those methods, RPC requires an RPC protocol sequence, a network address, and an endpoint. Both the RPC protocol sequence and the endpoint are bound to the RPC interface; they MUST be named pipes and **\pipe\spoolss**, respectively. The network address MUST be defined by the printer or print server name. The printer name can be in the form **\\server\printer** (for rules governing printer names, see section [2.2.4.14](#Section_24fcd124035c4988a8583a7d8d6f7b43)), and the server MUST be used as the network address.

#### Containers

##### DEVMODE\_CONTAINER

The DEVMODE\_CONTAINER structure specifies a [\_DEVMODE](#Section_ccc2a501794e4d2bb312f69c75131c2e) structure (section 2.2.2.1), which contains data for the initialization of a print device by a printer driver.

1. typedef struct \_DEVMODE\_CONTAINER {
2. DWORD cbBuf;
3. [size\_is(cbBuf), unique] BYTE\* pDevMode;
4. } DEVMODE\_CONTAINER;

**cbBuf:**  The size, in bytes, of the buffer pointed to by the **pDevMode** member.

**pDevMode:**  An optional pointer to a variable-length, custom-marshaled \_DEVMODE structure. The NULL value MUST be used to indicate that the default initialization data for the printer driver SHOULD be used.

##### DOC\_INFO\_CONTAINER

The DOC\_INFO\_CONTAINER structure provides information about the document to be printed, using the [DOC\_INFO\_1](#Section_20f7b510e6a04f6c97a05630d1e70879) structure.

1. typedef struct \_DOC\_INFO\_CONTAINER {
2. DWORD Level;
3. [switch\_is(Level)] union {
4. [case(1)]
5. DOC\_INFO\_1\* pDocInfo1;
6. } DocInfo;
7. } DOC\_INFO\_CONTAINER;

**Level:**  This member specifies the information level of the **DocInfo** member data. The value of this member MUST be set to 0x00000001.

**DocInfo:**  This member MUST define document properties, using an information structure that MUST correspond to the value of the **Level** member.

**pDocInfo1:**  This member MUST be a non-NULL pointer to a DOC\_INFO\_1 structure that describes the document that will be printed. Details are specified in section 2.2.1.4.

##### DRIVER\_CONTAINER

The DRIVER\_CONTAINER structure provides information about printer drivers by using **DRIVER\_INFO** structures (section [2.2.1.5](#Section_17943579a5144d0b8ded422265b23f45)). The **DriverInfo** member specifies the structure that defines the properties of a printer driver.

1. typedef struct \_DRIVER\_CONTAINER {
2. DWORD Level;
3. [switch\_is(Level)] union {
4. [case(1)]
5. DRIVER\_INFO\_1\* pNotUsed;
6. [case(2)]
7. DRIVER\_INFO\_2\* Level2;
8. [case(3)]
9. RPC\_DRIVER\_INFO\_3\* Level3;
10. [case(4)]
11. RPC\_DRIVER\_INFO\_4\* Level4;
12. [case(6)]
13. RPC\_DRIVER\_INFO\_6\* Level6;
14. [case(8)]
15. RPC\_DRIVER\_INFO\_8\* Level8;
16. } DriverInfo;
17. } DRIVER\_CONTAINER;

**Level:**  Specifies the information level of the **DriverInfo** data. The value of this member MUST be in the range 0x00000002 to 0x00000004 inclusive, 0x00000006, or 0x00000008.

**DriverInfo:**  Defines printer driver properties by using an information structure that corresponds to the value of the **Level** member.

**pNotUsed:**  A pointer to a structure that is specified only as a placeholder in the IDL and MUST be ignored.

**Level2:**  If the **Level** member is 0x00000002, this member is a pointer to a [DRIVER\_INFO\_2](#Section_39bbfc3087684cd49930434857e2c2a2) structure that provides printer driver information. For details, see section 2.2.1.5.2.

**Level3:**  If the **Level** member is 0x00000003, this member is a pointer to an [RPC\_DRIVER\_INFO\_3](#Section_b4d3b5f3c6314fc393484c983590a0d8) structure that provides printer driver information. For details, see section 2.2.1.5.3.

**Level4:**  If the **Level** member is 0x00000004, this member is a pointer to an [RPC\_DRIVER\_INFO\_4](#Section_459d24e40b754fc7b246fe40a1981eed) structure that provides printer driver information. For details, see section 2.2.1.5.4.

**Level6:**  If the **Level** member is 0x00000006, this member is a pointer to an [RPC\_DRIVER\_INFO\_6](#Section_ae12474015ef482c824e56e5436ed44b) structure that provides printer driver information. For details, see section 2.2.1.5.5.

**Level8:**  If the **Level** member is 0x00000008, this member is a pointer to an [RPC\_DRIVER\_INFO\_8](#Section_937d5fc867f145f8ab878b56a8fca558) structure that provides printer driver information. For details, see section 2.2.1.5.6.

##### FORM\_CONTAINER

The FORM\_CONTAINER structure provides information about printer forms, using [FORM\_INFO](#Section_2fef1fa4518f4e958572c6491442cf2a) structures. The **FormInfo** member specifies the structure that defines the printer form properties.

1. typedef struct \_FORM\_CONTAINER {
2. DWORD Level;
3. [switch\_is(Level)] union {
4. [case(1)]
5. FORM\_INFO\_1\* pFormInfo1;
6. [case(2)]
7. RPC\_FORM\_INFO\_2\* pFormInfo2;
8. } FormInfo;
9. } FORM\_CONTAINER;

**Level:**  This member MUST specify the information level of the **FormInfo** data. The value of this member MUST be 0x00000001 or 0x00000002.

**FormInfo:**  This member MUST define printer form properties, using an information structure that MUST correspond to the value of the **Level** member.

**pFormInfo1:**  If the **Level** member is 0x00000001, this member MUST be a non-NULL pointer to a [FORM\_INFO\_1](#Section_7821930259dd4d3f82d73a2e95466c28) structure, which provides information about a printer form. For details, see section 2.2.1.6.1.

**pFormInfo2:**  If the **Level** member is 0x00000002, this member MUST be a non-NULL pointer to a [RPC\_FORM\_INFO\_2](#Section_dca49b4f3eb146569e972f0b0c569bf6) structure, which provides information about a printer form. For details, see section 2.2.1.6.2.

##### JOB\_CONTAINER

The JOB\_CONTAINER structure provides information about print jobs, using [JOB\_INFO](#Section_87ea3e7722fb4914b803ef3b2619195a) structures. The **JobInfo** member specifies the structure that defines the print job properties.

1. typedef struct \_JOB\_CONTAINER {
2. DWORD Level;
3. [switch\_is(Level)] union {
4. [case(1)]
5. JOB\_INFO\_1\* Level1;
6. [case(2)]
7. JOB\_INFO\_2\* Level2;
8. [case(3)]
9. JOB\_INFO\_3\* Level3;
10. [case(4)]
11. JOB\_INFO\_4\* Level4;
12. } JobInfo;
13. } JOB\_CONTAINER;

**Level:**  Specifies the information level of the **JobInfo** data. The value of this member MUST be in the range 0x00000001 to 0x00000004 inclusive.

**JobInfo:**  Defines print job properties, using an information structure that corresponds to the value of the **Level** member.

**Level1:**  If the **Level** member is 0x00000001, this member is a pointer to a [JOB\_INFO\_1](#Section_794c1ba9ba0a4498ba9c2635563460dd) structure that provides print job information. For details, see section 2.2.1.7.1.

**Level2:**  If the **Level** member is 0x00000002, this member is a pointer to a [JOB\_INFO\_2](#Section_5a48c6c9fc5f42ffa5a7f111fee58d1a) structure that provides print job information. For details, see section 2.2.1.7.2.

**Level3:**  If the **Level** member is 0x00000003, this member is a pointer to a [JOB\_INFO\_3](#Section_1385aef9d3234e8d9d285f2f011bd39c) structure that provides print job information. For details, see section 2.2.1.7.3.

**Level4:**  If the **Level** member is 0x00000004, this member is a pointer to a [JOB\_INFO\_4](#Section_99086f6e3db14034ac6ccd4a5a77f22c) structure that provides print job information. For details, see section 2.2.1.7.4.

##### MONITOR\_CONTAINER

The MONITOR\_CONTAINER structure provides information about [**port monitors**](#gt_1b5f368b-942b-40dd-84ea-fa9082916216), using [MONITOR\_INFO](#Section_083e6e86c87a49a0a67975827d993511) structures. The **MonitorInfo** member specifies the structure that defines the port monitor properties.

1. typedef struct \_MONITOR\_CONTAINER {
2. DWORD Level;
3. [switch\_is(Level)] union {
4. [case(1)]
5. MONITOR\_INFO\_1\* pMonitorInfo1;
6. [case(2)]
7. MONITOR\_INFO\_2\* pMonitorInfo2;
8. } MonitorInfo;
9. } MONITOR\_CONTAINER;

**Level:**  Specifies the information level of the **MonitorInfo** data. The value of this member MUST be 0x00000001 or 0x00000002.

**MonitorInfo:**  Defines port monitor properties, using an information structure that corresponds to the value of the **Level** member.

**pMonitorInfo1:**  If the **Level** member is 0x00000001, this member is a pointer to a [MONITOR\_INFO\_1](#Section_7ee6a75f988941a68b1490d4e378b956) structure that provides information about a port monitor. For details, see section 2.2.1.8.1.

**pMonitorInfo2:**  If the **Level** member is 0x00000002, this member is a pointer to a [MONITOR\_INFO\_2](#Section_f05b335b203c454f87da69ad96930bd7) structure that provides information about a port monitor. For details, see section 2.2.1.8.2.

##### PORT\_CONTAINER

The PORT\_CONTAINER structure provides information about printer ports, using [PORT\_INFO](#Section_05c1a8d20bcb4292bc65c8cc2e7f90cf) structures.[<4>](#Appendix_A_4) The **PortInfo** member specifies the structure that defines the port properties.

1. typedef struct \_PORT\_CONTAINER {
2. DWORD Level;
3. [switch\_is(0x00FFFFFF & Level)]
4. union {
5. [case(1)]
6. PORT\_INFO\_1\* pPortInfo1;
7. [case(2)]
8. PORT\_INFO\_2\* pPortInfo2;
9. [case(3)]
10. PORT\_INFO\_3\* pPortInfo3;
11. [case(0x00FFFFFF)]
12. PORT\_INFO\_FF\* pPortInfoFF;
13. } PortInfo;
14. } PORT\_CONTAINER;

**Level:**  Specifies the information level of the **PortInfo** data. The value of this member MUST be in the range 0x00000001 to 0x00000003 inclusive, or 0xFFFFFFFF.

**PortInfo:**  Defines port properties, using an information structure that corresponds to the value of the **Level** member.

**Note**  Despite the bitwise AND of **Level** with 0x00FFFFFF, no values for **Level** are valid besides those specified.

**pPortInfo1:**  If the **Level** member is 0x00000001, this member is a pointer to a [PORT\_INFO\_1](#Section_9bab3fb5b36948f28f083082ca13e739) structure that provides information about the printer port. For details, see section 2.2.1.9.1.

**pPortInfo2:**  If the **Level** member is 0x00000002, this member is a pointer to a [PORT\_INFO\_2](#Section_7a923364dffc40b3bca5bf2eed610f31) structure that provides information about the printer port. For details, see section 2.2.1.9.2.

**pPortInfo3:**  If the **Level** member is 0x00000003, this member is a pointer to a [PORT\_INFO\_3](#Section_411e442325e44f6184b745336e40094e) structure that provides information about the printer port. For details, see section 2.2.1.9.3.

**pPortInfoFF:**  If the **Level** member is 0xFFFFFFFF, this member is a pointer to a [PORT\_INFO\_FF](#Section_6366af02dc304987a1a3f0ec4176ff94) structure that provides information about the printer port. For details, see section 2.2.1.9.4.

##### PORT\_VAR\_CONTAINER

The PORT\_VAR\_CONTAINER structure provides information for supported printer port monitors.[<5>](#Appendix_A_5)

1. typedef struct \_PORT\_VAR\_CONTAINER {
2. DWORD cbMonitorData;
3. [size\_is(cbMonitorData), unique, disable\_consistency\_check]
4. BYTE\* pMonitorData;
5. } PORT\_VAR\_CONTAINER;

**cbMonitorData:**  The size, in bytes, of the buffer that is pointed to by the **pMonitorData** member.

**pMonitorData:**  An optional pointer to a block of data that is passed to the port monitor.

##### PRINTER\_CONTAINER

The PRINTER\_CONTAINER structure provides information about printer properties and state information, using [PRINTER\_INFO](#Section_125fb29b3c5141fca333b1e0e8794a91) structures. The **PrinterInfo** member specifies the structure that defines the printer properties.

1. typedef struct \_PRINTER\_CONTAINER {
2. DWORD Level;
3. [switch\_is(Level)] union {
4. [case(0)]
5. PRINTER\_INFO\_STRESS\* pPrinterInfoStress;
6. [case(1)]
7. PRINTER\_INFO\_1\* pPrinterInfo1;
8. [case(2)]
9. PRINTER\_INFO\_2\* pPrinterInfo2;
10. [case(3)]
11. PRINTER\_INFO\_3\* pPrinterInfo3;
12. [case(4)]
13. PRINTER\_INFO\_4\* pPrinterInfo4;
14. [case(5)]
15. PRINTER\_INFO\_5\* pPrinterInfo5;
16. [case(6)]
17. PRINTER\_INFO\_6\* pPrinterInfo6;
18. [case(7)]
19. PRINTER\_INFO\_7\* pPrinterInfo7;
20. [case(8)]
21. PRINTER\_INFO\_8\* pPrinterInfo8;
22. [case(9)]
23. PRINTER\_INFO\_9\* pPrinterInfo9;
24. } PrinterInfo;
25. } PRINTER\_CONTAINER;

**Level:**  Specifies the information level of the **PrinterInfo** data. The value of this member MUST be in the range 0x00000000 to 0x00000009 inclusive.

**PrinterInfo:**  Provides printer information using a container structure that corresponds to the value specified by the **Level** member.

**pPrinterInfoStress:**  If the **Level** member is 0x00000000, this member is a pointer to a [PRINTER\_INFO\_STRESS](#Section_9e94b8f4a84b4c96b9ae4b4acb3469b8) structure (section 2.2.1.10.1), which provides diagnostic printer information.

**pPrinterInfo1:**  If the **Level** member is 0x00000001, this member is a pointer to a [PRINTER\_INFO\_1 (section 2.2.1.10.2)](#Section_0d1a296b7027441c83938aa5c3764b27) structure, which provides printer information.

**pPrinterInfo2:**  If the **Level** member is 0x00000002, this member is a pointer to a [PRINTER\_INFO\_2 (section 2.2.1.10.3)](#Section_d40734e1ac9b4b2a862883a1bc7bbeed) structure, which provides detailed printer information.

**pPrinterInfo3:**  If the **Level** member is 0x00000003, this member is a pointer to a [PRINTER\_INFO\_3 (section 2.2.1.10.4)](#Section_00af009b231e4ae6bfd7cb8476c32c19) structure, which provides printer security information.

**pPrinterInfo4:**  If the **Level** member is 0x00000004, this member is a pointer to a [PRINTER\_INFO\_4 (section 2.2.1.10.5)](#Section_ac391864cd9945099ff3ea0366dd9b2b) structure, which provides a subset of the printer information.

**pPrinterInfo5:**  If the **Level** member is 0x00000005, this member is a pointer to a [PRINTER\_INFO\_5 (section 2.2.1.10.6)](#Section_d7e5feaa16be4dbdb6d922f62f69afbb) structure, which provides information about the printer attributes.

**pPrinterInfo6:**  If the **Level** member is 0x00000006, this member is a pointer to a [PRINTER\_INFO\_6 (section 2.2.1.10.7)](#Section_436ef38839cd499295ac02ddd2a51193) structure, which provides information about the status of the printer.

**pPrinterInfo7:**  If the **Level** member is 0x00000007, this member is a pointer to a [PRINTER\_INFO\_7 (section 2.2.1.10.8)](#Section_91a6cb808df743d08a5d8f1d23f96de6) structure, which provides [**directory service (DS)**](#gt_c36db657-3138-4d9a-9289-ded5cbb8b40e) information.

**pPrinterInfo8:**  If the **Level** member is 0x00000008, this member is a pointer to a [PRINTER\_INFO\_8 (section 2.2.1.10.9)](#Section_abe75b29b0924398a5af9d639f430cde) structure, which provides information about the global printer driver settings for a printer.

**pPrinterInfo9:**  If the **Level** member is 0x00000009, this member is a pointer to a [PRINTER\_INFO\_9 (section 2.2.1.10.10)](#Section_6a5df26640c94661966f27caa209b1fd) structure. The PRINTER\_INFO\_9 structure is not used remotely, but it is included in this structure to yield a compatible IDL file. The print server MUST respond with ERROR\_NOT\_SUPPORTED if it receives a PRINTER\_CONTAINER structure with a **Level** value equal to 0x00000009.

##### RPC\_BIDI\_REQUEST\_CONTAINER

The RPC\_BIDI\_REQUEST\_CONTAINER structure is a container for a list of [**bidirectional**](#gt_604aaee0-7ad4-46a3-932f-8d2def6615ad) requests.[<6>](#Appendix_A_6)

1. typedef struct \_RPC\_BIDI\_REQUEST\_CONTAINER {
2. DWORD Version;
3. DWORD Flags;
4. DWORD Count;
5. [size\_is(Count), unique] RPC\_BIDI\_REQUEST\_DATA aData[];
6. } RPC\_BIDI\_REQUEST\_CONTAINER;

**Version:**  The version of the bidirectional API schema. The value of this member MUST be 0x00000001.

**Flags:**  A value that MUST be set to zero when sent and MUST be ignored on receipt.

**Count:**  The number of bidirectional requests in the **aData** member.

**aData:**  An array of [RPC\_BIDI\_REQUEST\_DATA](#Section_c3b78db75fec440584ff412e82d491f1) structures. Each structure in this member contains a single bidirectional request. For details, see section 2.2.1.12.1.

##### RPC\_BIDI\_RESPONSE\_CONTAINER

The RPC\_BIDI\_RESPONSE\_CONTAINER structure is a container for a list of bidirectional responses.[<7>](#Appendix_A_7)

1. typedef struct \_RPC\_BIDI\_RESPONSE\_CONTAINER {
2. DWORD Version;
3. DWORD Flags;
4. DWORD Count;
5. [size\_is(Count), unique] RPC\_BIDI\_RESPONSE\_DATA aData[];
6. } RPC\_BIDI\_RESPONSE\_CONTAINER;

**Version:**  This member MUST contain the value that specifies the version of the bidirectional API [**schema**](#gt_fd49ea36-576c-4417-93bd-d1ac63e71093). The value of this member MUST be 0x00000001.

**Flags:**  This member is a set of flags that are reserved for system use. The value of this member MUST be set to zero when sent and MUST be ignored on receipt.

**Count:**  This member MUST specify the number of bidirectional responses in the **aData** member.

**aData:**  This member is an array of [RPC\_BIDI\_RESPONSE\_DATA](#Section_f42d17bd05d643a4872e6e6513ee0922) structures. Each structure in this member MUST contain a single bidirectional response. For more information, see section 2.2.1.12.2.

##### RPC\_BINARY\_CONTAINER

The RPC\_BINARY\_CONTAINER structure is a container for binary printer data and is used in the [RPC\_BIDI\_DATA (section 2.2.1.12.3)](#Section_d6e679dc5a244936ac6858251dbd1555) structure.[<8>](#Appendix_A_8)

1. typedef struct \_RPC\_BINARY\_CONTAINER {
2. DWORD cbBuf;
3. [size\_is(cbBuf), unique] BYTE\* pszString;
4. } RPC\_BINARY\_CONTAINER;

**cbBuf:**  This member MUST specify the size, in bytes, of the buffer that is pointed to by the **pszString** member.

**pszString:**  This member MUST be a non-NULL pointer to an array of bytes that contain binary printer data.

##### SECURITY\_CONTAINER

The SECURITY\_CONTAINER structure specifies a SECURITY\_DESCRIPTOR structure ([[MS-DTYP]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-DTYP%5d.pdf) section 2.4.6), which contains security information.

1. typedef struct SECURITY\_CONTAINER {
2. DWORD cbBuf;
3. [size\_is(cbBuf), unique] BYTE\* pSecurity;
4. } SECURITY\_CONTAINER;

**cbBuf:**  The size, in bytes, of the buffer that is pointed to by the **pSecurity** member.

**pSecurity:**  An optional pointer to a self-relative SECURITY\_DESCRIPTOR structure.

##### SPLCLIENT\_CONTAINER

The SPLCLIENT\_CONTAINER structure contains an information structure that provides data about the connecting client.[<9>](#Appendix_A_9)

1. typedef struct \_SPLCLIENT\_CONTAINER {
2. DWORD Level;
3. [switch\_is(Level)] union {
4. [case(1)]
5. SPLCLIENT\_INFO\_1\* pClientInfo1;
6. [case(2)]
7. SPLCLIENT\_INFO\_2\* pNotUsed1;
8. [case(3)]
9. SPLCLIENT\_INFO\_3\* pNotUsed2;
10. } ClientInfo;
11. } SPLCLIENT\_CONTAINER;

**Level:**  The information level that is used by the **ClientInfo** member to determine the information structure. The value MUST be 0x00000001.

**ClientInfo:**  Client information in a structure that corresponds to the information level specified by the **Level** member.

**pClientInfo1:**  A pointer to an [SPLCLIENT\_INFO\_1 (section 2.2.1.11.1)](#Section_dd99a589aecd4c259111ded01b10f347) information structure.

**pNotUsed1:**  A pointer to a structure that is specified only as a placeholder in the IDL and MUST be ignored.

**pNotUsed2:**  A pointer to a structure that is specified only as a placeholder in the IDL and MUST be ignored.

##### STRING\_CONTAINER

The STRING\_CONTAINER structure contains a string.[<10>](#Appendix_A_10)

1. typedef struct \_STRING\_CONTAINER {
2. DWORD cbBuf;
3. [size\_is(cbBuf/2), unique] WCHAR\* pszString;
4. } STRING\_CONTAINER;

**cbBuf:**  This member MUST specify the size, in bytes, of the buffer that is pointed to by the **pszString** member. The value of this number MUST be an even number.

**pszString:**  This member MUST be a non-NULL pointer to a string. The string that is referenced by this member MUST NOT be empty.

##### SYSTEMTIME\_CONTAINER

The SYSTEMTIME\_CONTAINER structure is a container for a SYSTEMTIME structure that specifies a date and time using individual members for the month, day, year, weekday, hour, minute, second, and millisecond.[<11>](#Appendix_A_11)

1. typedef struct \_SYSTEMTIME\_CONTAINER {
2. DWORD cbBuf;
3. SYSTEMTIME\* pSystemTime;
4. } SYSTEMTIME\_CONTAINER;

**cbBuf:**  This member MUST specify the size, in bytes, of the buffer that is pointed to by the **pSystemTime** member.

**pSystemTime:**  This member MUST be a non-NULL pointer to a SYSTEMTIME structure.

##### RPC\_BranchOfficeJobDataContainer

The RPC\_BranchOfficeJobDataContainer structure is a container for an array of [RPC\_BranchOfficeJobData (section 2.2.1.15.2)](#Section_6e442be62c54497593c685895289ba0d) structures.[<12>](#Appendix_A_12)

1. typedef struct {
2. DWORD cJobDataEntries;
3. [size\_is(cJobDataEntries), unique]
4. RPC\_BranchOfficeJobData JobData[];
5. } RPC\_BranchOfficeJobDataContainer;

**cJobDataEntries:**  The number of RPC\_BranchOfficeJobData structures in the **JobData** member.

**JobData:**  An array of RPC\_BranchOfficeJobData structures. Each structure in the array contains a single **Branch Office Print Remote Log Entry** (section [3.1.1](#Section_1bd61448db4b4284a4b18346732402d4)).

#### Members in INFO Structures

This section specifies members that are commonly used in a consistent fashion in IDL-marshaled and custom-marshaled [INFO](#Section_f29e9057df2940d49c51c615c0da62a1) structures. The individual INFO sections provide definitions only for the following:

* Members that are not defined in this section.
* Members that are not defined in corresponding INFO subsections within this section.
* Members whose definitions in their corresponding INFO structures differ from their definitions in this section and subsections.

The type of each member is specified in its corresponding INFO structure section.

**pPrinterName**: This member MUST be a non-NULL pointer to a string that MUST specify the name of a printer. For rules governing printer names, see section [2.2.4.14](#Section_24fcd124035c4988a8583a7d8d6f7b43).

**pServerName**: This member MUST be a non-NULL pointer to a string that MUST specify the name of the server that hosts the printer. For rules governing server names, see section [2.2.4.16](#Section_9b3f813570224b72accbaefcc360c83b).

**Reserved**: This member is reserved for future use. The value of this member SHOULD be set to zero when sent and MUST be ignored on receipt.

**dwReserved2**: This member is reserved for future use. The value of this member SHOULD be set to zero when sent and MUST be ignored on receipt.

**dwReserved3**: This member is reserved for future use. The value of this member SHOULD be set to zero when sent and MUST be ignored on receipt.

##### DRIVER\_INFO and RPC\_DRIVER\_INFO Members

This section describes members commonly used in [DRIVER\_INFO (section 2.2.1.5)](#Section_17943579a5144d0b8ded422265b23f45) and RPC\_DRIVER\_INFO (section 2.2.1.3.1) structures.

**pName**: A pointer to a string that specifies the name of the printer driver; for example, "QMS 810". For rules governing printer driver names, see section [2.2.4.3](#Section_a4c7686fa88c48f996ebbf4dfc3fe519).

**cVersion**: An implementation-specific value that identifies the driver version and the operating system version for which the printer driver was written. The driver version contained by each printer driver object in the "List of Printer Drivers" described in section [3.1.1](#Section_1bd61448db4b4284a4b18346732402d4).[<13>](#Appendix_A_13)

**pEnvironment**: A pointer to a string that specifies the environment that the printer driver supports. For rules governing environment names, see section [2.2.4.4](#Section_ad7758938a194786ab4634731f52d1af).

**pDriverPath**: A pointer to a string that specifies a file name or full path and file name for the file that contains the printer driver. For further information on driver files, see [[MSDN-MPD]](http://go.microsoft.com/fwlink/?LinkId=91840). For rules governing path names, see section [2.2.4.9](#Section_1d75f622d07c45ccbf1ea6191e408516).

**pDataFile**: A pointer to a string that specifies a file name or a full path and file name for the file that contains printer driver data. For further information on driver files, see [MSDN-MPD]. For rules governing path names, see section 2.2.4.9.

**pConfigFile**: A pointer to a string that specifies a file name or a full path and file name for the printer driver configuration module. For further information on driver files, see [MSDN-MPD]. For rules governing path names, see section 2.2.4.9.

**pHelpFile**: An optional pointer to a string that specifies a file name or a full path and file name for the printer driver help file. For further information on driver files, see [MSDN-MPD]. For rules governing path names, see section 2.2.4.9.

**pMonitorName**: An optional pointer to a string that specifies a [**language monitor**](#gt_28b48680-595a-405b-a857-7c2dc95ee7b4). For rules governing monitor names, see section [2.2.4.8](#Section_808e2e6c7f914bdd92da56fbca513404).[<14>](#Appendix_A_14)

**pDefaultDataType**: An optional pointer to a string that specifies the default data type of print jobs created with this driver (for example, [**enhanced metafile spool format (EMFSPOOL)**](#gt_3bd355a2-4faa-40ab-bc79-0d158eba43b6) or RAW Format). For rules governing data type names, see section [2.2.4.2](#Section_f40cbad088dd4e86a16bcf62c3b17c52).

**cchDependentFiles**: The number of characters in the multisz pointed to by **pDependentFiles**.

**pDependentFiles**: An optional pointer to a multisz that MUST specify the names of the files that the printer driver is dependent on. If specified, this list MUST include at least one file name and SHOULD be ordered as follows:

* The file name of the [**printer driver manifest**](#gt_566759dc-6e10-4b36-b710-f4296c1fbc83), if present.
* If the printer driver is a [**derived printer driver**](#gt_3101a6be-4528-4314-afe4-ea9d26a106eb), the names of all the files the derived printer driver depends on. If the printer driver is not a derived printer driver, all of the other files the printer driver depends on.
* If the printer driver is a derived printer driver, the file name of the printer driver manifest of the corresponding [**class printer driver**](#gt_f4048a04-2516-44fc-ba42-2b5bee2782d5).
* If the printer driver is a derived printer driver, the names of all of the files the corresponding class printer driver depends on.[<15>](#Appendix_A_15)

**cchPreviousNames**: The value of this member MUST be the number of characters in the multisz pointed to by **pszzPreviousNames**.

**pszzPreviousNames**: An optional pointer to a multisz that MUST specify any previous printer drivers that are compatible with this driver.

**dwlDriverVersion**: The printer driver version number. The format of this number is specified by each printer driver manufacturer. A print client MAY use this value to determine whether a printer driver on the print server matches the version available on the client.[<16>](#Appendix_A_16)

**ftDriverDate**: The value of this member MUST be the manufacturer build date of the printer driver. The FILETIME format is specified in [[MS-DTYP]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-DTYP%5d.pdf) section 2.3.3.

**pMfgName**: An optional pointer to a string that specifies the manufacturer's name.

**pOEMUrl**: An optional pointer to a string that specifies the URL for the manufacturer of the printer driver.

**pHardwareID**: An optional pointer to a string that specifies the hardware identifier for the printer driver.

**pProvider**: An optional pointer to a string that specifies the publisher of the printer driver.

##### FORM\_INFO and RPC\_FORM\_INFO Members

This section describes the members that are commonly used in [FORM\_INFO](#Section_2fef1fa4518f4e958572c6491442cf2a) and RPC\_FORM\_INFO structures.

**Flags**: This member MUST specify the form property. The value of this member MUST be a value from the following table.

| Name/Value | Meaning |
| --- | --- |
| FORM\_USER  0x00000000 | If the value of this member is **FORM\_USER**, the form has been defined by the user. Forms that have this flag set are defined in the registry. |
| FORM\_BUILTIN  0x00000001 | If the value of this member is **FORM\_BUILTIN**, the form is part of the spooler. Form definitions that have this flag set do not appear in the registry. |
| FORM\_PRINTER  0x00000002 | If the value of this member is **FORM\_PRINTER**, the form is associated with a particular printer and its definition appears in the registry. |

**pName**: This member MUST be a non-NULL pointer to a string that MUST specify the form name. For rules governing form names, see section [2.2.4.5](#Section_cd213afb1b674479ac15316ea0fb537e).

**Size**: The value of this member MUST specify the form's width and height in thousandths of millimeters using a [SIZE](#Section_c76b61d3166543f59584a40d42bf05c5) structure.

**ImageableArea**: This member MUST specify the part of the form that the printer can print on as a rectangle in thousandths of millimeters using a [RECTL](#Section_d81607da4aa744b0ae4407d0c2227b34) structure.

##### JOB\_INFO Members

This section describes members commonly used in [JOB\_INFO](#Section_87ea3e7722fb4914b803ef3b2619195a) structures.

**pMachineName**: This member is a pointer to a string that specifies the name of a server that hosts a printer. For rules governing server names, see section [2.2.4.16](#Section_9b3f813570224b72accbaefcc360c83b).

**pUserName**: This member is an optional pointer to a string that specifies the name of a user that owns a print job. For rules governing user names, see section [2.2.4.17](#Section_9b19bbdf64da407fa5c6c6533a041d19).

**pNotifyName**: This member is an optional pointer to a string that specifies the name of a user to be notified when a job is complete or when an error occurs while printing a job. For rules governing user names, see section 2.2.4.17.

**pDocument**: This member is an optional pointer to a string that specifies the name of a print job.

**pDatatype**: This member is a pointer to a string that specifies the type of data that a printing application sends to a printer in a print job. The identified data type MUST be supported by the print processor that is associated with the printer that is processing the job. For rules governing data type names, see section [2.2.4.2](#Section_f40cbad088dd4e86a16bcf62c3b17c52).

**pPrintProcessor**: This member is a pointer to a string that specifies the name of a print processor that is used to print a job. For rules governing print processor names, see section [2.2.4.11](#Section_807e0bdd21a54a729a6a1a7fb1b08a8e).

**pParameters**: This member is an optional pointer to a string that specifies default print processor parameters.

**pDriverName**: This member is an optional pointer to a string that specifies the name of a printer driver to process a print job. For rules governing printer driver names, see section [2.2.4.3](#Section_a4c7686fa88c48f996ebbf4dfc3fe519).

**pDevMode**: This member is an optional pointer to a truncated [DEVMODE](#Section_0136fad4f02c44dd857f820477b19c1d) structure (section 2.2.1.1.1), and MUST be ignored on receipt. Actual **DEVMODE** data is passed to a method via a custom-marshaled [\_DEVMODE](#Section_ccc2a501794e4d2bb312f69c75131c2e) structure (section 2.2.2.1) in a [DEVMODE\_CONTAINER](#Section_c2b14fe764794460a7ba633a845dd61a) (section 2.2.1.2.1).

**pSecurityDescriptor**: This member is an optional pointer to a SECURITY\_DESCRIPTOR structure ([[MS-DTYP]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-DTYP%5d.pdf) section 2.4.6), and MUST be ignored on receipt. Actual **SECURITY\_DESCRIPTOR** data is passed to a method via a self-relative SECURITY\_DESCRIPTOR structure in a [SECURITY\_CONTAINER](#Section_6cf2b4ff1b2b453c87f985243823cea8) (section 2.2.1.2.13).

**JobId**: This member contains an identifier for a print job.

**pStatus**: This member is an optional pointer to a string that describes job status. The text is implementation-specific and can be displayed to the user, but it MUST NOT have any other functional effect. An example of job status is "Cannot print - Black ink must be replaced."

**Status**: This member specifies job status. The value of this member is the result of a bitwise OR of zero or more of the job status values defined in section [2.2.3.12](#Section_1625e9d929e448f4b83d3bd0fdaea787).

Client applications can display the job status to a user. It is an implementation-specific string and SHOULD support all job status descriptions specified in section 2.2.3.12 for all corresponding status bits. If **pStatus** is not NULL, the string that is pointed to by **pStatus** SHOULD be displayed instead.

**Priority**: This member specifies information about job priority. The value of this member MUST be a decimal number from 0 through 99, inclusive.

**Position**: This member specifies a job's position in a queue, where one represents the next job that will be printed.

**TotalPages**: This member specifies the number of pages a document contains. It can be zero.

**PagesPrinted**: This member specifies the number of pages that have been printed. It can be zero.

**Submitted**: This member is a SYSTEMTIME structure ([MS-DTYP] section 2.3.13) that specifies when a document was spooled.

**StartTime**: This member specifies the earliest time that a printer can print a job. The time is expressed as the number of minutes after 12:00 AM GMT within a 24-hour boundary.

**UntilTime**: This member specifies the latest time that the printer can print a job. The time is expressed as the number of minutes after 12:00 AM GMT within a 24-hour boundary.

**Size**: This member specifies the size of a job, in bytes.

**Time**: This member specifies the number of milliseconds that have elapsed since printing began.

##### MONITOR\_INFO Members

This section describes the members that are commonly used in [MONITOR\_INFO](#Section_822a6b4d37814a579eefd572dcf4d86f) structures.

**pName**: This member MUST be a non-NULL pointer to a string that MUST specify the name of the port monitor. For rules governing port monitor names, see section [2.2.4.8](#Section_808e2e6c7f914bdd92da56fbca513404).

##### PORT\_INFO Members

This section describes members commonly used in [PORT\_INFO](#Section_05c1a8d20bcb4292bc65c8cc2e7f90cf) structures.

**pPortName**: A pointer to a string that specifies a supported printer port. For rules governing port names, see section [2.2.4.10](#Section_03c5feb74fbb4569b0649eb6ad833f50).

##### PRINTER\_INFO Members

This section describes members commonly used in [PRINTER\_INFO](#Section_125fb29b3c5141fca333b1e0e8794a91) structures.

**pDescription**: This member is an optional pointer to a string that specifies a description of the printer.[<17>](#Appendix_A_17)

**pComment**: This member is an optional pointer to a string that MUST specify additional information about the printer.[<18>](#Appendix_A_18)

**Status**: This member specifies the printer status. It is the result of a bitwise OR of zero or more printer status values (section [2.2.3.12](#Section_1625e9d929e448f4b83d3bd0fdaea787)).

**Attributes**: This member specifies printer attributes. It is the result of a bitwise OR of zero or more printer attribute values (section 2.2.3.12).

**pDevMode**: This member is an optional pointer to a truncated [DEVMODE](#Section_0136fad4f02c44dd857f820477b19c1d) structure (section 2.2.1.1.1), and MUST be ignored on receipt. Actual **DEVMODE** data is passed to a method via a custom-marshaled [\_DEVMODE](#Section_ccc2a501794e4d2bb312f69c75131c2e) structure (section 2.2.2.1) in a [DEVMODE\_CONTAINER](#Section_c2b14fe764794460a7ba633a845dd61a) (section 2.2.1.2.1).

**pSecurityDescriptor**: This member is an optional pointer to a SECURITY\_DESCRIPTOR structure ([[MS-DTYP]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-DTYP%5d.pdf) section 2.4.6), and MUST be ignored on receipt. Actual **SECURITY\_DESCRIPTOR** data is passed to a method via a self-relative SECURITY\_DESCRIPTOR structure in a [SECURITY\_CONTAINER](#Section_6cf2b4ff1b2b453c87f985243823cea8) (section 2.2.1.2.13).

**pPortName**: This member is a pointer to a string that specifies the port(s) used to transmit data to a printer. For rules governing port names, see section [2.2.4.10](#Section_03c5feb74fbb4569b0649eb6ad833f50).

##### SPLCLIENT\_INFO Members

This section describes members commonly used in [SPLCLIENT\_INFO](#Section_d23fee1a3f624cbb8a06d01180fe2ff9) structures.

**pMachineName**: This member is a pointer to a string that provides the client computer name. Client computer names are governed by the same rules as server names (section [2.2.4.16](#Section_9b3f813570224b72accbaefcc360c83b)).

**pUserName**: This member is a pointer to a string that provides a user name.

**dwBuildNum**: The value of this member specifies the build number of the client operating system.

**dwMajorVersion**: The value of this member is the implementation-specific major version number of the client operating system.[<19>](#Appendix_A_19)

**dwMinorVersion**: The value of this member is the implementation-specific minor version number of the client operating system.[<20>](#Appendix_A_20)

**wProcessorArchitecture**: The value of this member is the implementation-specific identifier for the client system's processor architecture.[<21>](#Appendix_A_21) The value of this member SHOULD be ignored on receipt.

#### DOC\_INFO\_1

The DOC\_INFO\_1 structure describes a document that will be printed.

1. typedef struct \_DOC\_INFO\_1 {
2. [string] wchar\_t\* pDocName;
3. [string] wchar\_t\* pOutputFile;
4. [string] wchar\_t\* pDatatype;
5. } DOC\_INFO\_1;

**pDocName:**  An optional pointer to a string that provides the name of the document. If this member is NULL, the print server SHOULD use an implementation-specific default job name.[<22>](#Appendix_A_22)

**pOutputFile:**  An optional pointer to a string that specifies the name of an output file. For rules governing path names, see section [2.2.4.9](#Section_1d75f622d07c45ccbf1ea6191e408516).

**pDatatype:**  An optional pointer to a string that identifies the type of data used to record the document. For rules governing data type names, see section [2.2.4.2](#Section_f40cbad088dd4e86a16bcf62c3b17c52).

#### DRIVER\_INFO

##### DRIVER\_INFO\_1

The DRIVER\_INFO\_1 structure provides information about a printer driver.

1. typedef struct \_DRIVER\_INFO\_1 {
2. [string] wchar\_t\* pName;
3. } DRIVER\_INFO\_1;

All members not defined in this section are specified in sections [2.2.1.3.1](#Section_4464eaf0f34f40d5b970736437a21913) and [2.2.1.3](#Section_964b517f67304b4a99edee9648e7f1f7).

##### DRIVER\_INFO\_2

The DRIVER\_INFO\_2 structure provides information about a printer driver.

1. typedef struct \_DRIVER\_INFO\_2 {
2. DWORD cVersion;
3. [string] wchar\_t\* pName;
4. [string] wchar\_t\* pEnvironment;
5. [string] wchar\_t\* pDriverPath;
6. [string] wchar\_t\* pDataFile;
7. [string] wchar\_t\* pConfigFile;
8. } DRIVER\_INFO\_2;

All members not defined in this section are specified in sections [2.2.1.3.1](#Section_4464eaf0f34f40d5b970736437a21913) and [2.2.1.3](#Section_964b517f67304b4a99edee9648e7f1f7).

##### RPC\_DRIVER\_INFO\_3

The RPC\_DRIVER\_INFO\_3 structure provides information about a printer driver.[<23>](#Appendix_A_23)

1. typedef struct \_RPC\_DRIVER\_INFO\_3 {
2. DWORD cVersion;
3. [string] wchar\_t\* pName;
4. [string] wchar\_t\* pEnvironment;
5. [string] wchar\_t\* pDriverPath;
6. [string] wchar\_t\* pDataFile;
7. [string] wchar\_t\* pConfigFile;
8. [string] wchar\_t\* pHelpFile;
9. [string] wchar\_t\* pMonitorName;
10. [string] wchar\_t\* pDefaultDataType;
11. DWORD cchDependentFiles;
12. [size\_is(cchDependentFiles), unique]
13. wchar\_t\* pDependentFiles;
14. } RPC\_DRIVER\_INFO\_3;

All members not defined in this section are specified in sections [2.2.1.3.1](#Section_4464eaf0f34f40d5b970736437a21913) and [2.2.1.3](#Section_964b517f67304b4a99edee9648e7f1f7).

##### RPC\_DRIVER\_INFO\_4

The RPC\_DRIVER\_INFO\_4 structure provides information about a printer driver.[<24>](#Appendix_A_24)

1. typedef struct \_RPC\_DRIVER\_INFO\_4 {
2. DWORD cVersion;
3. [string] wchar\_t\* pName;
4. [string] wchar\_t\* pEnvironment;
5. [string] wchar\_t\* pDriverPath;
6. [string] wchar\_t\* pDataFile;
7. [string] wchar\_t\* pConfigFile;
8. [string] wchar\_t\* pHelpFile;
9. [string] wchar\_t\* pMonitorName;
10. [string] wchar\_t\* pDefaultDataType;
11. DWORD cchDependentFiles;
12. [size\_is(cchDependentFiles), unique]
13. wchar\_t\* pDependentFiles;
14. DWORD cchPreviousNames;
15. [size\_is(cchPreviousNames), unique]
16. wchar\_t\* pszzPreviousNames;
17. } RPC\_DRIVER\_INFO\_4;

All members not defined in this section are specified in sections [2.2.1.3.1](#Section_4464eaf0f34f40d5b970736437a21913) and [2.2.1.3](#Section_964b517f67304b4a99edee9648e7f1f7).

##### RPC\_DRIVER\_INFO\_6

The RPC\_DRIVER\_INFO\_6 structure provides extended printer driver information.[<25>](#Appendix_A_25)

1. typedef struct \_RPC\_DRIVER\_INFO\_6 {
2. DWORD cVersion;
3. [string] wchar\_t\* pName;
4. [string] wchar\_t\* pEnvironment;
5. [string] wchar\_t\* pDriverPath;
6. [string] wchar\_t\* pDataFile;
7. [string] wchar\_t\* pConfigFile;
8. [string] wchar\_t\* pHelpFile;
9. [string] wchar\_t\* pMonitorName;
10. [string] wchar\_t\* pDefaultDataType;
11. DWORD cchDependentFiles;
12. [size\_is(cchDependentFiles), unique]
13. wchar\_t\* pDependentFiles;
14. DWORD cchPreviousNames;
15. [size\_is(cchPreviousNames), unique]
16. wchar\_t\* pszzPreviousNames;
17. FILETIME ftDriverDate;
18. DWORDLONG dwlDriverVersion;
19. [string] wchar\_t\* pMfgName;
20. [string] wchar\_t\* pOEMUrl;
21. [string] wchar\_t\* pHardwareID;
22. [string] wchar\_t\* pProvider;
23. } RPC\_DRIVER\_INFO\_6;

All members not defined in this section are specified in sections [2.2.1.3.1](#Section_4464eaf0f34f40d5b970736437a21913) and [2.2.1.3](#Section_964b517f67304b4a99edee9648e7f1f7).

##### RPC\_DRIVER\_INFO\_8

The RPC\_DRIVER\_INFO\_8 structure specifies extended printer driver information.[<26>](#Appendix_A_26)

1. typedef struct \_RPC\_DRIVER\_INFO\_8 {
2. DWORD cVersion;
3. [string] wchar\_t\* pName;
4. [string] wchar\_t\* pEnvironment;
5. [string] wchar\_t\* pDriverPath;
6. [string] wchar\_t\* pDataFile;
7. [string] wchar\_t\* pConfigFile;
8. [string] wchar\_t\* pHelpFile;
9. [string] wchar\_t\* pMonitorName;
10. [string] wchar\_t\* pDefaultDataType;
11. DWORD cchDependentFiles;
12. [size\_is(cchDependentFiles), unique]
13. wchar\_t\* pDependentFiles;
14. DWORD cchPreviousNames;
15. [size\_is(cchPreviousNames), unique]
16. wchar\_t\* pszzPreviousNames;
17. FILETIME ftDriverDate;
18. DWORDLONG dwlDriverVersion;
19. [string] wchar\_t\* pMfgName;
20. [string] wchar\_t\* pOEMUrl;
21. [string] wchar\_t\* pHardwareID;
22. [string] wchar\_t\* pProvider;
23. [string] wchar\_t\* pPrintProcessor;
24. [string] wchar\_t\* pVendorSetup;
25. DWORD cchColorProfiles;
26. [size\_is(cchColorProfiles), unique]
27. wchar\_t\* pszzColorProfiles;
28. [string] wchar\_t\* pInfPath;
29. DWORD dwPrinterDriverAttributes;
30. DWORD cchCoreDependencies;
31. [size\_is(cchCoreDependencies), unique]
32. wchar\_t\* pszzCoreDriverDependencies;
33. FILETIME ftMinInboxDriverVerDate;
34. DWORDLONG dwlMinInboxDriverVerVersion;
35. } RPC\_DRIVER\_INFO\_8;

**pPrintProcessor:**  A pointer to a string that specifies the print processor for this printer. For rules governing print processor names, see section [2.2.4.11](#Section_807e0bdd21a54a729a6a1a7fb1b08a8e).

**pVendorSetup:**  An optional pointer to a string that specifies the name of the vendor setup file used for hardware vendor-provided custom setup.

**cchColorProfiles:**  The number of characters in the multisz structure pointed to by the **pszzColorProfiles** member.

**pszzColorProfiles:**  An optional pointer to a multisz structure that contains the names of all [**color profile**](#gt_cffe5dfa-5408-4302-8ab0-400cb94f5e32) files for this driver.

**pInfPath:**  An optional pointer to a string that specifies the path to the installation configuration file in the [**driver store**](#gt_55d56bf9-1542-43e4-b603-2e08ad5cdd3f) that identifies the printer driver for installation.[<27>](#Appendix_A_27)

When used as an input parameter in a call to [RpcAddPrinterDriverEx (section 3.1.4.4.8)](#Section_b96cc49759e54510ab045484993b259b), this pointer MUST be NULL. When used as output in the custom-marshaled form of this structure ([\_DRIVER\_INFO\_8](#Section_f9b749d608fc48208d7bfe0ee33e42c5) section 2.2.2.4.8), the server SHOULD set this value for package-aware drivers.

**dwPrinterDriverAttributes:**  A bit field that specifies attributes of the printer driver.

When used as an input parameter in a call to RpcAddPrinterDriverEx, this value MUST be zero. When used as output in the custom-marshaled form of this structure (\_DRIVER\_INFO\_8), this bit field is defined as one or more of the bit flags described in the following table.

| Value | Meaning |
| --- | --- |
| PRINTER\_DRIVER\_PACKAGE\_AWARE  0x00000001 | The printer driver is part of a driver package.[<28>](#Appendix_A_28) |
| PRINTER\_DRIVER\_XPS  0x00000002 | The printer driver supports the Microsoft [**XML Paper Specification (XPS)**](#gt_0b8be35a-bc52-4a6d-b711-3ee441b1bf60) format described in [[MSFT-XMLPAPER]](http://go.microsoft.com/fwlink/?LinkId=90214).[<29>](#Appendix_A_29)[<30>](#Appendix_A_30) |
| PRINTER\_DRIVER\_SANDBOX\_ENABLED  0x00000004 | The printer driver is compatible with [**printer driver isolation**](#gt_caf22b1c-0b1b-447e-863e-30ee95fd57c7).[<31>](#Appendix_A_31)[<32>](#Appendix_A_32) |
| PRINTER\_DRIVER\_CLASS  0x00000008 | The printer driver is a class printer driver.[<33>](#Appendix_A_33)[<34>](#Appendix_A_34) |
| PRINTER\_DRIVER\_DERIVED  0x00000010 | The printer driver is a derived printer driver.[<35>](#Appendix_A_35)[<36>](#Appendix_A_36) |
| PRINTER\_DRIVER\_NOT\_SHAREABLE  0x00000020 | Printers using the printer driver cannot be shared.[<37>](#Appendix_A_37)[<38>](#Appendix_A_38) |
| PRINTER\_DRIVER\_CATEGORY\_FAX  0x00000040 | The printer driver is intended for use with [**fax printers**](#gt_ae9f38f1-e339-4eba-aacb-c0593a9ac177).[<39>](#Appendix_A_39)[<40>](#Appendix_A_40) |
| PRINTER\_DRIVER\_CATEGORY\_FILE  00x00000080 | The printer driver is intended for use with [**file printers**](#gt_c55516ca-bd87-4ced-ad49-5cd31099832b).[<41>](#Appendix_A_41)[<42>](#Appendix_A_42) |
| PRINTER\_DRIVER\_CATEGORY\_VIRTUAL  0x00000100 | The printer driver is intended for use with [**virtual printers**](#gt_baa16582-ddb9-4c52-91f7-889dec3229e2).[<43>](#Appendix_A_43)[<44>](#Appendix_A_44) |
| PRINTER\_DRIVER\_CATEGORY\_SERVICE  0x00000200 | The printer driver is intended for use with [**service printers**](#gt_6ebe6443-5eaa-4a1d-b586-bbb6ae38c5bf).[<45>](#Appendix_A_45)[<46>](#Appendix_A_46) |
| PRINTER\_DRIVER\_SOFT\_RESET\_REQUIRED  0x00000400 | Printers using this printer driver SHOULD use an implementation-specific mechanism to reset the printer when a print job is canceled.[<47>](#Appendix_A_47)[<48>](#Appendix_A_48) |
| PRINTER\_DRIVER\_CATEGORY\_3D  0x00000200 | The printer driver is intended for use with [**3D printers**](#gt_59823646-3e13-4a01-b7be-11e455520969).[<49>](#Appendix_A_49)[<50>](#Appendix_A_50) |

**cchCoreDependencies:**  The number of characters in the multisz structure pointed to by the **pszzCoreDriverDependencies** member.

**pszzCoreDriverDependencies:**  An optional pointer to a multisz structure that contains the names of the core dependencies as specified by the installation configuration file. These names specify the core sections of the installation configuration file that are required by the printer driver.[<51>](#Appendix_A_51)

**ftMinInboxDriverVerDate:**  The minimum date version that is required in order for any [**core printer driver**](#gt_d4010501-f6e4-4cba-817b-403a9c5ea41c) to be used as a dependency as listed in the multisz structure pointed to by the **pszzCoreDriverDependencies** member. The value of this member MUST be specified in the same format as the **ftDriverDate** member.[<52>](#Appendix_A_52)

**dwlMinInboxDriverVerVersion:**  The minimum file version that is required for any core printer driver to be used as a dependency as listed in the multisz structure pointed to by the **pszzCoreDriverDependencies** member. The value of this member MUST be specified in the same format as the **dwlDriverVersion** member.[<53>](#Appendix_A_53)

All members not defined in this section are specified in sections [2.2.1.3.1](#Section_4464eaf0f34f40d5b970736437a21913) and [2.2.1.3](#Section_964b517f67304b4a99edee9648e7f1f7).

#### FORM\_INFO

##### FORM\_INFO\_1

The FORM\_INFO\_1 structure provides information about a printer form.

1. typedef struct \_FORM\_INFO\_1 {
2. DWORD Flags;
3. [string] wchar\_t\* pName;
4. SIZE Size;
5. RECTL ImageableArea;
6. } FORM\_INFO\_1;

All members not defined in this section are specified in sections [2.2.1.3.2](#Section_84fe09faea4b4cc58ae49c8582fd1ffe) and [2.2.1.3](#Section_964b517f67304b4a99edee9648e7f1f7).

##### RPC\_FORM\_INFO\_2

The RPC\_FORM\_INFO\_2 structure provides information about a printer form that includes its origin, dimensions, the dimensions of its printable area, and its display name.[<54>](#Appendix_A_54)

1. typedef struct \_RPC\_FORM\_INFO\_2 {
2. DWORD Flags;
3. [string, unique] const wchar\_t\* pName;
4. SIZE Size;
5. RECTL ImageableArea;
6. [string, unique] const char\* pKeyword;
7. DWORD StringType;
8. [string, unique] const wchar\_t\* pMuiDll;
9. DWORD dwResourceId;
10. [string, unique] const wchar\_t\* pDisplayName;
11. LANGID wLangID;
12. } RPC\_FORM\_INFO\_2;

**pKeyword:**  This member MUST be set to NULL by the client if the value of the **Flags** member is set to **FORM\_BUILTIN**, or this member MUST be a non-NULL pointer to a string that MUST specify a unique, localization-independent identifier for this form.[<55>](#Appendix_A_55)

**StringType:**  The value of this member MUST specify how a form's display name is passed. The value of this member MUST be a value from the following table.

| Name/Value | Meaning |
| --- | --- |
| STRING\_NONE  0x00000001 | Use the default display name, a string that is pointed to by the **pName** member. No localized display name exists. |
| STRING\_MUIDLL  0x00000002 | Load the form name from the library of string resources that is identified by the **pMuiDll** member. The **dwResourceId** member specifies the ID of the form name string in that library. |
| STRING\_LANGPAIR  0x00000004 | Use the form name, a string that is pointed to by the **pDisplayName** member, and the language that is identified by the **wLangID** member. |

**pMuiDll:**  This member MUST be a NULL pointer and MUST be ignored on receipt if **StringType** is not equal to **STRING\_MUIDLL**, or it MUST be a non-NULL pointer to a string that contains the name of a library of string resources. String resources MAY[<56>](#Appendix_A_56) be localized into multiple languages.

**dwResourceId:**  The value of this member SHOULD be set to zero when sent and ignored on receipt if the value of the **StringType** member is not equal to **STRING\_MUIDLL**; otherwise, the value of this member MUST specify the string resource ID of the form name in the library that is identified by the string that is pointed to by the **pMuiDll** member.

**pDisplayName:**  This member MUST be a NULL pointer and ignored on receipt if **StringType** is not equal to **STRING\_LANGPAIR**; otherwise, this member MUST be a non-NULL pointer to a string that MUST specify the form name.

**wLangID:**  The value of this member SHOULD be set to zero when sent and ignored on receipt if **StringType** is not equal to **STRING\_LANGPAIR**; otherwise, the value of this member MUST be the Language Identifier of the **pDisplayName** member as specified in [[MS-LCID]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-LCID%5d.pdf).

All members not defined in this section are specified in sections [2.2.1.3.2](#Section_84fe09faea4b4cc58ae49c8582fd1ffe) and [2.2.1.3](#Section_964b517f67304b4a99edee9648e7f1f7).

#### JOB\_INFO

##### JOB\_INFO\_1

The JOB\_INFO\_1 structure provides information about a print job.

1. typedef struct \_JOB\_INFO\_1 {
2. DWORD JobId;
3. [string] wchar\_t\* pPrinterName;
4. [string] wchar\_t\* pMachineName;
5. [string] wchar\_t\* pUserName;
6. [string] wchar\_t\* pDocument;
7. [string] wchar\_t\* pDatatype;
8. [string] wchar\_t\* pStatus;
9. DWORD Status;
10. DWORD Priority;
11. DWORD Position;
12. DWORD TotalPages;
13. DWORD PagesPrinted;
14. SYSTEMTIME Submitted;
15. } JOB\_INFO\_1;

All members not defined in this section are specified in sections [2.2.1.3.3](#Section_94d1857ec44449b18fb483eac6696060) and [2.2.1.3](#Section_964b517f67304b4a99edee9648e7f1f7).

##### JOB\_INFO\_2

The JOB\_INFO\_2 structure provides information about a print job.

1. typedef struct \_JOB\_INFO\_2 {
2. DWORD JobId;
3. [string] wchar\_t\* pPrinterName;
4. [string] wchar\_t\* pMachineName;
5. [string] wchar\_t\* pUserName;
6. [string] wchar\_t\* pDocument;
7. [string] wchar\_t\* pNotifyName;
8. [string] wchar\_t\* pDatatype;
9. [string] wchar\_t\* pPrintProcessor;
10. [string] wchar\_t\* pParameters;
11. [string] wchar\_t\* pDriverName;
12. ULONG\_PTR pDevMode;
13. [string] wchar\_t\* pStatus;
14. ULONG\_PTR pSecurityDescriptor;
15. DWORD Status;
16. DWORD Priority;
17. DWORD Position;
18. DWORD StartTime;
19. DWORD UntilTime;
20. DWORD TotalPages;
21. DWORD Size;
22. SYSTEMTIME Submitted;
23. DWORD Time;
24. DWORD PagesPrinted;
25. } JOB\_INFO\_2;

All members not defined in this section are specified in sections [2.2.1.3.3](#Section_94d1857ec44449b18fb483eac6696060) and [2.2.1.3](#Section_964b517f67304b4a99edee9648e7f1f7).

##### JOB\_INFO\_3

The JOB\_INFO\_3 structure provides information about a print job.[<57>](#Appendix_A_57)

1. typedef struct \_JOB\_INFO\_3 {
2. DWORD JobId;
3. DWORD NextJobId;
4. DWORD Reserved;
5. } JOB\_INFO\_3;

**NextJobId:**  An identifier that specifies the print job in the queue following the job identified by the **JobId** member. A value of zero indicates that there are no jobs following the job identified by the **JobId** member.

When used as input to [RpcSetJob (section 3.1.4.3.1)](#Section_f978bd2ce99a43ffba0802e892948dac) to alter the order of print jobs and link them together, **JobId** and **NextJobId** MUST be nonzero and SHOULD be obtained through [RpcEnumJobs (section 3.1.4.3.3)](#Section_0c77f1ff08274f4da06d5a3d802ffc0c) or [RpcGetJob (section 3.1.4.3.2)](#Section_f7090127287b4aa7ba8b2b6b7247dfc6).

All members not defined in this section are specified in sections [2.2.1.3.3](#Section_94d1857ec44449b18fb483eac6696060) and [2.2.1.3](#Section_964b517f67304b4a99edee9648e7f1f7).

##### JOB\_INFO\_4

The JOB\_INFO\_4 structure provides information about a print job.[<58>](#Appendix_A_58)

1. typedef struct \_JOB\_INFO\_4 {
2. DWORD JobId;
3. [string] wchar\_t\* pPrinterName;
4. [string] wchar\_t\* pMachineName;
5. [string] wchar\_t\* pUserName;
6. [string] wchar\_t\* pDocument;
7. [string] wchar\_t\* pNotifyName;
8. [string] wchar\_t\* pDatatype;
9. [string] wchar\_t\* pPrintProcessor;
10. [string] wchar\_t\* pParameters;
11. [string] wchar\_t\* pDriverName;
12. ULONG\_PTR pDevMode;
13. [string] wchar\_t\* pStatus;
14. ULONG\_PTR pSecurityDescriptor;
15. DWORD Status;
16. DWORD Priority;
17. DWORD Position;
18. DWORD StartTime;
19. DWORD UntilTime;
20. DWORD TotalPages;
21. DWORD Size;
22. SYSTEMTIME Submitted;
23. DWORD Time;
24. DWORD PagesPrinted;
25. long SizeHigh;
26. } JOB\_INFO\_4;

**SizeHigh:**  This member specifies the high-order 32 bits of a 64-bit unsigned integer that specifies the size of the job, in bytes.

All members not defined in this section are specified in sections [2.2.1.3.3](#Section_94d1857ec44449b18fb483eac6696060) and [2.2.1.3](#Section_964b517f67304b4a99edee9648e7f1f7).

#### MONITOR\_INFO

##### MONITOR\_INFO\_1

The MONITOR\_INFO\_1 structure provides information about a monitor.

1. typedef struct \_MONITOR\_INFO\_1 {
2. [string] wchar\_t\* pName;
3. } MONITOR\_INFO\_1;

All members not defined in this section are specified in sections [2.2.1.3.4](#Section_70e8e10238754ce8a292aaabc88e1458) and [2.2.1.3](#Section_964b517f67304b4a99edee9648e7f1f7).

##### MONITOR\_INFO\_2

The MONITOR\_INFO\_2 structure provides information about a monitor.

1. typedef struct \_MONITOR\_INFO\_2 {
2. [string] wchar\_t\* pName;
3. [string] wchar\_t\* pEnvironment;
4. [string] wchar\_t\* pDLLName;
5. } MONITOR\_INFO\_2;

**pEnvironment:**  This member MUST be a non-NULL pointer to a string that MUST specify the environment that the monitor supports. The environment specified MUST match the print server's operating system. For rules governing environment names and Windows behaviors, see section [2.2.4.4](#Section_ad7758938a194786ab4634731f52d1af).

**pDLLName:**  This member MUST be a non-NULL pointer to a string that MUST specify the name of the port monitor executable object.

All members not defined in this section are specified in sections [2.2.1.3.4](#Section_70e8e10238754ce8a292aaabc88e1458) and [2.2.1.3](#Section_964b517f67304b4a99edee9648e7f1f7).

#### PORT\_INFO

##### PORT\_INFO\_1

The PORT\_INFO\_1 structure provides information about a port.

1. typedef struct \_PORT\_INFO\_1 {
2. [string] wchar\_t\* pPortName;
3. } PORT\_INFO\_1;

All members not defined in this section are specified in sections [2.2.1.3.5](#Section_2607e270ebe64d84a2f712b056fae964) and [2.2.1.3](#Section_964b517f67304b4a99edee9648e7f1f7).

##### PORT\_INFO\_2

The PORT\_INFO\_2 structure provides information about a port.

1. typedef struct \_PORT\_INFO\_2 {
2. [string] wchar\_t\* pPortName;
3. [string] wchar\_t\* pMonitorName;
4. [string] wchar\_t\* pDescription;
5. DWORD fPortType;
6. DWORD Reserved;
7. } PORT\_INFO\_2;

**pMonitorName:**  A pointer to a string that specifies an installed port monitor. For rules governing port monitor names, see section [2.2.4.8](#Section_808e2e6c7f914bdd92da56fbca513404).

**pDescription:**  An optional pointer to a string that specifies additional implementation-specific information about the printer port.[<59>](#Appendix_A_59)

**fPortType:**  A bit field that specifies attributes of the printer port. These flags can be combined to specify multiple attributes.

| Value | Meaning |
| --- | --- |
| PORT\_TYPE\_WRITE  0x00000001 | The port can be written to. |
| PORT\_TYPE\_READ  0x00000002 | The port can be read from. |
| PORT\_TYPE\_REDIRECTED  0x00000004 | The port is a [**Terminal Services**](#gt_ffff3f01-8c21-44d3-bbda-0062a1fbda4b) redirected port. |
| PORT\_TYPE\_NET\_ATTACHED  0x00000008 | The port is a network TCP/IP port. |

All members not defined in this section are specified in sections [2.2.1.3.5](#Section_2607e270ebe64d84a2f712b056fae964) and [2.2.1.3](#Section_964b517f67304b4a99edee9648e7f1f7).

##### PORT\_INFO\_3

The PORT\_INFO\_3 structure provides information about a port.[<60>](#Appendix_A_60)

1. typedef struct \_PORT\_INFO\_3 {
2. DWORD dwStatus;
3. [string] wchar\_t\* pszStatus;
4. DWORD dwSeverity;
5. } PORT\_INFO\_3;

**dwStatus:**  The new port status. This value MUST be one of the following.

| Name/value | Meaning |
| --- | --- |
| PORT\_STATUS\_CLEAR  0x00000000 | Clears the printer port status. |
| PORT\_STATUS\_OFFLINE  0x00000001 | The port’s printer is offline. |
| PORT\_STATUS\_PAPER\_JAM  0x00000002 | The port’s printer has a paper jam. |
| PORT\_STATUS\_PAPER\_OUT  0x00000003 | The port’s printer is out of paper. |
| PORT\_STATUS\_OUTPUT\_BIN\_FULL  0x00000004 | The port’s printer's output bin is full. |
| PORT\_STATUS\_PAPER\_PROBLEM  0x00000005 | The port’s printer has a paper problem. |
| PORT\_STATUS\_NO\_TONER  0x00000006 | The port’s printer is out of toner. |
| PORT\_STATUS\_DOOR\_OPEN  0x00000007 | The door of the port’s printer is open. |
| PORT\_STATUS\_USER\_INTERVENTION  0x00000008 | The port’s printer requires user intervention. |
| PORT\_STATUS\_OUT\_OF\_MEMORY  0x00000009 | The port’s printer is out of memory. |
| PORT\_STATUS\_TONER\_LOW  0x0000000A | The port’s printer is low on toner. |
| PORT\_STATUS\_WARMING\_UP  0x0000000B | The port’s printer is warming up. |
| PORT\_STATUS\_POWER\_SAVE  0x0000000C | The port’s printer is in a power-conservation mode. |

**pszStatus:**  An optional pointer to a string that specifies a status description.

**dwSeverity:**  The severity of the port status value. This value MUST be one of the following.

| Name/value | Meaning |
| --- | --- |
| PORT\_STATUS\_TYPE\_ERROR  0x00000001 | The port status value indicates an error. |
| PORT\_STATUS\_TYPE\_WARNING  0x00000002 | The port status value is a warning. |
| PORT\_STATUS\_TYPE\_INFO  0x00000003 | The port status value is informational. |

All members not defined in this section are specified in sections [2.2.1.3.5](#Section_2607e270ebe64d84a2f712b056fae964) and [2.2.1.3](#Section_964b517f67304b4a99edee9648e7f1f7).

##### PORT\_INFO\_FF

The PORT\_INFO\_FF is used to communicate port information to a local port monitor.[<61>](#Appendix_A_61)

1. typedef struct \_PORT\_INFO\_FF {
2. [string] wchar\_t\* pPortName;
3. DWORD cbMonitorData;
4. BYTE\* pMonitorData;
5. } PORT\_INFO\_FF;

**cbMonitorData:**  A value that SHOULD be set to zero when sent and MUST be ignored on receipt.

**pMonitorData:**  A pointer that SHOULD be set to NULL when sent and MUST be ignored on receipt.

All members not defined in this section are specified in sections [2.2.1.3.5](#Section_2607e270ebe64d84a2f712b056fae964) and [2.2.1.3](#Section_964b517f67304b4a99edee9648e7f1f7).

#### PRINTER\_INFO

##### PRINTER\_INFO\_STRESS

The PRINTER\_INFO\_STRESS structure provides diagnostic printer information used for [**print system remote protocol stress analysis**](#gt_753687cf-c0e6-462e-b3bf-ea70a7ceadd1).[<62>](#Appendix_A_62)

1. typedef struct \_PRINTER\_INFO\_STRESS {
2. [string] wchar\_t\* pPrinterName;
3. [string] wchar\_t\* pServerName;
4. DWORD cJobs;
5. DWORD cTotalJobs;
6. DWORD cTotalBytes;
7. SYSTEMTIME stUpTime;
8. DWORD MaxcRef;
9. DWORD cTotalPagesPrinted;
10. DWORD dwGetVersion;
11. DWORD fFreeBuild;
12. DWORD cSpooling;
13. DWORD cMaxSpooling;
14. DWORD cRef;
15. DWORD cErrorOutOfPaper;
16. DWORD cErrorNotReady;
17. DWORD cJobError;
18. DWORD dwNumberOfProcessors;
19. DWORD dwProcessorType;
20. DWORD dwHighPartTotalBytes;
21. DWORD cChangeID;
22. DWORD dwLastError;
23. DWORD Status;
24. DWORD cEnumerateNetworkPrinters;
25. DWORD cAddNetPrinters;
26. unsigned short wProcessorArchitecture;
27. unsigned short wProcessorLevel;
28. DWORD cRefIC;
29. DWORD dwReserved2;
30. DWORD dwReserved3;
31. } PRINTER\_INFO\_STRESS;

**cJobs:**  The number of jobs that are currently in the print queue.

**cTotalJobs:**  The total number of jobs that have been spooled since the print server was started.

**cTotalBytes:**  The low-order 32 bits of an unsigned 64-bit value that specifies the total number of bytes that have been printed since system startup. The high-order 32 bits are specified by the **dwHighPartTotalBytes** member.

**stUpTime:**  The time the printer data structure was created, in SYSTEMTIME format.

**MaxcRef:**  The historic maximum value of the **cRef** member.

**cTotalPagesPrinted:**  The total number of pages printed.

**dwGetVersion:**  An implementation-specific value that specifies the version of the operating system.[<63>](#Appendix_A_63)

**fFreeBuild:**  An implementation-specific value that MUST be ignored on receipt.[<64>](#Appendix_A_64)

**cSpooling:**  The number of actively spooling jobs.

**cMaxSpooling:**  The historic maximum number of actively spooling jobs.

**cRef:**  The reference count for opened printer objects.

**cErrorOutOfPaper:**  The total number of out-of-paper errors.

**cErrorNotReady:**  The total number of not-ready errors.

**cJobError:**  The total number of job errors.

**dwNumberOfProcessors:**  The number of processors in the computer on which the print server is running.

**dwProcessorType:**  An implementation-specific value that identifies the type of processor in the computer.[<65>](#Appendix_A_65)

**dwHighPartTotalBytes:**  The high-order 32 bits of an unsigned 64-bit value that specifies the total number of bytes that have been printed since system startup. The low-order 32 bits are specified by the **cTotalBytes** member.

**cChangeID:**  A unique number that identifies the last change.

**dwLastError:**  An implementation-specific error code for the last error that occurred with this printer.[<66>](#Appendix_A_66)

**Status:**  The current printer status (section [2.2.3.12](#Section_1625e9d929e448f4b83d3bd0fdaea787)).

**cEnumerateNetworkPrinters:**  The number of times the network printers in the "List of Known Printers" have been requested.

**cAddNetPrinters:**  The number of network printers added, per server.

**wProcessorArchitecture:**  An implementation-specific value that identifies the system's processor architecture. This value SHOULD be ignored on receipt.[<67>](#Appendix_A_67)

**wProcessorLevel:**  An implementation-specific value that identifies the system's architecture-dependent processor level. This value SHOULD be ignored on receipt.[<68>](#Appendix_A_68)

**cRefIC:**  The number of open information context handles.

All members not defined in this section are specified in sections [2.2.1.3.6](#Section_c07c2bfde2c94d369259354a229d645d) and [2.2.1.3](#Section_964b517f67304b4a99edee9648e7f1f7).

##### PRINTER\_INFO\_1

The PRINTER\_INFO\_1 structure provides information about a printer.

1. typedef struct \_PRINTER\_INFO\_1 {
2. DWORD Flags;
3. [string] wchar\_t\* pDescription;
4. [string] wchar\_t\* pName;
5. [string] wchar\_t\* pComment;
6. } PRINTER\_INFO\_1;

**Flags:**  The value of this member MUST be the result of a bitwise OR of zero or more of the [Printer Enumeration Flags (section 2.2.3.7)](#Section_2a1fe8a4e8be4cf38b378d19f9a2edcd).

If the PRINTER\_INFO\_1 structure is used in a [PRINTER\_CONTAINER (section 2.2.1.2.9)](#Section_ad7d4d4e0feb45e093edb3b2b9a313d1) as input to [RpcAddPrinter (section 3.1.4.2.3)](#Section_bd0bb8d526fd470581f5614d32b9f38f) or [RpcAddPrinterEx (section 3.1.4.2.15)](#Section_3af45717308f4b4f8428430d12856110), **Flags** MUST be a bitwise OR of zero or more of the PRINTER\_ATTRIBUTE values defined in [Status and Attribute Values (section 2.2.3.12)](#Section_1625e9d929e448f4b83d3bd0fdaea787).

**pName:**  This member is synonymous with **pPrinterName**, as specified in section [3.1.4.1.5](#Section_3f2d41eb30e147e193bb534ea50cf8eb).

All members not defined in this section are specified in sections [2.2.1.3.6](#Section_c07c2bfde2c94d369259354a229d645d) and [2.2.1.3](#Section_964b517f67304b4a99edee9648e7f1f7).

##### PRINTER\_INFO\_2

The PRINTER\_INFO\_2 structure provides information about a printer.

1. typedef struct \_PRINTER\_INFO\_2 {
2. [string] wchar\_t\* pServerName;
3. [string] wchar\_t\* pPrinterName;
4. [string] wchar\_t\* pShareName;
5. [string] wchar\_t\* pPortName;
6. [string] wchar\_t\* pDriverName;
7. [string] wchar\_t\* pComment;
8. [string] wchar\_t\* pLocation;
9. ULONG\_PTR pDevMode;
10. [string] wchar\_t\* pSepFile;
11. [string] wchar\_t\* pPrintProcessor;
12. [string] wchar\_t\* pDatatype;
13. [string] wchar\_t\* pParameters;
14. ULONG\_PTR pSecurityDescriptor;
15. DWORD Attributes;
16. DWORD Priority;
17. DWORD DefaultPriority;
18. DWORD StartTime;
19. DWORD UntilTime;
20. DWORD Status;
21. DWORD cJobs;
22. DWORD AveragePPM;
23. } PRINTER\_INFO\_2;

**pShareName:**  This member is an optional pointer to a string that specifies the share name for the printer. This string MUST be ignored unless the **Attributes** member contains the **PRINTER\_ATTRIBUTED\_SHARED** flag. For rules governing path names, see section [2.2.4.9](#Section_1d75f622d07c45ccbf1ea6191e408516).

**pDriverName:**  This member is a pointer to a string that specifies the name of the printer driver. For rules governing printer driver names, see section [2.2.4.3](#Section_a4c7686fa88c48f996ebbf4dfc3fe519).

**pLocation:**  This member is an optional pointer to a string that specifies the location of the printer.

**pSepFile:**  This member is an optional pointer to a string that specifies the name of a file whose contents are used to create a separator page. This page is used to separate print jobs sent to the printer. For rules governing path names, see section 2.2.4.9.

**pPrintProcessor:**  This member is an optional pointer to a string that specifies the name of the print processor used by the printer. For rules governing print processor names, see section [2.2.4.11](#Section_807e0bdd21a54a729a6a1a7fb1b08a8e).

If this member is NULL on input, the server SHOULD use the print processor that is associated with the printer driver identified by the string pointed to by the **pDriverName** member.

**pDatatype:**  This member is an optional pointer to a string that specifies the default data format used to record print jobs on the printer. For rules governing data type names, see section [2.2.4.2](#Section_f40cbad088dd4e86a16bcf62c3b17c52).

If this member is NULL on input, the server MUST choose a default data type from one of the data types supported by the print processor associated with the printer.[<69>](#Appendix_A_69)

**pParameters:**  This member is an optional pointer to a string that specifies the default print processor parameters.

**Priority:**  The value of this member specifies a priority value that the spooler uses to route each print job. The value of this member MUST be from 0 through 99, inclusive.

**DefaultPriority:**  The value of this member specifies the default priority value assigned to each print job. The value of this member MUST be from 0 through 99, inclusive.

**StartTime:**  The value of this member specifies the earliest time that a job can be printed. The time is expressed as the number of minutes after 12:00 AM GMT within a 24-hour boundary.

**UntilTime:**  The value of this member specifies the latest time that a job can be printed. The time is expressed as the number of minutes after 12:00 AM GMT within a 24-hour boundary.

**cJobs:**  The value of this member specifies the number of print jobs that have been queued for the printer.

**AveragePPM:**  The value of this member specifies the average pages per minute that have been printed on the printer.

All members not defined in this section are specified in sections [2.2.1.3.6](#Section_c07c2bfde2c94d369259354a229d645d) and [2.2.1.3](#Section_964b517f67304b4a99edee9648e7f1f7).

##### PRINTER\_INFO\_3

The PRINTER\_INFO\_3 structure provides information about a printer.

1. typedef struct \_PRINTER\_INFO\_3 {
2. ULONG\_PTR pSecurityDescriptor;
3. } PRINTER\_INFO\_3;

All members not defined in this section are specified in sections [2.2.1.3.6](#Section_c07c2bfde2c94d369259354a229d645d) and [2.2.1.3](#Section_964b517f67304b4a99edee9648e7f1f7).

##### PRINTER\_INFO\_4

The PRINTER\_INFO\_4 structure provides information about a printer.[<70>](#Appendix_A_70)

1. typedef struct \_PRINTER\_INFO\_4 {
2. [string] wchar\_t\* pPrinterName;
3. [string] wchar\_t\* pServerName;
4. DWORD Attributes;
5. } PRINTER\_INFO\_4;

All members not defined in this section are specified in sections [2.2.1.3.6](#Section_c07c2bfde2c94d369259354a229d645d) and [2.2.1.3](#Section_964b517f67304b4a99edee9648e7f1f7).

##### PRINTER\_INFO\_5

The PRINTER\_INFO\_5 structure provides information about a printer.[<71>](#Appendix_A_71)

1. typedef struct \_PRINTER\_INFO\_5 {
2. [string] wchar\_t\* pPrinterName;
3. [string] wchar\_t\* pPortName;
4. DWORD Attributes;
5. DWORD DeviceNotSelectedTimeout;
6. DWORD TransmissionRetryTimeout;
7. } PRINTER\_INFO\_5;

**DeviceNotSelectedTimeout:**  The value of this member MUST specify the maximum number of milliseconds between select attempts. The **DeviceNotSelectedTimeout** value controls communication between the print server and a print device. It does not have any effect on communication between the print client and the print server.

**TransmissionRetryTimeout:**  The value of this member MUST specify the maximum number of milliseconds between retransmission attempts. The **TransmissionRetryTimeout** value controls communication between the print server and a print device. It does not have any effect on communication between the print client and the print server.

All members not defined in this section are specified in sections [2.2.1.3.6](#Section_c07c2bfde2c94d369259354a229d645d) and [2.2.1.3](#Section_964b517f67304b4a99edee9648e7f1f7).

##### PRINTER\_INFO\_6

The PRINTER\_INFO\_6 structure provides information about a printer.[<72>](#Appendix_A_72)

1. typedef struct \_PRINTER\_INFO\_6 {
2. DWORD dwStatus;
3. } PRINTER\_INFO\_6;

**dwStatus:**  The value of this member MUST specify the printer status. It MUST be the result of a bitwise OR of zero or more of the printer status values defined in section [2.2.3.12](#Section_1625e9d929e448f4b83d3bd0fdaea787).

All members not defined in this section are specified in sections [2.2.1.3.6](#Section_c07c2bfde2c94d369259354a229d645d) and [2.2.1.3](#Section_964b517f67304b4a99edee9648e7f1f7).

##### PRINTER\_INFO\_7

The PRINTER\_INFO\_7 structure provides directory service (DS) information about a printer.[<73>](#Appendix_A_73)

1. typedef struct \_PRINTER\_INFO\_7 {
2. [string] wchar\_t\* pszObjectGUID;
3. DWORD dwAction;
4. } PRINTER\_INFO\_7;

**pszObjectGUID:**  A pointer to a string that represents the GUID that is used by the DS to identify this printer, if it is used in a response to [RpcGetPrinter (section 3.1.4.2.6)](#Section_ccbf288a0768416dbede0ef1987fc286).

The GUID string MUST conform to the UUID grammar, as specified in [[RFC4122]](http://go.microsoft.com/fwlink/?LinkId=90460) section 3. The string representation of a 128-bit GUID is a [**GUIDString**](#gt_b330cb96-68e2-41be-a53f-cae98c2b5d32).

This member SHOULD be set to NULL when sent and MUST be ignored on receipt by the server, if it is used by the client in a call to [RpcSetPrinter (section 3.1.4.2.5)](#Section_382997b5b02a4b6a8c6849a64b8c45f8).

**dwAction:**  An action for the printer to perform, if it used by the client in a call to RpcSetPrinter.

The value of this member MUST represent a DS-specific publishing state by the server if it is used in a response to RpcGetPrinter.

The value of this member MUST be a constant from the following table:

| Name | Meaning |
| --- | --- |
| DSPRINT\_PUBLISH  0x00000001 | RpcSetPrinter: The server MUST publish the printer's data in the DS as described in section [2.3.3.1](#Section_3f4f0a52e417495aad460466a9d0ef9d).  RpcGetPrinter: The server MUST set this value to indicate the printer is published in the DS. |
| DSPRINT\_UPDATE  0x00000002 | RpcSetPrinter: The server MUST update the printer's published data in the DS as described in section [2.3.3.2](#Section_1c49f0840f634ac49e566be4c47915e6).  RpcGetPrinter: This value MUST NOT be returned by the server. |
| DSPRINT\_UNPUBLISH  0x00000004 | RpcSetPrinter: The server MUST remove the printer's published data from the DS as described in section 2.3.3.2.  RpcGetPrinter: The server MUST set this value to indicate the printer is not published. |
| DSPRINT\_REPUBLISH  0x00000008 | RpcSetPrinter: The server MUST unpublish (as described in section 2.3.3.2) and publish again (as described in section 2.3.3.1) the DS data for the printer. Republishing also MUST change the GUID of the published printer.  RpcGetPrinter: The server MUST NOT set this value. |
| DSPRINT\_PENDING  0x80000000 | RpcSetPrinter: This value MUST NOT be used by the client.  RpcGetPrinter: The server MUST return this value, if a previous publish or unpublish action initiated by RpcSetPrinter is still in progress. |

All members not defined in this section are specified in sections [2.2.1.3.6](#Section_c07c2bfde2c94d369259354a229d645d) and [2.2.1.3](#Section_964b517f67304b4a99edee9648e7f1f7).

##### PRINTER\_INFO\_8

The PRINTER\_INFO\_8 structure provides information about a printer.[<74>](#Appendix_A_74)

This structure is used for the global default settings of a printer.

1. typedef struct \_PRINTER\_INFO\_8 {
2. ULONG\_PTR pDevMode;
3. } PRINTER\_INFO\_8;

All members not defined in this section are specified in sections [2.2.1.3.6](#Section_c07c2bfde2c94d369259354a229d645d) and [2.2.1.3](#Section_964b517f67304b4a99edee9648e7f1f7).

##### PRINTER\_INFO\_9

The PRINTER\_INFO\_9 structure is not used remotely.[<75>](#Appendix_A_75)

1. typedef struct \_PRINTER\_INFO\_9 {
2. ULONG\_PTR pDevMode;
3. } PRINTER\_INFO\_9;

#### SPLCLIENT\_INFO

##### SPLCLIENT\_INFO\_1

The SPLCLIENT\_INFO\_1 structure provides information about the calling client of the print server.[<76>](#Appendix_A_76)

1. typedef struct \_SPLCLIENT\_INFO\_1 {
2. DWORD dwSize;
3. [string] wchar\_t\* pMachineName;
4. [string] wchar\_t\* pUserName;
5. DWORD dwBuildNum;
6. DWORD dwMajorVersion;
7. DWORD dwMinorVersion;
8. unsigned short wProcessorArchitecture;
9. } SPLCLIENT\_INFO\_1;

**dwSize:**  The value of this member MUST specify the size, in bytes, of the structure.

All members not defined in this section are specified in sections [2.2.1.3.7](#Section_fae766fc31d343cc902f8e6cc4c75f31) and [2.2.1.3](#Section_964b517f67304b4a99edee9648e7f1f7).

##### SPLCLIENT\_INFO\_2

The SPLCLIENT\_INFO\_2 structure is specified only as a placeholder in the IDL. It is not sent over the wire.

1. typedef struct \_SPLCLIENT\_INFO\_2 {
2. LONG\_PTR notUsed;
3. } SPLCLIENT\_INFO\_2;

**notUsed:**  A value that MUST be ignored.

##### SPLCLIENT\_INFO\_3

The SPLCLIENT\_INFO\_3 structure provides information about the calling client of the print server.[<77>](#Appendix_A_77)

1. typedef struct \_SPLCLIENT\_INFO\_3 {
2. unsigned int cbSize;
3. DWORD dwFlags;
4. DWORD dwSize;
5. [string] wchar\_t\* pMachineName;
6. [string] wchar\_t\* pUserName;
7. DWORD dwBuildNum;
8. DWORD dwMajorVersion;
9. DWORD dwMinorVersion;
10. unsigned short wProcessorArchitecture;
11. unsigned \_\_int64 hSplPrinter;
12. } SPLCLIENT\_INFO\_3;

**cbSize:**  The value of this member MUST specify the size, in bytes, of the structure.

**dwFlags:**  This member is reserved for future use. The value of this member SHOULD be set to zero when sent and MUST be ignored on receipt.

**dwSize:**  This member is reserved for future use. The value of this member SHOULD be set to zero when sent and MUST be ignored on receipt.

**hSplPrinter:**  This member MUST NOT be used remotely and the value of this member SHOULD be set to zero for calls that are made remotely.

All members not defined in this section are specified in sections [2.2.1.3.7](#Section_fae766fc31d343cc902f8e6cc4c75f31) and [2.2.1.3](#Section_964b517f67304b4a99edee9648e7f1f7).

#### Bidirectional Communication Data

##### RPC\_BIDI\_REQUEST\_DATA

The RPC\_BIDI\_REQUEST\_DATA structure holds a single bidirectional request.[<78>](#Appendix_A_78) The request is part of a bidirectional communication request using the [RpcSendRecvBidiData (section 3.1.4.2.27)](#Section_5aec83472ebf464eb6d1540aa60b5c71) method. One or more RPC\_BIDI\_REQUEST\_DATA structures MUST be contained in a [RPC\_BIDI\_REQUEST\_CONTAINER (section 2.2.1.2.10)](#Section_4ea856c44c374485868ae7ac8e331e5e).

1. typedef struct \_RPC\_BIDI\_REQUEST\_DATA {
2. DWORD dwReqNumber;
3. [string, unique] wchar\_t\* pSchema;
4. RPC\_BIDI\_DATA data;
5. } RPC\_BIDI\_REQUEST\_DATA;

**dwReqNumber:**  The index of the request, which is used to match a response to a request in a multi-request operation.

**pSchema:**  A pointer to the schema string that identifies the requested information. [<79>](#Appendix_A_79)

**data:**  The data that is associated with the schema.

##### RPC\_BIDI\_RESPONSE\_DATA

The RPC\_BIDI\_RESPONSE\_DATA structure holds a single bidirectional response.[<80>](#Appendix_A_80)

1. typedef struct \_RPC\_BIDI\_RESPONSE\_DATA {
2. DWORD dwResult;
3. DWORD dwReqNumber;
4. [string, unique] wchar\_t\* pSchema;
5. RPC\_BIDI\_DATA data;
6. } RPC\_BIDI\_RESPONSE\_DATA;

**dwResult:**  The result of the operation that used this structure. If the operation was successful, the value of this member MUST be set to zero; otherwise, the value of this member MUST be set to a nonzero value.[<81>](#Appendix_A_81)

**dwReqNumber:**  The index of the response, which is used to match the response to the request in a multi-request operation.

**pSchema:**  A pointer to the schema string that identifies the requested information.[<82>](#Appendix_A_82)

**data:**  The data that is associated with the schema. This can be a single piece of data or a homogeneous data list. The data MUST be composed of a name, a type, and a value; for example, "\Printer.Stapler:CurrentValue". It is referenced by its name under **Properties**.

##### RPC\_BIDI\_DATA

The RPC\_BIDI\_DATA structure is used to store the values of a bidirectional schema.[<83>](#Appendix_A_83)

1. typedef struct \_RPC\_BIDI\_DATA {
2. DWORD dwBidiType;
3. [switch\_is(dwBidiType)] union {
4. [case(BIDI\_NULL,BIDI\_BOOL)]
5. int bData;
6. [case(BIDI\_INT)]
7. long iData;
8. [case(BIDI\_STRING,BIDI\_TEXT,BIDI\_ENUM)]
9. [unique, string] wchar\_t\* sData;
10. [case(BIDI\_FLOAT)]
11. float fData;
12. [case(BIDI\_BLOB)]
13. RPC\_BINARY\_CONTAINER biData;
14. } u;
15. } RPC\_BIDI\_DATA;

**dwBidiType:**  The type of data in a bidirectional request. The value of this member specifies a valid structure for the **u** union. The value of this member MUST be one of the **BIDI\_TYPE** enumeration values specified in section [2.2.3.13](#Section_4ebaf54761f64b2eb32872c1a5dc685d).

**u:**  The bidirectional data in the format specified by the value of the **dwBidiType** member.

**bData:**  This case indicates that either there is no bidirectional data, or the bidirectional data is a Boolean value.

**iData:**  This case indicates that the bidirectional data is an integer.

**sData:**  This case indicates that the bidirectional data is either a string, text data, or an enumeration.

**fData:**  The bidirectional data is a floating-point number.

**biData:**  This case indicates that the bidirectional data is an [RPC\_BINARY\_CONTAINER](#Section_bc9f774d66fc45209563091553c3cabd) structure.

#### Printer Notification Data

##### RPC\_V2\_NOTIFY\_OPTIONS

The RPC\_V2\_NOTIFY\_OPTIONS structure specifies options for a change notification object that monitors a printer or print server for any changes in state.[<84>](#Appendix_A_84)

1. typedef struct \_RPC\_V2\_NOTIFY\_OPTIONS {
2. DWORD Version;
3. DWORD Reserved;
4. DWORD Count;
5. [size\_is(Count), unique] RPC\_V2\_NOTIFY\_OPTIONS\_TYPE\* pTypes;
6. } RPC\_V2\_NOTIFY\_OPTIONS;

**Version:**  The version of the structure. The value of this member MUST be 0x00000002.

**Reserved:**  A bit field that specifies attributes of the change notification. The name "Reserved" is intentional.

| Value | Meaning |
| --- | --- |
| PRINTER\_NOTIFY\_OPTIONS\_REFRESH  0x00000001 | Refreshed data is requested from the server for all monitored members. |

**Count:**  The number of [RPC\_V2\_NOTIFY\_OPTIONS\_TYPE structures (section 2.2.1.13.2)](#Section_3cce2507707d4a8e8ce8388463a24952) in the array pointed to by the **pTypes** member.

**pTypes:**  A pointer to an array of RPC\_V2\_NOTIFY\_OPTIONS\_TYPE structures, each of which identifies a set of print job or printer information members to be monitored by a printer change notification object.

##### RPC\_V2\_NOTIFY\_OPTIONS\_TYPE

The RPC\_V2\_NOTIFY\_OPTIONS\_TYPE structure MUST specify the set of printer or job information members to be monitored by a printer change notification object.[<85>](#Appendix_A_85)

1. typedef struct \_RPC\_V2\_NOTIFY\_OPTIONS\_TYPE {
2. unsigned short Type;
3. unsigned short Reserved0;
4. DWORD Reserved1;
5. DWORD Reserved2;
6. DWORD Count;
7. [size\_is(Count), unique] unsigned short\* pFields;
8. } RPC\_V2\_NOTIFY\_OPTIONS\_TYPE;

**Type:**  The value of this member specifies the type of notification to watch for. The value of this member MUST be one of the constant values from the following table.

| Name/Value | Meaning |
| --- | --- |
| PRINTER\_NOTIFY\_TYPE  0x0000 | Indicates that the members specified in the array that is pointed to by the **pFields** member are printer notification constants. |
| JOB\_NOTIFY\_TYPE  0x0001 | Indicates that the members specified in the array that is pointed to by the **pFields** member are job notification constants. |
| SERVER\_NOTIFY\_TYPE  0x0002 | Indicates that the members specified in the array that is pointed to by the **pFields** member are server notification constants.[<86>](#Appendix_A_86) |

**Reserved0:**  The value of this member MUST be set to zero when sent and MUST be ignored on receipt.

**Reserved1:**  The value of this member MUST be set to zero when sent and MUST be ignored on receipt.

**Reserved2:**  The value of this member MUST be set to zero when sent and MUST be ignored on receipt.

**Count:**  The value of this member MUST specify the number of elements in the **pFields** array.

**pFields:**  This member MUST be a non-NULL pointer to an array that MUST identify the job or printer information members to be monitored. The array MUST consist entirely of elements that are either job notification values (as specified in section [2.2.3.3](#Section_76acb7fbb63f4cad88dd6920096a2f13)) or printer notification values (as specified in section [2.2.3.8](#Section_41d5c622ec7749ff95e369b325ce4e77)), depending on the value of the **Type** member. The two types of notification values MUST NOT be mixed within a given instance of the array.

##### RPC\_V2\_NOTIFY\_INFO

The RPC\_V2\_NOTIFY\_INFO structure specifies printer or print job notification information.[<87>](#Appendix_A_87)

1. typedef struct \_RPC\_V2\_NOTIFY\_INFO {
2. DWORD Version;
3. DWORD Flags;
4. DWORD Count;
5. [size\_is(Count), unique] RPC\_V2\_NOTIFY\_INFO\_DATA aData[];
6. } RPC\_V2\_NOTIFY\_INFO;

**Version:**  The version of the structure. The value of this member MUST be 0x00000002.

**Flags:**  A bit field that specifies the state of the notification structure.

| Value | Meaning |
| --- | --- |
| PRINTER\_NOTIFY\_INFO\_DISCARDED  0x00000001 | An overflow or error has occurred, and notifications have been lost. The print server MUST NOT send additional notifications until the client has made a call to [RpcRouterRefreshPrinterChangeNotification (section 3.1.4.10.5)](#Section_cb5d22ed0352430383d5b5afd32fb461). |

**Count:**  The number of [RPC\_V2\_NOTIFY\_INFO\_DATA structures (section 2.2.1.13.4)](#Section_440ed0c046a4420193cdcc1e5d1fd022) in the **aData** array.

**aData:**  An array of RPC\_V2\_NOTIFY\_INFO\_DATA structures, each of which identifies a single print job or printer information member and specifies the current data for that member.

##### RPC\_V2\_NOTIFY\_INFO\_DATA

The RPC\_V2\_NOTIFY\_INFO\_DATA structure specifies printer or print job notification information data.[<88>](#Appendix_A_88)

1. typedef struct \_RPC\_V2\_NOTIFY\_INFO\_DATA {
2. unsigned short Type;
3. unsigned short Field;
4. DWORD Reserved;
5. DWORD Id;
6. [switch\_is(Reserved & 0x0000FFFF)]
7. RPC\_V2\_NOTIFY\_INFO\_DATA\_DATA Data;
8. } RPC\_V2\_NOTIFY\_INFO\_DATA;

**Type:**  The type of notification information that is contained in this structure. This MUST be one of the following values:

| Name/value | Meaning |
| --- | --- |
| PRINTER\_NOTIFY\_TYPE  0x0000 | Printer-related notifications |
| JOB\_NOTIFY\_TYPE  0x0001 | Job-related notifications |

**Field:**  The member that changed using the printer notification values and job notification values in sections [2.2.3.8](#Section_41d5c622ec7749ff95e369b325ce4e77) and [2.2.3.3](#Section_76acb7fbb63f4cad88dd6920096a2f13).

**Reserved:**  The member of the [RPC\_V2\_NOTIFY\_INFO\_DATA\_DATA union (section 2.2.1.13.5)](#Section_7567e260f0044427832c4eda6449bfa0) that is used to specify the data type of the **Data** member. Only the 16 least-significant bits of this member are used. The value of these bits MUST be one of the [Notification Data Type values (section 2.2.3.5)](#Section_6e1ac55984a34b3ea89707159974661e).

**Id:**  The job identifier if the **Type** member specifies **JOB\_NOTIFY\_TYPE**; otherwise, this value MUST be ignored.

**Data:**  The data determined by the values of the **Type** and **Reserved** members of this structure. The data is in an **RPC\_V2\_NOTIFY\_INFO\_DATA\_DATA** structure using the data type described by the value of the **Reserved** member.

##### RPC\_V2\_NOTIFY\_INFO\_DATA\_DATA

The RPC\_V2\_NOTIFY\_INFO\_DATA\_DATA union specifies the data information container for the current notification.[<89>](#Appendix_A_89) The case attributes of this union are specified by [Notification Data Type values (section 2.2.3.5)](#Section_6e1ac55984a34b3ea89707159974661e).

1. typedef
2. [switch\_type(DWORD)]
3. union \_RPC\_V2\_NOTIFY\_INFO\_DATA\_DATA {
4. [case(TABLE\_STRING)]
5. STRING\_CONTAINER String;
6. [case(TABLE\_DWORD)]
7. DWORD dwData[2];
8. [case(TABLE\_TIME)]
9. SYSTEMTIME\_CONTAINER SystemTime;
10. [case(TABLE\_DEVMODE)]
11. DEVMODE\_CONTAINER DevMode;
12. [case(TABLE\_SECURITYDESCRIPTOR)]
13. SECURITY\_CONTAINER SecurityDescriptor;
14. } RPC\_V2\_NOTIFY\_INFO\_DATA\_DATA;

**String:**  Case **TABLE\_STRING**: This member specifies a [STRING\_CONTAINER structure (section 2.2.1.2.15)](#Section_5f6aaa8dad83427892fa2afaaf8b75fc).

**dwData:**  Case **TABLE\_DWORD**: This member specifies an array of two **DWORD** values that contain the member's current data.

**SystemTime:**  Case **TABLE\_TIME** This member specifies a [SYSTEMTIME\_CONTAINER structure (section 2.2.1.2.16)](#Section_1438066b4b56488898213e2432dc0f07).

**DevMode:**  Case **TABLE\_DEVMODE**: This member specifies a [DEVMODE\_CONTAINER structure (section 2.2.1.2.1)](#Section_c2b14fe764794460a7ba633a845dd61a) that defines default printer attributes such as the paper orientation and printing resolution.

**SecurityDescriptor:**  Case **TABLE\_SECURITYDESCRIPTOR**: This member specifies a [SECURITY\_CONTAINER structure (section 2.2.1.2.13)](#Section_6cf2b4ff1b2b453c87f985243823cea8), in which the **pSecurity** member is a pointer to a SECURITY\_DESCRIPTOR structure ([[MS-DTYP]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-DTYP%5d.pdf) section 2.4.6) in self-relative form.

##### RPC\_V2\_UREPLY\_PRINTER

The RPC\_V2\_UREPLY\_PRINTER union defines printer notification responses.[<90>](#Appendix_A_90)

1. typedef
2. [switch\_type(DWORD)]
3. union \_RPC\_V2\_UREPLY\_PRINTER {
4. [case(0x00000000)]
5. RPC\_V2\_NOTIFY\_INFO\* pInfo;
6. } RPC\_V2\_UREPLY\_PRINTER;

**pInfo:**  This member MUST be a non-NULL pointer to an [RPC\_V2\_NOTIFY\_INFO](#Section_79db335febe74a6f8533efdda32cc5d8) structure, which MUST contain notification information.

#### Job Named Properties

##### RPC\_PrintPropertyValue

The **RPC\_PrintPropertyValue** structure specifies the value of a Job Named Property (section [3.1.1](#Section_1bd61448db4b4284a4b18346732402d4)).[<91>](#Appendix_A_91)

1. typedef struct RPC\_EPrintPropertyType {
2. ePropertyType;
3. [switch\_is(ePropertyType)] union {
4. [case(kRpcPropertyTypeString), string]
5. wchar\_t\* propertyString;
6. [case(kRpcPropertyTypeInt32)] LONG propertyInt32;
7. [case(kRpcPropertyTypeInt64)] LONGLONG propertyInt64;
8. [case(kRpcPropertyTypeByte)] BYTE propertyByte;
9. [case(kRpcPropertyTypeBuffer] struct {
10. DWORD cbBuf;
11. [size\_is(cbBuf)] BYTE\* pBuf;
12. } propertyBlob;
13. } value;
14. } RPC\_PrintPropertyValue;

**ePropertyType:**  The type of the value. All enumeration values described in section [2.2.1.14.3](#Section_0360160c35514c5bbcee9164122d34e2) are valid.

**propertyString:**  A pointer to a string containing the property value. Valid only if **ePropertyType** is set to **kRpcPropertyTypeString**.

**propertyInt32:**  The property value as a signed 32-bit integer. Valid only if **ePropertyType** is set to **kRpcPropertyTypeInt32**.

**propertyInt64:**  The property value as a signed 64-bit integer. Valid only if **ePropertyType** is set to **kRpcPropertyTypeInt64**.

**propertyByte:**  The property value as a byte. Valid only if **ePropertyType** is set to kRpcPropertyTypeByte.

**propertyBlob:**  An embedded structure that describes the buffer containing the property value as an array of bytes. Valid only if **ePropertyType** is set to **kRpcPropertyTypeBuffer**.

**cbBuf:**  Member of the **propertyBlob** structure that specifies the length, in bytes, of the property value contained in the **pBuf** buffer when **ePropertyType** is set to **kRpcPropertyTypeBuffer**.

**pBuf:**  Member of the **propertyBlob** structure that contains a pointer to the buffer containing the property value when **ePropertyType** is set to **kRpcPropertyTypeBuffer**.

##### RPC\_PrintNamedProperty

The **RPC\_PrintNamedProperty** structure specifies a Job Named Property (section [3.1.1](#Section_1bd61448db4b4284a4b18346732402d4)).[<92>](#Appendix_A_92)

1. typedef struct {
2. [string] wchar\_t\* propertyName;
3. RPC\_PrintPropertyValue propertyValue;
4. } RPC\_PrintNamedProperty;

**propertyName:**  A pointer to a string containing the name of the property.

**propertyValue:**  An [RPC\_PrintPropertyValue structure (section 2.2.1.14.1)](#Section_2ed8404520264696b2e263dbffdc6d21) containing the value of the property.

##### RPC\_EPrintPropertyType

The **RPC\_EPrintPropertyType** enumeration specifies the type of the value contained by a Job Named Property (section [3.1.1](#Section_1bd61448db4b4284a4b18346732402d4)).[<93>](#Appendix_A_93)

1. typedef enum
2. {
3. kRpcPropertyTypeString = 1,
4. kRpcPropertyTypeInt32,
5. kRpcPropertyTypeInt64,
6. kRpcPropertyTypeByte,
7. kRpcPropertyTypeBuffer
8. } RPC\_EPrintPropertyType;

**kRpcPropertyTypeString:** The property value is a string.

**kRpcPropertyTypeInt32:** The property value is a signed 32-bit integer.

**kRpcPropertyTypeInt64:** The property value is a signed 64-bit integer.

**kRpcPropertyTypeByte:** The property value is a byte.

**kRpcPropertyTypeBuffer:** The property value consists of an array of bytes contained in a buffer.

##### SPLFILE\_CONTENT\_TYPE\_PROP\_NAME

The SPL\_CONTENT\_TYPE\_PROP\_NAME constant defines the name of a standard **Job Named Property** (section [3.1.1](#Section_1bd61448db4b4284a4b18346732402d4)) that specifies the spool file format for the print data that is sent by the client.[<94>](#Appendix_A_94)

1. #define SPLFILE\_CONTENT\_TYPE\_PROP\_NAME L"Spool File Contents"

The following are the valid string values for this property.

| Property Value | Meaning |
| --- | --- |
| L"TYPE\_XPS\_MS" | Microsoft XML Paper Specification (XPS) format. For more information, see [[MSFT-XMLPAPER]](http://go.microsoft.com/fwlink/?LinkId=90214). |
| L"TYPE\_XPS\_OPEN" | [**OpenXPS**](#gt_14f30cc4-fd62-46b3-ad42-c0203b61a1b4) format. For more information, see [[ECMA-388]](http://go.microsoft.com/fwlink/?LinkId=158846). |
| L"TYPE\_PDL\_POSTSCRIPT" | Page description language (PDL) [**PostScript**](#gt_31971c63-f974-47de-8de9-6b76d093890d) format. |
| L"SPLFILE\_CONTENT\_TYPE\_PDL\_UNKNOWN" | Unknown PDL format. |

Print servers are not required to support this property. Print clients are not required to support or set this property. If a print client does not set this property on a print job that the client submits to a print server, but the print server supports this property, the print server SHOULD by default select the value L"TYPE\_XPS\_MS".

#### Branch Office Print Remote Logging Structures

##### EBranchOfficeJobEventType

The EBranchOfficeJobEventType enumeration specifies the type of [**Windows Event**](#gt_b9456a1c-aee2-41aa-af78-a083db7a27c7) contained by a **Branch Office Print Remote Log Entry** (section [3.1.1](#Section_1bd61448db4b4284a4b18346732402d4)).[<95>](#Appendix_A_95)

1. typedef enum
2. {
3. kInvalidJobState = 0,
4. kLogJobPrinted,
5. kLogJobRendered,
6. kLogJobError,
7. kLogJobPipelineError,
8. kLogOfflineFileFull
9. } EBranchOfficeJobEventType;

**kInvalidJobState:** The Windows Event is an unknown type.

**kLogJobPrinted:** The **Branch Office Print Remote Log Entry** contains a Windows Event that corresponds to [**event ID**](#gt_0c6539af-aed7-4baa-9d21-4aa5b41d448a) 307.

**kLogJobRendered:** The Branch Office Print Remote Log Entry contains a Windows Event that corresponds to event ID 805.

**kLogJobError:** The Branch Office Print Remote Log Entry contains a Windows Event that corresponds to event ID 372.

**kLogJobPipelineError:** The Branch Office Print Remote Log Entry contains a Windows Event that corresponds to event ID 824.

**kLogOfflineFileFull:** The Branch Office Print Remote Log Entry contains a Windows Event that corresponds to event ID 868.

##### RPC\_BranchOfficeJobData

The RPC\_BranchOfficeJobData structure holds a [**branch office print remote logging**](#gt_9b511320-74d4-4c21-9ee1-935a6476d492) structure that contains the data required to log a single **Branch Office Print Remote Log Entry** (section [3.1.1](#Section_1bd61448db4b4284a4b18346732402d4)) corresponding to a specific type of Windows Event.[<96>](#Appendix_A_96)

1. typedef struct {
2. EBranchOfficeJobEventType eEventType;
3. DWORD JobId;
4. [switch\_type(EBranchOfficeJobEventType), switch\_is(eEventType)]
5. union {
6. [case(kLogJobPrinted)]
7. RPC\_BranchOfficeJobDataPrinted LogJobPrinted;
8. [case(kLogJobRendered)]
9. RPC\_BranchOfficeJobDataRendered LogJobRendered;
10. [case(kLogJobError)]
11. RPC\_BranchOfficeJobDataError LogJobError;
12. [case(kLogJobPipelineError)]
13. RPC\_BranchOfficeJobDataPipelineFailed LogJobPipelineFailed;
14. [case(kLogOfflineFileFull)]
15. RPC\_BranchOfficeLogOfflineFileFull LogOfflineFileFull;
16. } JobInfo;
17. } RPC\_BranchOfficeJobData;

**eEventType:**  The type of Windows Event to which the **Branch Office Print Remote Log Entry** corresponds, which MUST be an [EBranchOfficeJobEventType (section 2.2.1.15.1)](#Section_d070101180b3473d843cedf7e4c5bc86) value.

**JobId:**  The identifier of a print job.

**JobInfo:**  The branch office print remote logging structure that contains the data required to log a **Branch Office Print Remote Log Entry** corresponding to the **eEventType** member value.

**LogJobPrinted:**  An [RPC\_BranchOfficeJobDataPrinted (section 2.2.1.15.5)](#Section_b4a220487837460eb1bd8d34f4270f29) structure for a **Branch Office Print Remote Log Entry** corresponding to event ID 307. This structure is present only if **eEventType** is set to **kLogJobPrinted**.

**LogJobRendered:**  An [RPC\_BranchOfficeJobDataRendered (section 2.2.1.15.6)](#Section_e90ee32f12f44310923871aca7dd9cb1) structure for a **Branch Office Print Remote Log Entry** corresponding to event ID 805. This structure is present only if **eEventType** is set to **kLogJobRendered**.

**LogJobError:**  An [RPC\_BranchOfficeJobDataError (section 2.2.1.15.3)](#Section_d86fd29b95124c80a14b5f5fb25e5975) structure for a **Branch Office Print Remote Log Entry** corresponding to event ID 372. This structure is present only if **eEventType** is set to **kLogJobError**.

**LogJobPipelineFailed:**  An [RPC\_BranchOfficeJobDataPipelineFailed (section 2.2.1.15.4)](#Section_31a2aecb2d7a4846a93be4fd708863e0) structure for a **Branch Office Print Remote Log Entry** corresponding to event ID 824. This structure is present only if **eEventType** is set to **kLogJobPipelineError**.

**LogOfflineFileFull:**  An [RPC\_BranchOfficeLogOfflineFileFull (section 2.2.1.15.7)](#Section_55fcc7c1d63e448888cd8ce5d9dd945c) structure for a **Branch Office Print Remote Log Entry** corresponding to event ID 868. This structure is present only if **eEventType** is set to **kLogOfflineFileFull**.

##### RPC\_BranchOfficeJobDataError

The RPC\_BranchOfficeJobDataError structure holds a single **Branch Office Remote Logging Entry** (section [3.1.1](#Section_1bd61448db4b4284a4b18346732402d4)).[<97>](#Appendix_A_97) This entry contains the information needed to create event ID 372 in the [**Microsoft-Windows-PrintService**](#gt_e6ccb1c9-7b3e-473c-9e14-0806d19c4337)/Admin [**event channel**](#gt_43907602-849d-4d1c-9ff8-a2c3228b3735).

1. typedef struct {
2. DWORD LastError;
3. [string] wchar\_t\* pDocumentName;
4. [string] wchar\_t\* pUserName;
5. [string] wchar\_t\* pPrinterName;
6. [string] wchar\_t\* pDataType;
7. LONGLONG TotalSize;
8. LONGLONG PrintedSize;
9. DWORD TotalPages;
10. DWORD PrintedPages;
11. [string] wchar\_t\* pMachineName;
12. [string] wchar\_t\* pJobError;
13. [string] wchar\_t\* pErrorDescription;
14. } RPC\_BranchOfficeJobDataError;

**LastError:**  A 32-bit unsigned integer that specifies an implementation-specific error code for the last error that occurred during processing of this print job.

**pDocumentName:**  A pointer to a string that specifies the name of the print document for this print job.

**pUserName:**  A pointer to a string that specifies the name of the user that owns the print job. For rules governing user names, see section [2.2.4.17](#Section_9b19bbdf64da407fa5c6c6533a041d19).

**pPrinterName:**  A pointer to a string that specifies the name of the printer used for the print job. For rules governing printer names, see section [2.2.4.14](#Section_24fcd124035c4988a8583a7d8d6f7b43).

**pDataType:**  A pointer to a string that specifies the type of data that the printing application sent to the printer in the print job. The identified data type MUST be supported by the print processor that is associated with the printer that is processing the job. For rules governing data type names, see section [2.2.4.2](#Section_f40cbad088dd4e86a16bcf62c3b17c52).

**TotalSize:**  A 64-bit signed integer that specifies the size of the print job, in bytes. This value MUST be greater than zero.

**PrintedSize:**  A 64-bit signed integer that specifies the amount of data for the print job that actually got processed and sent to the printer, in bytes. This value MUST be zero or greater.

**TotalPages:**  A 32-bit unsigned integer that specifies the number of pages the document contains.

**PrintedPages:**  A 32-bit unsigned integer that specifies the number of pages of the document that actually got processed and sent to the printer.

**pMachineName:**  A pointer to a string that specifies the name of the client computer that owns the print job. For rules governing computer names, see section [2.2.4.16](#Section_9b3f813570224b72accbaefcc360c83b).

**pJobError:**  A pointer to a string that specifies the text representation of the value of the **LastError** error code.

**pErrorDescription:**  A pointer to an optional string that specifies message text for a system-defined error corresponding to the value of the **LastError** error code.

##### RPC\_BranchOfficeJobDataPipelineFailed

The RPC\_BranchOfficeJobDataPipelineFailed structure holds a single **Branch Office Print Remote Log Entry** (section [3.1.1](#Section_1bd61448db4b4284a4b18346732402d4)).[<98>](#Appendix_A_98) This entry contains the information needed to create event ID 824 in the Microsoft-Windows-PrintService/Operational event channel.

1. typedef struct {
2. [string] wchar\_t\* pDocumentName;
3. [string] wchar\_t\* pPrinterName;
4. [string] wchar\_t\* pExtraErrorInfo;
5. } RPC\_BranchOfficeJobDataPipelineFailed;

**pDocumentName:**  A pointer to a string that specifies the name of the print document for this print job.

**pPrinterName:**  A pointer to a string that specifies the name of the printer used for the print job. For rules governing printer names, see section [2.2.4.14](#Section_24fcd124035c4988a8583a7d8d6f7b43).

**pExtraErrorInfo:**  A pointer to an optional string that specifies additional text associated with the failure in the [**Print Pipeline**](#gt_0cf6559d-e22e-4d3d-a62d-7d7a0b021c37).

##### RPC\_BranchOfficeJobDataPrinted

The RPC\_BranchOfficeJobDataPrinted structure holds a single **Branch Office Print Remote Log Entry** (section [3.1.1](#Section_1bd61448db4b4284a4b18346732402d4)).[<99>](#Appendix_A_99) This entry contains the information needed to create event ID 307 in the Microsoft-Windows-PrintService/Operational event channel.

1. typedef struct {
2. DWORD Status;
3. [string] wchar\_t\* pDocumentName;
4. [string] wchar\_t\* pUserName;
5. [string] wchar\_t\* pMachineName;
6. [string] wchar\_t\* pPrinterName;
7. [string] wchar\_t\* pPortName;
8. LONGLONG Size;
9. DWORD TotalPages;
10. } RPC\_BranchOfficeJobDataPrinted;

**Status:**  A 32-bit unsigned integer that specifies an implementation-specific error code for the last error that occurred during the processing of this print job.

**pDocumentName:**  A string that specifies the name of the print document for this print job.

**pUserName:**  A pointer to a string that specifies the name of the user that owns the print job. For rules governing user names, see section [2.2.4.17](#Section_9b19bbdf64da407fa5c6c6533a041d19).

**pMachineName:**  A pointer to a string that specifies the name of the client computer that owns the print job. For rules governing computer names, see section [2.2.4.16](#Section_9b3f813570224b72accbaefcc360c83b).

**pPrinterName:**  A pointer to a string that specifies the name of the printer used for the print job. For rules governing printer names, see section [2.2.4.14](#Section_24fcd124035c4988a8583a7d8d6f7b43).

**pPortName:**  A pointer to a string that specifies a supported printer port. For rules governing port names, see section [2.2.4.10](#Section_03c5feb74fbb4569b0649eb6ad833f50).

**Size:**  A 64-bit signed integer that specifies the size of the print job, in bytes.

**TotalPages:**  A 32-bit unsigned integer that specifies the number of pages the document contains.

##### RPC\_BranchOfficeJobDataRendered

The RPC\_BranchOfficeJobDataRendered structure holds a single **Branch Office Print Remote Log Entry** (section [3.1.1](#Section_1bd61448db4b4284a4b18346732402d4)).[<100>](#Appendix_A_100) This entry contains the information needed to create event ID 805 in the Microsoft-Windows-PrintService/Operational event channel.

1. typedef struct {
2. LONGLONG Size;
3. DWORD ICMMethod;
4. short Color;
5. short PrintQuality;
6. short YResolution;
7. short Copies;
8. short TTOption;
9. } RPC\_BranchOfficeJobDataRendered;

**Size:**  A 64-bit signed integer that specifies the size of the print job, in bytes.

**ICMMethod:**  A 32-bit unsigned integer that specifies how [**Image Color Management (ICM)**](#gt_fcd5ee55-57ff-45d6-b51e-2ccba47329e3) is handled for the print job. See the definition of the **dmICMMethod** field in section [2.2.2.1](#Section_ccc2a501794e4d2bb312f69c75131c2e).

**Color:**  A 16-bit signed integer that specifies the color mode to use for the print job. See the definition of the **dmColor** field in section 2.2.2.1.

**PrintQuality:**  A 16-bit signed integer that specifies the printer resolution for the print job. See the definition of the **dmPrintQuality** field in section 2.2.2.1.

**YResolution:**  A 16-bit signed integer that specifies the vertical-resolution in dots per inch for the print job. See the definition of the **dmYResolution** field in section 2.2.2.1.

**Copies:**  A 16-bit signed integer that specifies the number of copies to be printed for the print job. See the definition of the **dmCopies** field in section 2.2.2.1.

**TTOption:**  A 16-bit signed integer that specifies how TrueType fonts are printed for the print job. See the definition of the **dmTTOption** field in section 2.2.2.1.

##### RPC\_BranchOfficeLogOfflineFileFull

The RPC\_BranchOfficeLogOfflineFileFull structure holds a single **Branch Office Print Remote Log Entry** (section [3.1.1](#Section_1bd61448db4b4284a4b18346732402d4))[<101>](#Appendix_A_101) that indicates the **Branch Office Print Remote Log Offline Archive** (section [3.2.1](#Section_fa5a51d585cc40dc951a1e288b62d820)) has exceeded the maximum allowed size.[<102>](#Appendix_A_102) This entry contains the information needed to create event ID 868 in the Microsoft-Windows-PrintService/Admin event channel.

1. typedef struct {
2. [string] wchar\_t\* pMachineName;
3. } RPC\_BranchOfficeLogOfflineFileFull;

**pMachineName:**  A pointer to a string that specifies the name of the client computer that encountered the **Branch Office Print Remote Log Offline Archive** full condition. For rules governing computer names, see section [2.2.4.16](#Section_9b3f813570224b72accbaefcc360c83b).

### Custom-Marshaled Data Types

This section specifies data structures that are custom-marshaled, including those that contain "\_INFO" in their names. All custom-marshaled **INFO** data structures MUST be completely ignored on input, and validation of their contents MUST NOT take place.

Custom-marshaled **INFO** data structures consist of single **Fixed\_Portion** blocks for methods accepting or returning a single structure, and arrays of one or more **Fixed\_Portion** blocks for methods accepting or returning an array of structures. The size of the **Fixed\_Portion** data is the size of a single **Fixed\_Portion** block multiplied by the number of **Fixed\_Portion** blocks returned. The **Fixed\_Portion** data is followed by a single **Variable\_Data** block, which contains variable-length fields. The size of the **Variable\_Data** block is the size specified by the caller in the *cbBuf* parameter of the call minus the size of the **Fixed\_Portion** data.

For each field in a **Variable\_Data** block, a corresponding offset value is specified in a field of a **Fixed\_Portion** block. A **Variable\_Data** field is located by adding that offset value to the address of the start of the **Fixed\_Portion** block in which that offset is defined.

This generic structure of custom-marshaled **INFO** data structures is represented by the following diagram.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| Fixed\_Portion (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Variable\_Data (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Fixed\_Portion (variable):** An array of one or more **Fixed\_Portion** blocks, each consisting of one or more fixed-length fields. The specific structure of the **Fixed\_Portion** block is defined for each **INFO** structure.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| Fixed\_Portion\_Block\_1 (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Fixed\_Portion\_Block\_2 (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Fixed\_Portion\_Block\_1 (variable): Fixed\_Portion** block 1.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| Fixed\_Portion\_Block\_1\_Field\_1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Fixed\_Portion\_Block\_1\_Field\_2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Fixed\_Portion\_Block\_1\_Field\_1 (4 bytes):** Fixed-length field 1 of **Fixed\_Portion** block 1. Although the length of this field is shown as 4 bytes, its actual length is indeterminate in this generic structure.

**Fixed\_Portion\_Block\_1\_Field\_2 (4 bytes):** Fixed-length field 2 of **Fixed\_Portion** block 1. This field contains an offset to **Variable\_Data\_Field\_1**, which is relative to the start of **Fixed\_Portion** block 1.

**Fixed\_Portion\_Block\_2 (variable):** Optional **Fixed\_Portion** block 2.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| Fixed\_Portion\_Block\_2\_Field\_1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Fixed\_Portion\_Block\_2\_Field\_2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Fixed\_Portion\_Block\_2\_Field\_1 (4 bytes):** Fixed-length field 1 of **Fixed\_Portion** block 2. Although the length of this field is shown as 4 bytes, its actual length is indeterminate in this generic structure.

**Fixed\_Portion\_Block\_2\_Field\_2 (4 bytes):** Fixed-length field 2 of **Fixed\_Portion** block 2. This field contains an offset to **Variable\_Data\_Field\_2**, which is relative to the start of **Fixed\_Portion** block 2.

**Variable\_Data (variable):** A data block of variable length. Because the data is not necessarily aligned on 16-bit boundaries, it is specified as an array of bytes of arbitrary length; however, data fields in the **Variable\_Data** block MUST be aligned on natural boundaries matching their data type. That is, WCHAR fields MUST be aligned on 2-byte boundaries, DWORD fields MUST be aligned on 4-byte boundaries, and so on.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| Variable\_Data\_Field\_1 (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Variable\_Data\_Field\_2 (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Variable\_Data\_Field\_1 (variable):** Variable-length field 1 of the **Variable\_Data** block.

**Variable\_Data\_Field\_2 (variable):** Variable-length field 2 of the **Variable\_Data** block.

The following characteristics apply to the fields in custom-marshaled **INFO** data structures:

* The start of the **Fixed\_Portion** block MUST be 32-bit aligned.
* The order of fields in the **Fixed\_Portion** block is defined by the specific **INFO** structure layout.
* Data fields in the **Variable\_Data** block can appear in arbitrary order.
* One or more offsets in **Fixed\_Portion** blocks can locate the same field in the **Variable\_Data** block; or there can be a one-to-one correspondence between offsets and **Variable\_Data** fields.
* The **Variable\_Data** fields SHOULD be packed tightly in the **Variable\_Data** block, filling the **Variable\_Data** block from the end toward the beginning, such that, if the *cbBuf* parameter specified by the caller is larger than the sum of all **Fixed\_Portion** blocks and all **Variable\_Data** fields, the unused space in the **[out]** buffer receiving the custom-marshaled **INFO** structure will form a gap between the end of the last **Fixed\_Portion** block and the beginning of the first **Variable\_Data** field; however, client-side unmarshaling code that processes a custom-marshaled **INFO** structure SHOULD be prepared to correctly handle data that does not fill the **Variable\_Data** block from the end toward the beginning, or is not tightly packed and includes unused space in arbitrary positions of the **Variable\_Data** block.[<103>](#Appendix_A_103)

#### \_DEVMODE

The \_DEVMODE structure defines initialization data for a printer. Although the \_DEVMODE structure does not contain any pointers, it is still custom-marshaled, because the size of the structure is version-specific and implementation-specific, and cannot be expressed using IDL attributes. It has the following message format.

The print server MUST accept \_DEVMODE structures with truncated public information. A truncated \_DEVMODE structure contains a subset of fields, from **dmDeviceName** up to and including **dmFields**, plus at least those fields that are initialized as specified by **dmFields**. The truncated \_DEVMODE structure is followed by private, printer driver-specific data, the size of which is specified by the value of the **dmDriverExtra** field.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| dmDeviceName (64 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dmSpecVersion | | | | | | | | | | | | | | | | dmDriverVersion | | | | | | | | | | | | | | | |
| dmSize | | | | | | | | | | | | | | | | dmDriverExtra | | | | | | | | | | | | | | | |
| dmFields | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dmOrientation | | | | | | | | | | | | | | | | dmPaperSize | | | | | | | | | | | | | | | |
| dmPaperLength | | | | | | | | | | | | | | | | dmPaperWidth | | | | | | | | | | | | | | | |
| dmScale | | | | | | | | | | | | | | | | dmCopies | | | | | | | | | | | | | | | |
| dmDefaultSource | | | | | | | | | | | | | | | | dmPrintQuality | | | | | | | | | | | | | | | |
| dmColor | | | | | | | | | | | | | | | | dmDuplex | | | | | | | | | | | | | | | |
| dmYResolution | | | | | | | | | | | | | | | | dmTTOption | | | | | | | | | | | | | | | |
| dmCollate | | | | | | | | | | | | | | | | dmFormName (64 bytes) | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | reserved0 | | | | | | | | | | | | | | | |
| reserved1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| reserved2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| reserved3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dmNup | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| reserved4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dmICMMethod | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dmICMIntent | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dmMediaType | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dmDitherType | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| reserved5 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| reserved6 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| reserved7 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| reserved8 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dmDriverExtraData (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**dmDeviceName (64 bytes):** A 32-element array of 16-bit Unicode characters that form a null-terminated string that specifies the name of the printer. Printer name strings that are longer than 32 characters MUST be truncated to fit the array. For more rules governing printer names, see section [2.2.4.14](#Section_24fcd124035c4988a8583a7d8d6f7b43).

**dmSpecVersion (2 bytes):** The version of initialization data specification on which the \_DEVMODE structure is based. This value SHOULD be 0x0401.[<104>](#Appendix_A_104)

**dmDriverVersion (2 bytes):** For printers, an optional, implementation-defined version of the printer driver.[<105>](#Appendix_A_105)

**dmSize (2 bytes):** The size, in bytes, of the \_DEVMODE structure, which MUST be a multiple of 4 bytes. This value MUST NOT include the length of any private, printer driver–specific data that might follow the \_DEVMODE structure's public fields. The size, in bytes, of such private data is specified by the **dmDriverExtra** value.

If the \_DEVMODE structure contains truncated public information, the value of **dmSize** is at least the size, in bytes, of a subset of fields, from **dmDeviceName** up to and including **dmFields**, plus fields that are initialized as specified by **dmFields**.

**dmDriverExtra (2 bytes):** The size, in bytes, of the private printer drivers data that follows this structure.

**dmFields (4 bytes):** A bitfield that specifies the fields of the \_DEVMODE structure that have been initialized. If a bit is set, the corresponding field MUST be initialized and MUST be processed on receipt. If a bit is not set, the value of the corresponding field SHOULD be set to zero when sent and MUST be ignored on receipt.

The value of this field is the result of a bitwise OR of the following bits.

| 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 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | U P | 0 | S C | P W | P L | P S | O R | C L | T T | Y | D X | C R | P Q | D S | C P | C M | 0 | 0 | 0 | 0 | 0 | 0 | F M | 0 | 0 | 0 | 0 | 0 | D T | M T | C I |

| Value | Description |
| --- | --- |
| CI  DM\_ICMINTENT | If this bit is set, the **dmICMIntent** field MUST be initialized. |
| MT  DM\_MEDIATYPE | If this bit is set, the **dmMediaType** field MUST be initialized. |
| DT  DM\_DITHERTYPE | If this bit is set, the **dmDitherType** field MUST be initialized. |
| FM  DM\_FORMNAME | If this bit is set, the **dmFormName** field MUST be initialized. |
| CM  DM\_ICMMETHOD | If this bit is set, the **dmICMMethod** field MUST be initialized. |
| CP  DM\_COPIES | If this bit is set, the **dmCopies** field MUST be initialized. |
| DS  DM\_DEFAULTSOURCE | If this bit is set, the **dmDefaultSource** field MUST be initialized. |
| PQ  DM\_PRINTQUALITY | If this bit is set, the **dmPrintQuality** field MUST be initialized. |
| CR  DM\_COLOR | If this bit is set, the **dmColor** field MUST be initialized. |
| DX  DM\_DUPLEX | If this bit is set, the **dmDuplex** field MUST be initialized. |
| Y  DM\_YRESOLUTION | If this bit is set, the **dmYResolution** field MUST be initialized. |
| TT  DM\_TTOPTION | If this bit is set, the **dmTTOption** field MUST be initialized. |
| CL  DM\_COLLATE | If this bit is set, the **dmCollate** field MUST be initialized. |
| OR  DM\_ORIENTATION | If this bit is set, the **dmOrientation** field MUST be initialized. |
| PS  DM\_PAPERSIZE | If this bit is set, the **dmPaperSize** field MUST be initialized. This bit MUST NOT be set if either DM\_PAPERLENGTH or DM\_PAPERWIDTH are set. |
| PL  DM\_PAPERLENGTH | If this bit is set, the **dmPaperLength** field MUST be initialized. This bit MUST NOT be set if DM\_PAPERSIZE is set. |
| PW  DM\_PAPERWIDTH | If this bit is set, the **dmPaperWidth** field MUST be initialized. This bit MUST NOT be set if DM\_PAPERSIZE is set. |
| SC  DM\_SCALE | If this bit is set, the **dmScale** field MUST be initialized. |
| UP  DM\_NUP | If this bit is set, the **dmNup** field MUST be initialized. |

**dmOrientation (2 bytes):** For printers, the orientation for output. If the DM\_ORIENTATION bit is set in **dmFields**, this value MUST be specified. The value of this field SHOULD be one of the following.

| Name/value | Meaning |
| --- | --- |
| DMORIENT\_POTRAIT  0x0001 | "Portrait" orientation. |
| DMORIENT\_LANDSCAPE  0x0002 | "Landscape" orientation. |

**dmPaperSize (2 bytes):** For printers, the size of the output media. If the DM\_PAPERSIZE bit is set in **dmFields**, this value MUST be specified. The value of this field SHOULD be one of the following, or it MAY be a device-specific value that is greater than or equal to 0x0100.

| Name/value | Meaning |
| --- | --- |
| DMPAPER\_LETTER  0x0001 | Letter, 8 1/2 x 11 inches |
| DMPAPER\_LEGAL  0x0005 | Legal, 8 1/2 x 14 inches |
| DMPAPER\_10X14  0x0010 | 10 x 14-inch sheet |
| DMPAPER\_11X17  0x0011 | 11 x 17-inch sheet |
| DMPAPER\_12X11  0x005A | 12 x 11-inch sheet[<106>](#Appendix_A_106) |
| DMPAPER\_A3  0x0008 | A3 sheet, 297 x 420 millimeters |
| DMPAPER\_A3\_ROTATED  0x004C | A3 rotated sheet, 420 x 297 millimeters[<107>](#Appendix_A_107) |
| DMPAPER\_A4  0x0009 | A4 sheet, 210 x 297 millimeters |
| DMPAPER\_A4\_ROTATED  0x004D | A4 rotated sheet, 297 x 210 millimeters[<108>](#Appendix_A_108) |
| DMPAPER\_A4SMALL  0x000A | A4 small sheet, 210 x 297 millimeters |
| DMPAPER\_A5  0x000B | A5 sheet, 148 x 210 millimeters |
| DMPAPER\_A5\_ROTATED  0x004E | A5 rotated sheet, 210 x 148 millimeters[<109>](#Appendix_A_109) |
| DMPAPER\_A6  0x0046 | A6 sheet, 105 x 148 millimeters[<110>](#Appendix_A_110) |
| DMPAPER\_A6\_ROTATED  0x0053 | A6 rotated sheet, 148 x 105 millimeters[<111>](#Appendix_A_111) |
| DMPAPER\_B4  0x000C | B4 sheet, 250 x 354 millimeters |
| DMPAPER\_B4\_JIS\_ROTATED  0x004F | B4 (JIS) rotated sheet, 364 x 257 millimeters[<112>](#Appendix_A_112) |
| DMPAPER\_B5  0x000D | B5 sheet, 182 x 257-millimeter paper |
| DMPAPER\_B5\_JIS\_ROTATED  0x0050 | B5 (JIS) rotated sheet, 257 x 182 millimeters[<113>](#Appendix_A_113) |
| DMPAPER\_B6\_JIS  0x0058 | B6 (JIS) sheet, 128 x 182 millimeters[<114>](#Appendix_A_114) |
| DMPAPER\_B6\_JIS\_ROTATED  0x0059 | B6 (JIS) rotated sheet, 182 x 128 millimeters[<115>](#Appendix_A_115) |
| DMPAPER\_CSHEET  0x0018 | C Sheet, 17 x 22 inches |
| DMPAPER\_DBL\_JAPANESE\_POSTCARD  0x0045 | Double Japanese Postcard, 200 x 148 millimeters[<116>](#Appendix_A_116) |
| DMPAPER\_DBL\_JAPANESE\_POSTCARD\_ROTATED  0x0052 | Double Japanese Postcard Rotated, 148 x 200 millimeters[<117>](#Appendix_A_117) |
| DMPAPER\_DSHEET  0x0019 | D Sheet, 22 x 34 inches |
| DMPAPER\_ENV\_9  0x0013 | #9 Envelope, 3 7/8 x 8 7/8 inches |
| DMPAPER\_ENV\_10  0x0014 | #10 Envelope, 4 1/8 x 9 1/2 inches |
| DMPAPER\_ENV\_11  0x0015 | #11 Envelope, 4 1/2 x 10 3/8 inches |
| DMPAPER\_ENV\_12  0x0016 | #12 Envelope, 4 3/4 x 11 inches |
| DMPAPER\_ENV\_14  0x0017 | #14 Envelope, 5 x 11 1/2 inches |
| DMPAPER\_ENV\_C5  0x001C | C5 Envelope, 162 x 229 millimeters |
| DMPAPER\_ENV\_C3  0x001D | C3 Envelope, 324 x 458 millimeters |
| DMPAPER\_ENV\_C4  0x001E | C4 Envelope, 229 x 324 millimeters |
| DMPAPER\_ENV\_C6  0x001F | C6 Envelope, 114 x 162 millimeters |
| DMPAPER\_ENV\_C65  0x0020 | C65 Envelope, 114 x 229 millimeters |
| DMPAPER\_ENV\_B4  0x0021 | B4 Envelope, 250 x 353 millimeters |
| DMPAPER\_ENV\_B5  0x0022 | B5 Envelope, 176 x 250 millimeters |
| DMPAPER\_ENV\_B6  0x0023 | B6 Envelope, 176 x 125 millimeters |
| DMPAPER\_ENV\_DL  0x001B | DL Envelope, 110 x 220 millimeters |
| DMPAPER\_ENV\_ITALY  0x0024 | Italy Envelope, 110 x 230 millimeters |
| DMPAPER\_ENV\_MONARCH  0x0025 | Monarch Envelope, 3 7/8 x 7 1/2 inches |
| DMPAPER\_ENV\_PERSONAL  0x0026 | 6 3/4 Envelope, 3 5/8 x 6 1/2 inches |
| DMPAPER\_ESHEET  0x001A | E Sheet, 34 x 44 inches |
| DMPAPER\_EXECUTIVE  0x0007 | Executive, 7 1/4 x 10 1/2 inches |
| DMPAPER\_FANFOLD\_US  0x0027 | US Std Fanfold, 14 7/8 x 11 inches |
| DMPAPER\_FANFOLD\_STD\_GERMAN  0x0028 | German Std Fanfold, 8 1/2 x 12 inches |
| DMPAPER\_FANFOLD\_LGL\_GERMAN  0x0029 | German Legal Fanfold, 8 x 13 inches |
| DMPAPER\_FOLIO  0x000E | Folio, 8 1/2 x 13-inch paper |
| DMPAPER\_JAPANESE\_POSTCARD\_ROTATED  0x0051 | Japanese Postcard Rotated, 148 x 100 millimeters[<118>](#Appendix_A_118) |
| DMPAPER\_JENV\_CHOU3  0x0049 | Japanese Envelope Chou #3[<119>](#Appendix_A_119) |
| DMPAPER\_JENV\_CHOU3\_ROTATED  0x0056 | Japanese Envelope Chou #3 Rotated[<120>](#Appendix_A_120) |
| DMPAPER\_JENV\_CHOU4  0x004A | Japanese Envelope Chou #4[<121>](#Appendix_A_121) |
| DMPAPER\_JENV\_CHOU4\_ROTATED  0x0057 | Japanese Envelope Chou #4 Rotated[<122>](#Appendix_A_122) |
| DMPAPER\_JENV\_KAKU2  0x0047 | Japanese Envelope Kaku #2[<123>](#Appendix_A_123) |
| DMPAPER\_JENV\_KAKU2\_ROTATED  0x0054 | Japanese Envelope Kaku #2 Rotated[<124>](#Appendix_A_124) |
| DMPAPER\_JENV\_KAKU3  0x0048 | Japanese Envelope Kaku #3[<125>](#Appendix_A_125) |
| DMPAPER\_JENV\_KAKU3\_ROTATED  0x0055 | Japanese Envelope Kaku #3 Rotated[<126>](#Appendix_A_126) |
| DMPAPER\_JENV\_YOU4  0x005B | Japanese Envelope You #4[<127>](#Appendix_A_127) |
| DMPAPER\_JENV\_YOU4\_ROTATED  0x005C | Japanese Envelope You #4[<128>](#Appendix_A_128) |
| DMPAPER\_LEDGER  0x0004 | Ledger, 17 x 11 inches |
| DMPAPER\_LETTER\_ROTATED  0x004B | Letter Rotated, 11 by 8 1/2 inches |
| DMPAPER\_LETTERSMALL  0x0002 | Letter Small, 8 1/2 x 11 inches |
| DMPAPER\_NOTE  0x0012 | Note, 8 1/2 x 11-inches |
| DMPAPER\_P16K  0x005D | PRC 16K, 146 x 215 millimeters[<129>](#Appendix_A_129) |
| DMPAPER\_P16K\_ROTATED  0x006A | PRC 16K Rotated, 215 x 146 millimeters[<130>](#Appendix_A_130) |
| DMPAPER\_P32K  0x005E | PRC 32K, 97 x 151 millimeters[<131>](#Appendix_A_131) |
| DMPAPER\_P32K\_ROTATED  0x006B | PRC 32K Rotated, 151 x 97 millimeters[<132>](#Appendix_A_132) |
| DMPAPER\_P32KBIG  0x005F | PRC 32K(Big) 97 x 151 millimeters[<133>](#Appendix_A_133) |
| DMPAPER\_P32KBIG\_ROTATED  0x006C | PRC 32K(Big) Rotated, 151 x 97 millimeters[<134>](#Appendix_A_134) |
| DMPAPER\_PENV\_1  0x0060 | PRC Envelope #1, 102 by 165 millimeters[<135>](#Appendix_A_135) |
| DMPAPER\_PENV\_1\_ROTATED  0x006D | PRC Envelope #1 Rotated, 165 x 102 millimeters[<136>](#Appendix_A_136) |
| DMPAPER\_PENV\_2  0x0061 | PRC Envelope #2, 102 x 176 millimeters[<137>](#Appendix_A_137) |
| DMPAPER\_PENV\_2\_ROTATED  0x006E | PRC Envelope #2 Rotated, 176 x 102 millimeters[<138>](#Appendix_A_138) |
| DMPAPER\_PENV\_3  0x0062 | PRC Envelope #3, 125 x 176 millimeters[<139>](#Appendix_A_139) |
| DMPAPER\_PENV\_3\_ROTATED  0x006F | PRC Envelope #3 Rotated, 176 x 125 millimeters[<140>](#Appendix_A_140) |
| DMPAPER\_PENV\_4  0x0063 | PRC Envelope #4, 110 x 208 millimeters[<141>](#Appendix_A_141) |
| DMPAPER\_PENV\_4\_ROTATED  0x0070 | PRC Envelope #4 Rotated, 208 x 110 millimeters[<142>](#Appendix_A_142) |
| DMPAPER\_PENV\_5  0x0064 | PRC Envelope #5, 110 x 220 millimeters[<143>](#Appendix_A_143) |
| DMPAPER\_PENV\_5\_ROTATED  0x0071 | PRC Envelope #5 Rotated, 220 x 110 millimeters[<144>](#Appendix_A_144) |
| DMPAPER\_PENV\_6  0x0065 | PRC Envelope #6, 120 x 230 millimeters[<145>](#Appendix_A_145) |
| DMPAPER\_PENV\_6\_ROTATED  0x0072 | PRC Envelope #6 Rotated, 230 x 120 millimeters[<146>](#Appendix_A_146) |
| DMPAPER\_PENV\_7  0x0066 | PRC Envelope #7, 160 x 230 millimeters[<147>](#Appendix_A_147) |
| DMPAPER\_PENV\_7\_ROTATED  0x0073 | PRC Envelope #7 Rotated, 230 x 160 millimeters[<148>](#Appendix_A_148) |
| DMPAPER\_PENV\_8  0x0067 | PRC Envelope #8, 120 x 309 millimeters[<149>](#Appendix_A_149) |
| DMPAPER\_PENV\_8\_ROTATED  0x0074 | PRC Envelope #8 Rotated, 309 x 120 millimeters[<150>](#Appendix_A_150) |
| DMPAPER\_PENV\_9  0x0068 | PRC Envelope #9, 229 x 324 millimeters[<151>](#Appendix_A_151) |
| DMPAPER\_PENV\_9\_ROTATED  0x0075 | PRC Envelope #9 Rotated, 324 x 229 millimeters[<152>](#Appendix_A_152) |
| DMPAPER\_PENV\_10  0x0069 | PRC Envelope #10, 324 x 458 millimeters[<153>](#Appendix_A_153) |
| DMPAPER\_PENV\_10\_ROTATED  0x0076 | PRC Envelope #10 Rotated, 458 x 324 millimeters[<154>](#Appendix_A_154) |
| DMPAPER\_QUARTO  0x000F | Quarto, 215 x 275 millimeter paper |
| DMPAPER\_STATEMENT  0x0006 | Statement, 5 1/2 x 8 1/2 inches |
| DMPAPER\_TABLOID  0x0003 | Tabloid, 11 x 17 inches |
| 0x0100 ≤ *value* | The value is device-specific. |

**dmPaperLength (2 bytes):** If the DM\_PAPERLENGTH bit is set in the **dmFields** field, the value of this field specifies the length of the paper, in tenths of a millimeter, to use in the printer for which the job is destined.

**dmPaperWidth (2 bytes):** If the DM\_PAPERWIDTH bit is set in the **dmFields** field, the value of this field specifies the width of the paper, in tenths of a millimeter, to use in the printer for which the job is destined.

**dmScale (2 bytes):** If the DM\_SCALE bit is set in the **dmFields** field, the value of this field specifies the percentage factor by which the printed output is to be scaled.

**dmCopies (2 bytes):** If the DM\_COPIES bit is set in the **dmFields** field, the value of this field specifies the number of copies to be printed, if the device supports multiple-page copies.

**dmDefaultSource (2 bytes):** If the DM\_DEFAULTSOURCE bit is set in the **dmFields** field, the value of this field specifies the paper source.

The value of this field SHOULD be one of the following, or it MAY be a device-specific value that is greater than or equal to 0x0100.

| Name/value | Meaning |
| --- | --- |
| DMBIN\_UPPER  0x0001 | Select the upper paper bin. This value is also used for the paper source for printers that only have one paper bin. |
| DMBIN\_LOWER  0x0002 | Select the lower bin. |
| DMBIN\_MIDDLE  0x0003 | Select the middle paper bin. |
| DMBIN\_MANUAL  0x0004 | Manually select the paper bin. |
| DMBIN\_ENVELOPE  0x0005 | Select the auto envelope bin. |
| DMBIN\_ENVMANUAL  0x0006 | Select the manual envelope bin. |
| DMBIN\_AUTO  0x0007 | Auto-select the bin. |
| DMBIN\_TRACTOR  0x0008 | Select the bin with the tractor paper. |
| DMBIN\_SMALLFMT  0x0009 | Select the bin with the smaller paper format. |
| DMBIN\_LARGEFMT  0x000A | Select the bin with the larger paper format. |
| DMBIN\_LARGECAPACITY  0x000B | Select the bin with large capacity. |
| DMBIN\_CASSETTE  0x000E | Select the cassette bin. |
| DMBIN\_FORMSOURCE  0x000F | Select the bin with the required form. |

**dmPrintQuality (2 bytes):** If the DM\_PRINTQUALITY bit is set in the **dmFields** field, the value of this field specifies the printer resolution. The value of this field MUST be either a positive value that specifies a device-dependent resolution in dots per inch (DPI) or one of the following four predefined device-independent values that are mapped to a device-specific resolution in an implementation-specific manner.

| Name/value | Meaning |
| --- | --- |
| DMRES\_HIGH  0xFFFC | High-resolution printouts |
| DMRES\_MEDIUM  0xFFFD | Medium-resolution printouts |
| DMRES\_LOW  0xFFFE | Low-resolution printouts |
| DMRES\_DRAFT  0xFFFF | Draft-resolution printouts |

**dmColor (2 bytes):** If the DM\_COLOR bit is set in the **dmFields** field, the value of this field specifies the color mode to use with color printers. The value of this field MUST be one of the following.

| Name/value | Meaning |
| --- | --- |
| DMRES\_MONOCHROME  0x0001 | Use monochrome printing mode. |
| DMRES\_COLOR  0x0002 | Use color printing mode. |

**dmDuplex (2 bytes):** If the DM\_DUPLEX bit is set in the **dmFields** field, the value of this field specifies duplex or double-sided printing for printers that are capable of duplex printing. The value of this field MUST be one of the following.

| Name/value | Meaning |
| --- | --- |
| DMDUP\_SIMPLEX  0x0001 | Normal (non-duplex) printing. |
| DMDUP\_VERTICAL  0x0002 | Long-edge binding; that is, the long edge of the page is vertical. |
| DMDUP\_HORIZONTAL  0x0003 | Short-edge binding; that is, the long edge of the page is horizontal. |

**dmYResolution (2 bytes):** If the DM\_YRESOLUTION bit is set in the **dmFields**, the value of this field specifies the y-resolution, in dots per inch, of the printer.

**dmTTOption (2 bytes):** If the DM\_TTOPTION bit is set in the **dmFields** field, the value of this field specifies how TrueType fonts MUST be printed. The value of this field MUST be one of the following.

| Name/value | Meaning |
| --- | --- |
| DMTT\_BITMAP  0x0001 | Prints TrueType fonts as graphics. This is the default action for dot-matrix printers. |
| DMTT\_DOWNLOAD  0x0002 | Downloads TrueType fonts as soft fonts. This is the default action for Hewlett-Packard printers that use [**printer control language (PCL)**](#gt_ffc25745-85c9-40bc-b2d3-f19ab7b48570). |
| DMTT\_SUBDEV  0x0003 | Substitutes device fonts for TrueType fonts. This is the default action for PostScript printers. |
| DMTT\_DOWNLOAD\_OUTLINE  0x0004 | Downloads TrueType fonts as outline soft fonts.[<155>](#Appendix_A_155) |

**dmCollate (2 bytes):** If the DM\_COLLATE bit is set in the **dmFields** field, the value of this field specifies whether collation MUST be used when printing multiple copies. The value of this field MUST be one of the following:

| Value | Meaning |
| --- | --- |
| DMCOLLATE\_FALSE  0x0000 | Do not collate when printing multiple copies. |
| DMCOLLATE\_TRUE  0x0001 | Collate when printing multiple copies. |

**dmFormName (64 bytes):** This field is a 32-element array of 16-bit Unicode characters. If the DM\_FORMNAME bit is set in the **dmFields** field, the value of this field specifies the name of the form to use, for example, "Letter" or "Legal". The value of this field is restricted to 32 characters, including the trailing null. Form names that are longer than 32 characters, including the trailing null, MUST be truncated to fit the array.

**reserved0 (2 bytes):** A value that SHOULD be set to zero when sent and MUST be ignored on receipt.

**reserved1 (4 bytes):** A value that SHOULD be set to zero when sent and MUST be ignored on receipt.

**reserved2 (4 bytes):** A value that SHOULD be set to zero when sent and MUST be ignored on receipt.

**reserved3 (4 bytes):** A value that SHOULD be set to zero when sent and MUST be ignored on receipt.

**dmNup (4 bytes):** If the DM\_NUP bit is set in the **dmFields**, the value of this field specifies the responsibility for performing page layout for [**N-Up Printing**](#gt_a1da4916-9932-4274-a4db-dd3994a9ba1b). It MUST be one of the following values:

| Name/Value | Meaning |
| --- | --- |
| DMNUP\_SYSTEM  0x00000001 | The print server does the page layout. |
| DMNUP\_ONEUP  0x00000002 | The application does the page layout. |

**reserved4 (4 bytes):** A value that SHOULD be set to zero when sent and MUST be ignored on receipt.

**dmICMMethod (4 bytes):** If the DM\_ICMMETHOD bit is set in the **dmFields** field, the value of this field specifies how Image Color Management (ICM) is handled. For a non-ICM application, this field determines if ICM is enabled or disabled. For ICM applications, the system examines this field to determine how to handle ICM support. The value of this field MUST be one of the following predefined values or a printer driver-defined value greater than or equal to the value of 0x00000100.

| Name/Value | Meaning |
| --- | --- |
| DMICMMETHOD\_NONE  0x00000001 | Specifies that ICM is disabled. |
| DMICMMETHOD\_SYSTEM  0x00000002 | Specifies that ICM is handled by the system on which the page description language (PDL) data is generated. |
| DMICMMETHOD\_DRIVER  0x00000003 | Specifies that ICM is handled by the printer driver. |
| DMICMMETHOD\_DEVICE  0x00000004 | Specifies that ICM is handled by the destination device. |

**dmICMIntent (4 bytes):** If the DM\_ICMINTENT bit is set in the **dmFields** field, the value of this field specifies which [**color matching**](#gt_a3322d29-7dc5-4696-b17a-55311db1cb33) method, or intent, MUST be used by default. This field is primarily for non-ICM applications. ICM applications can establish intents by using the ICM functions. The value of this field MUST be one of the following predefined values, or a printer driver defined value greater than or equal to the value of 0x00000100.

| Name/Value | Meaning |
| --- | --- |
| DMICM\_SATURATE  0x00000001 | Color matching SHOULD optimize for color saturation. |
| DMICM\_CONTRAST  0x00000002 | Color matching SHOULD optimize for color contrast. |
| DMICM\_COLORIMETRIC  0x00000003 | Color matching SHOULD optimize to match the exact color requested. |
| DMICM\_ABS\_COLORIMETRIC  0x00000004 | Color matching SHOULD optimize to match the exact color requested without [**white point**](#gt_d2287556-8c49-4b96-86b2-9bdf91da1860) mapping. |

**dmMediaType (4 bytes):** If the DM\_MEDIATYPE bit is set in the **dmFields** field, the value of this field specifies the type of media to print on. The value of this field MUST be one of the following predefined values or else a printer driver-defined value greater than or equal to the value of 0x00000100.

| Name/Value | Meaning |
| --- | --- |
| DMMEDIA\_STANDARD  0x00000001 | Plain paper |
| DMMEDIA\_TRANSPARENCY  0x00000002 | Transparent film |
| DMMEDIA\_GLOSSY  0x00000003 | Glossy paper |

**dmDitherType (4 bytes):** If the DM\_DITHERTYPE bit is set in the **dmFields** field, the value of this field specifies how [**dithering**](#gt_0540a76e-7353-4f83-a800-377d1fb00903) is to be done. The value of this field MUST be one of the following predefined values or else a printer driver-defined value greater than or equal to 0x00000100.

| Name/Value | Meaning |
| --- | --- |
| DMDITHER\_NONE  0x00000001 | No dithering. |
| DMDITHER\_COARSE  0x00000002 | Dithering with a coarse brush. |
| DMDITHER\_FINE  0x00000003 | Dithering with a fine brush. |
| DMDITHER\_LINEART  0x00000004 | Line art dithering, a special dithering method that produces well defined borders between black, white, and gray scaling. |
| DMDITHER\_ERRORDIFFUSION  0x00000005 | Windows 95/98/ME: Error diffusion dithering. |
| DMDITHER\_RESERVED6  0x00000006 | Same as DMDITHER\_LINEART. |
| DMDITHER\_RESERVED7  0x00000007 | Same as DMDITHER\_LINEART. |
| DMDITHER\_RESERVED8  0x00000008 | Same as DMDITHER\_LINEART. |
| DMDITHER\_RESERVED9  0x00000009 | Same as DMDITHER\_LINEART. |
| DMDITHER\_GRAYSCALE  0x0000000A | Device does gray scaling. |

**reserved5 (4 bytes):** A value that SHOULD be set to zero when sent and MUST be ignored on receipt.

**reserved6 (4 bytes):** A value that SHOULD be set to zero when sent and MUST be ignored on receipt.

**reserved7 (4 bytes):** A value that SHOULD be set to zero when sent and MUST be ignored on receipt.

**reserved8 (4 bytes):** A value that SHOULD be set to zero when sent and MUST be ignored on receipt.

**dmDriverExtraData (variable):** This field MAY contain implementation-specific printer driver data. Its size in bytes is specified by the value of the **dmDriverExtra** field.[<156>](#Appendix_A_156)

##### PostScript Driver Extra Data

Information about PostScript Driver Extra Data can be found in [Appendix B: Product Behavior](#Section_e81cbc09ab054a32ae4a8ec57b436c43).[<157>](#Appendix_A_157)

##### Generic Driver Extra Data

Information about Generic Driver Extra Data can be found in [Appendix B: Product Behavior](#Section_e81cbc09ab054a32ae4a8ec57b436c43).[<158>](#Appendix_A_158)

##### OEM Driver Extra Data

Information about OEM (vendor-supplied) Driver Extra Data can be found in [Appendix B: Product Behavior](#Section_e81cbc09ab054a32ae4a8ec57b436c43) [<159>](#Appendix_A_159).

##### Print Ticket Driver Extra Data

Information about Print Ticket Driver Extra Data can be found in [Appendix B: Product Behavior](#Section_e81cbc09ab054a32ae4a8ec57b436c43) [<160>](#Appendix_A_160)

#### Members in Custom-Marshaled INFO structures

Unless noted otherwise, pointer and string pointer members of IDL-marshaled [INFO](#Section_f29e9057df2940d49c51c615c0da62a1) structure forms are represented in the corresponding custom-marshaled INFO structure forms according to the following rules:

* **MEMBERNAME** is derived by trimming the leading "p" from the pointer, string pointer, or multisz pointer member of the IDL-marshaled INFO structure.
* The custom-marshaled INFO structure form contains an **Offset** member whose name is derived by appending "Offset" to **MEMBERNAME**.
* The **Offset** member is a 32-bit unsigned integer that specifies the number of bytes from the start of the structure to the start of the bytes making up the pointed-to data, string, or multisz. That data, string, or multisz area is represented in the custom-marshaled structure by a member whose name is derived by appending "Array" to **MEMBERNAME**. The length of that member is variable and includes the terminating null character for string data or the two terminating null characters for multisz data, respectively.
* If the pointer, string pointer, or multisz pointer member in the IDL-marshaled structure form is optional (that is, it can be NULL), it can be represented by a zero **Offset** in a custom-marshaled structure. The corresponding **Array** member is then considered optional and is present only if the **Offset** is not zero.

Unless noted otherwise, all other members of IDL-marshaled INFO structure forms are represented identically in the corresponding custom-marshaled INFO structure forms.

#### DATATYPES\_INFO\_1

The DATATYPES\_INFO\_1 structure contains information about the data type used to record a print job.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| Fixed\_Portion (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Variable\_Data (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Fixed\_Portion (variable):** An array of one or more fixed-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| NameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**NameOffset (4 bytes):** A 32-bit unsigned integer that specifies the number of bytes from the start of the structure to the **NameArray** member.

**Variable\_Data (variable):** An array of zero or more optional, variable-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| NameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**NameArray (variable):** A string that specifies the data type used to record a print job. The location of this buffer is determined by the value of the **NameOffset** member. For rules governing data type names, see section [2.2.4.2](#Section_f40cbad088dd4e86a16bcf62c3b17c52).

#### \_DRIVER\_INFO

##### \_DRIVER\_INFO\_1

The \_DRIVER\_INFO\_1 structure specifies printer driver information. It is a custom-marshaled form of the [DRIVER\_INFO\_1 (section 2.2.1.5.1)](#Section_df308bbf8ff74b0f930bcb9b90080b90) structure.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| Fixed\_Portion (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Variable\_Data (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Fixed\_Portion (variable):** An array of one or more fixed-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| NameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**NameOffset (4 bytes):** This member is a 32-bit unsigned integer that MUST specify the number of bytes from the start of the structure to the **NameArray** member.

**Variable\_Data (variable):** An array of zero or more optional, variable-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| NameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**NameArray (variable):** This is a string that MUST specify the Printer Driver Name. The location of this buffer is determined by the value of the **NameOffset** member.

##### \_DRIVER\_INFO\_2

The \_DRIVER\_INFO\_2 structure specifies printer driver information. It is a custom-marshaled form of the [DRIVER\_INFO\_2 (section 2.2.1.5.2)](#Section_39bbfc3087684cd49930434857e2c2a2) structure.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| Fixed\_Portion (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Variable\_Data (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Fixed\_Portion (variable):** An array of one or more groups of fixed-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| cVersion | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| NameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| EnvironmentOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DriverPathOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DataFileOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ConfigFileOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Variable\_Data (variable):** An array of zero or more groups of optional, variable-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| ConfigFileArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DataFileArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DriverPathArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| EnvironmentArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| NameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

##### \_DRIVER\_INFO\_3

The \_DRIVER\_INFO\_3 structure specifies printer driver information. It is a custom-marshaled form of the [RPC\_DRIVER\_INFO\_3 (section 2.2.1.5.3)](#Section_b4d3b5f3c6314fc393484c983590a0d8) structure.[<161>](#Appendix_A_161)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| Fixed\_Portion (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Variable\_Data (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Fixed\_Portion (variable):** An array of one or more groups of fixed-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| cVersion | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| NameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| EnvironmentOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DriverPathOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DataFileOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ConfigFileOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| HelpFileOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DependentFilesOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| MonitorNameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DefaultDataTypeOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Variable\_Data (variable):** An array of zero or more groups of optional, variable-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| DefaultDataTypeArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| MonitorNameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DependentFilesArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| HelpFileArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ConfigFileArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DataFileArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DriverPathArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| EnvironmentArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| NameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

##### \_DRIVER\_INFO\_4

The \_DRIVER\_INFO\_4 structure specifies printer driver information.[<162>](#Appendix_A_162) It is a custom-marshaled form of the [RPC\_DRIVER\_INFO\_4 (section 2.2.1.5.4)](#Section_459d24e40b754fc7b246fe40a1981eed) structure.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| Fixed\_Portion (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Variable\_Data (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Fixed\_Portion (variable):** An array of one or more groups of fixed-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| cVersion | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| NameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| EnvironmentOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DriverPathOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DataFileOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ConfigFileOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| HelpFileOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DependentFilesOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| MonitorNameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DefaultDataTypeOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| szzPreviousNamesOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Variable\_Data (variable):** An array of zero or more groups of optional, variable-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| szzPreviousNamesArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DefaultDataTypeArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| MonitorNameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DependentFilesArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| HelpFileArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ConfigFileArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DataFileArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DriverPathArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| EnvironmentArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| NameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

##### \_DRIVER\_INFO\_5

The \_DRIVER\_INFO\_5 structure specifies printer driver information.[<163>](#Appendix_A_163)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| Fixed\_Portion (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Variable\_Data (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Fixed\_Portion (variable):** An array of one or more groups of fixed-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| cVersion | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| NameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| EnvironmentOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DriverPathOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DataFileOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ConfigFileOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwDriverAttributes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwConfigVersion | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwDriverVersion | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**dwDriverAttributes (4 bytes):** A bit field that specifies attributes of the printer driver.

| 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 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | A | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Where the bits are defined as:

| Value | Description |
| --- | --- |
| A  PRINTER\_DRIVER\_PACKAGE\_AWARE | The printer driver is part of a driver package. |

**dwConfigVersion (4 bytes):** The number of times the printer driver configuration file has been upgraded (replaced with a newer binary) or downgraded (replaced with an older binary) since the system was restarted.

**dwDriverVersion (4 bytes):** The number of times the printer driver executable file has been upgraded (replaced with a newer binary) or downgraded (replaced with an older binary) since the system was restarted.

**Variable\_Data (variable):** An array of zero or more groups of optional, variable-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| ConfigFileArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DataFileArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DriverPathArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| EnvironmentArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| NameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Members not defined in this section are identical to members in the [\_DRIVER\_INFO\_4 structure (section 2.2.2.4.4)](#Section_2a53e91004c34aba905a26741dcd59dd).

##### \_DRIVER\_INFO\_6

The DRIVER\_INFO\_6 structure specifies printer driver information.[<164>](#Appendix_A_164) It is a custom-marshaled form of the [RPC\_DRIVER\_INFO\_6 (section 2.2.1.5.5)](#Section_ae12474015ef482c824e56e5436ed44b) structure.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| Fixed\_Portion (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Variable\_Data (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Fixed\_Portion (variable):** An array of one or more groups of fixed-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| cVersion | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| NameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| EnvironmentOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DriverPathOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DataFileOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ConfigFileOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| HelpFileOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DependentFilesOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| MonitorNameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DefaultDataTypeOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| szzPreviousNamesOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ftDriverDate.dwLowDateTime | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ftDriverDate.dwHighDateTime | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PaddingForAlignment | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwlDriverVersion | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| MfgNameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| OEMUrlOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| HardwareIDOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ProviderOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**PaddingForAlignment (4 bytes):** 4 bytes of padding to align the **dwlDriverVersion** field on an 8-byte boundary. The contents of this field MUST be ignored.

**Variable\_Data (variable):** An array of zero or more groups of optional, variable-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| ProviderArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| HardwareIDArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| OEMUrlArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| MfgNameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| szzPreviousNamesArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DefaultDataTypeArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| MonitorNameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DependentFilesArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| HelpFileArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ConfigFileArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DataFileArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DriverPathArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| EnvironmentArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| NameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

##### \_DRIVER\_INFO\_7

The \_DRIVER\_INFO\_7 structure specifies printer driver information.[<165>](#Appendix_A_165)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| Fixed\_Portion (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Variable\_Data (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Fixed\_Portion (variable):** An array of one or more groups of fixed-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| cbSize | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cVersion | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| szDriverNameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| szInfNameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| szInstallSourceRootOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**cbSize (4 bytes):** The value of this member MUST specify the size, in bytes, of the \_DRIVER\_INFO\_7 data structure.

**szDriverNameOffset (4 bytes):** This member is synonymous with the **NameOffset** member.

**szInfNameOffset (4 bytes):** This member is a 32-bit unsigned integer that MUST specify the number of bytes from the start of the structure to the start of the **szInfNameArray** member.

**szInstallSourceRootOffset (4 bytes):** This member is a 32-bit unsigned integer that MUST specify the number of bytes from the start of the structure to the start of the **szInstallSourceRootArray** member.

**Variable\_Data (variable):** An array of zero or more groups of optional, variable-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| szInstallSourceRootArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| szInfNameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| szDriverNameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**szInstallSourceRootArray (variable):** This member MUST be a string that specifies the path from which the driver MUST be installed. The location of this buffer is determined by the value of the **szInstallSourceRootOffset** member.

**szInfNameArray (variable):** This member MUST be a string that specifies the name of the driver's installation configuration file. The location of this buffer is determined by the value of the **szInfNameOffset** member.[<166>](#Appendix_A_166)

**szDriverNameArray (variable):** This member is synonymous with the **NameArray** member.

##### \_DRIVER\_INFO\_8

The \_DRIVER\_INFO\_8 structure specifies printer driver information. It is a custom-marshaled form of the [RPC\_DRIVER\_INFO\_8 (section 2.2.1.5.6)](#Section_937d5fc867f145f8ab878b56a8fca558) structure.[<167>](#Appendix_A_167)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| Fixed\_Portion (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Variable\_Data (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Fixed\_Portion (variable):** An array of one or more groups of fixed-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| cVersion | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| NameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| EnvironmentOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DriverPathOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DataFileOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ConfigFileOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| HelpFileOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DependentFilesOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| MonitorNameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DefaultDataTypeOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| szzPreviousNamesOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ftDriverDate.dwLowDateTime | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ftDriverDate.dwHighDateTime | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PaddingForAlignment | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwlDriverVersion | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| MfgNameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| OEMUrlOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| HardwareIDOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ProviderOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PrintProcessorOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| VendorSetupOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| szzColorProfilesOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| InfPathOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwPrinterDriverAttributes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| szzCoreDependenciesOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ftMinInboxDriverVerDate | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwlMinInboxDriverVerVersion | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**PaddingForAlignment (4 bytes):** 4 bytes of padding to align the **dwlDriverVersion** field on an 8-byte boundary. The contents of this field MUST be ignored.

**Variable\_Data (variable):** An array of zero or more groups of optional, variable-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| InfPathArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| szzColorProfilesArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| VendorSetupArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PrintProcessorArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ProviderArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| HardwareIDArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| OEMUrlArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| MfgNameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| szzPreviousNamesArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DefaultDataTypeArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| MonitorNameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DependentFilesArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| HelpFileArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ConfigFileArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DataFileArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DriverPathArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| EnvironmentArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| NameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| szzCoreDependenciesArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

##### \_DRIVER\_INFO\_101

The DRIVER\_INFO\_101 structure specifies printer driver information.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| Fixed\_Portion (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Variable\_Data (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Fixed\_Portion (variable):** An array of one or more groups of fixed-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| cVersion | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| NameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| EnvironmentOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| FileInfoOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwFileCount | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| MonitorNameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DefaultDataTypeOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| szzPreviousNamesOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ftDriverDate | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwlDriverVersion | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| MfgNameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| OEMUrlOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| HardwareIDOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ProviderOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**FileInfoOffset (4 bytes):** This member is a 32-bit unsigned integer that MUST specify the number of bytes from the start of the structure to the start of the **FileInfoArray** member.

**dwFileCount (4 bytes):** This member is a 32-bit unsigned integer that MUST specify the number of [DRIVER\_FILE\_INFO](#Section_37ac9d7f9ee54e1a98e81186007ca52e) structures in the **FileInfoArray** member.

**Variable\_Data (variable):** An array of zero or more groups of optional, variable-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| ProviderArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| HardwareIDArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| OEMUrlArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| MfgNameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| szzPreviousNamesArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DefaultDataTypeArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| MonitorNameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| FileInfoArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| EnvironmentArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| NameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**FileInfoArray (variable):** This member MUST be an array of DRIVER\_FILE\_INFO structures. The number of elements in the array MUST be the same as the value of the **dwFileCount** member.

**Note:** Member definitions for all members not defined in this section are identical to members in [DRIVER\_INFO\_8 (section 2.2.2.4.8)](#Section_f9b749d608fc48208d7bfe0ee33e42c5).

##### \_DRIVER\_FILE\_INFO

The \_DRIVER\_FILE\_INFO structure specifies information about a file belonging to a printer driver.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| Fixed\_Portion (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Variable\_Data (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Fixed\_Portion (variable):** An array of one or more groups of fixed-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| FileNameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| FileType | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| FileVersion | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**FileNameOffset (4 bytes):** A 32-bit unsigned integer that specifies the number of bytes from the start of the [\_DRIVER\_INFO\_101 (section 2.2.2.4.9)](#Section_821d675e39444c77b4ec3b519883cfa6) structure that contains this structure to the start of the **FileNameArray** field.

**FileType (4 bytes):** A 32-bit unsigned integer that specifies the file type using one of the constant values from the following table.

| Value | Meaning |
| --- | --- |
| 0x00000000 | The file is a rendering driver module executable. |
| 0x00000001 | The file is a configuration module executable. |
| 0x00000002 | The file is a driver data file. |
| 0x00000003 | The file is a driver help file. |
| 0x00000004 | The file is a dependent file with a type other than the preceding file types. |

**FileVersion (4 bytes):** The version of the printer driver file. The actual value is specific to the implementation of the printer driver. Since printer driver are developed by third parties, it is not practical to list all possible values.

A print client MAY use this field to detect a change to the printer driver on a print server.[<168>](#Appendix_A_168)

**Variable\_Data (variable):** An array of zero or more optional, variable-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| FileNameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**FileNameArray (variable):** A null-terminated string that specifies the name of the file in Unicode UTF-16LE characters. The location of this buffer is determined by the value of the **FileNameOffset** field.

#### \_FORM\_INFO

##### \_FORM\_INFO\_1

The \_FORM\_INFO\_1 structure specifies printer media information. It is a custom-marshaled form of the [FORM\_INFO\_1 (section 2.2.1.6.1)](#Section_7821930259dd4d3f82d73a2e95466c28) structure.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| Fixed\_Portion (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Variable\_Data (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Fixed\_Portion (variable):** An array of one or more groups of fixed-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| Flags | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| NameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size.cx | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size.cy | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ImageableArea.left | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ImageableArea.top | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ImageableArea.right | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ImageableArea.bottom | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Variable\_Data (variable):** An array of zero or more optional, variable-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| NameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

All members not defined in this section are specified in sections [2.2.1.3.2](#Section_84fe09faea4b4cc58ae49c8582fd1ffe) and [2.2.2.3](#Section_7224d5eac0e1474b8e24034f0032e09c).

##### \_FORM\_INFO\_2

The \_FORM\_INFO\_2 structure specifies printer media information. It is a custom-marshaled form of the [RPC\_FORM\_INFO\_2 (section 2.2.1.6.2)](#Section_dca49b4f3eb146569e972f0b0c569bf6) structure.[<169>](#Appendix_A_169)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| Fixed\_Portion (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Variable\_Data (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Fixed\_Portion (variable):** An array of one or more groups of fixed-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| Flags | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| NameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size.cx | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size.cy | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ImageableArea.left | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ImageableArea.top | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ImageableArea.right | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ImageableArea.bottom | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| KeywordOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| StringType | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| MuiDllOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwResourceID | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DisplayNameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| wLangID | | | | | | | | | | | | | | | | unused | | | | | | | | | | | | | | | |

**unused (2 bytes):** A value that SHOULD be set to zero when sent and MUST be ignored on receipt.

**Variable\_Data (variable):** An array of zero or more groups of optional, variable-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| NameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| KeywordArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| MuiDllArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DisplayNameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

#### \_JOB\_INFO

##### \_JOB\_INFO\_1

The \_JOB\_INFO\_1 structure specifies print job information. It is a custom-marshaled form of the [JOB\_INFO\_1 (section 2.2.1.7.1)](#Section_794c1ba9ba0a4498ba9c2635563460dd) structure.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| Fixed\_Portion (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Variable\_Data (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Fixed\_Portion (variable):** An array of one or more groups of fixed-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| JobId | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PrinterNameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| MachineNameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| UserNameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DocumentOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DatatypeOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| StatusOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Status | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Priority | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| TotalPages | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PagesPrinted | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Submitted.wYear | | | | | | | | | | | | | | | | Submitted.wMonth | | | | | | | | | | | | | | | |
| Submitted.wDayOfWeek | | | | | | | | | | | | | | | | Submitted.wDay | | | | | | | | | | | | | | | |
| Submitted.wHour | | | | | | | | | | | | | | | | Submitted.wMinute | | | | | | | | | | | | | | | |
| Submitted.wSecond | | | | | | | | | | | | | | | | Submitted.wMilliseconds | | | | | | | | | | | | | | | |

**Variable\_Data (variable):** An array of zero or more groups of optional, variable-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| StatusArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DatatypeArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DocumentArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| UserNameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| MachineNameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PrinterNameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Fields that are not defined in this section are specified in section [2.2.1.3.3](#Section_94d1857ec44449b18fb483eac6696060).

##### \_JOB\_INFO\_2

The \_JOB\_INFO\_2 structure specifies print job information. It is a custom-marshaled form of the [JOB\_INFO\_2 (section 2.2.1.7.2)](#Section_5a48c6c9fc5f42ffa5a7f111fee58d1a) structure.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| Fixed\_Portion (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Variable\_Data (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Fixed\_Portion (variable):** An array of one or more groups of fixed-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| JobId | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PrinterNameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| MachineNameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| UserNameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DocumentOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| NotifyNameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DatatypeOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PrintProcessorOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ParametersOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DriverNameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DevModeOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| StatusOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SecurityDescriptorOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Status | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Priority | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| StartTime | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| UntilTime | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| TotalPages | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Submitted.wYear | | | | | | | | | | | | | | | | Submitted.wMonth | | | | | | | | | | | | | | | |
| Submitted.wDayOfWeek | | | | | | | | | | | | | | | | Submitted.wDay | | | | | | | | | | | | | | | |
| Submitted.wHour | | | | | | | | | | | | | | | | Submitted.wMinute | | | | | | | | | | | | | | | |
| Submitted.wSecond | | | | | | | | | | | | | | | | Submitted.wMilliseconds | | | | | | | | | | | | | | | |
| Time | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PagesPrinted | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Variable\_Data (variable):** An array of zero or more groups of optional, variable-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| SecurityDescriptorArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| StatusArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DevModeArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DriverNameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ParametersArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PrintProcessorArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DatatypeArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| NotifyNameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DocumentArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| UserNameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| MachineNameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PrinterNameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Fields that are not defined in this section are specified in section [2.2.1.3.3](#Section_94d1857ec44449b18fb483eac6696060).

##### \_JOB\_INFO\_3

The \_JOB\_INFO\_3 structure specifies information about the order of print jobs, and it is used to alter the order of print jobs.[<170>](#Appendix_A_170) It is a custom-marshaled form of the [JOB\_INFO\_3 (section 2.2.1.7.3)](#Section_1385aef9d3234e8d9d285f2f011bd39c) structure.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| Fixed\_Portion (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Fixed\_Portion (variable):** An array of one or more groups of fixed-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| JobId | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| NextJobId | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Fields that are not defined in this section are specified in sections 2.2.1.7.3, and [2.2.1.3](#Section_964b517f67304b4a99edee9648e7f1f7).

##### \_JOB\_INFO\_4

The \_JOB\_INFO\_4 structure specifies print job information. It is a custom-marshaled form of the [JOB\_INFO\_4 (section 2.2.1.7.4)](#Section_99086f6e3db14034ac6ccd4a5a77f22c) structure.[<171>](#Appendix_A_171)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| Fixed\_Portion (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Variable\_Data (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Fixed\_Portion (variable):** An array of one or more groups of fixed-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| JobId | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PrinterNameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| MachineNameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| UserNameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DocumentOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| NotifyNameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DatatypeOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PrintProcessorOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ParametersOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DriverNameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DevModeOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| StatusOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SecurityDescriptorOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Status | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Priority | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| StartTime | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| UntilTime | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| TotalPages | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Submitted.wYear | | | | | | | | | | | | | | | | Submitted.wMonth | | | | | | | | | | | | | | | |
| Submitted.wDayOfWeek | | | | | | | | | | | | | | | | Submitted.wDay | | | | | | | | | | | | | | | |
| Submitted.wHour | | | | | | | | | | | | | | | | Submitted.wMinute | | | | | | | | | | | | | | | |
| Submitted.wSecond | | | | | | | | | | | | | | | | Submitted.wMilliseconds | | | | | | | | | | | | | | | |
| Time | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PagesPrinted | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SizeHigh | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Variable\_Data (variable):** An array of zero or more groups of optional, variable-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| SecurityDescriptorArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| StatusArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DevModeArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DriverNameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ParametersArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PrintProcessorArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DatatypeArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| NotifyNameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DocumentArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| UserNameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| MachineNameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PrinterNameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Fields that are not defined in this section are specified in section [2.2.1.3.3](#Section_94d1857ec44449b18fb483eac6696060).

#### \_MONITOR\_INFO

##### \_MONITOR\_INFO\_1

The \_MONITOR\_INFO\_1 structure identifies an installed port monitor. It is a custom-marshaled form of the [MONITOR\_INFO\_1 (section 2.2.1.8.1)](#Section_7ee6a75f988941a68b1490d4e378b956) structure.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| Fixed\_Portion (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Variable\_Data (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Fixed\_Portion (variable):** An array of one or more fixed-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| NameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Variable\_Data (variable):** An array of zero or more optional, variable-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| NameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

##### \_MONITOR\_INFO\_2

The \_MONITOR\_INFO\_2 structure is used to identify a port monitor. It is a custom-marshaled form of the [MONITOR\_INFO\_2 (section 2.2.1.8.2)](#Section_f05b335b203c454f87da69ad96930bd7) structure.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| Fixed\_Portion (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Variable\_Data (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Fixed\_Portion (variable):** An array of one or more groups of fixed-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| NameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| EnvironmentOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DLLNameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Variable\_Data (variable):** An array of zero or more groups of optional, variable-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| DLLNameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| EnvironmentArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| NameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

#### \_PORT\_INFO

##### \_PORT\_INFO\_1

The \_PORT\_INFO\_1 structure specifies information about a printer port. It is a custom-marshaled form of the [PORT\_INFO\_1 (section 2.2.1.9.1)](#Section_9bab3fb5b36948f28f083082ca13e739) structure.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| Fixed\_Portion (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Variable\_Data (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Fixed\_Portion (variable):** An array of one or more fixed-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| NameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Variable\_Data (variable):** An array of zero or more optional, variable-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| NameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

##### \_PORT\_INFO\_2

The \_PORT\_INFO\_2 structure specifies information about a printer port. It is a custom-marshaled form of the [PORT\_INFO\_2 (section 2.2.1.9.2)](#Section_7a923364dffc40b3bca5bf2eed610f31) structure.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| Fixed\_Portion (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Variable\_Data (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Fixed\_Portion (variable):** An array of one or more groups of fixed-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| PortNameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| MonitorNameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DescriptionOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| fPortType | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Variable\_Data (variable):** An array of zero or more groups of optional, variable-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| DescriptionArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| MonitorNameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PortNameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

#### \_PRINTER\_INFO

##### \_PRINTER\_INFO\_STRESS

The \_PRINTER\_INFO\_STRESS structure specifies printer diagnostic information. It is a custom-marshaled form of the [PRINTER\_INFO\_STRESS (section 2.2.1.10.1)](#Section_9e94b8f4a84b4c96b9ae4b4acb3469b8) structure.[<172>](#Appendix_A_172) This form of the \_PRINTER\_INFO\_STRESS structure corresponds to an information **Level** value of 0x00000000.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| Fixed\_Portion (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Variable\_Data (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Fixed\_Portion (variable):** An array of one or more groups of fixed-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| PrinterNameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ServerNameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cJobs | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cTotalJobs | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cTotalBytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| stUpTime.wYear | | | | | | | | | | | | | | | | stUpTime.wMonth | | | | | | | | | | | | | | | |
| stUpTime.wDayOfWeek | | | | | | | | | | | | | | | | stUpTime.wDay | | | | | | | | | | | | | | | |
| stUpTime.wHour | | | | | | | | | | | | | | | | stUpTime.wMinute | | | | | | | | | | | | | | | |
| stUpTime.wSecond | | | | | | | | | | | | | | | | stUpTime.wMilliseconds | | | | | | | | | | | | | | | |
| MaxcRef | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cTotalPagesPrinted | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwGetVersion | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| fFreeBuild | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cSpooling | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cMaxSpooling | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cRef | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cErrorOutOfPaper | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cErrorNotReady | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cJobError | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwNumberOfProcessors | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwProcessorType | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwHighPartTotalBytes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cChangeID | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwLastError | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Status | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cEnumerateNetworkPrinters | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cAddNetPrinters | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| wProcessorArchitecture | | | | | | | | | | | | | | | | wProcessorLevel | | | | | | | | | | | | | | | |
| cRefIC | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwReserved2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwReserved3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Variable\_Data (variable):** An array of zero or more groups of optional, variable-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| PrinterNameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ServerNameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

##### \_PRINTER\_INFO\_1

The \_PRINTER\_INFO\_1 structure specifies printer information. It is a custom-marshaled form of the [PRINTER\_INFO\_1 (section 2.2.1.10.2)](#Section_0d1a296b7027441c83938aa5c3764b27) structure.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| Fixed\_Portion (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Variable\_Data (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Fixed\_Portion (variable):** An array of one or more groups of fixed-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| Flags | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DescriptionOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| NameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| CommentOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Variable\_Data (variable):** An array of zero or more groups of optional, variable-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| CommentArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| NameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DescriptionArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

##### \_PRINTER\_INFO\_2

The \_PRINTER\_INFO\_2 structure specifies printer information. It is a custom-marshaled form of the [PRINTER\_INFO\_2 (section 2.2.1.10.3)](#Section_d40734e1ac9b4b2a862883a1bc7bbeed) structure.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| Fixed\_Portion (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Variable\_Data (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Fixed\_Portion (variable):** An array of one or more groups of fixed-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| ServerNameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PrinterNameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ShareNameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PortNameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DriverNameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| CommentOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| LocationOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DevModeOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SepFileOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PrintProcessorOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DatatypeOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ParametersOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SecurityDescriptorOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Attributes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Priority | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DefaultPriority | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| StartTime | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| UntilTime | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Status | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cJobs | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| AveragePPM | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Variable\_Data (variable):** An array of zero or more groups of optional, variable-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| SecurityDescriptorArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ParametersArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DatatypeArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PrintProcessorArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SepFileArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DevModeArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| LocationArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| CommentArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DriverNameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PortNameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ShareNameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PrinterNameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ServerNameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

##### \_PRINTER\_INFO\_3

The \_PRINTER\_INFO\_3 structure specifies printer information. It is a custom-marshaled form of the [PRINTER\_INFO\_3 (section 2.2.1.10.4)](#Section_00af009b231e4ae6bfd7cb8476c32c19) structure.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| Fixed\_Portion (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Variable\_Data (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Fixed\_Portion (variable):** An array of one or more fixed-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| SecurityDescriptorOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Variable\_Data (variable):** An array of zero or more optional, variable-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| SecurityDescriptorArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

##### \_PRINTER\_INFO\_4

The \_PRINTER\_INFO\_4 structure specifies printer information. It is a custom-marshaled form of the [PRINTER\_INFO\_4 (section 2.2.1.10.5)](#Section_ac391864cd9945099ff3ea0366dd9b2b) structure.[<173>](#Appendix_A_173)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| Fixed\_Portion (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Variable\_Data (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Fixed\_Portion (variable):** An array of one or more groups of fixed-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| PrinterNameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ServerNameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Attributes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Variable\_Data (variable):** An array of zero or more groups of optional, variable-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| ServerNameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PrinterNameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

##### \_PRINTER\_INFO\_5

The \_PRINTER\_INFO\_5 structure specifies printer information. It is a custom-marshaled form of the [PRINTER\_INFO\_5 (section 2.2.1.10.6)](#Section_d7e5feaa16be4dbdb6d922f62f69afbb) structure.[<174>](#Appendix_A_174)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| Fixed\_Portion (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Variable\_Data (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Fixed\_Portion (variable):** An array of one or more groups of fixed-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| PrinterNameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PortNameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Attributes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DeviceNotSelectedTimeout | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| TransmissionRetryTimeout | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Variable\_Data (variable):** An array of zero or more groups of optional, variable-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| PortNameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PrinterNameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

##### \_PRINTER\_INFO\_6

The \_PRINTER\_INFO\_6 structure specifies printer information. It is a custom-marshaled form of the [PRINTER\_INFO\_6 (section 2.2.1.10.7)](#Section_436ef38839cd499295ac02ddd2a51193) structure.[<175>](#Appendix_A_175)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| Fixed\_Portion (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Fixed\_Portion (variable):** An array of one or more fixed-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| dwStatus | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**dwStatus (4 bytes):** The new printer status, which is a value specified in [Status and Attribute Values (section 2.2.3.12)](#Section_1625e9d929e448f4b83d3bd0fdaea787).

##### \_PRINTER\_INFO\_7

The \_PRINTER\_INFO\_7 structure specifies printer information.[<176>](#Appendix_A_176) It is a custom-marshaled form of the [PRINTER\_INFO\_7 (section 2.2.1.10.8)](#Section_91a6cb808df743d08a5d8f1d23f96de6) structure.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| Fixed\_Portion (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Variable\_Data (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Fixed\_Portion (variable):** An array of one or more groups of fixed-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| szObjectGUIDOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwAction | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Variable\_Data (variable):** An array of zero or more optional, variable-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| szObjectGUIDArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

##### \_PRINTER\_INFO\_8

The \_PRINTER\_INFO\_8 structure specifies printer information.[<177>](#Appendix_A_177) It is a custom-marshaled form of the [PRINTER\_INFO\_8 (section 2.2.1.10.9)](#Section_abe75b29b0924398a5af9d639f430cde) structure.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| Fixed\_Portion (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Variable\_Data (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Fixed\_Portion (variable):** An array of one or more fixed-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| DevModeOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Variable\_Data (variable):** An array of zero or more optional, variable-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| DevModeArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

#### PRINTPROCESSOR\_INFO\_1

The PRINTPROCESSOR\_INFO\_1 structure specifies printer information.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| Fixed\_Portion (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Variable\_Data (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Fixed\_Portion (variable):** An array of one or more fixed-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| NameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**NameOffset (4 bytes):** This member is a 32-bit unsigned integer that MUST specify the number of bytes from the start of the structure to the start of the **NameArray** member.

**Variable\_Data (variable):** An array of zero or more optional, variable-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| NameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**NameArray (variable):** This member MUST contain a string that specifies the print processor name. The location of this buffer is determined by the value of the **NameOffset** member. For rules governing print processor names, see section [2.2.4.11](#Section_807e0bdd21a54a729a6a1a7fb1b08a8e).

#### PRINTER\_ENUM\_VALUES

The PRINTER\_ENUM\_VALUES structure specifies the value name, type, and data for a printer configuration value.[<178>](#Appendix_A_178)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| Fixed\_Portion (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Variable\_Data (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Fixed\_Portion (variable):** An array of one or more groups of fixed-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| ValueNameOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cbValueName | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwType | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DataOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| cbData | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**ValueNameOffset (4 bytes):** This member is a 32-bit unsigned integer that MUST specify the number of bytes from the start of the structure to the start of the **ValueNameArray** member.

**cbValueName (4 bytes):** The value of this member MUST specify the size of the **ValueNameArray**, in bytes.

**dwType (4 bytes):** The value of this member MUST specify the data type of the data in the **DataArray** member. For a list of the possible type codes, see section [2.2.3.9](#Section_25cce7007fcf4bb6a2f30f6d08430a55). For rules governing [**registry**](#gt_2d67ea57-10e6-422f-985e-694834945a02) type values, see section [2.2.4.15](#Section_f3ded7900d0a462b9bd173051033cb8a).

**DataOffset (4 bytes):** This member is a 32-bit unsigned integer that MUST specify the number of bytes from the start of the structure to the start of the **Data** member.

**cbData (4 bytes):** The value of this member MUST specify the number of bytes retrieved in the **DataArray** buffer.

**Variable\_Data (variable):** An array of zero or more groups of optional, variable-size fields, which are defined as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| ValueNameArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DataArray (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**ValueNameArray (variable):** This member MUST contain a string that specifies the value name. The location of this buffer is determined by the value of the **ValueNameOffset** member. For rules governing value names, see section [2.2.4.18](#Section_cd9850cf1e6c48759b65e159cf2d83d6).

**DataArray (variable):** This member MUST contain the data for the retrieved value.

#### UNIVERSAL\_FONT\_ID

The UNIVERSAL\_FONT\_ID structure identifies a font.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| Checksum | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Index | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Checksum (4 bytes):** A 32-bit unsigned integer that is the implementation-specific [**checksum**](#gt_fa444149-ef93-4512-a278-2e756295630c) of the font. [<179>](#Appendix_A_179)

**Index (4 bytes):** A 32-bit unsigned integer that is an index associated with the font. The meaning of this field is determined by the type of font.

**Note**  The UNIVERSAL\_FONT\_ID structure is equivalent to the [**enhanced metafile format (EMF)**](#gt_d9d0bff9-d270-4528-9081-fe51db809c36) UniversalFontId object ([[MS-EMF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-EMF%5d.pdf) section 2.2.27).

#### CORE\_PRINTER\_DRIVER

The CORE\_PRINTER\_DRIVER structure defines information that identifies a specific core printer driver. See the [RpcGetCorePrinterDrivers (section 3.1.4.4.9)](#Section_3d6282322aa944f484e01168d3208f27) method for an example of its use.[<180>](#Appendix_A_180)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| CoreDriverGUID (16 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ftDriverDate | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwlDriverVersion | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| szPackageID (520 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**CoreDriverGUID (16 bytes):** A GUID value that MUST uniquely identify the package.

**ftDriverDate (8 bytes):** A FILETIME value that MUST specify the date this package was published.

**dwlDriverVersion (8 bytes):** A 64-bit value that MUST specify the version of the core printer driver that MAY[<181>](#Appendix_A_181) be used to match the driver version in the driver installation control file.

**szPackageID (520 bytes):** A **string** that MUST specify the package name.

#### TCPMON Structures

##### CONFIG\_INFO\_DATA\_1

The CONFIG\_INFO\_DATA\_1 structure specifies printer configuration data.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| PortName (128 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Version | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**PortName (128 bytes):** A null-terminated Unicode string, which is the name of the port to be queried for configuration information.

**Version (4 bytes):** A 32-bit unsigned integer that is the level of the **PORT\_DATA** structure that will contain the configuration information. This value MUST be 0x00000001 or 0x00000002.

##### DELETE\_PORT\_DATA\_1

The DELETE\_PORT\_DATA\_1 structure specifies the port to be deleted.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| PortName (128 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Name (98 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | PaddingForAlignment | | | | | | | | | | | | | | | |
| Version | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**PortName (128 bytes):** A null-terminated Unicode string, which is the name of the port to be deleted.

**Name (98 bytes):** A null-terminated Unicode string, which is the server name for the port to be deleted.

**PaddingForAlignment (2 bytes):** 2 bytes of padding to align **Version** field on a 4-byte boundary. The contents of this field MUST be ignored.

**Version (4 bytes):** A 32-bit unsigned integer that is the version of this structure. This value MUST be 0x00000001.

**Reserved (4 bytes):** A 32-bit unsigned integer that is set to zero.

##### PORT\_DATA\_1

The PORT\_DATA\_1 structure specifies **PORT\_DATA** level 1 data.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| PortName (128 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Version | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Protocol | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| HostAddress (98 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | SNMPCommunity (66 bytes) | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DoubleSpool | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Queue (66 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | IPAddress (32 bytes) | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | HardwareAddress (26 bytes) | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DeviceType (514 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | PaddingForAlignment | | | | | | | | | | | | | | | |
| PortNumber | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SNMPEnabled | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SNMPDevIndex | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**PortName (128 bytes):** A null-terminated Unicode string, which is the name of the port.

**Version (4 bytes):** A 32-bit unsigned integer that specifies the version number of the PORT\_DATA\_1 structure. This value MUST be 0x00000001.

**Protocol (4 bytes):** A 32-bit unsigned integer that specifies the protocol to use for the port. This value MUST be either **PROTOCOL\_RAWTCP\_TYPE** (0x00000001), indicating that the port expects RAW print data, or **PROTOCOL\_LPR\_TYPE** (0x00000002), indicating that the port expects to be driven as an LPR port.

**Size (4 bytes):** A 32-bit unsigned integer that specifies the size, in bytes, of the PORT\_DATA\_1 structure.

**Reserved (4 bytes):** A 32-bit unsigned integer that is set to zero.

**HostAddress (98 bytes):** A null-terminated Unicode string, which is the IP address or host name of the printer.

**SNMPCommunity (66 bytes):** A null-terminated Unicode string, which is the Simple Network Management Protocol (SNMP) [[RFC1157]](http://go.microsoft.com/fwlink/?LinkId=121934) community name of the printer.

**DoubleSpool (4 bytes):** A 32-bit unsigned integer that, if nonzero, indicates that double spooling is enabled or, if zero, indicates that double spooling is disabled.

**Queue (66 bytes):** A null-terminated Unicode string, which is the LPR queue name.

**IPAddress (32 bytes):** A null-terminated Unicode string, which is the [**IPv4**](#gt_0f25c9b5-dc73-4c3e-9433-f09d1f62ea8e) address of the printer.

**HardwareAddress (26 bytes):** A null-terminated Unicode string, which is the MAC address of the printer.

**DeviceType (514 bytes):** A null-terminated Unicode string, which is the generic SNMP device description ([**object identifier (OID)**](#gt_aaaf2f1a-0b0a-487e-a0f0-c3510a6091b2) 1.3.6.1.2.1.1.1).

**PaddingForAlignment (2 bytes):** 2 bytes of padding to align the **PortNumber** field on a 4-byte boundary. The contents of this field MUST be ignored.

**PortNumber (4 bytes):** A 32-bit unsigned integer that is the port number of the device.

**SNMPEnabled (4 bytes):** A 32-bit unsigned integer that MUST be nonzero if the device supports SNMP.

**SNMPDevIndex (4 bytes):** A 32-bit unsigned integer that is the SNMP device index.

##### PORT\_DATA\_2

The PORT\_DATA\_2 structure specifies **PORT\_DATA** level 2 data.[<182>](#Appendix_A_182)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| PortName (128 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Version | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Protocol | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Reserved | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| HostAddress (256 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SNMPCommunity (66 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | PaddingForAlignment | | | | | | | | | | | | | | | |
| DoubleSpool | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Queue (66 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | DeviceType (514 bytes) | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PortNumber | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SNMPEnabled | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SNMPDevIndex | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PortMonitorMibIndex | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**PortName (128 bytes):** A null-terminated Unicode string, which is the name of the port.

**Version (4 bytes):** A 32-bit unsigned integer that specifies the version number of the PORT\_DATA\_2 structure. This value MUST be 0x00000002.

**Protocol (4 bytes):** A 32-bit unsigned integer that specifies the protocol to use for the port. This value MUST be either **PROTOCOL\_RAWTCP\_TYPE** (0x00000001), indicating that the port expects RAW print data, or **PROTOCOL\_LPR\_TYPE** (0x00000002), indicating that the port expects to be driven as an LPR port.

| Value | Meaning |
| --- | --- |
| PROTOCOL\_RAWTCP\_TYPE  0x00000001 | The port expects RAW print data. |
| PROTOCOL\_LPR\_TYPE  0x00000002 | The port expects to be driven as an LPR port. |

**Size (4 bytes):** A 32-bit unsigned integer that specifies the size, in bytes, of the PORT\_DATA\_2 structure.

**Reserved (4 bytes):** A 32-bit unsigned integer that is set to zero.

**HostAddress (256 bytes):** A null-terminated Unicode string, which is the IP address or host name of the printer.

**SNMPCommunity (66 bytes):** A null-terminated Unicode string, which is the Simple Network Management Protocol (SNMP) [[RFC1157]](http://go.microsoft.com/fwlink/?LinkId=121934) community name of the printer.

**PaddingForAlignment (2 bytes):** 2 bytes of padding to align the **DoubleSpool** field on a 4-byte boundary. The contents of this field MUST be ignored.

**DoubleSpool (4 bytes):** A 32-bit unsigned integer that, if nonzero, indicates double spooling is enabled or, if zero, indicates that double spooling is disabled.

**Queue (66 bytes):** A null-terminated Unicode string, which is the LPR queue name.

**DeviceType (514 bytes):** A null-terminated Unicode string, which is the generic SNMP device description (object identifier (OID) 1.3.6.1.2.1.1.1).

**PortNumber (4 bytes):** A 32-bit unsigned integer that is the port number of the device.

**SNMPEnabled (4 bytes):** A 32-bit unsigned integer that, if nonzero, indicates that the device supports SNMP.

**SNMPDevIndex (4 bytes):** A 32-bit unsigned integer that is the SNMP device index.

**PortMonitorMibIndex (4 bytes):** A 32-bit unsigned integer that specifies the index in the network devices PWG port Monitor MIB for the current [**TCPMON**](#gt_7e4cced1-30e5-4baf-8848-8a050df1be4e) port. This index is used to query the IEEE 1284 device ID for the attached printer. For details, see [[IEEE1284]](http://go.microsoft.com/fwlink/?LinkId=89898).

##### PORT\_DATA\_LIST\_1

The PORT\_DATA\_LIST\_1 structure specifies an array of [PORT\_DATA\_2](#Section_2d256a4b819c4b848e8fb1edb1dcf0c2) structures.[<183>](#Appendix_A_183)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| Version | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PortDataNum | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PortData (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Version (4 bytes):** A 32-bit unsigned integer that specifies the version number of the PORT\_DATA\_LIST\_1 structure. This value MUST be 0x00000001.

**PortDataNum (4 bytes):** A 32-bit unsigned integer that is the number of PORT\_DATA\_2 structures contained in the **PortData** array of this PORT\_DATA\_LIST\_1 structure.

**PortData (variable):** An array of PORT\_DATA\_2 structures.

#### WSDMON Structures

##### WSD\_DRIVER\_DATA

The WSD\_DRIVER\_DATA structure holds information on the discovered printer.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| IEEE1284Id (520 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ModelName (520 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**IEEE1284Id (520 bytes):** A null-terminated Unicode string, which is the IEEE 1284 device ID for the discovered printer (for details, see [[IEEE1284]](http://go.microsoft.com/fwlink/?LinkId=89898)). This can be used to generate the correct [**Universal Plug and Play (UPnP)**](#gt_ca90ad1b-0ee4-41de-bb60-7839243bce61) ID for the printer driver.

**ModelName (520 bytes):** A null-terminated Unicode string, which is the name of the printer model discovered.

##### WSD\_BACKUP\_PORT\_DATA

The WSD\_BACKUP\_PORT\_DATA structure specifies information about the [**Web Services for Devices (WSD)**](#gt_f3d1ea15-54a3-4912-ae10-3a28ebf8c723) backup port.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| DiscoveryMethod | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| GlobalID (2048 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| RemoteURL (2048 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**DiscoveryMethod (4 bytes):** A 32-bit unsigned integer that specifies how the [**WSD**](#gt_734b2ac6-73db-47d5-ae14-213ddd6e958b) port was initially discovered.

| Value | Meaning |
| --- | --- |
| kMulticast  0x00000000 | The WSD port was initially discovered by using [**multicast discovery**](#gt_a0dad6fc-8bf0-4c7a-86f3-7197555b02b3). |
| kDirected  0x00000001 | The WSD port was initially discovered by using [**directed discovery**](#gt_36a0dc12-3449-4085-8af7-6a1e2b18f0ee). |

**GlobalID (2048 bytes):** A null-terminated [**Unicode string**](#gt_b069acb4-e364-453e-ac83-42d469bb339e) which, if **DiscoveryMethod** is kMulticast, specifies the **PKEY\_PNPX\_GlobalIdentify** of the device attached to the WSD port; otherwise, this field MUST be a NULL pointer.

**RemoteURL (2048 bytes):** A null-terminated Unicode string which, if **DiscoveryMethod** is kDirected, specifies the [**URL**](#gt_433a4fb7-ef84-46b0-ab65-905f5e3a80b1) of the device attached to the WSD port; otherwise, this field MUST be a NULL pointer.

##### WSD\_BACKUP\_PORT\_DATA\_EX

The WSD\_BACKUP\_PORT\_DATA\_EX structure specifies information to restore for the Web Services for Devices (WSD) port.[<184>](#Appendix_A_184)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| DiscoveryMethod | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| GlobalID (2048 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ServiceID (2048 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| RemoteURL (2048 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**DiscoveryMethod (4 bytes):** A 32-bit unsigned integer that specifies how the WSD port was initially discovered.

| Value | Meaning |
| --- | --- |
| kMulticast  0x00000000 | The WSD port was discovered by using multicast discovery. |
| kDirected  0x00000001 | The WSD port was discovered by using directed discovery. |

**GlobalID (2048 bytes):** A null-terminated Unicode string that specifies the **PKEY\_PNPX\_GlobalIdentify** of the device attached to the WSD port.

**ServiceID (2048 bytes):** A null-terminated Unicode string that specifies the **PKEY\_PNPX\_ServiceId** of the [**WS-Print**](#gt_db5db720-f17a-434f-8ecf-c236f368faeb) printer service within the device represented by the WSD port.

**RemoteURL (2048 bytes):** A null-terminated Unicode string which, if **DiscoveryMethod** is kDirected, specifies the URL of the device attached to the WSD port; otherwise, this field MUST be NULL.

### Constants

#### Access Values

The following table lists access level values that a print client can specify when opening a print job object, port object, printer object, or print server object. The type of object that an access value applies to is indicated in its name, as follows:

* Access values that are named starting with "JOB" are used for opening print job objects.
* Access values that are named starting with "PRINTER" are used for opening printer objects and port objects.
* Access values that are named starting with "SERVER" are used for opening print server objects.

The ACCESS\_MASK data type ([[MS-DTYP]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-DTYP%5d.pdf) section 2.4.3) specifies standard access rights and reserves a range of values for protocol-specific use. The following table defines printing-specific access rights in the reserved range and in combination with standard access rights. For more information concerning access rights, see [[MS-AZOD]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-AZOD%5d.pdf) section 1.1.1.5.

For completeness, the table lists all defined access values and identifies those that MUST NOT be used with the Print System Remote Protocol.

| Name/value | Description |
| --- | --- |
| JOB\_ACCESS\_ADMINISTER  0x00000010 | Printing-specific authorization to cancel, pause, resume, or restart the job ([MS-DTYP] ACCESS\_MASK Bit 27). |
| JOB\_ACCESS\_READ  0x00000020 | Printing-specific read rights for the spool file ([MS-DTYP] ACCESS\_MASK Bit 26).[<185>](#Appendix_A_185) |
| JOB\_EXECUTE  0x00020010 | Access rights for jobs combining **RC** (Read Control) of ACCESS\_MASK with printing-specific **JOB\_ACCESS\_ADMINISTER**.  This value MUST NOT be passed over the wire. If it is, the server SHOULD return ERROR\_ACCESS\_DENIED. |
| JOB\_READ  0x00020020 | Access rights for jobs combining **RC** (Read Control) of ACCESS\_MASK with printing-specific **JOB\_ACCESS\_READ**. |
| JOB\_WRITE  0x00020010 | Access rights for jobs combining **RC** (Read Control) of ACCESS\_MASK with printing-specific **JOB\_ACCESS\_ADMINISTER**.  This value MUST NOT be passed over the wire. If it is, the server SHOULD return ERROR\_ACCESS\_DENIED. |
| JOB\_ALL\_ACCESS  0x000F0030 | Access rights for printers to perform all administrative tasks and basic printing operations except SYNCHRONIZE ([MS-DTYP] ACCESS\_MASK Bit 'SY'). Combines STANDARD\_RIGHTS\_REQUIRED (ACCESS\_MASK Bits 'RC', 'DE', 'WD', 'WO'), JOB\_ACCESS\_ADMINISTER (ACCESS\_MASK Bit 27), and JOB\_ACCESS\_READ (ACCESS\_MASK Bit 26). |
| PRINTER\_ACCESS\_ADMINISTER  0x00000004 | Printing-specific access rights for printers to perform administrative tasks ([MS-DTYP] ACCESS\_MASK Bit 29). |
| PRINTER\_ACCESS\_USE  0x00000008 | Printing-specific access rights for printers to perform basic printing operations ([MS-DTYP] ACCESS\_MASK Bit 28). |
| PRINTER\_ACCESS\_MANAGE\_LIMITED  0x00000040 | Printing-specific access rights for printers to perform printer data management operations ([MS-DTYP] ACCESS\_MASK Bit 25).[<186>](#Appendix_A_186) |
| PRINTER\_ALL\_ACCESS  0x000F000C | Access rights for printers to perform all administrative tasks and basic printing operations except synchronization. Combines **WO** (Write Owner), **WD** (Write DACL), **RC** (Read Control), and **DE** (Delete) of ACCESS\_MASK with printing-specific **PRINTER\_ACCESS\_ADMINISTER** and printing-specific **PRINTER\_ACCESS\_USE**. |
| PRINTER\_EXECUTE  0x00020008 | Access rights for printers combining **RC** (Read Control) of ACCESS\_MASK with printing-specific **PRINTER\_ACCESS\_USE**. |
| PRINTER\_READ  0x00020008 | Access rights for printers combining **RC** (Read Control) of ACCESS\_MASK with printing-specific **PRINTER\_ACCESS\_USE**. |
| PRINTER\_WRITE  0x00020008 | Access rights for printers combining **RC** (Read Control) of ACCESS\_MASK with printing-specific **PRINTER\_ACCESS\_USE**. |
| SERVER\_ACCESS\_ADMINISTER  0x00000001 | Printing-specific access rights to administer print servers ([MS-DTYP] ACCESS\_MASK Bit 31). |
| SERVER\_ACCESS\_ENUMERATE  0x00000002 | Printing-specific access rights to enumerate print servers ([MS-DTYP] ACCESS\_MASK Bit 30). |
| SERVER\_ALL\_ACCESS  0x000F0003 | Access rights for print servers to perform all administrative tasks and basic printing operations except synchronization. Combines **WO** (Write Owner), **WD** (Write DACL), **RC** (Read Control), and **DE** (Delete) of ACCESS\_MASK with printing-specific **SERVER\_ACCESS\_ADMINISTER** and printing-specific **SERVER\_ACCESS\_ENUMERATE**. |
| SERVER\_EXECUTE  0x00020002 | Access rights for print servers combining **RC** (Read Control) of ACCESS\_MASK with printing-specific **SERVER\_ACCESS\_ENUMERATE**. |
| SERVER\_READ  0x00020002 | Access rights for print servers combining **RC** (Read Control) of ACCESS\_MASK with printing-specific **SERVER\_ACCESS\_ENUMERATE**. |
| SERVER\_WRITE  0x00020003 | Access rights for print servers combining **RC** (Read Control) of ACCESS\_MASK with printing-specific **SERVER\_ACCESS\_ADMINISTER** and printing-specific **SERVER\_ACCESS\_ENUMERATE**. |
| SPECIFIC\_RIGHTS\_ALL  0x0000FFFF | All specific rights.[<187>](#Appendix_A_187)  This value MUST NOT be passed over the wire. It SHOULD only be used locally, as a mask to determine the protocol-specific subset of access values. |
| STANDARD\_RIGHTS\_ALL  0x001F0000 | Combines **SY** (Synchronize), **WO** (Write Owner), **WD** (Write DACL), **RC** (Read Control), and **DE** (Delete) of ACCESS\_MASK.[<188>](#Appendix_A_188)  This value MUST NOT be passed over the wire. It SHOULD only be used locally, as a mask to determine the standard set of access values. |
| STANDARD\_RIGHTS\_EXECUTE  0x00020000 | Standard rights, set to **RC** (Read Control) of ACCESS\_MASK. |
| STANDARD\_RIGHTS\_READ  0x00020000 | Standard read rights, set to **RC** (Read Control) of ACCESS\_MASK. |
| STANDARD\_RIGHTS\_REQUIRED  0x000F0000 | Standard rights, combines **WO** (Write Owner), **WD** (Write DACL), **RC** (Read Control), and **DE** (Delete) of ACCESS\_MASK. |
| STANDARD\_RIGHTS\_WRITE  0x00020000 | Standard write rights, set to **RC** (Read Control) of ACCESS\_MASK. |
| SYNCHRONIZE  0x00100000 | The right to use the object for synchronization, set to **SY** (Synchronize) of ACCESS\_MASK.[<189>](#Appendix_A_189)  This value MUST NOT be passed over the wire. If it is, the server SHOULD return ERROR\_ACCESS\_DENIED. |
| DELETE  0x00010000 | The right to delete an object, set to **DE** (Delete) of ACCESS\_MASK. |
| READ\_CONTROL  0x00020000 | The right to read the information in the object's [**security descriptor**](#gt_e5213722-75a9-44e7-b026-8e4833f0d350), not including the information in the [**system access control list (SACL)**](#gt_c189801e-3752-4715-88f4-17804dad5782), set to **RC** (Read Control) of ACCESS\_MASK. |
| WRITE\_DAC  0x00040000 | The right to modify the [**discretionary access control list (DACL)**](#gt_d727f612-7a45-48e4-9d87-71735d62b321) in the object's security descriptor, set to **WD** (Write DACL) of ACCESS\_MASK. |
| WRITE\_OWNER  0x00080000 | The right to change the owner in the object's security descriptor, set to **WO** (Write Owner) of ACCESS\_MASK. |
| GENERIC\_READ  0x80000000 | **GR** (Generic Read) of ACCESS\_MASK:  For server object, access is mapped to **SERVER\_READ**.  For printer object, access is mapped to **PRINTER\_READ**.  For print job, access is mapped to **JOB\_READ**. |
| GENERIC\_WRITE  0x40000000 | **GW** (Generic Write) of ACCESS\_MASK:  For server object, access is mapped to **SERVER\_WRITE**.  For printer object, access is mapped to **PRINTER\_WRITE**.  For print job, access is mapped to **JOB\_WRITE**. |
| GENERIC\_EXECUTE  0x20000000 | **GX** (Generic Execute) of ACCESS\_MASK:  For server object, access is mapped to **SERVER\_EXECUTE**.  For printer object, access is mapped to **PRINTER\_EXECUTE**.  For print job, access is mapped to **JOB\_EXECUTE**. |
| GENERIC\_ALL  0x10000000 | **GA** (Generic All) of ACCESS\_MASK:  For server object, access is mapped to **SERVER\_ALL\_ACCESS**.  For printer object, access is mapped to **PRINTER\_ALL\_ACCESS**.  For print job, access is mapped to **JOB\_ALL\_ACCESS**. |

#### Change Notification Flags

Change Notification Flags specify change notification information and options.[<190>](#Appendix_A_190)

The following bit flag is set by a print server in the **Flags** member of an [RPC\_V2\_NOTIFY\_INFO structure (section 2.2.1.13.3)](#Section_79db335febe74a6f8533efdda32cc5d8).

| Name/value | Description |
| --- | --- |
| PRINTER\_NOTIFY\_INFO\_DISCARDED  0x00000001 | An overflow or error has occurred, and notifications have been lost. The print server MUST NOT send additional notifications until the client has made a call to [RpcRouterRefreshPrinterChangeNotification (section 3.1.4.10.5)](#Section_cb5d22ed0352430383d5b5afd32fb461). |

The following bit flags are set by a print client in the variable pointed to by the *pdwResult* parameter in a call to [RpcRouterReplyPrinterEx (section 3.2.4.1.4)](#Section_1b44cb8295b5485c9ec90360c3a3ecb0). They are used to indicate the results of processing the state of an RPC\_V2\_NOTIFY\_INFO structure.

These flags can be combined to specify multiple results.

| Name/Value | Description |
| --- | --- |
| PRINTER\_NOTIFY\_INFO\_DISCARDNOTED  0x00010000 | The client acknowledges receiving and processing the **PRINTER\_NOTIFY\_INFO\_DISCARDED** notification. |
| PRINTER\_NOTIFY\_INFO\_COLORMISMATCH  0x00080000 | The value of the *dwColor* parameter in a call to RpcRouterReplyPrinterEx does not match the value the client previously passed to the server in a call to RpcRouterRefreshPrinterChangeNotification. |

The following bit flag is set by a print client in the **Reserved** member of an [RPC\_V2\_NOTIFY\_OPTIONS structure (section 2.2.1.13.1)](#Section_4599c052e81441749bd500473329a9f8).

| Name/value | Description |
| --- | --- |
| PRINTER\_NOTIFY\_OPTIONS\_REFRESH  0x00000001 | Refreshed data is requested from the server for all monitored members. |

#### Job Notification Values

Job Notification Values specify types of changes in the **Data** member of an [RPC\_V2\_NOTIFY\_INFO\_DATA (section 2.2.1.13.4)](#Section_440ed0c046a4420193cdcc1e5d1fd022) structure.[<191>](#Appendix_A_191)**Reserved** members are specified by [Notification Data Type Values (section 2.2.3.5)](#Section_6e1ac55984a34b3ea89707159974661e), and **String** values are specified in [JOB\_INFO (section 2.2.2.6)](#Section_aed5b928e9d24c65b4136644531ab8d5) structures. The print server passes these notification values to a print client using [RpcRouterReplyPrinterEx (section 3.2.4.1.4)](#Section_1b44cb8295b5485c9ec90360c3a3ecb0).

| Constant/value | Description |
| --- | --- |
| JOB\_NOTIFY\_FIELD\_PRINTER\_NAME  0x0000 | Specifies that the printer name for the print job has changed.  **Reserved** contains **TABLE\_STRING**, and **Data.String** contains the new printer name (see **pPrinterName** in JOB\_INFO structures). |
| JOB\_NOTIFY\_FIELD\_MACHINE\_NAME  0x0001 | Specifies that the server name for the job has changed.  **Reserved** contains **TABLE\_STRING**, and **Data.String** contains the new server name (see **pMachineName** in JOB\_INFO structures). |
| JOB\_NOTIFY\_FIELD\_PORT\_NAME  0x0002 | Specifies that the port for the job has changed.  **Reserved** contains **TABLE\_STRING**, and **Data.String** contains a string specifying the new port the job is printed on (see **pPortName** in JOB\_INFO structures). |
| JOB\_NOTIFY\_FIELD\_USER\_NAME  0x0003 | Specifies that the user name for the job has changed.  **Reserved** contains **TABLE\_STRING**, and **Data.String** contains the new user name (see **pUserName** in JOB\_INFO structures). |
| JOB\_NOTIFY\_FIELD\_NOTIFY\_NAME  0x0004 | Specifies that the notify name for the job has changed.  **Reserved** contains **TABLE\_STRING**, and **Data.String** contains the new notify name (see **pNotifyName** in JOB\_INFO structures). |
| JOB\_NOTIFY\_FIELD\_DATATYPE  0x0005 | Specifies that the default data type for the job has changed.  **Reserved** contains **TABLE\_STRING**, and **Data.String** contains the new default data type (see **pDatatype** in JOB\_INFO structures). |
| JOB\_NOTIFY\_FIELD\_PRINT\_PROCESSOR  0x0006 | Specifies that the print processor associated with the job has changed.  **Reserved** contains **TABLE\_STRING**, and **Data.String** contains the new print processor name (see **pPrintProcessor** in JOB\_INFO structures). |
| JOB\_NOTIFY\_FIELD\_PARAMETERS  0x0007 | Specifies that the default print processor parameters for the job have changed.  **Reserved** contains **TABLE\_STRING**, and **Data.String** contains the new print processor parameters value (see **pParameters** in JOB\_INFO structures). |
| JOB\_NOTIFY\_FIELD\_DRIVER\_NAME  0x0008 | Specifies that the printer driver for the job has changed.  **Reserved** contains **TABLE\_STRING**, and **Data.String** contains the new printer driver name (see **pDriverName** in JOB\_INFO structures). |
| JOB\_NOTIFY\_FIELD\_DEVMODE  0x0009 | Specifies that the default [\_DEVMODE (section 2.2.2.1)](#Section_ccc2a501794e4d2bb312f69c75131c2e) structure for the job has changed.  **Reserved** contains **TABLE\_DEVMODE**, and **Data.DevMode** contains the new **DEVMODE** structure (see **pDevMode** in JOB\_INFO structures). |
| JOB\_NOTIFY\_FIELD\_STATUS  0x000A | Specifies that the status for the job has changed.  **Reserved** contains **TABLE\_DWORD**, and **Data.dwData[0]** contains the new **Status**. |
| JOB\_NOTIFY\_FIELD\_STATUS\_STRING  0x000B | The textual representation for the job status has changed.  **Reserved** contains **TABLE\_STRING**, and **Data.String** contains the new status string (see **pStatus** in JOB\_INFO structures). |
| JOB\_NOTIFY\_FIELD\_SECURITY\_DESCRIPTOR  0x000C | Security descriptor for the job has changed.  **Reserved** contains **TABLE\_SECURITYDESCRIPTOR**, and **Data.SecurityDescriptor** contains the new security descriptor (see **pSecurityDescriptor** in JOB\_INFO structures). |
| JOB\_NOTIFY\_FIELD\_DOCUMENT  0x000D | Specifies that the document name for the job has changed.  **Reserved** contains **TABLE\_STRING**, and **Data.String** contains the new document name (see **pDocument** in JOB\_INFO structures). |
| JOB\_NOTIFY\_FIELD\_PRIORITY  0x000E | Specifies that the current priority for the job has changed.  **Reserved** contains **TABLE\_DWORD**, and **Data.dwData[0]** contains the new **Priority** value. |
| JOB\_NOTIFY\_FIELD\_POSITION  0x000F | Specifies that the position in the queue for the job has changed.  **Reserved** contains **TABLE\_DWORD**, and **Data.dwData[0]** contains the new **Position** value. |
| JOB\_NOTIFY\_FIELD\_SUBMITTED  0x0010 | Specifies that the submitted time for the job has changed.  **Reserved** contains **TABLE\_TIME**, and **Data.SystemTime** contains the new **Submitted** value. |
| JOB\_NOTIFY\_FIELD\_START\_TIME  0x0011 | Specifies that the earliest start time for the job has changed.  **Reserved** contains **TABLE\_DWORD**, and **Data.dwData[0]** contains the new **StartTime** value. |
| JOB\_NOTIFY\_FIELD\_UNTIL\_TIME  0x0012 | Specifies that the latest print time for the job has changed.  **Reserved** contains **TABLE\_DWORD**, and **Data.dwData[0]** contains the new **UntilTime** value. |
| JOB\_NOTIFY\_FIELD\_TIME  0x0013 | Specifies that the total print time for the job has changed.  **Reserved** contains **TABLE\_DWORD**, and **Data.dwData[0]** contains the new **Time** value. |
| JOB\_NOTIFY\_FIELD\_TOTAL\_PAGES  0x0014 | Specifies that the total number of pages of the job has changed.  **Reserved** contains **TABLE\_DWORD**, and **Data.dwData[0]** contains the new **cTotalPages** value. |
| JOB\_NOTIFY\_FIELD\_PAGES\_PRINTED  0x0015 | Specifies that the number of pages that have been printed has changed.  **Reserved** contains **TABLE\_DWORD**, and **Data.dwData[0]** contains the new **PagesPrinted** value. |
| JOB\_NOTIFY\_FIELD\_TOTAL\_BYTES  0x0016 | Specifies that the total number of bytes of the job has changed.  **Reserved** contains **TABLE\_DWORD**, **Data.dwData[0]** contains the new **Size**, and **Data.dwData[1]** contains the new **SizeHigh** value. |
| JOB\_NOTIFY\_FIELD\_BYTES\_PRINTED  0x0017 | Specifies that the total number of bytes that have been printed has changed.  **Reserved** contains **TABLE\_DWORD**, **Data.dwData[0]** contains the 32 low-order bits of the number of bytes printed, and **Data.dwData[1]** contains the 32 high-order bits of the number of bytes printed. |

#### Server Notification Values

Server notification values specify types of changes in the **Data** member of an [RPC\_V2\_NOTIFY\_INFO\_DATA](#Section_440ed0c046a4420193cdcc1e5d1fd022) structure (section 2.2.1.13.4). **Reserved** members are specified by notification data type values (section [2.2.3.5](#Section_6e1ac55984a34b3ea89707159974661e)). The print server passes these notification values to a print client using [RpcRouterReplyPrinterEx (section 3.2.4.1.4)](#Section_1b44cb8295b5485c9ec90360c3a3ecb0).

| Constant/value | Description |
| --- | --- |
| SERVER\_NOTIFY\_FIELD\_PRINT\_DRIVER\_ISOLATION\_GROUP  0x0000 | Specifies that the printer driver isolation group setting for the print server has changed. **Reserved** contains **TABLE\_STRING** and **Data.String** contains the updated value of the print server's "PrintDriverIsolationGroups" configuration data. For details, see [Server Handle Key Values (section 2.2.3.10)](#Section_1393a76f585149a5bf147188ca6763b2). [<192>](#Appendix_A_192) |

#### Notification Data Type Values

| Constant/value | Description |
| --- | --- |
| TABLE\_DWORD  0x0001 | The notification data member MUST contain a two-DWORD array. |
| TABLE\_STRING  0x0002 | The notification data member MUST contain a string. |
| TABLE\_DEVMODE  0x0003 | The notification data member MUST be a non-NULL pointer to a [DEVMODE\_CONTAINER structure (section 2.2.1.2.1)](#Section_c2b14fe764794460a7ba633a845dd61a). |
| TABLE\_TIME  0x0004 | The notification data member MUST contain a [SYSTEMTIME\_CONTAINER structure (section 2.2.1.2.16)](#Section_1438066b4b56488898213e2432dc0f07). |
| TABLE\_SECURITYDESCRIPTOR  0x0005 | The notification data member MUST contain a [SECURITY\_CONTAINER structure (section 2.2.1.2.13)](#Section_6cf2b4ff1b2b453c87f985243823cea8). |

#### Printer Change Flags

Printer Change Flags specify changes to a printer configuration. These flags can be combined to specify multiple changes.

##### Printer Change Flags for Use with a Printer Handle

These flags are for use with a printer handle, and all other bits MUST be ignored when used with a printer handle:

| Constant/value | Description |
| --- | --- |
| PRINTER\_CHANGE\_SET\_PRINTER  0x00000002 | Printer object properties were configured. |
| PRINTER\_CHANGE\_DELETE\_PRINTER  0x00000004 | A printer object was deleted. |
| PRINTER\_CHANGE\_PRINTER  0x000000FF | A printer object changed in some way.  A client can use this value to indicate that it accepts all change notifications regarding printers. The server SHOULD set only the individual flags corresponding to the changes that actually occurred. |
| PRINTER\_CHANGE\_ADD\_JOB  0x00000100 | A print job was added. |
| PRINTER\_CHANGE\_SET\_JOB  0x00000200 | Print job properties were configured. |
| PRINTER\_CHANGE\_DELETE\_JOB  0x00000400 | A print job was deleted. |
| PRINTER\_CHANGE\_WRITE\_JOB  0x00000800 | A print job was written. |
| PRINTER\_CHANGE\_JOB  0x0000FF00 | A print job changed in some way.  A client can use this value to indicate that it accepts all change notifications regarding print jobs. The server SHOULD set only the individual flags corresponding to the changes that actually occurred. |
| PRINTER\_CHANGE\_SET\_PRINTER\_DRIVER  0x20000000 | A printer driver was specified.[<193>](#Appendix_A_193) |
| PRINTER\_CHANGE\_TIMEOUT  0x80000000 | Returned by [RpcWaitForPrinterChange (section 3.1.4.10.1)](#Section_e40cb68f679e4b84a33e86a94a141d83) if the implementation-specific timeout has expired. |
| PRINTER\_CHANGE\_ALL  0x7777FFFF | A change was made to one or more printer-related objects, including print job, form, port, **processor**, or printer driver, or to the printer object itself.  A client can use this value to indicate that it is interested in all change notifications. The server SHOULD set only the individual flags corresponding to the changes that actually occurred.[<194>](#Appendix_A_194) |
| PRINTER\_CHANGE\_ALL\_2  0x7F77FFFF | Identical with PRINTER\_CHANGE\_ALL (0x7777FFFF).[<195>](#Appendix_A_195) |

For more information about the rules governing printer change values, see section [2.2.4.13](#Section_f7e72f6c8ba345738a7edd12f2cff50e).

##### Printer Change Flags for Use with a Server Handle

These flags are for use with a server handle, and all other bits MUST be ignored when used with a server handle:

| Constant/value | Description |
| --- | --- |
| PRINTER\_CHANGE\_ADD\_PRINTER\_DRIVER  0x10000000 | A printer driver was added. |
| PRINTER\_CHANGE\_DELETE\_PRINTER\_DRIVER  0x40000000 | A printer driver was deleted. |
| PRINTER\_CHANGE\_PRINTER\_DRIVER  0x70000000 | A printer driver was changed in some way.  A client can use this to indicate that it accepts all change notifications regarding printer drivers. The server SHOULD set only the individual flags corresponding to the changes that actually occurred. |
| PRINTER\_CHANGE\_ADD\_FORM  0x00010000 | A form was added. |
| PRINTER\_CHANGE\_DELETE\_FORM  0x00040000 | A form was deleted. |
| PRINTER\_CHANGE\_SET\_FORM  0x00020000 | Form properties were configured. |
| PRINTER\_CHANGE\_FORM  0x00070000 | A form was changed in some way.  A client can use this to indicate that it accepts all change notifications regarding forms. The server SHOULD set only the individual flags corresponding to the changes that actually occurred. |
| PRINTER\_CHANGE\_ADD\_PORT  0x00100000 | A port was added. |
| PRINTER\_CHANGE\_CONFIGURE\_PORT  0x00200000 | Port properties were configured. |
| PRINTER\_CHANGE\_DELETE\_PORT  0x00400000 | A port was deleted. |
| PRINTER\_CHANGE\_PORT  0x00700000 | A port was changed in some way.  A client can use this to indicate that it accepts all change notifications regarding ports. The server SHOULD set only the individual flags corresponding to the changes that actually occurred. |
| PRINTER\_CHANGE\_ADD\_PRINT\_PROCESSOR  0x01000000 | A print processor was added. |
| PRINTER\_CHANGE\_DELETE\_PRINT\_PROCESSOR  0x04000000 | A print processor was deleted. |
| PRINTER\_CHANGE\_PRINT\_PROCESSOR  0x07000000 | The properties for a print processor were updated.  A client can use this to indicate that it accepts all change notifications regarding print processors. The server SHOULD set only the individual flags corresponding to the changes that actually occurred. |
| PRINTER\_CHANGE\_ADD\_PRINTER  0x00000001 | A printer object was added. |
| PRINTER\_CHANGE\_FAILED\_CONNECTION\_PRINTER  0x00000008 | A connection to a printer object failed.[<196>](#Appendix_A_196) |
| PRINTER\_CHANGE\_SERVER  0x08000000 | A change was made to one or more of the monitored server configuration settings.[<197>](#Appendix_A_197) |

For more information about the rules governing printer change values, see section [2.2.4.13](#Section_f7e72f6c8ba345738a7edd12f2cff50e).

#### Printer Enumeration Flags

Printer Enumeration Flags specify types of printers to enumerate. These flags can be combined to specify multiple printer types.

| Constant/value | Description |
| --- | --- |
| PRINTER\_ENUM\_LOCAL  0x00000002 | Enumerate local printer objects. |
| PRINTER\_ENUM\_CONNECTIONS  0x00000004 | Enumerate printer connections previously added through [RpcAddPerMachineConnection](#Section_3648c131b1554cc69a2d79b95b422a5b). |
| PRINTER\_ENUM\_NAME  0x00000008 | Enumerate printers on the print server, network [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca), or a specific [**print provider**](#gt_8ba4eb41-4adc-4272-ae1a-456870ad8cde). |
| PRINTER\_ENUM\_REMOTE  0x00000010 | Enumerate network printers and other print servers that are in the same domain as the print server. |
| PRINTER\_ENUM\_SHARED  0x00000020 | Only enumerate printers with the shared attribute set. This flag MUST be combined with one or more of the other flags. |
| PRINTER\_ENUM\_NETWORK  0x00000040 | Enumerate network printers that are in the same domain as the print server. |
| PRINTER\_ENUM\_EXPAND  0x00004000 | Indicates that the printer object contains further enumerable child objects. When a server enumerates print servers (see [RpcEnumPrinters (section 3.1.4.2.1)](#Section_0d83c60dec7546718a3b99517651ff1b) the server can set this bit for each enumerated server whose name matches the server's [**domain name**](#gt_45a1c9f1-0263-49a8-97c7-7aca1a99308c). |
| PRINTER\_ENUM\_CONTAINER  0x00008000 | Indicates that the printer object is capable of containing enumerable objects. One such object is a print provider, which is a print server that contains printers. |
| PRINTER\_ENUM\_ICON1  0x00010000 | Indicates that, where appropriate, an application treats the printer object as a top-level network name, such as Windows network. A GUI application can [<198>](#Appendix_A_198) choose to display an icon of choice for this type of object. |
| PRINTER\_ENUM\_ICON2  0x00020000 | Indicates that, where appropriate, an application treats an object as a network domain name. A GUI application can[<199>](#Appendix_A_199) choose to display an icon of choice for this type of object. |
| PRINTER\_ENUM\_ICON3  0x00040000 | Indicates that, where appropriate, an application treats an object as a print server. A GUI application can[<200>](#Appendix_A_200) choose to display an icon of choice for this type of object. |
| PRINTER\_ENUM\_ICON8  0x00800000 | Indicates that, where appropriate, an application treats an object as a print server. A GUI application can[<201>](#Appendix_A_201) choose to display an icon of choice for this type of object. |
| PRINTER\_ENUM\_HIDE  0x01000000 | Indicates that an application cannot display the printer object.[<202>](#Appendix_A_202) |

#### Printer Notification Values

Printer notification values specify types of changes in printer data and/or state for which print clients can be notified.

The following constants can be used in the *fdwOptions* parameters of the RpcRemoteFindFirstPrinterChangeNotification (section [3.1.4.10.3](#Section_b8b414d9f1cd4191bb6b87d09ab2fd83)) and RpcRemoteFindFirstPrinterChangeNotificationEx (section [3.1.4.10.4](#Section_eb66b2211c1f4249b8bcc5befec2314d)) methods. They specify the categories of printers for which change notifications are returned.

| Name/value | Description |
| --- | --- |
| 0x00000000 | Return notifications for 2D printers only. |
| PRINTER\_NOTIFY\_CATEGORY\_ALL  0x00010000 | Return notifications for both 2D and 3D printers.[<203>](#Appendix_A_203)[<204>](#Appendix_A_204) |
| PRINTER\_NOTIFY\_CATEGORY\_3D  0x00020000 | Return notifications for 3D printers only.[<205>](#Appendix_A_205) |

The following constants can be used in the **Data** member of an [RPC\_V2\_NOTIFY\_INFO\_DATA (section 2.2.1.13.4)](#Section_440ed0c046a4420193cdcc1e5d1fd022) structure.[<206>](#Appendix_A_206) **Reserved** members are specified by [Notification Data Type Values (section 2.2.3.5)](#Section_6e1ac55984a34b3ea89707159974661e), and **String** values are specified in [PRINTER\_INFO (section 2.2.1.10)](#Section_125fb29b3c5141fca333b1e0e8794a91) structures. The print server passes these notification values to a print client using [RpcRouterReplyPrinterEx (section 3.2.4.1.4)](#Section_1b44cb8295b5485c9ec90360c3a3ecb0).

| Name/value | Description |
| --- | --- |
| PRINTER\_NOTIFY\_FIELD\_ATTRIBUTES  0x000D | Specifies that printer attributes have changed.  **Reserved** contains **TABLE\_DWORD**, and **Data.dwData[0]** contains the new **Attributes** value. |
| PRINTER\_NOTIFY\_FIELD\_AVERAGE\_PPM  0x0015 | Specifies that the average pages per minute for the printer has changed.  **Reserved** contains **TABLE\_DWORD**, and **Data.dwData[0]** contains the new **AveragePPM** value. |
| PRINTER\_NOTIFY\_FIELD\_BRANCH\_OFFICE\_PRINTING  0x001C | Specifies that the EnableBranchOfficePrinting [printer data value (section 2.2.3.11)](#Section_4a81aa1c469e49418c4db8c4093b7fc2) for the printer has changed.  **Reserved** contains **TABLE\_DWORD**, and **Data.dwData[0]** contains the new value for the EnableBranchOfficePrinting printer data value.[<207>](#Appendix_A_207) |
| PRINTER\_NOTIFY\_FIELD\_BYTES\_PRINTED  0x0019 | Specifies that the number of bytes that have been printed has changed.  **Reserved** contains **TABLE\_DWORD**, and **Data.dwData[0]** contains the new **cTotalBytes** value. |
| PRINTER\_NOTIFY\_FIELD\_CJOBS  0x0014 | Specifies that the number of print jobs that are queued for the printer has changed.  **Reserved** contains **TABLE\_DWORD**, and **Data.dwData[0]** contains the new **cJobs** value. |
| PRINTER\_NOTIFY\_FIELD\_COMMENT  0x0005 | Specifies that the printer comment has changed.  **Reserved** contains **TABLE\_STRING**, and **Data.String** contains the new comment (see **pComment** in [PRINTER\_INFO](#Section_fc749a2738ec4063b0c1dc477c120ffd) structures). |
| PRINTER\_NOTIFY\_FIELD\_DATATYPE  0x000B | Specifies that the printer default data type has changed.  **Reserved** contains **TABLE\_STRING**, and **Data.String** contains the new comment (see **pDatatype** in PRINTER\_INFO structures). |
| PRINTER\_NOTIFY\_FIELD\_DEFAULT\_PRIORITY  0x000F | Specifies that the default priority for the printer has changed.  **Reserved** contains **TABLE\_DWORD**, and **Data.dwData[0]** contains the new **DefaultPriority**. |
| PRINTER\_NOTIFY\_FIELD\_DEVMODE  0x0007 | Specifies that the default [\_DEVMODE (section 2.2.2.1)](#Section_ccc2a501794e4d2bb312f69c75131c2e) structure for the printer has changed.  **Reserved** contains **TABLE\_DEVMODE**, and **Data.DevMode** contains the new **DEVMODE** structure (see **pDevMode** in PRINTER\_INFO structures). |
| PRINTER\_NOTIFY\_FIELD\_DRIVER\_NAME  0x0004 | Specifies that the printer driver for the printer has changed.  **Reserved** contains **TABLE\_STRING**, and **Data.String** contains the new printer driver name (see **pDriverName** in PRINTER\_INFO structures). |
| PRINTER\_NOTIFY\_FIELD\_LOCATION  0x0006 | Specifies that the printer location has changed.  **Reserved** contains **TABLE\_STRING**, and **Data.String** contains the new location description value (see **pLocation** in PRINTER\_INFO structures). |
| PRINTER\_NOTIFY\_FIELD\_OBJECT\_GUID  0x001A | Specifies that the printer object GUID has changed.[<208>](#Appendix_A_208)  **Reserved** contains **TABLE\_STRING**, and **Data.String** contains the new printer GUID value (see **pszObjectGUID** in PRINTER\_INFO structures). |
| PRINTER\_NOTIFY\_FIELD\_PAGES\_PRINTED  0x0017 | Specifies that the number of pages that have been printed for the printer has changed.  **Reserved** contains **TABLE\_DWORD**, and **Data.dwData[0]** contains the new **PagesPrinted** value. |
| PRINTER\_NOTIFY\_FIELD\_PARAMETERS  0x000A | Specifies that the default print processor parameters for the printer have changed.  **Reserved** contains **TABLE\_STRING**, and **Data.String** contains the new print processor parameters value (see **pParameters** in PRINTER\_INFO structures). |
| PRINTER\_NOTIFY\_FIELD\_PORT\_NAME  0x0003 | Specifies that the default port name for the printer has changed.  **Reserved** contains **TABLE\_STRING**, and **Data.String** contains the new port name value (see **pPortName** in PRINTER\_INFO structures). |
| PRINTER\_NOTIFY\_FIELD\_PRINTER\_NAME  0x0001 | Specifies that the printer name has changed.  **Reserved** contains **TABLE\_STRING**, and **Data.String** contains the new printer name value (see **pPrinterName** in PRINTER\_INFO structures). |
| PRINTER\_NOTIFY\_FIELD\_PRINT\_PROCESSOR  0x0009 | Specifies that the print processor associated with the printer has changed.  **Reserved** contains **TABLE\_STRING**, and **Data.String** contains the new print processor name value (see **pPrintProcessor** in PRINTER\_INFO structures). |
| PRINTER\_NOTIFY\_FIELD\_PRIORITY  0x000E | Specifies that the current priority for the printer has changed.  **Reserved** contains **TABLE\_DWORD**, and **Data.dwData[0]** contains the new **Priority** value. |
| PRINTER\_NOTIFY\_FIELD\_SECURITY\_DESCRIPTOR  0x000C | Specifies that the security descriptor for the printer has changed.  **Reserved** contains **TABLE\_SECURITYDESCRIPTOR**, and **Data.SecurityDescriptor** contains the new security descriptor value (see **pSecurityDescriptor** in PRINTER\_INFO structures). |
| PRINTER\_NOTIFY\_FIELD\_SEPFILE  0x0008 | Specifies that the separator page for the printer has changed.  **Reserved** contains **TABLE\_STRING**, and **Data.String** contains the new separator page value (see **pSepFile** in PRINTER\_INFO structures). |
| PRINTER\_NOTIFY\_FIELD\_SERVER\_NAME  0x0000 | Specifies that the server name for the printer has changed.  **Reserved** contains **TABLE\_STRING**, and **Data.String** contains the new server name value (see **pServerName** in PRINTER\_INFO structures). |
| PRINTER\_NOTIFY\_FIELD\_SHARE\_NAME  0x0002 | Specifies that the printer share name has changed.  **Reserved** contains **TABLE\_STRING**, and **Data.String** contains the new share name value (see **pShareName** in PRINTER\_INFO structures). |
| PRINTER\_NOTIFY\_FIELD\_START\_TIME  0x0010 | Specifies that the earliest start time for the printer has changed.  **Reserved** contains **TABLE\_DWORD**, and **Data.dwData[0]** contains the new **StartTime** value. |
| PRINTER\_NOTIFY\_FIELD\_STATUS  0x0012 | Specifies that the status for the printer has changed.  **Reserved** contains **TABLE\_DWORD**, and **Data.dwData[0]** contains the new **Status** value. |
| PRINTER\_NOTIFY\_FIELD\_TOTAL\_BYTES  0x0018 | Specifies that the total number of bytes that have been printed on the printer has changed.  **Reserved** contains **TABLE\_DWORD**, and **Data.dwData[0]** contains the new **cTotalBytes** value. |
| PRINTER\_NOTIFY\_FIELD\_TOTAL\_PAGES  0x0016 | Specifies that the total number of pages that have been printed on the printer has changed.  **Reserved** contains **TABLE\_DWORD**, and **Data.dwData[0]** contains the new **cTotalPagesPrinted** value. |
| PRINTER\_NOTIFY\_FIELD\_UNTIL\_TIME  0x0011 | Specifies that the latest print time for the printer has changed.  **Reserved** contains **TABLE\_DWORD**, and **Data.dwData[0]** contains the new **UntilTime** value. |

#### Registry Type Values

| Registry type name/value | Description |
| --- | --- |
| REG\_NONE  0x00000000 | No value type is defined. |
| REG\_SZ  0x00000001 | A string. |
| REG\_EXPAND\_SZ  0x00000002 | A string that can contain unexpanded references to environment variables, for example, "%PATH%". |
| REG\_BINARY  0x00000003 | Binary data in any form. |
| REG\_DWORD  0x00000004 | A 32-bit number. |
| REG\_DWORD\_LITTLE\_ENDIAN  0x00000004 | A 32-bit number in little-endian format; equivalent to REG\_DWORD. |
| REG\_DWORD\_BIG\_ENDIAN  0x00000005 | A 32-bit number in [**big-endian**](#gt_6f6f9e8e-5966-4727-8527-7e02fb864e7e) format. |
| REG\_LINK  0x00000006 | Symbolic link to a registry key. |
| REG\_MULTI\_SZ  0x00000007 | A REG\_MULTI\_SZ structure as specified in [[MS-RRP]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-RRP%5d.pdf) section 2.2.6. |
| REG\_RESOURCE\_LIST  0x00000008 | A [**device driver**](#gt_07b00435-bda8-4d5f-9403-ea2c40eb7e21) resource list. |
| REG\_QWORD  0x0000000B | A 64-bit number. |
| REG\_QWORD\_LITTLE\_ENDIAN  0x0000000B | A 64-bit number in little-endian format; equivalent to REG\_QWORD. |

#### Server Handle Key Values

Server Handle Key Values are used to store printer configuration data. The values named in the following table MUST be supported by print servers as follows:

* In a call to [RpcGetPrinterData](#Section_9c04a0f2adc443938df47c36a14183f5) or [RpcGetPrinterDataEx](#Section_74a9f79910bf4ef6a2a40dcd9c43fb5c), the *pValueName* parameter identifies the data that MUST be returned in the *pData* parameter. The value pointed to by *pValueName* MUST be one of the strings specified in the "Server handle key value name" column.

The specified registry type values are defined in section [2.2.3.9](#Section_25cce7007fcf4bb6a2f30f6d08430a55).

* If the "Read-write" column is checked, a print server SHOULD keep track of the value set by a call to [RpcSetPrinterData](#Section_e0d3e41e097c459da7fbacaf1343ea27) or [RpcSetPrinterDataEx](#Section_58520b8b44ce46fc90d30362457098e0) in the *pData* parameter, and it SHOULD return the same value in a subsequent call to RpcGetPrinterData or RpcGetPrinterDataEx.
* The "Meaning" column describes the printer configuration data that is associated with the server handle key value name, and in some cases it specifies print server behavior. If the "Behavior optional" column is checked, that behavior is implementation-specific and not mandatory.

| Server handle key value name/registry type | Read-write | Behavior optional | Meaning |
| --- | --- | --- | --- |
| "Architecture"  REG\_SZ |  |  | A string that specifies the OS [Environment Name (section 2.2.4.4)](#Section_ad7758938a194786ab4634731f52d1af), as determined by the processor architecture. |
| "BeepEnabled"  REG\_DWORD | X | X | If this value is nonzero, the print server issues a beep sound on the local console. |
| "DefaultSpoolDirectory"  REG\_SZ | X |  | The [**UNC**](#gt_c9507dca-291d-4fd6-9cba-a9ee7da8c908) path for the directory in which the print server stores spooled print jobs. |
| "DNSMachineName"  REG\_SZ |  |  | [**Domain Name System (DNS)**](#gt_604dcfcd-72f5-46e5-85c1-f3ce69956700) computer name. |
| "DsPresent"  REG\_DWORD |  |  | 0x0001 if the print server is joined to a domain with directory services, zero if not. |
| "DsPresentForUser"  REG\_DWORD |  |  | 0x0001 if the user is logged on to a domain with directory services, zero if not. |
| "EventLog"  REG\_DWORD | X | X | A bit mask specifying which events for a print server to log in its internal event log. It is a bitwise OR of zero or more of the [Event Log Flags (section 2.2.3.10.4)](#Section_34284809abff4ecea5ad71f47160e5e4). |
| "MajorVersion"  REG\_DWORD |  |  | The major OS version. See **dwMajorVersion** in [SPLCLIENT\_INFO Members (section 2.2.1.3.7)](#Section_fae766fc31d343cc902f8e6cc4c75f31) for details. |
| "MinorVersion"  REG\_DWORD |  |  | The minor OS version. See **dwMinorVersion** in SPLCLIENT\_INFO Members (section 2.2.1.3.7) for details. |
| "NetPopup"  REG\_DWORD | X | X | If this value is nonzero, the print server MAY alert the print client of the status of a print job.[<209>](#Appendix_A_209) |
| "NetPopupToComputer"  REG\_DWORD | X | X | If this value is nonzero, the print server MAY alert the print client of changes to the status of a print job.[<210>](#Appendix_A_210) |
| "OSVersion"  REG\_BINARY |  |  | OS version information, in the form of an [OSVERSIONINFO structure (section 2.2.3.10.1)](#Section_5942648eb54f4e22a0e22d000e084b23). |
| "OSVersionEx"  REG\_BINARY |  |  | Extended OS version information, in the form of an [OSVERSIONINFOEX structure (section 2.2.3.10.2)](#Section_cd8e00e84c444e4f98d30db33d75d06d). |
| "PortThreadPriority"  REG\_DWORD | X | X | The current priority of the thread on which the print server sends data to printers. This value MUST be one of the [Thread Priority values (section 2.2.3.10.6)](#Section_f0f64e02a00a4b72832f4cc9a035d8de) constants.  This key name MAY be used remotely.[<211>](#Appendix_A_211) |
| "PortThreadPriorityDefault"  REG\_DWORD |  | X | The default priority of the thread on which the print server sends data to printers. This value MUST be one of the Thread Priority values constants.  This key name MAY be used remotely.[<212>](#Appendix_A_212) |
| "RemoteFax"  REG\_DWORD |  |  | 0x0001 if the Microsoft Fax printer driver supports remote clients, zero otherwise. If this value is zero, print client connections to the Microsoft Fax printer driver SHOULD NOT be made. |
| "RestartJobOnPoolEnabled"  REG\_DWORD | X |  | A nonzero value indicates that **SPLREG\_RESTART\_JOB\_ON\_POOL\_ERROR** is enabled.  This key name MAY be used remotely.[<213>](#Appendix_A_213) |
| "RestartJobOnPoolError"  REG\_DWORD | X |  | The minimum time, in seconds, when a print job is restarted on another port after an error occurs.  This key name MAY be used remotely.[<214>](#Appendix_A_214) |
| "RetryPopup"  REG\_DWORD | X | X | If this value is nonzero, the print server MAY offer the print client an option to retry a print job.[<215>](#Appendix_A_215)  This key name MAY be used remotely.[<216>](#Appendix_A_216) |
| "SchedulerThreadPriority"  REG\_DWORD | X | X | The current priority of the thread on which the print server schedules jobs for sending to printers. This value MUST be one of the Thread Priority values constants.  This key name MAY be used remotely.[<217>](#Appendix_A_217) |
| "SchedulerThreadPriorityDefault"  REG\_DWORD |  | X | The default priority of the thread on which the print server schedules jobs for sending to printers. This value MUST be one of the Thread Priority values constants.  This key name MAY be used remotely.[<218>](#Appendix_A_218) |
| "W3SvcInstalled"  REG\_DWORD |  |  | 0x0001 if the web printing services are installed on the machine that hosts the print server. |
| "PrintDriverIsolationGroups"  REG\_SZ | X | X | A string that specifies groups of printer driver names.[<219>](#Appendix_A_219) The printer drivers in each group are executed in the same process, but printer drivers in different groups are executed in separate processes. The format of the string is as follows:   * Printer driver names within a group are separated from each other by a single backslash "\". * Groups are separated from each other by two backslashes "\\". * The last group is terminated by a null character. |
| "PrintDriverIsolationTimeBeforeRecycle"  REG\_DWORD | X | X | A time in milliseconds that specifies the maximum time span a printer driver isolation process should be used for before it is shut down and restarted; the shut down and restart sequence reclaims memory potentially leaked by drivers.[<220>](#Appendix_A_220) |
| "PrintDriverIsolationMaxobjsBeforeRecycle"  REG\_DWORD | X | X | A count that specifies the maximum number of operations a printer driver isolation process should be used for before it is shut down and restarted; the shut down and restart sequence reclaims memory potentially leaked by drivers.[<221>](#Appendix_A_221) |
| "PrintDriverIsolationIdleTimeout"  REG\_DWORD | X | X | A time in milliseconds that specifies the maximum time a printer driver isolation process should remain idle before it is shut down.[<222>](#Appendix_A_222) |
| "PrintDriverIsolationExecutionPolicy"  REG\_DWORD |  | X | An integer that specifies if printer driver isolation is enabled on the print server.  0x00000000 indicates that printer driver isolation is disabled.  0x00000001 indicates that printer driver isolation is enabled.[<223>](#Appendix_A_223) |
| "PrintDriverIsolationOverrideCompat"  REG\_DWORD |  | X | An integer value that specifies if the print server overrides the printer driver's indication of printer driver isolation compatibility.[<224>](#Appendix_A_224)  0x00000001 indicates that the print server runs the printer driver in isolation mode, even if they do not indicate that they are compatible through the [PRINTER\_DRIVER\_SANDBOX\_ENABLED](#Section_937d5fc867f145f8ab878b56a8fca558) driver attribute flag. |
| "V4DriverDisallowPrinterUIApp"  REG\_DWORD | X | X | An integer value that specifies if printer drivers with a driver version (see *cVersion* in section [2.2.1.3.1](#Section_4464eaf0f34f40d5b970736437a21913)) of 0x00000004 are enabled to run their [**printer UI applications**](#gt_c6573f31-a905-4f22-bb69-5fc48d91ee92), if any. [<225>](#Appendix_A_225)  If the value is not configured, or if the value is set to 0x00000000, running of printer UI applications is enabled.  If the value is set to 0x00000001, running of printer UI applications is disabled.  This value has no effect on drivers with a driver version other than 0x00000004. By default this value is not configured. |

##### OSVERSIONINFO

The OSVERSIONINFO structure specifies operating system (OS) version information for use with [Server Handle Key Values (section 2.2.3.10)](#Section_1393a76f585149a5bf147188ca6763b2).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| dwOSVersionInfoSize | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwMajorVersion | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwMinorVersion | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwBuildNumber | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwPlatformId | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| szCSDVersion (256 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**dwOSVersionInfoSize (4 bytes):** The size of the OSVERSIONINFO structure in bytes.

**dwMajorVersion (4 bytes):** The major OS version. See **dwMajorVersion** in [SPLCLIENT\_INFO Members (section 2.2.1.3.7)](#Section_fae766fc31d343cc902f8e6cc4c75f31) for details.

**dwMinorVersion (4 bytes):** The minor OS version. See **dwMinorVersion** in SPLCLIENT\_INFO Members (section 2.2.1.3.7) for details.

**dwBuildNumber (4 bytes):** The build number of the OS.[<226>](#Appendix_A_226)

**dwPlatformId (4 bytes):** The OS platform. See **wProcessorArchitecture** in SPLCLIENT\_INFO Members (section 2.2.1.3.7) for details.

**szCSDVersion (256 bytes):** A maintenance string for Microsoft Product Support Services (PSS) use.

##### OSVERSIONINFOEX

The OSVERSIONINFOEX structure specifies extended operating system (OS) version information for use with [Server Handle Key Values (section 2.2.3.10)](#Section_1393a76f585149a5bf147188ca6763b2).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| OSVersionInfo (276 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| wServicePackMajor | | | | | | | | | | | | | | | | wServicePackMinor | | | | | | | | | | | | | | | |
| wSuiteMask | | | | | | | | | | | | | | | | wProductType | | | | | | | | wReserved | | | | | | | |

**OSVersionInfo (276 bytes):** An [OSVERSIONINFO structure (section 2.2.3.10.1)](#Section_5942648eb54f4e22a0e22d000e084b23), which specifies basic OS version information.

**wServicePackMajor (2 bytes):** The major version number of the latest Service Pack installed on the system. For example, for Service Pack 3, the major version number is 3. If no Service Pack has been installed, the value is zero.

**wServicePackMinor (2 bytes):** The minor version number of the latest Service Pack installed on the system. For example, for Service Pack 3, the minor version number is 0.

**wSuiteMask (2 bytes):** A value that identifies the product suites available on the system, consisting of [Product Suite Flags (section 2.2.3.10.5)](#Section_90be334ae4ab46a0ba72ec6ed8706606).

**wProductType (1 byte):** Additional information about the OS, which MUST be an [OS\_TYPE enumeration (section 2.2.3.10.3)](#Section_682158a5b093400b8dd9b31f3fdd04d5) value.

**wReserved (1 byte):** A field that SHOULD be initialized to zero when sent and MUST be ignored on receipt.

##### OS\_TYPE Enumeration

The OS\_TYPE enumeration specifies information about the operating system (OS) type for use with [Server Handle Key Values (section 2.2.3.10)](#Section_1393a76f585149a5bf147188ca6763b2).

1. typedef enum
2. {
3. VER\_NT\_WORKSTATION = 0x00000001,
4. VER\_NT\_DOMAIN\_CONTROLLER = 0x00000002,
5. VER\_NT\_SERVER = 0x00000003
6. } OS\_TYPE;

**VER\_NT\_WORKSTATION:** The OS is a Windows NT operating system workstation.[<227>](#Appendix_A_227)

**VER\_NT\_DOMAIN\_CONTROLLER:** The OS is a Windows NT [**domain controller**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd).[<228>](#Appendix_A_228)

**VER\_NT\_SERVER:** The OS is a Windows NT server.[<229>](#Appendix_A_229) A server that is also a domain controller is reported as **VER\_NT\_DOMAIN\_CONTROLLER**, not **VER\_NT\_SERVER**.

##### Event Log Flags

The Event Log Flags specify events for a print server to log in its internal event log, for use with [Server Handle Key Values (section 2.2.3.10)](#Section_1393a76f585149a5bf147188ca6763b2). These flags can be combined to specify multiple options.

| Constant/value | Description |
| --- | --- |
| EVENTLOG\_ERROR\_TYPE  0x00000001 | An event that indicates a significant problem such as loss of data or loss of functionality. For example, a service fails to load during startup. |
| EVENTLOG\_WARNING\_TYPE  0x00000002 | An event that is not necessarily significant, but may indicate a possible future problem. For example, disk space is low.  If an application can recover from an event without loss of functionality or data, it can classify the event as a warning. |
| EVENTLOG\_INFORMATION\_TYPE  0x00000004 | An event that describes the successful operation of an application, driver, or service. For example, a network driver loads successfully. |
| EVENTLOG\_AUDIT\_SUCCESS  0x00000008 | An event that records an audited security access attempt that is successful. For example, a user's attempt to log onto a system is successful. |
| EVENTLOG\_AUDIT\_FAILURE  0x00000010 | An event that records an audited security access attempt that fails. For example, a user's attempt to access a network drive fails. |

##### Product Suite Flags

The Product Suite Flags are implementation-specific values for the product suites that are available on the operating system (OS). They are used with [Server Handle Key Values (section 2.2.3.10)](#Section_1393a76f585149a5bf147188ca6763b2).[<230>](#Appendix_A_230)

##### Thread Priority Values

The Thread Priority values specify priorities for threads on which print servers schedule jobs or send data to printers, for use with [Server Handle Key Values (section 2.2.3.10)](#Section_1393a76f585149a5bf147188ca6763b2).

| Constant/value | Description |
| --- | --- |
| THREAD\_PRIORITY\_LOWEST  0xFFFFFFFE | The thread can be scheduled after threads with any other priority. |
| THREAD\_PRIORITY\_BELOW\_NORMAL  0xFFFFFFFF | The thread can be scheduled after threads with normal priority and before those with lowest priority. |
| THREAD\_PRIORITY\_NORMAL  0x00000000 | The thread can be scheduled after threads with above normal priority and before those with below normal priority. Threads have normal priority by default. |
| THREAD\_PRIORITY\_ABOVE\_NORMAL  0x00000001 | The thread can be scheduled after threads with highest priority and before those with normal priority. |
| THREAD\_PRIORITY\_HIGHEST  0x00000002 | The thread can be scheduled before threads with any other priority. |

#### Printer Data Values

Printer Data Values are used to store printer configuration data. The values named in the following table MUST be supported on print servers for each printer as follows:

* In a call to [RpcGetPrinterData (section 3.1.4.2.7)](#Section_9c04a0f2adc443938df47c36a14183f5) or [RpcGetPrinterDataEx (section 3.1.4.2.19)](#Section_74a9f79910bf4ef6a2a40dcd9c43fb5c), the *pValueName* parameter identifies the data that MUST be returned in the *pData* parameter. The key name pointed to by *pKeyName* MUST be NULL, and the value pointed to by *pValueName* MUST be one of the strings specified in the "Printer data value name" column.

The specified registry type values are defined in section [2.2.3.9](#Section_25cce7007fcf4bb6a2f30f6d08430a55).

* If the "Read-write" column is checked, a print server SHOULD keep track of the value set by a call to [RpcSetPrinterData (section 3.1.4.2.8)](#Section_e0d3e41e097c459da7fbacaf1343ea27) or [RpcSetPrinterDataEx (section 3.1.4.2.18)](#Section_58520b8b44ce46fc90d30362457098e0) in the *pData* parameter, and it SHOULD return the same value in a subsequent call to RpcGetPrinterData or RpcGetPrinterDataEx.
* The "Meaning" column describes the printer configuration data that is associated with the printer data value name, and in some cases it specifies print server behavior. If the "Behavior optional" column is checked, that behavior is implementation-specific and not mandatory.[<231>](#Appendix_A_231)

| Printer data value name / registry type | Read-write | Behavior optional | Meaning |
| --- | --- | --- | --- |
| "HardwareId"  REG\_SZ |  | X | A string that identifies compatible printer drivers for the printer. The value of this identifier is implementation-specific.[<232>](#Appendix_A_232) |
| "EnableBranchOfficePrinting" REG\_DWORD |  | X | An integer that specifies whether print clients connect directly to the printer. If the value exists and contains a REG\_DWORD value of 0x00000001, print clients render print jobs and send printed data directly to the printer instead of using the [Job Management Methods (section 3.1.4.3)](#Section_bd8e1517a5b14850933866575d2b8cf4) and [Document Printing Methods (section 3.1.4.9)](#Section_93e6c5a794d6487ebe20d2c91e5277ad) on the print server.[<233>](#Appendix_A_233) |
| "SeparatorFileData" |  | X | If the printer driver for the printer has a driver version of 0x00000004, and the EnableBranchOfficePrinting printer data value exists and contains a REG\_DWORD value of 0x00000001, this value contains the contents of the separator page, if any, associated with the printer.[<234>](#Appendix_A_234) |
| "V4\_Driver\_Hardware\_IDs"  REG\_MULTI\_SZ |  |  | A multisz containing one or more curly-braced GUID strings that identify the compatible printer drivers with a driver version (see *cVersion* in section [2.2.1.3.1](#Section_4464eaf0f34f40d5b970736437a21913)) of 0x00000004.[<235>](#Appendix_A_235)  This value is required for printers with drivers having a driver version of 0x00000004. |
| "XpsFormat"  REG\_BINARY |  | X | A binary value that specifies the XML Paper Specification (XPS) formats that the server supports: Microsoft XPS [[MSFT-XMLPAPER]](http://go.microsoft.com/fwlink/?LinkId=90214) and/or OpenXPS [[ECMA-388]](http://go.microsoft.com/fwlink/?LinkId=158846).[<236>](#Appendix_A_236)  The binary value MUST be either one of the following two:   * A single DWORD (4 bytes) if only one XPS format is supported. * A pair of DWORDs (8 bytes) if both XPS formats are supported.   Each DWORD MUST contain one of the following two values:   * 0x00000001 for Microsoft XPS. * 0x00000002 for OpenXPS.   When both formats are supported, the two DWORD values can be specified in any order. |
| "MergedData"  REG\_BINARY |  |  | A binary value that specifies the contents of the GPD ([[MSDN-GPDFiles]](http://go.microsoft.com/fwlink/?LinkId=221046)) or PPD ([[PS-PPD4.3]](http://go.microsoft.com/fwlink/?LinkId=221049)) file for a printer driver with a driver version of 0x00000004.[<237>](#Appendix_A_237)  This value is required for printers with printer drivers having a driver version of 0x00000004. |
| "MergedDataName"  REG\_SZ |  |  | A string value that specifies the name of the GPD ([MSDN-GPDFiles] or PPD ([PS-PPD4.3]) file for a printer driver with a driver version of 0x00000004.[<238>](#Appendix_A_238)  This value is required for printers with printer drivers having a driver version of 0x00000004. |
| "BranchOfficeLoggingEnabled"  REG\_DWORD |  | X | An integer that specifies whether branch office print remote logging is enabled, which means that print clients operating in [**branch office print mode**](#gt_88d553c4-8182-4c27-8e7f-9f66c480215e) SHOULD log printing-related Windows Events on the print server.[<239>](#Appendix_A_239)  If this value is not configured, or if it is set to 0x00000001, the client SHOULD create printing events on the server as well as the client.  If the value is set to zero, the client SHOULD NOT create printing events on the server. |
| "BranchOfficeOfflineLogSize"  REG\_DWORD |  | X | The maximum size, in megabytes (MB), of the **Branch Office Print Remote Log Offline Archive** (section [3.2.1](#Section_fa5a51d585cc40dc951a1e288b62d820)).[<240>](#Appendix_A_240) This value MUST be 1 MB or more.  If the print client is unable to contact the print server when processing a branch office print job, and branch office print remote logging is enabled, the printing event SHOULD be written to the **Branch Office Print Remote Log Offline Archive**. If the offline archive exceeds this maximum size, printing events SHOULD be discarded, and the server, when it can be contacted, SHOULD be informed of the overflow condition.  If this value is not configured, an implementation-defined default value SHOULD[<241>](#Appendix_A_241) be used.  If this value is set to zero, the client SHOULD NOT create printing events on the server. |
| "MinimumSupportedClientBuild"  REG\_DWORD |  | X | An integer value that specifies the minimum print client version required in order to connect to this printer. This is expressed as a Windows build number and is provided by printer drivers having a version of 0x00000004 which have known dependencies that cannot be met on clients with versions prior to this build number. If the build number of the print client is less than this value, the client MUST NOT create a connection to this printer. If this value is not supplied or is zero, any client, regardless of build number, can connect to this printer. |

#### Status and Attribute Values

| Printer status | Description |
| --- | --- |
| PRINTER\_STATUS\_BUSY  0x00000200 | The printer is busy. |
| PRINTER\_STATUS\_DOOR\_OPEN  0x00400000 | The printer door is open. |
| PRINTER\_STATUS\_ERROR  0x00000002 | The printer is in an error state. |
| PRINTER\_STATUS\_INITIALIZING  0x00008000 | The printer is initializing. |
| PRINTER\_STATUS\_IO\_ACTIVE  0x00000100 | The printer is in an active input or output state. |
| PRINTER\_STATUS\_MANUAL\_FEED  0x00000020 | The printer is in a manual feed state. |
| PRINTER\_STATUS\_NOT\_AVAILABLE  0x00001000 | The printer is not available for printing. |
| PRINTER\_STATUS\_NO\_TONER  0x00040000 | The printer is out of toner. |
| PRINTER\_STATUS\_OFFLINE  0x00000080 | The printer is offline. |
| PRINTER\_STATUS\_OUTPUT\_BIN\_FULL  0x00000800 | The printer's output bin is full. |
| PRINTER\_STATUS\_OUT\_OF\_MEMORY  0x00200000 | The printer has run out of memory. |
| PRINTER\_STATUS\_PAGE\_PUNT  0x00080000 | The printer cannot print the current page. |
| PRINTER\_STATUS\_PAPER\_JAM  0x00000008 | Paper is stuck in the printer. |
| PRINTER\_STATUS\_PAPER\_OUT  0x00000010 | The printer is out of paper. |
| PRINTER\_STATUS\_PAPER\_PROBLEM  0x00000040 | The printer has an unspecified paper problem. |
| PRINTER\_STATUS\_PAUSED  0x00000001 | The printer is paused. |
| PRINTER\_STATUS\_PENDING\_DELETION  0x00000004 | The printer is being deleted as a result of a client's call to [RpcDeletePrinter](#Section_e46c727493db443e8787ad49e5a08466). No new jobs can be submitted on existing printer objects for that printer. |
| PRINTER\_STATUS\_POWER\_SAVE  0x01000000 | The printer is in power-save mode.[<242>](#Appendix_A_242) |
| PRINTER\_STATUS\_PRINTING  0x00000400 | The printer is printing. |
| PRINTER\_STATUS\_PROCESSING  0x00004000 | The printer is processing a print job. |
| PRINTER\_STATUS\_SERVER\_OFFLINE  0x02000000 | The printer is offline.[<243>](#Appendix_A_243) |
| PRINTER\_STATUS\_SERVER\_UNKNOWN  0x00800000 | The printer status is unknown.[<244>](#Appendix_A_244) |
| PRINTER\_STATUS\_TONER\_LOW  0x00020000 | The printer is low on toner. |
| PRINTER\_STATUS\_USER\_INTERVENTION  0x00100000 | The printer has an error that requires the user to do something. |
| PRINTER\_STATUS\_WAITING  0x00002000 | The printer is waiting. |
| PRINTER\_STATUS\_WARMING\_UP  0x00010000 | The printer is warming up. |

| Printer attribute | Description |
| --- | --- |
| PRINTER\_ATTRIBUTE\_DEFAULT  0x00000004 | Indicates the printer is the default printer in the system. |
| PRINTER\_ATTRIBUTE\_DIRECT  0x00000002 | Job is sent directly to the printer (it is not spooled). |
| PRINTER\_ATTRIBUTE\_DO\_COMPLETE\_FIRST  0x00000200 | If set and printer is set for print-while-spooling, any jobs that have completed spooling are scheduled to print before jobs that have not completed spooling.[<245>](#Appendix_A_245) |
| PRINTER\_ATTRIBUTE\_ENABLE\_BIDI  0x00000800 | Indicates whether bidirectional communications are enabled for the printer.[<246>](#Appendix_A_246) |
| PRINTER\_ATTRIBUTE\_ENABLE\_DEVQ  0x00000080 | Setting this flag causes mismatched documents to be held in the queue.[<247>](#Appendix_A_247) |
| PRINTER\_ATTRIBUTE\_FAX  0x00004000 | If set, printer is a fax printer. |
| PRINTER\_ATTRIBUTE\_KEEPPRINTEDJOBS  0x00000100 | If set, jobs are kept after they are printed. If cleared, jobs are deleted.[<248>](#Appendix_A_248) |
| PRINTER\_ATTRIBUTE\_LOCAL  0x00000040 | Printer is a local printer. |
| PRINTER\_ATTRIBUTE\_NETWORK  0x00000010 | Printer is a network printer connection. |
| PRINTER\_ATTRIBUTE\_PUBLISHED  0x00002000 | Indicates whether the printer is published in the directory service (DS).[<249>](#Appendix_A_249) |
| PRINTER\_ATTRIBUTE\_QUEUED  0x00000001 | If set, the printer spools and starts printing after the last page is spooled. If cleared, and **PRINTER\_ATTRIBUTE\_DIRECT** is not set, the printer spools and prints while spooling. |
| PRINTER\_ATTRIBUTE\_RAW\_ONLY  0x00001000 | Indicates that only RAW data type print jobs MUST be spooled.[<250>](#Appendix_A_250) |
| PRINTER\_ATTRIBUTE\_SHARED  0x00000008 | Printer is shared. |
| PRINTER\_ATTRIBUTE\_TS  0x00008000 | Printer is a redirected terminal server printer. |
| PRINTER\_ATTRIBUTE\_WORK\_OFFLINE  0x00000400 | Indicates whether the printer is currently connected. If the printer is not currently connected, print jobs will continue to spool.[<251>](#Appendix_A_251) |

| Job status | Description |
| --- | --- |
| JOB\_STATUS\_BLOCKED\_DEVQ  0x00000200 | Printer driver cannot print the job.[<252>](#Appendix_A_252) |
| JOB\_STATUS\_COMPLETE  0x00001000 | Job has been delivered to the printer. |
| JOB\_STATUS\_DELETED  0x00000100 | Job has been deleted.[<253>](#Appendix_A_253) |
| JOB\_STATUS\_DELETING  0x00000004 | Job is being deleted. |
| JOB\_STATUS\_ERROR  0x00000002 | An error is associated with the job. |
| JOB\_STATUS\_OFFLINE  0x00000020 | Printer is offline. |
| JOB\_STATUS\_PAPEROUT  0x00000040 | Printer is out of paper. |
| JOB\_STATUS\_PAUSED  0x00000001 | Job is paused. |
| JOB\_STATUS\_PRINTED  0x00000080 | Job has printed. |
| JOB\_STATUS\_PRINTING  0x00000010 | Job is printing. |
| JOB\_STATUS\_RESTART  0x00000800 | Job has been restarted.[<254>](#Appendix_A_254) |
| JOB\_STATUS\_SPOOLING  0x00000008 | Job is spooling. |
| JOB\_STATUS\_USER\_INTERVENTION  0x00000400 | Printer has an error that requires the user to do something.[<255>](#Appendix_A_255) |

#### BIDI\_TYPE Enumeration

The BIDI\_TYPE enumeration specifies the type of data transferred in a bidirectional operation.

1. typedef enum
2. {
3. BIDI\_NULL = 0x00000000,
4. BIDI\_INT = 0x00000001,
5. BIDI\_FLOAT = 0x00000002,
6. BIDI\_BOOL = 0x00000003,
7. BIDI\_STRING = 0x00000004,
8. BIDI\_TEXT = 0x00000005,
9. BIDI\_ENUM = 0x00000006,
10. BIDI\_BLOB = 0x00000007
11. } BIDI\_TYPE;

**BIDI\_NULL:** No bidirectional data.

**BIDI\_INT:** Bidirectional data is an integer.

**BIDI\_FLOAT:** Bidirectional data is a floating-point number.

**BIDI\_BOOL:** Bidirectional data is a Boolean value.

**BIDI\_STRING:** Bidirectional data is a string.

**BIDI\_TEXT:** Bidirectional data is text data.

**BIDI\_ENUM:** Bidirectional data is an enumeration.

**BIDI\_BLOB:** Bidirectional data is a data **BLOB**.

### Rules for Members

The following sections specify rules for the common string and flag members that are passed as parameters, or are parts of structures that are passed as parameters, to methods in this protocol.

#### Access Values

Access values specify the access rights that a caller is requesting.

#### Datatype Names

A Datatype Name is a string that contains the name of a data type. It MUST uniquely identify a format for print data that is supported by a print processor. The string MUST NOT be empty.

The data type SHOULD be specified by the implementation.[<256>](#Appendix_A_256)

#### Driver Names

A Driver Name is a string that contains the name of a printer driver. The string MUST NOT be empty.

An implementation MAY restrict the length of driver name strings.[<257>](#Appendix_A_257)

#### Environment Names

An Environment Name is a string that contains the name of the operating system environment. The string MUST permit white space and MUST NOT be empty.

Environment name strings MAY be specified by the implementation.[<258>](#Appendix_A_258)

#### Form Names

A Form Name is a string that contains the name of a printer form. It MUST uniquely identify a printer form on the system. The string MUST NOT be empty.

An implementation MAY restrict the length of form name strings.[<259>](#Appendix_A_259)

#### Job Control Values

Job control values specify actions such as pause and cancel (see [RpcSetJob (section 3.1.4.3.1)](#Section_f978bd2ce99a43ffba0802e892948dac) for a list of job control values and their meanings). A job control value MUST NOT be zero if a job control action is required.

#### Key Names

A Key Name is a string that contains the name of a [**printer key**](#gt_f6141c2f-6b05-4f7c-856a-a635bfd64d8e). It MUST uniquely identify a path under the main registry key where printer configuration data is kept. The string MUST permit a backslash ("\") as delimiter for paths with one or more subkeys.

An implementation MAY restrict the length of key name strings.[<260>](#Appendix_A_260)

#### Monitor Names

A Monitor Name is a string that contains the name of a port monitor. It MUST uniquely identify a port monitor on the system. The string MUST NOT be empty.

An implementation MAY restrict the length of monitor name strings.[<261>](#Appendix_A_261)

#### Path Names

A Path Name is a string that contains the file name, or full path and file name, including subdirectories, for the identified file. The string MUST permit white space and MUST NOT be empty.

If the path name string identifies a network-addressable file, it MUST be a **DNS**, **NetBIOS**, IPv4, or universal naming convention (UNC) name, or it SHOULD be an [**IPv6**](#gt_64c29bb6-c8b2-4281-9f3a-c1eb5d2288aa) name.[<262>](#Appendix_A_262) The string SHOULD be of the form "\\ServerName\ShareName" and MUST identify a unique shared folder on the machine.

The pattern followed by path name strings of local files SHOULD be specified by the implementation.[<263>](#Appendix_A_263)

An implementation SHOULD restrict the length of path name strings.[<264>](#Appendix_A_264)

For further information on **DNS** names, see [[RFC819]](http://go.microsoft.com/fwlink/?LinkId=90495) section 2. **NetBIOS** names are specified in [[RFC1001]](http://go.microsoft.com/fwlink/?LinkId=90260) section 14.

#### Port Names

A Port Name is a string that contains the name of a printer port. The string MUST NOT be empty.

The pattern followed by port name strings MAY be specified by the port monitor implementation that they belong to. [<265>](#Appendix_A_265)

#### Print Processor Names

A Print Processor Name specifies a string that contains the name of a print processor. It MUST uniquely identify a print processor for a given operating system environment. The string SHOULD NOT be empty.

An implementation MAY restrict the length of print processor name strings.[<266>](#Appendix_A_266)

#### Print Provider Names

A Print Provider Name is a string that contains the name of a print provider. It MUST uniquely identify a print provider on the system. The string MUST NOT be empty.

An implementation MAY restrict the length of print provider strings.[<267>](#Appendix_A_267)

#### Printer Change Values

Printer Change values specify printing-related events that occur on a print server. These values consist of [Printer Change Flags (section 2.2.3.6)](#Section_ef51a734e033467eb45d4b5c06f440e7).

#### Printer Names

A Printer Name is a string that contains the name of a facility for output. It MUST uniquely identify a destination that is local or on a print server. The string MUST NOT be empty.

In [RpcOpenPrinter](#Section_989357e2446e4872bb381dce21e1313f) and [RpcOpenPrinterEx](#Section_0e81ce1872b146c38584a205393b04ff) parameters, all instances of printer names MUST follow either the **PRINTER\_NAME** or **PRINTER\_NAME\_EX** pattern. In any other context, printer names MUST follow the **PRINTER\_NAME** pattern.

The **PRINTER\_NAME** pattern is defined as follows

1. UNICODE\_NOCOMMA\_NOBACKSLASH = <Any UTF-16LE character except ","
2. and "\">
3. UNICODE\_NOBACKSLASH = <Any UTF-16LE character, except "\">
4. PRINTER\_NAME = (SERVER\_NAME LOCAL\_PRINTER\_NAME) |
5. (WEB\_PRINT\_SERVER "/" "printers" "/" LOCAL\_PRINTER\_NAME "/"
6. ".printer")
7. WEB\_PRINT\_SERVER = "http: " "//" host [":" port]
8. LOCAL\_PRINTER\_NAME = 1#UNICODE\_NOCOMMA\_NOBACKSLASH

where:

* **SERVER\_NAME** is defined in section [2.2.4.16](#Section_9b3f813570224b72accbaefcc360c83b).
* **WEB\_PRINT\_SERVER** specifies the address of the web print server.
* **LOCAL\_PRINTER\_NAME** is a string specifying the local printer name or share name of the printer. Printer names MUST NOT contain the characters ',' and '\'.

Basic notational conventions are specified in [[RFC2616]](http://go.microsoft.com/fwlink/?LinkId=90372) section 2, and the terms **host** and **port** are defined in [[RFC3986]](http://go.microsoft.com/fwlink/?LinkId=90453) section 3.2.2.

The **PRINTER\_NAME\_EX** pattern extends the **PRINTER\_NAME** pattern and is defined as follows:

1. PRINTER\_NAME\_EX = PRINTER\_NAME\_EX1 | PRINTER\_NAME\_EX2 |
2. PRINTER\_NAME\_EX3 | PRINTER\_NAME\_EX4 | PRINTER\_NAME\_EX5
3. PRINTER\_NAME\_EX1 = SERVER\_NAME LOCAL\_PRINTER\_NAME "," #" "
4. "Job " #" " JOB\_IDENTIFIER
5. JOB\_IDENTIFIER = MUST be between 1 and 2,147,483,648, inclusive.
6. PRINTER\_NAME\_EX2 = SERVER\_NAME LOCAL\_PORT\_NAME "," #" " "Port"
7. PORT\_NAME = 1#UNICODE\_NOCOMMA
8. PRINTER\_NAME\_EX3 = SERVER\_NAME\_NE "\," #" " "XcvPort "
9. PORT\_NAME
10. PORT\_NAME = 1#UNICODE\_NOCOMMA
11. PRINTER\_NAME\_EX4 = SERVER\_NAME\_NE "\," #" " "XcvMonitor "
12. PORT\_MONITOR\_NAME
13. PORT\_MONITOR\_NAME = 1#UNICODE\_NOBACKSLASH
14. PRINTER\_NAME\_EX5 = SERVER\_NAME\_NE

where:

* **SERVER\_NAME\_NE** is defined in section 2.2.4.16
* **PRINTER\_NAME\_EX1** specifies the print job identified by **JOB\_IDENTIFIER** on the printer specified with **PrinterName**.

When the **PRINTER\_NAME\_EX1** form is used as the name parameter with RpcOpenPrinter or RpcOpenPrinterEx, a job object will be returned that can be used with [RpcReadPrinter](#Section_58fc87857f004f00bc784c3b438acbb7) and [RpcWritePrinter](#Section_6dd263aac8b547748d211daa0d0d9920) to read and write job content.

* **JOB\_IDENTIFIER** specifies a server-wide unique decimal identifier for the print job.
* **PRINTER\_NAME\_EX2** specifies the name of the port to be opened.

When the **PRINTER\_NAME\_EX2** form is used as the name parameter with RpcOpenPrinter or RpcOpenPrinterEx, a port object is returned that can be used with [RpcStartDocPrinter](#Section_90f95ff03ae04ddea83d35f457c710a6) and RpcWritePrinter to print directly to a port without intermediate spooling.

* **PORT\_NAME** is a port name, as specified in section [2.2.4.10](#Section_03c5feb74fbb4569b0649eb6ad833f50).
* **PRINTER\_NAME\_EX3** specifies the name of the port to be opened.

When the **PRINTER\_NAME\_EX3** form is used as the name parameter with RpcOpenPrinter or RpcOpenPrinterEx, a port object is returned that can be used with [RpcXcvData](#Section_bf9d2351aac0481e9d3705eb5cc961da) to communicate directly with a port.

* **PRINTER\_NAME\_EX4** specifies the name of the port monitor to be opened.

When this form is used as the name parameter with RpcOpenPrinter or RpcOpenPrinterEx, a port monitor object is returned that can be used with RpcXcvData to communicate directly with a port monitor.

* **PORT\_MONITOR\_NAME** is a monitor name, as specified in section [2.2.4.8](#Section_808e2e6c7f914bdd92da56fbca513404).
* **PRINTER\_NAME\_EX5** specifies the print server to be opened.

The **PRINTER\_NAME\_WITH\_POSTFIX** pattern extends the **PRINTER\_NAME** and **PRINTER\_NAME\_EX** patterns and is defined as follows:

1. PRINTER\_NAME\_WITH\_POSTFIX = PRINTER\_NAME\_PREFIX "," PRINTER\_NAME\_POSTFIX

where:

* **PRINTER\_NAME\_PREFIX** is the **PRINTER\_NAME** or **PRINTER\_NAME\_EX** pattern.
* **PRINTER\_NAME\_POSTFIX** is an implementation-defined string.

An implementation MAY not use the **PRINTER\_NAME\_WITH\_POSTFIX** pattern and instead use only the **PRINTER\_NAME** and **PRINTER\_NAME\_EX** patterns.[<268>](#Appendix_A_268)

An implementation MAY restrict the length of printer name strings. [<269>](#Appendix_A_269)

#### Registry Type Values

A Registry Type Value specifies the type of a data value in the registry using one of the constants specified in [Registry Type Value (section 2.2.3.9)](#Section_25cce7007fcf4bb6a2f30f6d08430a55).

#### Server Names

A Server Name is a string that contains the name of a print server. It MUST be a **DNS**, **NetBIOS**, IPv4, or Universal Naming Convention (UNC) name, or it MAY be an IPv6 name.[<270>](#Appendix_A_270) An empty server name string is interpreted as identifying the server that the [**RPC server endpoint**](#gt_8be6a1fb-bc3c-4ee3-8018-c236f351222a) is bound to.

Server names MUST follow the **SERVER\_NAME** pattern, which is defined as follows.

1. SERVER\_NAME = "\\" host "\" | ""

The **SERVER\_NAME\_NE** pattern is used where a non-empty server name is required and is defined as follows.

1. SERVER\_NAME\_NE = "\\" host

An implementation MAY restrict the length of server name strings. [<271>](#Appendix_A_271)

For further information on **DNS** names, see [[RFC819]](http://go.microsoft.com/fwlink/?LinkId=90495) section 2. Details about **NetBIOS** names are specified in [[RFC1001]](http://go.microsoft.com/fwlink/?LinkId=90260) section 14. Details about basic notational conventions are specified in [[RFC2616]](http://go.microsoft.com/fwlink/?LinkId=90372) section 2. The definition of **host** is specified in [[RFC3986]](http://go.microsoft.com/fwlink/?LinkId=90453) section 3.2.2.

#### User Names

A User Name is a string that contains the name of a valid [**principal**](#gt_8492780e-99e2-47ba-8553-aedb8de9f9c0). It MUST be formatted as either an Active Directory name or a name with a fully qualified domain. The string MUST NOT be empty,

For more information on user names, see [[MSDN-ADOVRVW]](http://go.microsoft.com/fwlink/?LinkId=89951) Object Names and Identities.

#### Value Names

A Value Name is a string that contains the name of a value that is kept under a printer key. The string MUST NOT be empty.

An implementation MAY restrict the length of value name strings. [<272>](#Appendix_A_272)

## Directory Service Interaction

### Interaction Summary

The Print System Remote Protocol initiates interactions with the Active Directory in the following way:[<273>](#Appendix_A_273)

* The print client calls [RpcSetPrinter](#Section_382997b5b02a4b6a8c6849a64b8c45f8) with a [PRINTER\_CONTAINER](#Section_ad7d4d4e0feb45e093edb3b2b9a313d1) structure that is initialized to contain a Level value of 0x00000007 and a pointer to a [PRINTER\_INFO\_7](#Section_91a6cb808df743d08a5d8f1d23f96de6) structure, which specifies the publishing action for the server to take for a print queue. The possible actions are publishing data about a print queue to the Active Directory, modifying the directory data for a print queue already published in the Active Directory, and removing the data for a print queue from the Active Directory.
* The print client calls RpcSetPrinter with a [PRINTER\_HANDLE](#Section_9c5d49b8692047b0946687ba910e6071) that is a handle to a print queue that the print server has published to the Active Directory, and as a result of the print client request, the print server either changes the printer driver associated with the print queue or modifies print queue configuration settings that the print server has published in the Active Directory.
* The print client calls [RpcDeletePrinter](#Section_e46c727493db443e8787ad49e5a08466) with a PRINTER\_HANDLE that is a handle to a print queue that the print server has published to the Active Directory.

Print servers use LDAP, as specified in [[RFC2251]](http://go.microsoft.com/fwlink/?LinkId=90325), to access data in the Active Directory. The implementation of LDAP by the Active Directory is described in section 3.1.1.4 of [[MS-ADTS]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ADTS%5d.pdf).

### Directory Service Schema Elements

Print servers access the directory service (DS) schema classes and attributes that are listed in this section. For the syntactic specifications of the <Class> or <Class> <Attribute> pairs, refer to [[MS-ADSC]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ADSC%5d.pdf) and [[MS-ADA3]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ADA3%5d.pdf).

* The print server uses the printQueue Active Directory schema class to publish a print queue.
* When creating a print queue object in the Active Directory, the print server MUST specify the following mandatory attributes of the printQueue Active Directory schema object:

| Class | Attribute |
| --- | --- |
| printQueue | versionNumber  uNCName  shortServerName  serverName  printerName |

* When creating a print queue object in the Active Directory, the print server SHOULD specify the following optional attributes of the printQueue Active Directory schema object:

| Class | Attribute |
| --- | --- |
| printQueue | priority  printStatus  printStartTime  printStaplingSupported  printSpooling  printShareName  printSeparatorFile  printRateUnit  printRate  printPagesPerMinute  printOwner  printOrientationsSupported  printNumberUp  printNotify  printNetworkAddress  printMinYExtent  printMinXExtent  printMemory  printMediaSupported  printMediaReady  printMaxYExtent  printMaxXExtent  printMaxResolutionSupported  printMaxCopies  printMACAddress  printLanguage  printKeepPrintedJobs  printFormName  printEndTime  printDuplexSupported  printColor  printCollate  printBinNames  printAttributes  portName  physicalLocationObject  operatingSystemVersion  operatingSystemServicePack  operatingSystemHotfix  operatingSystem  location  driverVersion  driverName  defaultPriority  bytesPerMinute  assetNumber |

### Interaction Details

Print servers access data in the Active Directory through the domain controller, which acts as an LDAP server. Print servers locate the domain controller by using the mechanism described in [[MS-ADTS]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ADTS%5d.pdf) section 6.3.6.1. The remainder of this section describes the sequences of steps performed by print servers in their interactions with the Active Directory.

Unless otherwise noted, the print server SHOULD detect any LDAP failures that can occur during all of the following operations: publishing a print queue to the Active Directory (see section [2.3.3.1](#Section_3f4f0a52e417495aad460466a9d0ef9d)), modifying or deleting a print queue in the Active Directory (see section [2.3.3.2](#Section_1c49f0840f634ac49e566be4c47915e6)), or searching for print queues in the Active Directory (see section [2.3.3.3](#Section_5352a7b056744d3393c4c94b5d499d37)). If the print server detects an LDAP failure, it SHOULD retry the complete publish, modification, deletion, or search operation. Retries SHOULD be separated by time intervals decaying from 10 seconds to 2 hours.

Because of LDAP failures, print queue objects in the Active Directory will not necessarily match the state of the print server at all times. Aside from update operations executed at print server initialization or periodically as described in section [2.3.3.4](#Section_a0cddd7937f64f60a4bc321938df4736), print servers MAY attempt to maintain consistency between their internal state and the objects in the Active Directory using other processes.

#### Publishing a Print Queue to the Active Directory

Print servers perform the following steps when fulfilling a client request to publish a print queue to the Active Directory.

1. Locate the domain controller as specified in section [2.3.3](#Section_9e24210cdc8e474d8ffd7b9dbe168440).
2. Determine the [**distinguished name (DN)**](#gt_1175dd11-9368-41d5-98ed-d585f268ad4b) for the [**container**](#gt_c3143e71-2ada-417e-83f4-3ef10eff2c56) representing the print server in the directory. This container is created when the print server joins the domain, as described in [[MS-ADOD]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ADOD%5d.pdf) section 2.7.7.1. The print server uses the DRSR protocol [[MS-DRSR]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-DRSR%5d.pdf) to determine the DN, as follows.
   * The print server calls the RPC method IDL\_DRSCrackNames as defined in [MS-DRSR] section 4.1.4. The input arguments for this call are the following:

| Attribute | Value |
| --- | --- |
| hDrs | Context handle returned from calling IDL\_DRSBind, as defined in [MS-DRSR] section 4.1.3. |
| dwInVersion | 1 |
| pmsgIn | A request with a string containing one name. This is the name of the computer's domain account (<domain>\<computer>, where <domain> is the name of the domain and <computer> is the name of the computer). The **formatOffered** field is DS\_UNKNOWN\_NAME, a member of the DS\_NAME\_FORMAT enumeration defined in [MS-DRSR] section 4.1.4.1.3. The **formatDesired** field is DS\_FQDN\_1779\_NAME, also defined in the DS\_NAME\_FORMAT enumeration. |

As specified in [MS-DRSR] section 4.1.4, if it is successful, the IDL\_DRSCrackNames method returns the object name in the requested format, which is the [**fully qualified domain name (FQDN)**](#gt_1769aec9-237e-44ed-9014-1abb3ec6de6e) in this case.

1. Create the directory service object representing the printer by using the LDAP protocol. The specific steps are as follows:
   * The print server binds to the LDAP server by using the bind operation defined in section 4.2 of [[RFC2251]](http://go.microsoft.com/fwlink/?LinkId=90325). The **version** field MUST be set to 3, the **name** field MUST be a null value, and the authentication mechanism MUST be the GSS\_SPNEGO SASL mechanism, as documented in [[MS-ADTS]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ADTS%5d.pdf) section 3.1.1.3.4.5.2.
   * The print server generates a [**relative distinguished name (RDN)**](#gt_22198321-b40b-4c24-b8a2-29e44d9d92b9), as defined in section 3.2 of [RFC2251], for the printer object in the directory. This DN is relative to the DN of the computer's directory object determined previously. There are no restrictions on the RDN generated beyond those requirements stated in [RFC2251]. As specified in [RFC2251], the FQDN for the printer object is the RDN generated here, prepended to the DN of the computer's directory object.
   * The print server adds the printer object to the directory and sets the attributes on the object by using exactly one LDAP add operation and any number of LDAP modify operations, as defined in sections 4.7 and 4.9 of [RFC2251]. The printer object MUST conform to the schema for the printQueue Active Directory object class as discussed in section [2.3.2](#Section_8b0f1957cb264966b55f371858ea4406).

In accordance with the schema, the print server MUST set the required properties of the printer object as part of the LDAP add request. These required properties MUST be set as follows:

| Attribute | Value |
| --- | --- |
| versionNumber | 4 |
| shortServerName | A string containing the machine name of the print server. |
| serverName | A string containing the fully qualified DNS name, as defined in [[RFC819]](http://go.microsoft.com/fwlink/?LinkId=90495), of the print server. |
| printerName | A string containing the name of the printer. This corresponds to the **pPrinterName** field described in section [3.1.4.1.5](#Section_3f2d41eb30e147e193bb534ea50cf8eb). |
| uNCName | A string containing the UNC name of the printer. This is of the form "\\<serverName>\<printerName>", where <serverName> is the value of serverName as defined in this table, and <printerName> is the value of printerName. |

The print server SHOULD publish to the directory any of the optional attributes specified in the schema for the print queue object class. The print server determines the values for both the mandatory and optional attributes in the schema by querying its internal state or the printer driver associated with the printer, as necessary depending on the print server implementation.[<274>](#Appendix_A_274)

The print server SHOULD also publish to the directory values previously set under certain predefined keys ("DsSpooler", "DsDriver", and "DsUser") by print clients using calls to **RpcSetPrinterDataEx** (section [3.1.4.2.18](#Section_58520b8b44ce46fc90d30362457098e0)). If the name of a value under one of these keys matches the name of an attribute in the printQueue object class schema, the print server SHOULD set the value of this attribute in the directory object to the data stored in the printer data value.[<275>](#Appendix_A_275)

For all attribute values the print server determined by querying internal state or a printer driver, as specified earlier in this section, the print server SHOULD save these values into the printer data corresponding to the printer. Print clients can later access these data values using calls to **RpcGetPrinterDataEx** (section [3.1.4.2.19](#Section_74a9f79910bf4ef6a2a40dcd9c43fb5c)). The print server SHOULD save these values under the predefined "DsSpooler" key if it determined the value by querying internal state or the predefined "DsDriver" key if it queried a printer driver. The print server SHOULD use value names that are the same as the names of the corresponding LDAP attributes.

If the LDAP add operation fails, the print server SHOULD wait for the retry interval specified in section 2.3.3 and then perform all the steps in this section. If the LDAP add operation succeeds but any LDAP modify operation fails, the print server SHOULD retry, as specified in section [2.3.3.2](#Section_1c49f0840f634ac49e566be4c47915e6), each failed LDAP modify operation.

* + The print server unbinds from the LDAP server by using the unbind operation defined in [RFC2251] section 4.3.

#### Modifying or Deleting a Print Queue in the Active Directory

Print servers perform the following steps when modifying or deleting a print queue in the Active Directory.

* Locate the domain controller and bind to the LDAP server as specified in section [2.3.3.1](#Section_3f4f0a52e417495aad460466a9d0ef9d).
* Determine the fully qualified DN of the print queue object in the directory, either by searching for the print queue as described in section [2.3.3.3](#Section_5352a7b056744d3393c4c94b5d499d37), or by using implementation-specific state saved during the publish operation.[<276>](#Appendix_A_276)
* Initiate LDAP modify or delete operations (as defined in sections 4.6 and 4.8 of [[RFC2251]](http://go.microsoft.com/fwlink/?LinkId=90325)) to update or delete the object as desired.[<277>](#Appendix_A_277)
* Unbind from the LDAP server as specified in section 2.3.3.1.

#### Searching for Print Queues in the Active Directory

Print servers also use LDAP to search for print queues in the Active Directory. Print servers perform the following steps to search for print queues:

* Locate the domain controller and bind to the LDAP server as described in section [2.3.3.1](#Section_3f4f0a52e417495aad460466a9d0ef9d).
* Search for print queues with the desired attributes, using the LDAP search operation, as defined in section 4.5 of [[RFC2251]](http://go.microsoft.com/fwlink/?LinkId=90325). The format of the LDAP search request is described in section 4.5.1 of [RFC2251]. The parameters of the search request are set as follows:

| Parameter | Value |
| --- | --- |
| baseObject | Print servers MAY specify the default naming context retrieved from the root DSE. The root DSE is defined in [[MS-ADTS]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ADTS%5d.pdf) section 1.1, and the default naming context attribute is described in [MS-ADTS] section 3.1.1.3.2.3. |
| scope | Print servers SHOULD specify wholeSubtree. |
| derefAliases | Print servers SHOULD specify neverDerefAliases. |
| sizeLimit | This value is dependent on the print server implementation and does not affect the protocol. |
| timeLimit | This value is dependent on the print server implementation and does not affect the protocol. |
| filter | The filter parameter of the search request MUST contain the requirement that the returned objects be of the print queue object class. In the string representation of filters described in [[RFC2254]](http://go.microsoft.com/fwlink/?LinkId=90328), this requirement is written as "(objectClass = printQueue)". In conjunction with this requirement, the filter can contain any restrictions based on any combination of attributes of the schema for the print queue object class. |
| attributes | The print server can request any set of attributes in the schema, but it SHOULD NOT fail if any optional attribute is missing. |

* Unbind from the LDAP server as described in section 2.3.3.1.

Print servers can also search the [**global catalog (GC)**](#gt_4f5d605a-7b3f-4db7-8c21-b146856d7169) for print queues in other [**NCs**](#gt_784c7cce-f782-48d8-9444-c9030ba86942).[<278>](#Appendix_A_278) The GC and naming contexts are defined in section [1.1](#Section_831cd729be7c451eb729bd8d84ce4d24). The Active Directory implementation of the GC is described in [MS-ADTS] section 3.1.1.1.8. Print clients use LDAP to perform searches on the GC in the same way as above.

Print servers can negotiate encryption of LDAP messages as part of the SASL authentication during the binding process, as described in [MS-ADTS] section 5.1.1.1.2.[<279>](#Appendix_A_279)

If an Active Directory print queue object is enumerated by the search, but the mandatory information specified in section [2.3.2](#Section_8b0f1957cb264966b55f371858ea4406) is not present in that print queue object, the print server SHOULD ignore this object and continue to the next enumerated print queue object.

Print servers MAY retry failed LDAP search operations.

#### Initializing the Print Server for Active Directory

Print servers perform the following initialization when Active Directory is available:

* Enumerate through the **List of Printers** and, for each print queue with the PRINTER\_ATTRIBUTE\_PUBLISHED flag set, verify that the print queue is already published into the Active Directory. If the print queue is not already published, publish the print queue to the Active Directory as described in section [2.3.3.1](#Section_3f4f0a52e417495aad460466a9d0ef9d). If the print queue is already published, verify that the data in the Active Directory is current and, if not, update the Active Directory entry for this print queue as described in section [2.3.3.2](#Section_1c49f0840f634ac49e566be4c47915e6). If the print queue is marked as "Delete Pending", delete the Active Directory entry for the print queue as described in section 2.3.3.2. If an Active Directory operation fails for a print queue, the print server SHOULD continue enumerating through the **List of Printers**.

Print servers can also perform the following periodic operations when Active Directory is available, but only when the print server is configured to do so:

* Periodically enumerate through the **List of Printers** and update the Active Directory as described earlier in this section.[<280>](#Appendix_A_280)
* Periodically search for the print queues in the Active Directory (see section [2.3.3.3](#Section_5352a7b056744d3393c4c94b5d499d37)) and delete (see section 2.3.3.2) all print queues that are not present in the current **List of Printers**.[<281>](#Appendix_A_281)

# Protocol Details

## Server Details

### Abstract Data Model

This section describes a conceptual model of a possible data organization that a print server implementation might need to maintain in order to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This specification does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this specification.

This protocol depends on an abstract data model that maintains information about printers and related objects. These objects represent physical output devices, and they are used in the protocol to communicate with those devices, to print to them, and to manage their configurations.

A print server has a SECURITY\_DESCRIPTOR controlling access to the server object. By default, the SECURITY\_DESCRIPTOR allows SERVER\_ALL\_ACCESS to members of the Administrators group and SERVER\_EXECUTE access to all other users.

A print server behaves as if it hosted the following objects in the following hierarchy.

**Note**  The abstract data model specified for this protocol is identical to that specified for the Print System Asynchronous Remote Protocol specified in [[MS-PAR]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-PAR%5d.pdf). A print server maintains only one copy of the data underlying the implementation that exposes this protocol or [MS-PAR].

**List of Print Server Names:** It is assumed that a client of this protocol has obtained the name of at least one print server that supports this protocol before it is invoked. There are various ways a client can build a **list of print server names** ([[MS-ADLS]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ADLS%5d.pdf), [[MS-ADSC]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ADSC%5d.pdf), [[MS-RAP]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-RAP%5d.pdf), and [[MS-SMB2]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-SMB2%5d.pdf)).

The server name that is passed to the print server by the client can differ from the server name that the print server determined upon its own initialization; for example, when aliasing of server names via DNS or directory services (DS) takes place.

Other methods, such as [RpcAddPerMachineConnection (section 3.1.4.2.24)](#Section_3648c131b1554cc69a2d79b95b422a5b), pass in an additional server name parameter that identifies a print server that is different from the one handling the API call. To correctly resolve server names, each print server maintains a mapping between server names and server addresses. When composing a response to the client, the print server that handles the API call uses the same server name that the client passed as the parameter in this call.

A print server does not persist the **list of print server names** between [**server restarts**](#gt_53c6244f-4101-436b-bfef-ab55868a38cf).

**List of Form Objects:** Form objects represent information about units of physical media on output devices, which are available to print clients. Examples are printer forms, such as sheets of paper. Each form object also contains a data element indicating whether the form object was added by a user and if the form object is specific to a printer.

Form objects are added, removed, accessed, and enumerated using [Form Management Methods (section 3.1.4.5)](#Section_0b9ceec4c17747aea4bea5d3167fc205).

A print server initializes the **list of form objects** to contain an implementation-specific set of built-in form objects. A print server persists the **list of form objects** between server restarts.

**List of Printers:** Each printer (also referred to as print queue) represents a physical print device or a number of homogeneous physical devices installed on the print server. Each printer object maintains the following data elements:

* A name that uniquely identifies the printer.
* A reference to a printer driver object for the printer.
* A reference to a print processor object.
* References to one or more port objects.[<282>](#Appendix_A_282)
* A list of queued or printing print jobs, where each print job has a SECURITY\_DESCRIPTOR controlling access to the print job and a data type, which specifies the type of data provided by the printing application to the printer during the print job. By default, the SECURITY\_DESCRIPTOR allows JOB\_ALL\_ACCESS to members of the Administrators group and to the client who created the print job, as well as a separate JOB\_READ access to the client who created the job. Each print job can also have zero or more **Job Named Properties**.[<283>](#Appendix_A_283)[<284>](#Appendix_A_284)
* Global [\_DEVMODE](#Section_ccc2a501794e4d2bb312f69c75131c2e) settings.
* Per-user \_DEVMODE settings.
* A SECURITY\_DESCRIPTOR controlling access to the printer object. By default, the SECURITY\_DESCRIPTOR allows PRINTER\_ALL\_ACCESS to members of the Administrators group and PRINTER\_ACCESS\_USE to all other users.
* The name of the default data type for the printer.
* A list of clients to be notified by the server about printer changes. Each entry in this list contains the change notifications requested by one client (the notification filter settings from the client), the mechanism used to send notifications to this client, and the information about the client destination where the server sends these notifications. Clients can request printer change notifications using [Notification Methods (section 3.1.4.10)](#Section_72b88737108f4e398fa9ee3af45da015).

A print server initializes the list of printers to the persisted list of printers. The server does not persist the list of printers between server restarts.

**Persisted List of Printers:** A print server maintains in persistent storage a **persisted list of printers** that is identical to the **list of printers**. A print server updates the **persisted list of printers** whenever it updates the **list of printers**.

**List of Printer Drivers:** Each printer driver represents the software component responsible for converting print content submitted by applications into device-specific commands. Each printer driver object maintains the following data elements:

* A name that uniquely identifies the printer driver.
* A rendering module, a configuration module, and a data module.

In addition, each printer driver object can maintain a list of Boolean values indicating whether the manufacturer of the printer driver declares it to have any of the following attributes, where a Boolean value of TRUE indicates that the printer driver has that attribute:

* The printer driver is a class printer driver.
* The printer driver is a derived printer driver.
* Printers using the printer driver cannot be shared from the print server to print clients.
* The printer driver is intended for use with fax printers.
* The printer driver is intended for use with file printers.
* The printer driver is intended for use with service printers.
* The printer driver is intended for use with virtual printers.

In addition, each printer driver object maintains the following optional data elements:

* A list of dependent files.
* Information about the printer driver manufacturer, timestamp, and driver version.[<285>](#Appendix_A_285)

Drivers are added, removed, enumerated, and managed using [Printer Driver Management Methods (section 3.1.4.4)](#Section_84e21d0dd5da4db2ab19d7ece51e42aa).

A print server initializes the **list of printer drivers** to an empty list. The server persists the list of drivers between server restarts.

**List of Core Printer Drivers**: The server maintains a second list of printer drivers containing the core printer drivers installed on the [**server**](#gt_434b0234-e970-4e8c-bdfa-e16a30d96703).

**List of Language Monitors:** A language monitor is a module that is tightly coupled to a printer driver and is part of the printer driver installation. It filters RAW format data as it is being sent from the printer driver to a port monitor, which then sends the data to the port. Language monitors add control information to the data stream, such as commands defined by a page description language (PDL). Language monitors are optional and are only associated with a particular type of printer if specified in the [**INF file**](#gt_5f92d31c-3db8-4344-a7e0-a224983abe07) for the printer.

Language monitors are added, removed, and enumerated using [Port Monitor Management Methods (section 3.1.4.7)](#Section_2c505550d9b24123bfa9ae204f4d88d6).

A print server initializes the **list of language monitors** to an empty list. The server persists the **list of language monitors** between server restarts.

**List of Port Monitors:** A port monitor is a component that can send buffers of data to devices using supported protocols. Port monitors can manage extended communication with the device, such as collection status information from the device. Port monitors can expose zero or more ports.

[**Port monitor modules**](#gt_4c9689f8-854e-43e1-964e-5dd9a70b8aae) are implementation-specific for a given port type. A printer port identifies a device connected to the machine via an implementation-specific protocol understood by its parent port monitor.

Port monitors are added, removed, and enumerated using Port Monitor Management Methods (section 3.1.4.7).

A print server initializes the **list of port monitors** to an empty list.[<286>](#Appendix_A_286) The server persists the **list of port monitors** between server restarts.

**List of Ports:** A port represents a connection to an actual print device. Ports are exposed and managed by port monitors.

Each port monitor can persist the **list of ports** it manages between server restarts.

**List of Print Providers:** A print provider performs transparent routing of print system calls to a local or remote spooler. When [RpcEnumPrinters (section 3.1.4.2.1)](#Section_0d83c60dec7546718a3b99517651ff1b) is called, an implementation-defined print provider name can be returned.[<287>](#Appendix_A_287)

A print server initializes the **list of print providers** to contain an implementation-specific set of included print providers. A print server persists the **list of print providers** between server restarts.[<288>](#Appendix_A_288)

**List of Print Processors:** Print processors, provided by device manufacturers or generic suppliers, perform additional manipulation of print content before it is sent to the device.

A print server persists the **list of print processors** between server restarts.

**List of Known Printers:** The server can maintain a **list of known printers** that includes printers not installed on the print server but installed on other print servers reachable on the network. The printers in this list are only used when composing a response for RpcEnumPrinters with the appropriate flags set, as specified in RpcEnumPrinters. This list facilitates printer detection in networks without directory services.

A print server does not persist the **list of known printers** between server restarts.

**List of Warned Printer Drivers:** The server can maintain a **list of printer drivers** that cannot be added through calls to [RpcAddPrinterDriverEx (section 3.1.4.4.8)](#Section_b96cc49759e54510ab045484993b259b) unless the appropriate flag is set, as specified in RpcAddPrinterDriverEx.[<289>](#Appendix_A_289)

If a print server maintains a **list of warned printer drivers**, it persists the **list of warned printer drivers** between server restarts.[<290>](#Appendix_A_290)

**List of Notification Clients:** A print server maintains a list of print clients that are notified upon server changes. Each entry in this list contains the change notifications requested by one client (the notification filter settings from the client), the mechanism used to send notifications to that client, and the information about the client destination where the server sends these notifications. Clients can request server change notifications by using Notification Methods (section 3.1.4.10).

A print server does not persist the **list of notification clients** between server restarts.

**Job Named Properties:** Each print job in a print queue can have zero or more **Job Named Properties** (also referred to as named properties or properties). The **Job Named Properties** are created, written, read, and deleted by the client. Each **Job Named Property** contains a name and a value. Unless deleted by the client, the **Job Named Properties** of a print job persist as long as the respective print job exists in the print queue.[<291>](#Appendix_A_291)

**List of Per-Machine Connections:**  The server maintains a list of [**shared printers**](#gt_d096e0d4-7ea2-4cc1-9204-5369ceb703ae) on other print servers. When a print server is used as a print client, all users logging on to the machine can use printers in this list without explicitly specifying the print server name or printer name of the shared printer.

A print server persists the **list of per-machine connections** between server restarts.

**Branch Office Print Remote Log Entries:** Each shared printer on a print server can be configured to operate in branch office print mode.[<292>](#Appendix_A_292) This mode enables a print client to print documents directly to a print device as defined by the print server, instead of routing print data to the server and then to the print device.

When in branch office print mode, if branch office print remote logging is enabled, the print client creates certain event channel entries on the print server in response to Windows Events while processing a print job. To do this, the client creates a structure that contains a **Branch Office Print Remote Log Entry** with the information needed to create the event ID in the correct event channel on the server, and it sends it to the server by using [RpcLogJobInfoForBranchOffice (section 3.1.4.13.1)](#Section_65a54d4df30b41438299d61f086ac7fe).

If the print server cannot be contacted when the print job is being processed, the print client writes remote log entries to a **Branch Office Print Remote Log Offline Archive** (section [3.2.1](#Section_fa5a51d585cc40dc951a1e288b62d820)) for transmission at a later time.

The abstract data model associates each printer with a single printer driver, one or more printer ports, and exactly one print processor. Every object stored in the abstract data model defines an associated set of attributes, as specified in [IDL Data Types (section 2.2.1)](#Section_df6af0abee9745aaae87186f83ab2265) and [Custom-Marshaled Data Types (section 2.2.2)](#Section_99fdf2cf79bb43bf997c88b33538a5e1).

Where the preceding data model requires persistence, and unless specified otherwise, the print server stores one persistent copy of each object in the registry.[<293>](#Appendix_A_293)

The server is responsible for ensuring consistency among persistently stored objects and their transient copies.[<294>](#Appendix_A_294) The server is also responsible for managing any resources, including memory, disk space, locks, and physical ports, that are used for object representations, throughout the lifetimes of the objects.

**Note**  The previous conceptual data can be implemented using a variety of techniques. A print server can implement such data as needed.

### Timers

No protocol timer events are required on the server beyond timers for the underlying RPC protocol.

### Initialization

The server SHOULD listen on the well-known endpoints defined for this RPC interface in section [2.1](#Section_fdf1138af6b94b00ada61fbb6097d683).

The server MUST perform initialization according to the following rules, when calling an RPC notification method on the client:

* Create an RPC binding handle to the server RPC endpoint (the client implements a server endpoint for this protocol in order to process notifications), or use an RPC context handle, as specified in [[C706]](http://go.microsoft.com/fwlink/?LinkId=89824).
* Use RPC handles of the following types:
  + Context handles that are used across multiple calls to the client, for methods taking a [PRINTER\_HANDLE](#Section_9c5d49b8692047b0946687ba910e6071).
  + Handles that are bound to a single call to the client, for name-based methods taking a [STRING\_HANDLE](#Section_59883eb85ce64ad4bdd2599434cfdcf2). A **STRING\_HANDLE\_BIND** method MUST be implemented by the server.
* When creating the **RPC binding handle** on the named pipe **\pipe\spoolss**, the server MUST specify an **ImpersonationLevel** of 2 (**Impersonation**), as specified in [[MS-SMB2]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-SMB2%5d.pdf), section 2.2.13.

If Active Directory is available, the server SHOULD perform according to the rules described in section [2.3.3.4](#Section_a0cddd7937f64f60a4bc321938df4736) when starting additional initialization.

Both the server and the client MUST ignore all LDAP operation failures that occur during their initialization. For example, if LDAP initialization for one print queue fails, the server MUST ignore this failure and SHOULD continue to execute the remaining LDAP initializations for the remaining print queues.[<295>](#Appendix_A_295)

### Message Processing Events and Sequencing Rules

The Print System Remote Protocol MUST indicate the following to the RPC **runtime** specified in [[MS-RPCE]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-RPCE%5d.pdf) section 3.

* It is to perform a strict NDR data consistency check at target level 6.0.
* It is to reject a NULL unique or full pointer with non-zero conformant value.
* Using the **strict\_context\_handle** attribute, it is to reject the use of context handles that are created by the methods of a different RPC interface.

The methods that are defined by this protocol are grouped into functional categories, and their syntax and behavior are specified in sections, as shown in the following table.

| Functional category | Description | Section |
| --- | --- | --- |
| Printer management and discovery | Methods used for discovering and obtaining access to supported printers. | [3.1.4.2](#Section_5ea67df3c4e745efb42540b8b066d276) |
| Job management | Methods for discovering, defining, and scheduling print jobs. | [3.1.4.3](#Section_bd8e1517a5b14850933866575d2b8cf4) |
| Printer driver management | Methods for discovering and installing printer drivers. | [3.1.4.4](#Section_84e21d0dd5da4db2ab19d7ece51e42aa) |
| Form management | Methods for discovering and configuring printer forms. | [3.1.4.5](#Section_0b9ceec4c17747aea4bea5d3167fc205) |
| Printer port management | Methods for discovering and communicating with printer ports. | [3.1.4.6](#Section_1cf0e23868da43bdbaba44b35b115cc0) |
| Port monitor management | Methods for discovering and installing port monitor modules. | [3.1.4.7](#Section_2c505550d9b24123bfa9ae204f4d88d6) |
| Print processor management | Methods for discovering and manipulating print processor objects. | [3.1.4.8](#Section_701f22dc32934ef28aa64d03ad4ef868) |
| Document printing | Methods for printing documents, pages and data. | [3.1.4.9](#Section_93e6c5a794d6487ebe20d2c91e5277ad) |
| Notifications | Methods for obtaining notifications of printing events. | [3.1.4.10](#Section_72b88737108f4e398fa9ee3af45da015) |
| Monitor modules | Methods specified by executable language monitors. | [3.1.4.11](#Section_07ce205123c64d508f55c25d7ac09cb3) |
| Job named property management | Methods for creating, updating, deleting, and enumerating **Job Named Properties** (section [3.1.1](#Section_1bd61448db4b4284a4b18346732402d4)).[<296>](#Appendix_A_296) | [3.1.4.12](#Section_95aa7a44c15841a0b2ad2a08ccd0881f) |
| Branch office print remote logging | Methods for processing **Branch Office Print Remote Log Entries** (section 3.1.1).[<297>](#Appendix_A_297) | [3.1.4.13](#Section_fd04e0de264746e0a06aa3ccb138dbb6) |

The following table lists all the methods of the Print System Remote Protocol in ascending [**opnum**](#gt_e127848e-c66d-427d-b3aa-9f904fa4ada7) order.

Methods in RPC Opnum Order

| Method | Description |
| --- | --- |
| [RpcEnumPrinters](#Section_0d83c60dec7546718a3b99517651ff1b) | RpcEnumPrinters enumerates available printers, print servers, domains, or print providers.  Opnum: 0 |
| [RpcOpenPrinter](#Section_989357e2446e4872bb381dce21e1313f) | RpcOpenPrinter retrieves a handle for a printer, port, port monitor, print job, or print server.  Opnum: 1 |
| [RpcSetJob](#Section_f978bd2ce99a43ffba0802e892948dac) | RpcSetJob pauses, resumes, cancels, or restarts a print job. It also sets print job parameters, for example, the job priority and the document name.  Opnum: 2 |
| [RpcGetJob](#Section_f7090127287b4aa7ba8b2b6b7247dfc6) | RpcGetJob retrieves information about a specified print job.  Opnum: 3 |
| [RpcEnumJobs](#Section_0c77f1ff08274f4da06d5a3d802ffc0c) | RpcEnumJobs retrieves information about a specified set of print jobs for a specified printer.  Opnum: 4 |
| [RpcAddPrinter](#Section_bd0bb8d526fd470581f5614d32b9f38f) | RpcAddPrinter adds a printer to the list of supported printers for a specified server.  Opnum: 5 |
| [RpcDeletePrinter](#Section_e46c727493db443e8787ad49e5a08466) | RpcDeletePrinter deletes the specified printer object.  Opnum: 6 |
| [RpcSetPrinter](#Section_382997b5b02a4b6a8c6849a64b8c45f8) | RpcSetPrinter sets the data for a specified printer or sets the state of the specified printer by pausing or resuming printing, or clearing all print jobs.  Opnum: 7 |
| [RpcGetPrinter](#Section_ccbf288a0768416dbede0ef1987fc286) | RpcGetPrinter retrieves information about a specified printer.  Opnum: 8 |
| [RpcAddPrinterDriver](#Section_f23a75191c7740699acea6d8eae47c22) | RpcAddPrinterDriver installs a printer driver on the print server and links the configuration, data, and printer driver files.  Opnum: 9 |
| [RpcEnumPrinterDrivers](#Section_857d00ac36824a0d86ca3d3c372e5e4a) | RpcEnumPrinterDrivers enumerates the printer drivers installed on a specified print server.  Opnum: 10 |
| [RpcGetPrinterDriver](#Section_00d91852785d42e4a774b1d96361b0ce) | RpcGetPrinterDriver retrieves printer driver data for the specified printer.  Opnum: 11 |
| [RpcGetPrinterDriverDirectory](#Section_9df11cf440984852ad72d1f75a82bffe) | RpcGetPrinterDriverDirectory retrieves the path of the printer driver directory.  Opnum: 12 |
| [RpcDeletePrinterDriver](#Section_e03b14adabe84158ab0ce0033d820116) | RpcDeletePrinterDriver removes the specified printer driver from the list of supported drivers for a print server.  Opnum: 13 |
| [RpcAddPrintProcessor](#Section_d7e6fdb25256483592c127aa6f76758a) | RpcAddPrintProcessor installs a print processor on the specified server and adds its name to an internal list of supported print processors.  Opnum: 14 |
| [RpcEnumPrintProcessors](#Section_8c3c2db185d84d8894c11e09cda435c7) | RpcEnumPrintProcessors enumerates the print processors installed on a specified server.  Opnum: 15 |
| [RpcGetPrintProcessorDirectory](#Section_17eaee3be5754f82855ce60657d5b2c8) | RpcGetPrintProcessorDirectory retrieves the path for the print processor on the specified server.  Opnum: 16 |
| [RpcStartDocPrinter](#Section_90f95ff03ae04ddea83d35f457c710a6) | RpcStartDocPrinter notifies the print spooler that a document is being spooled for printing.  Opnum: 17 |
| [RpcStartPagePrinter](#Section_76261f544c634fc2a2dabaee584e076b) | RpcStartPagePrinter notifies the spooler that a page is about to be printed on the specified printer.  Opnum: 18 |
| [RpcWritePrinter](#Section_6dd263aac8b547748d211daa0d0d9920) | RpcWritePrinter sends data to the print spooler.  Opnum: 19 |
| [RpcEndPagePrinter](#Section_c4a17128584d402e9bac6a050e9da6d0) | RpcEndPagePrinter notifies the print spooler that the application is at the end of a page in a print job.  Opnum: 20 |
| [RpcAbortPrinter](#Section_924cd448b8b74dbaaa56d9d3244cd386) | RpcAbortPrinter aborts the currently spooling print document.  Opnum: 21 |
| [RpcReadPrinter](#Section_58fc87857f004f00bc784c3b438acbb7) | RpcReadPrinter retrieves data from the specified printer.  Opnum: 22 |
| [RpcEndDocPrinter](#Section_209fcf3205b444aba3a4779cfae181ca) | RpcEndDocPrinter notifies the print spooler that the application is at the end of the current print job.  Opnum: 23 |
| [RpcAddJob](#Section_7ba3267737294b5f96b5c8a9f99d34a5) | RpcAddJob returns ERROR\_INVALID\_PARAMETER.  Opnum: 24 |
| [RpcScheduleJob](#Section_4d6454a7f7d44d308ba18644cb6b2186) | RpcScheduleJob returns ERROR\_SPL\_NO\_ADDJOB.  Opnum: 25 |
| [RpcGetPrinterData](#Section_9c04a0f2adc443938df47c36a14183f5) | RpcGetPrinterData retrieves configuration data for a printer or print server.  Opnum: 26 |
| [RpcSetPrinterData](#Section_e0d3e41e097c459da7fbacaf1343ea27) | RpcSetPrinterData sets the configuration data for a printer or print server.  Opnum: 27 |
| [RpcWaitForPrinterChange](#Section_e40cb68f679e4b84a33e86a94a141d83) | RpcWaitForPrinterChange retrieves information about the most recent change notification associated with a printer or print server.  Opnum: 28 |
| [RpcClosePrinter](#Section_8482c18677734fb58034e9bae7a91e52) | RpcClosePrinter closes a handle to a printer object, server object, job object, or port object.  Opnum: 29 |
| [RpcAddForm](#Section_d6eb54304a274e5ab6988140a12fa889) | RpcAddForm adds a form name to the list of supported forms.  Opnum: 30 |
| [RpcDeleteForm](#Section_1b25a56720084ca1ac93feaeaec7ec54) | RpcDeleteForm removes a form name from the list of supported forms.  Opnum: 31 |
| [RpcGetForm](#Section_be369d1d1ede49b181c209c69411c25d) | RpcGetForm retrieves information about a specified form.  Opnum: 32 |
| [RpcSetForm](#Section_c17cbf6601a24ac0b57b1d2d8d1f73de) | RpcSetForm replaces the form information for the specified form.  Opnum: 33 |
| [RpcEnumForms](#Section_0b9dde1473d14f1a9f748e82255a9612) | RpcEnumForms enumerates the forms that the specified printer supports.  Opnum: 34 |
| [RpcEnumPorts](#Section_4f018733e78e4088ac48f6e19ee00863) | RpcEnumPorts enumerates the ports that are available for printing on a specified server.  Opnum: 35 |
| [RpcEnumMonitors](#Section_6c5836fed9ea41a0ad7e01152f39e8a1) | RpcEnumMonitors retrieves information about the port monitors installed on the specified server.  Opnum: 36 |
| Opnum37NotUsedOnWire | Reserved for local use.  Opnum: 37 |
| Opnum38NotUsedOnWire | Reserved for local use.  Opnum: 38 |
| [RpcDeletePort](#Section_cb13f514f95a4d00936bf934641b6a4b) | RpcDeletePort removes a port added by the [RpcAddPortEx](#Section_1303a48b47724226a8b8c495d255304e) method.  Opnum: 39 |
| [RpcCreatePrinterIC](#Section_b0686ba23a6b4a6c943fb26600dbd88c) | RpcCreatePrinterIC is called by the [**Graphics Device Interface (GDI)**](#gt_6c550e84-9a9c-46ef-8716-7260fe29cae3) to create an information context for a printer.  Opnum: 40 |
| [RpcPlayGdiScriptOnPrinterIC](#Section_eff167adf4004c6eb0e2c48b62ac0826) | RpcPlayGdiScriptOnPrinterIC returns identifying information for fonts available for printing to a printer object.  Opnum: 41 |
| [RpcDeletePrinterIC](#Section_d9792de2d19c4c9bb7a4034523b7784b) | RpcDeletePrinterIC deletes a printer information context (IC).  Opnum: 42 |
| Opnum43NotUsedOnWire | Reserved for local use.  Opnum: 43 |
| Opnum44NotUsedOnWire | Reserved for local use.  Opnum: 44 |
| Opnum45NotUsedOnWire | Reserved for local use.  Opnum: 45 |
| [RpcAddMonitor](#Section_dee4feea274940bdab7cc1c0ca1fbcf8) | RpcAddMonitor installs a local port monitor and links the configuration, data, and monitor files.  Opnum: 46 |
| [RpcDeleteMonitor](#Section_2483519c230c4e8bb52d29677d941ad8) | RpcDeleteMonitor removes a port monitor.  Opnum: 47 |
| [RpcDeletePrintProcessor](#Section_e08179d3d3c3421bab77a0ba50a4254e) | RpcDeletePrintProcessor removes a print processor.  Opnum: 48 |
| Opnum49NotUsedOnWire | Reserved for local use.  Opnum: 49 |
| Opnum50NotUsedOnWire | Reserved for local use.  Opnum: 50 |
| [RpcEnumPrintProcessorDatatypes](#Section_c4a35c2040824ce6840102c52ee3158e) | RpcEnumPrintProcessorDatatypes enumerates the data types that a specified print processor supports.  Opnum: 51 |
| [RpcResetPrinter](#Section_6d979ef765e848af94c76bbb9d2be06a) | RpcResetPrinter resets the data type and device mode values to use for printing documents submitted by RpcStartDocPrinter (section 3.1.4.9.1).  Opnum: 52 |
| [RpcGetPrinterDriver2](#Section_a73e77d07b2448cfa6d38b4061749495) | RpcGetPrinterDriver2 retrieves printer driver data for the specified printer.  Opnum: 53 |
| Opnum54NotUsedOnWire | Reserved for local use.  Opnum: 54 |
| Opnum55NotUsedOnWire | Reserved for local use.  Opnum: 55 |
| [RpcFindClosePrinterChangeNotification](#Section_f50c8a861b2d4124b2205b840aacd2c7) | RpcFindClosePrinterChangeNotification closes a change notification object created by calling [RpcRemoteFindFirstPrinterChangeNotification (section 3.1.4.10.3)](#Section_b8b414d9f1cd4191bb6b87d09ab2fd83). The printer or print server associated with the change notification object will no longer be monitored by that object.  Opnum: 56 |
| Opnum57NotUsedOnWire | Reserved for local use.  Opnum: 57 |
| [RpcReplyOpenPrinter](#Section_7fcd3036d45a4ec7b081f2b860e66676) | RpcReplyOpenPrinter establishes a context handle from the server to the client. The server uses the context handle to send notification data to the client machine.  Opnum: 58 |
| [RpcRouterReplyPrinter](#Section_38bf5d2ce4a744b2b16f91ff8226f836) | RpcRouterReplyPrinter handles the notification coming from a remote router, as opposed to one coming from a print provider.  Opnum: 59 |
| [RpcReplyClosePrinter](#Section_d6c73ecbc02b4150a01bad73a642990a) | RpcReplyClosePrinter closes the notification channel opened by the RpcReplyOpenPrinter (section 3.2.4.1.1) method between the server and client.  Opnum: 60 |
| RpcAddPortEx | RpcAddPortEx adds a port name to the list of supported ports.  Opnum: 61 |
| RpcRemoteFindFirstPrinterChangeNotification | RpcRemoteFindFirstPrinterChangeNotification creates a remote change notification object that monitors changes to printer objects and sends change notifications to the client using the method RpcRouterReplyPrinter (section 3.2.4.1.2).  Opnum: 62 |
| Opnum63NotUsedOnWire | Reserved for local use.  Opnum: 63 |
| Opnum64NotUsedOnWire | Reserved for local use.  Opnum: 64 |
| [RpcRemoteFindFirstPrinterChangeNotificationEx](#Section_eb66b2211c1f4249b8bcc5befec2314d) | RpcRemoteFindFirstPrinterChangeNotificationEx creates a remote change notification object that monitors changes to printer objects and sends change notifications to the client using the method [RpcRouterReplyPrinterEx (section 3.2.4.1.4)](#Section_1b44cb8295b5485c9ec90360c3a3ecb0).  Opnum: 65 |
| RpcRouterReplyPrinterEx | RpcRouterReplyPrinterEx handles the notification coming from a remote router, as opposed to one coming from a print provider. It is similar to RpcRouterReplyPrinter but an [RPC\_V2\_UREPLY\_PRINTER](#Section_1ccdac5b0b2a4bd38afdd2c2589130fc) structure to specify the set of notifications that the client requested.  Opnum: 66 |
| [RpcRouterRefreshPrinterChangeNotification](#Section_cb5d22ed0352430383d5b5afd32fb461) | RpcRouterRefreshPrinterChangeNotification returns change notification information.  Opnum: 67 |
| Opnum68NotUsedOnWire | Reserved for local use.  Opnum: 68 |
| [RpcOpenPrinterEx](#Section_0e81ce1872b146c38584a205393b04ff) | RpcOpenPrinterEx retrieves handle for a printer, port, port monitor, print job, or print server. This method is similar to RpcOpenPrinter but takes a pointer to an [SPLCLIENT\_CONTAINER (section 2.2.1.2.14)](#Section_cca15c7257d342fc91e749442b623c11) structure, which contains information about the connecting client.  Opnum: 69 |
| [RpcAddPrinterEx](#Section_3af45717308f4b4f8428430d12856110) | RpcAddPrinterEx installs a printer on the print server. This method is similar to RpcAddPrinter but takes a pointer to an SPLCLIENT\_CONTAINER structure, which contains information about the connecting client.  Opnum: 70 |
| [RpcSetPort](#Section_b0a0984e6c724dc8b8c9db98276c94cd) | RpcSetPort sets the status associated with a printer port.  Opnum: 71 |
| [RpcEnumPrinterData](#Section_51bfa051e7e44fc4b79eea9d4b08cb84) | RpcEnumPrinterData enumerates configuration data for a specified printer.  Opnum: 72 |
| [RpcDeletePrinterData](#Section_a67521b7c92d4ebf900836fb321196ff) | RpcDeletePrinterData deletes specified configuration data for a printer.  Opnum: 73 |
| Opnum74NotUsedOnWire | Reserved for local use.  Opnum: 74 |
| Opnum75NotUsedOnWire | Reserved for local use.  Opnum: 75 |
| Opnum76NotUsedOnWire | Reserved for local use.  Opnum: 76 |
| [RpcSetPrinterDataEx](#Section_58520b8b44ce46fc90d30362457098e0) | RpcSetPrinterDataEx sets the configuration data for a printer or print server. This method is similar to RpcSetPrinterData but also allows the caller to specify the registry key under which to store the data.  Opnum: 77 |
| [RpcGetPrinterDataEx](#Section_74a9f79910bf4ef6a2a40dcd9c43fb5c) | RpcGetPrinterDataEx retrieves configuration data for the specified printer or print server. This method extends RpcGetPrinterData and can retrieve values sorted under a specified key by RpcSetPrinterDataEx.  Opnum: 78 |
| [RpcEnumPrinterDataEx](#Section_9574b3e74a294bfaba67c35dab674162) | RpcEnumPrinterDataEx enumerates all value names and data for a specified printer and key. This method extends RpcEnumPrinterData by retrieving several values in a single call.  Opnum: 79 |
| [RpcEnumPrinterKey](#Section_128ff57da4a44a23b2ee36f71ca5c0d5) | RpcEnumPrinterKey enumerates the subkeys of a specified key for a specified printer. Printer data is stored in the registry.  Opnum: 80 |
| [RpcDeletePrinterDataEx](#Section_19c2e0e5b7ec4d068b211488f2275505) | RpcDeletePrinterDataEx deletes specified configuration data for a printer. This method is similar to RpcDeletePrinterData but accesses the configuration data using a set of named and typed values that are stored in a hierarchy of registry keys.  Opnum: 81 |
| [RpcDeletePrinterKey](#Section_cde0dcc363264509ad3fe170977931ba) | RpcDeletePrinterKey deletes a specified key and all of its subkeys for a specified printer.  Opnum: 82 |
| Opnum83NotUsedOnWire | Reserved for local use.  Opnum: 83 |
| [RpcDeletePrinterDriverEx](#Section_21d247cd66e24373a187b858c907e13f) | RpcDeletePrinterDriverEx removes the specified printer driver from the list of supported drivers for a print server and deletes the files associated with the driver. This method is similar to RpcDeletePrinterDriver but can also delete specific versions of the driver.  Opnum: 84 |
| [RpcAddPerMachineConnection](#Section_3648c131b1554cc69a2d79b95b422a5b) | RpcAddPerMachineConnection stores the print server name and the name of the binary executable used as a provider for a particular connection.  Opnum: 85 |
| [RpcDeletePerMachineConnection](#Section_8861fe40b2764c778ef7cc6a28063e0e) | RpcDeletePerMachineConnection deletes information about a server and connection provider.  Opnum: 86 |
| [RpcEnumPerMachineConnections](#Section_3b1e7385d1ca48dfb372461c524f78d9) | RpcEnumPerMachineConnections enumerates each of the connections and copies [PRINTER\_INFO\_4 (section 2.2.1.10.5)](#Section_ac391864cd9945099ff3ea0366dd9b2b) structures for all the per-machine connections to the buffer **pPrinterEnum**.  Opnum: 87 |
| [RpcXcvData](#Section_bf9d2351aac0481e9d3705eb5cc961da) | RpcXcvData provides an extensible mechanism by which a client can control ports on the server and exchange port-specific commands and data with the server. See section 3.1.4.11 for details on language monitor methods.  Opnum: 88 |
| [RpcAddPrinterDriverEx](#Section_b96cc49759e54510ab045484993b259b) | RpcAddPrinterDriverEx installs a printer driver on the print server. This method performs a function similar to RpcAddPrinterDriver and additionally allows options to be specified which permit [**printer driver upgrade**](#gt_642a4e0f-a3c9-4641-9b63-e622b46d490b), [**printer driver downgrade**](#gt_a4b3cd29-c2a1-4d08-a431-de57a0283628), copying of newer files only, and copying of all files regardless of their time stamps.  Opnum: 89 |
| Opnum90NotUsedOnWire | Reserved for local use.  Opnum: 90 |
| Opnum91NotUsedOnWire | Reserved for local use.  Opnum: 91 |
| Opnum92NotUsedOnWire | Reserved for local use.  Opnum: 92 |
| Opnum93NotUsedOnWire | Reserved for local use.  Opnum: 93 |
| Opnum94NotUsedOnWire | Reserved for local use.  Opnum: 94 |
| Opnum95NotUsedOnWire | Reserved for local use.  Opnum: 95 |
| [RpcFlushPrinter](#Section_be7ec83d52eb42e9bc5d7244e5b44b94) | RpcFlushPrinter is used by the printer driver to send a buffer of bytes to the port to cleanly abort a print job. It also allows delaying the I/O line to the printer.  Opnum: 96 |
| [RpcSendRecvBidiData](#Section_5aec83472ebf464eb6d1540aa60b5c71) | RpcSendRecvBidiData sends and receives bidirectional data. This method is used to communicate with port monitors that support such data.  Opnum: 97 |
| Opnum98NotUsedOnWire | Reserved for local use.  Opnum: 98 |
| Opnum99NotUsedOnWire | Reserved for local use.  Opnum: 99 |
| Opnum100NotUsedOnWire | Reserved for local use.  Opnum: 100 |
| Opnum101NotUsedOnWire | Reserved for local use.  Opnum: 101 |
| Opnum102NotUsedOnWire | Reserved for local use.  Opnum: 102 |
| [RpcGetCorePrinterDrivers](#Section_3d6282322aa944f484e01168d3208f27) | RpcGetCorePrinterDrivers gets the GUIDs, versions, and publish dates of the specified core printer drivers, and the paths to their packages.  Opnum: 102 |
| Opnum103NotUsedOnWire | Reserved for local use.  Opnum: 103 |
| Opnum104NotUsedOnWire | Reserved for local use.  Opnum: 104 |
| [RpcGetPrinterDriverPackagePath](#Section_e453b11350c64a129ca6e327b5accca6) | RpcGetPrinterDriverPackagePath gets the path to the specified printer driver package.  Opnum: 104 |
| Opnum105NotUsedOnWire | Reserved for local use.  Opnum: 105 |
| Opnum106NotUsedOnWire | Reserved for local use.  Opnum: 106 |
| Opnum107NotUsedOnWire | Reserved for local use.  Opnum: 107 |
| Opnum108NotUsedOnWire | Reserved for local use.  Opnum: 108 |
| Opnum109NotUsedOnWire | Reserved for local use.  Opnum: 109 |
| [RpcGetJobNamedPropertyValue](#Section_64d3b0c08eef4a2b859960904de7c823) | **RpcGetJobNamedPropertyValue** retrieves the value of the specified **Job Named Property** (section 3.1.1) for the specified print job.  Opnum: 110 |
| [RpcSetJobNamedProperty](#Section_a08c060ad4e24a88a4874f6aaecfcf09) | **RpcSetJobNamedProperty** creates a new **Job Named Property** or changes the value of an existent **Job Named Property** for the specified print job.  Opnum: 111 |
| [RpcDeleteJobNamedProperty](#Section_02b59e33f1534870af67199267dc0b64) | **RpcDeleteJobNamedProperty** deletes a **Job Named Property** for the specified print job.  Opnum: 112 |
| [RpcEnumJobNamedProperties](#Section_6b1559f993494d90b8cf24292ab736ad) | **RpcEnumJobNamedProperties** enumerates the **Job Named Properties** for the specified print job.  Opnum: 113 |
| Opnum114NotUsedOnWire | Reserved for local use.  Opnum: 114 |
| Opnum115NotUsedOnWire | Reserved for local use.  Opnum: 115 |
| [RpcLogJobInfoForBranchOffice](#Section_65a54d4df30b41438299d61f086ac7fe) | **RpcLogJobInfoForBranchOffice** processes one or more **Branch Office Print Remote Log Entries** (section 3.1.1).  Opnum: 116 |

In the preceding table, the term "Reserved for local use" means that the client MUST NOT send the opnum, and server behavior is undefined since it does not affect interoperability.[<298>](#Appendix_A_298)

All these methods are request/response RPC methods. They MUST return zero to indicate successful completion and nonzero values to indicate failure, except where specifically described.

Unless stated otherwise, if a method fails for any reason, returning a nonzero failure value, the server state as visible to the client through this or any other protocol MUST NOT change.

Two nonzero return codes have specific meanings in this protocol, ERROR\_MORE\_DATA and ERROR\_INSUFFICIENT\_BUFFER, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf). When a method declaration in this specification has an output parameter that supplies a needed buffer size, one of the values in the following table can be returned from a call to that method to enable the caller to discover that size. Thus, there are circumstances in which a nonzero return value SHOULD NOT be treated as an error, but, instead, the client SHOULD allocate a buffer with a larger size and retry the request. These cases are noted in the method definitions in this section.

| Error name/code | Meaning |
| --- | --- |
| ERROR\_INSUFFICIENT\_BUFFER  0x0000007A | The data area passed to a system call is too small. |
| ERROR\_MORE\_DATA  0x000000EA | More data is available. |

#### Commonly Used Parameters

This section describes parameters commonly used in method calls with consistent definitions. The type of each parameter is given by the method declaration in the relevant method sections.

Individual method sections specify only parameters whose definitions are different from, or that are not listed in, this section; however, they may impose additional restrictions on the values of parameters defined in this section.

##### Datatype Name Parameters

*pDatatype*: This parameter MUST be one of the following:

* NULL, to indicate that the default data type for the printer MUST be used.
* A pointer to a string that specifies the data type to be associated with the printer handle.

For rules governing data type names, see section [2.2.4.2](#Section_f40cbad088dd4e86a16bcf62c3b17c52).

The individual method sections include the following parameter validation steps by reference:

* If *pDatatype* is not NULL, verify that the string that is referenced by the *pDatatype* parameter identifies one of the data types that the printer or print server supports, and if that verification fails, return ERROR\_INVALID\_DATATYPE.

**Note**  A client SHOULD use [RpcEnumPrintProcessorDatatypes (section 3.1.4.8.5)](#Section_c4a35c2040824ce6840102c52ee3158e), specifying a print processor, to obtain a list of supported data types.

##### Dynamically Typed Query Parameters

Unless notified otherwise, methods returning one or more dynamically-typed values use a common API pattern, in which the caller passes in the following parameters:

* *BUFFER*: A buffer into which the server copies the requested dynamically-typed values. The term "BUFFER" is used here as a placeholder. Each method section defines which of its parameters is the pointer to the buffer.
* *pType*: An optional pointer to a variable that receives a code that indicates the type of data that is stored in the specified value. For a list of possible type values, see section [2.2.3.9](#Section_25cce7007fcf4bb6a2f30f6d08430a55).
* *nSize* or *cbData*: The size, in bytes, of the buffer. This value can be larger than the required size for the requested dynamically-typed values.
* *pcbNeeded* or *pcbData*: A pointer to a variable into which the server copies the number of bytes between the start of *BUFFER* and the last byte written by the server into *BUFFER*, both inclusive; or the required size of the buffer, in bytes, if the value of the buffer size parameter is smaller than the size of the data to return to the caller.

For methods capable of returning more than one dynamically-typed value, the caller also passes in:

* *pcReturned*: A pointer to a variable into which the server copies the number of dynamically-typed values that were written to the buffer, if the buffer was large enough to hold them. If the buffer is not large enough to hold these dynamically typed values, the client SHOULD ignore the value that the server can return through pcReturned.

The individual method sections include the following parameter validation steps by reference:

* The server MUST verify that the value of the buffer size parameter is not smaller than the number of bytes required to hold the dynamically-typed values to be written to the buffer. If that verification fails, the server MUST write the number of bytes required into the variable that is pointed to by *pcbNeeded* and return ERROR\_MORE\_DATA.
* If the value of the buffer size parameter is not zero, the server MUST verify that a non-NULL pointer to the buffer was passed in. If that verification fails, the server MUST return ERROR\_INVALID\_USER\_BUFFER.

The individual method sections include the following processing and response steps by reference:

* The server MUST populate *BUFFER* with dynamically-typed values enumerated according to method-specific enumeration steps.
* If *pType* is not a NULL pointer, the server MUST write the type of the data returned in *BUFFER* to the variable pointed to by *pType*.
* For methods capable of returning more than one dynamically-typed value, the server MUST store the number of values that were written to *BUFFER* into the variable pointed to by *pcReturned*.
* The server MUST return zero for success or a nonzero error code if the method was not successful.

Except for diagnostic purposes, the server state as visible to the client through this protocol MUST NOT change as a result of processing the method call.

##### Environment Name Parameters

*pEnvironment*: This parameter MUST either be NULL or a pointer to a string that MUST specify the environment name. For rules governing environment names and Windows behaviors, see section [2.2.4.4](#Section_ad7758938a194786ab4634731f52d1af).

The individual method sections include the following parameter validation steps by reference:

* If *pEnvironment* is NULL, use the local environment of the print server.
* If *pEnvironment* is a non-NULL pointer, verify that the string that is referenced by the *pEnvironment* parameter identifies an environment name that is supported on this server, and if that verification fails, return ERROR\_INVALID\_ENVIRONMENT.

##### Print Server Name Parameters

*pName*: This parameter is a pointer to a string that specifies the name of the print server that the method operates on. This MUST be a Domain Name System (DNS), [**NetBIOS**](#gt_b86c44e6-57df-4c48-8163-5e3fa7bdcff4), Internet Protocol version 4 (IPv4), Internet Protocol version 6 (IPv6), or Universal Naming Convention (UNC) name that remote procedure call (RPC) binds to, and it MUST uniquely identify a print server on the network.

For all methods taking a [STRING\_HANDLE](#Section_59883eb85ce64ad4bdd2599434cfdcf2) custom binding handle parameter, the Print System Remote Protocol assumes that the bind routine provided by the client uses the name provided through the *pName* parameter to create the **RPC binding**, although that is not strictly necessary from an RPC perspective. Although it is possible to create an **RPC binding** to a different server than that identified by the *pName* parameter, the Print System Remote Protocol has not been designed and tested for that usage pattern. However, server implementations MAY choose to implement support for server names not identical to the server name used to create the **RPC binding**, and as a result effectively route the call to another server.[<299>](#Appendix_A_299)

**Note**  Regardless of the preceding statement, server implementations MUST NOT assume that the server name passed via the *pName* parameter matches the name the server determined upon its own initialization; the server name passed in MAY differ from that name as a result of server name aliasing, for example, by use of **DNS** names or directory services. The server MUST use the passed-in name to compose names for responses because the client is not aware that aliasing occurred.

**RPC binding** handles are specified in [[C706]](http://go.microsoft.com/fwlink/?LinkId=89824). For rules governing server names, see section [2.2.4.16](#Section_9b3f813570224b72accbaefcc360c83b).

*pServer*: Synonymous with *pName*.

*pszServer*: Synonymous with *pName*.

*Name*: Synonymous with *pName*.

The individual method sections include the following parameter validation steps by reference:

* Verify that the string pointed to by the *Name* parameter is well-formed according to the rules governing server names (section 2.2.4.16). If that verification fails, return the error code ERROR\_INVALID\_NAME.
* Verify that the string pointed to by the *Name* parameter corresponds to a server name. If that verification fails, return any of the following error codes: ERROR\_INVALID\_NAME, ERROR\_INVALID\_PARAMETER, or ERROR\_INVALID\_PRINTER\_NAME.

The server SHOULD perform this validation step to ensure correctness with clients that do not derive the RPC binding directly from the STRING\_HANDLE parameter.[<300>](#Appendix_A_300)

##### Printer Name Parameters

*pPrinterName*: This parameter is a pointer to a string that specifies the name of the printer connection, printer object, server object, job object, or port object. The string that is referenced by this parameter MUST NOT be empty. For rules governing printer names, see section [2.2.4.14](#Section_24fcd124035c4988a8583a7d8d6f7b43).

The individual method sections include the following parameter validation steps by reference:

* Verify that the string pointed to by *pPrinterName* is well-formed according to the rules governing printer names. If that verification fails, return the error code ERROR\_INVALID\_PRINTER\_NAME. When the string pointed to by *pPrinterName* contains a printer name postfix string appended at the end, the server SHOULD:
  + Ignore both the comma character preceding the postfix string and the postfix string.
  + Use only the printer name specified by the prefix string.
  + Validate the prefix string according to the rules governing printer names.

The server can execute additional implementation-specific validation of the postfix string, including rejecting unsupported postfix string values by returning an implementation-specific error code.

* For server names, verify that the server name portion of the string corresponds to a server name. If that verification fails, return any of the following error codes: ERROR\_INVALID\_NAME, ERROR\_INVALID\_PARAMETER, or ERROR\_INVALID\_PRINTER\_NAME.

The server SHOULD perform this validation step to ensure correctness with clients that do not derive the RPC binding directly from the [STRING\_HANDLE](#Section_59883eb85ce64ad4bdd2599434cfdcf2) parameter.[<301>](#Appendix_A_301)

* For printer, job, or port names, verify that the remainder of the string corresponds to a printer, job, or port name. If that verification fails, return the error code ERROR\_INVALID\_PRINTER\_NAME.
* For port and port monitor, open requests by using [RpcOpenPrinter](#Section_989357e2446e4872bb381dce21e1313f) or [RpcOpenPrinterEx](#Section_0e81ce1872b146c38584a205393b04ff), and verify that the port monitor supports all the optional methods: **XcvOpenPort**, **XcvDataPort**, and **XcvClosePort**. And if that verification fails, return ERROR\_INVALID\_PRINT\_MONITOR.

##### Standard Parameter Validation

The implementation MUST apply the following validation rules to all parameters unless more specific statements appear in the individual method sections.

| Term used to describe parameter | Required validation |
| --- | --- |
| X MUST be a non-NULL pointer to a string. | Verify that X is not NULL. If that verification fails, return ERROR\_INVALID\_PARAMETER. |
| X MUST be A. | Verify that X is A. If that verification fails, return ERROR\_INVALID\_PARAMETER. |
| X MUST be a value from A through B, inclusive. | Verify that X is a value that is greater than or equal to A and less than or equal to B. If that verification fails, return ERROR\_INVALID\_PARAMETER. |
| X MUST NOT be A. | Verify that X is a value that is not A. If that verification fails, return ERROR\_INVALID\_PARAMETER. |
| X MUST be one of <list>. | Verify that X is a value that is a member of <list>. If that verification fails, return ERROR\_INVALID\_PARAMETER. |
| X MUST be the result of a bitwise OR of zero or more of the flags in <list>. | If <list> contains the statement "All other bits MUST be zero", verify that the only bits that are set are those that are specified in <list>. If that verification fails, return ERROR\_INVALID\_PARAMETER. |
| X MUST be the result of a bitwise OR of one or more of the flags in <list>. | Verify that at least one of the bit flags from <list> is set and if that verification fails, return ERROR\_INVALID\_PARAMETER. If <list> contains the statement "All other bits MUST be zero", verify that the only bits that are set are the bits that are specified in <list>. If that verification fails, return ERROR\_INVALID\_PARAMETER. |

##### String Query Parameters

Unless noted otherwise, methods that return one or more string values use a common API pattern, in which the caller passes in the following parameters:

* *Level*: The value 0x00000001, if this parameter is present in the method signature.
* *BUFFER*: A buffer into which the server copies the requested string values. The term BUFFER is used here as a placeholder. Each method section defines which of its parameters is the pointer to the buffer. Methods capable of returning more than one string value MUST write the values to the buffer as a multisz.
* *cbBuf*: The size, in bytes, of the buffer. This value can be larger than the required size for the requested string values.
* *pcbNeeded*: A pointer to a variable into which the server copies the number of bytes between the start of the buffer and the last byte written by the server into the buffer, both inclusive, or the required size of the buffer, in bytes, if the value of the *cbBuf* parameter is smaller than the actual size of the data to return to the caller.

For methods that are capable of returning more than one string value, the following parameter interpretation applies:

* If a *pcReturned* parameter is present in the method signature, it is a pointer to a variable into which the server copies the actual number of string values that are written to *BUFFER*, if the buffer is large enough to hold them.
* If a *pcReturned* parameter is not present in the method signature, the caller MUST interpret the data returned in *BUFFER* as a multisz.

The individual method sections include the following parameter validation steps by reference:

* The server MUST verify that the value of *cbBuf* is not smaller than the number of bytes required to hold the string values to be written to the buffer. If that verification fails, the server MUST write the number of bytes required to the variable pointed to by the *pcbNeeded* parameter and return ERROR\_INSUFFICIENT\_BUFFER.
* If the value of the *cbBuf* parameter is not zero, the server MUST verify that a non-NULL pointer to the buffer was passed in. If that verification fails, the server MUST return ERROR\_INVALID\_USER\_BUFFER.

The individual method sections include the following processing and response steps by reference:

* The server MUST populate *BUFFER* with string values enumerated according to method-specific enumeration steps. Multiple string values MUST be represented as a multisz.
* For methods that are capable of returning more than one string value, if a *pcReturned* parameter is present in the method signature, the server MUST store the number of string values written to *BUFFER* into the variable pointed to by *pcReturned*.
* The server MUST return zero for success, or a nonzero error code if the method was not successful.

Except for diagnostic purposes, the server state as visible to the client through this protocol MUST NOT change as a result of processing the method call.

##### CONTAINER Parameters

This section specifies common [CONTAINER](#Section_8ca20b730d864e57a009f797ef1b2de8) parameters and related validation and processing requirements.

###### DEVMODE\_CONTAINER Parameters

*pDevMode*: This parameter is a pointer to a [DEVMODE\_CONTAINER (section 2.2.1.2.1)](#Section_c2b14fe764794460a7ba633a845dd61a) structure.

*pDevModeContainer*: This parameter is synonymous with *pDevMode*.

The individual method sections include the following parameter validation steps by reference:

* If the *pDevModeContainer* parameter is declared with the "unique" IDL attribute, and its value is NULL, the server SHOULD skip the validation steps and assume validation success.
* If the *pDevModeContainer* parameter is not declared with the "unique" IDL attribute, the server MAY verify that its value is not NULL.[<302>](#Appendix_A_302)
* The server SHOULD verify that the *pDevMode* member of the DEVMODE\_CONTAINER that is pointed to by *pDevModeContainer* is NULL, or that the [\_DEVMODE (section 2.2.2.1)](#Section_ccc2a501794e4d2bb312f69c75131c2e) structure that is pointed to by the *pDevMode* member is valid, which means that the total size specified in \_DEVMODE MUST be less than or equal to the number of bytes specified by the value of the **cbBuf** member of the DEVMODE\_CONTAINER.
* The server SHOULD verify that the **dmSize** and **dmDriverExtra** members of the \_DEVMODE structure comply with the constraints defined in section 2.2.2.1, and that the sum of their values is not larger than the *cbBuf* member of the DEVMODE\_CONTAINER.

**Note**  The server uses the printer driver associated with a print queue to validate all other \_DEVMODE members. Incorrect settings for these other members are silently corrected by the printer driver. Therefore, the print client SHOULD obtain a valid \_DEVMODE from the printer driver or print queue to use as template for the DEVMODE\_CONTAINER parameter. The client SHOULD modify selected fields only as necessary and as indicated by the printer driver capabilities that are returned by a local call to the printer driver.

Unless noted otherwise, if any of the preceding validation steps fail, the server SHOULD return ERROR\_INVALID\_PARAMETER.

###### DOC\_INFO\_CONTAINER Parameters

*pDocInfoContainer*: This parameter is a pointer to a [DOC\_INFO\_CONTAINER (section 2.2.1.2.2)](#Section_c4f39f63301745bf8b8a81ab9458f052) structure that specifies the document to print.

The individual method sections include the following parameter validation steps by reference:

* If the *pDocInfoContainer* parameter is declared with the "unique" IDL attribute, and its value is NULL, skip the validation steps and assume validation success.
* If the *pDocInfoContainer* parameter is not declared with the "unique" IDL attribute, the server MAY verify that its value is not NULL.[<303>](#Appendix_A_303)
* Verify that the value of the **Level** member in the DOC\_INFO\_CONTAINER is 0x00000001.
* Verify that the **pDocInfo1** pointer in the DOC\_INFO\_CONTAINER is not NULL.
* Verify that all members of the structure that is pointed to by the **pDocInfo1** pointer in the DOC\_INFO\_CONTAINER comply with the constraints defined in section 2.2.1.2.2.

Unless noted otherwise, if any of the preceding validation steps fail, return ERROR\_INVALID\_PARAMETER.

###### DRIVER\_CONTAINER Parameters

*pDriverContainer*: This parameter is a pointer to a [DRIVER\_CONTAINER (section 2.2.1.2.3)](#Section_3a3f9cf78ec44921b1f686cf8d139bc2) structure that specifies printer driver information. The value of the **Level** member of the DRIVER\_CONTAINER structure MUST be 0x00000002, 0x00000003, 0x00000004, 0x00000006, or 0x00000008.

The individual method sections include the following parameter validation steps by reference:

* If the *pDriverContainer* parameter is declared with the "unique" IDL attribute, and its value is NULL, skip the validation steps and assume validation success.
* Verify that *pDriverContainer* points to a DRIVER\_CONTAINER structure that specifies an appropriate level, as defined in the referring method definition. If that verification fails, return ERROR\_INVALID\_LEVEL.
* Verify that, within the [DRIVER\_INFO (section 2.2.1.5)](#Section_17943579a5144d0b8ded422265b23f45) structure that is contained in DRIVER\_CONTAINER, the **pEnvironment** member specifies an environment name that is supported on the server (see section [2.2.4.4](#Section_ad7758938a194786ab4634731f52d1af)). If that verification fails, return ERROR\_INVALID\_ENVIRONMENT.
* Verify that all members of DRIVER\_CONTAINER comply with the constraints defined in section 2.2.1.2.3.

Unless noted otherwise, if any of the preceding validation steps fail, return ERROR\_INVALID\_PARAMETER.

###### FORM\_CONTAINER Parameters

*pFormInfoContainer*: This parameter is a pointer to a [FORM\_CONTAINER (section 2.2.1.2.4)](#Section_49508bc38fcb49f5b92d0724666d76a4) structure that specifies form information.

The individual method sections include the following parameter validation steps by reference:

* If the *pFormInfoContainer* parameter is declared with the "unique" IDL attribute, and its value is NULL, skip the validation steps and assume validation success.
* Verify that *pFormInfoContainer* points to a FORM\_CONTAINER that specifies an appropriate level as defined in the referring method definition. If that verification fails, return ERROR\_INVALID\_LEVEL.
* Verify that all members of the FORM\_CONTAINER structure comply with the constraints defined in section 2.2.1.2.4.

Unless noted otherwise, if any of the preceding validation steps fail, return ERROR\_INVALID\_PARAMETER.

###### PORT\_CONTAINER Parameters

*pPortContainer*: This parameter is a pointer to [PORT\_CONTAINER (section 2.2.1.2.7)](#Section_e05adf6ca04e4e398b34e7c118ce2bc8) structure that specifies port information.

The individual method sections include the following parameter validation steps by reference:

* If the *pPortContainer* parameter is declared with the "unique" IDL attribute, and its value is NULL, skip the validation steps and assume validation success.
* Verify that *pPortContainer* points to a PORT\_CONTAINER that specifies an appropriate level as defined in the referring method definition. If that verification fails, return ERROR\_INVALID\_LEVEL.
* Verify that all members of the PORT\_CONTAINER structure comply with the constraints specified in section 2.2.1.2.7.

Unless noted otherwise, if any of the above validation steps fail, return ERROR\_INVALID\_PARAMETER.

###### PRINTER\_CONTAINER Parameters

*pPrinterContainer*: This parameter is a pointer to a [PRINTER\_CONTAINER (section 2.2.1.2.9)](#Section_ad7d4d4e0feb45e093edb3b2b9a313d1) structure, which specifies printer information. The **Level** member of the PRINTER\_CONTAINER MUST be between 0x00000000 and 0x00000008, inclusive. When the **Level** member is 0x00000002, the **Status**, **cJobs**, and **AveragePPM** members of the [PRINTER\_INFO\_2](#Section_D40734E1AC9B4B2A862883A1BC7BBEED) structure (sections [2.2.1.3.6](#Section_C07C2BFDE2C94D369259354A229D645D) and 2.2.1.10.3) MUST be set to zero by the caller and MUST be ignored on receipt. For details concerning [PRINTER\_INFO](#Section_125fb29b3c5141fca333b1e0e8794a91) structures, see section 2.2.1.10.

The individual method sections include the following parameter validation steps by reference:

* If the *pPrinterContainer* parameter is declared with the "unique" IDL attribute, and its value is NULL, skip the validation steps and assume validation success.
* Verify that *pPrinterContainer* points to a PRINTER\_CONTAINER that specifies an appropriate level as defined in the referring method definition. If that verification fails, return ERROR\_INVALID\_LEVEL.
* If *pDatatype* is not NULL, verify that it points to a string that identifies a data type supported by the associated print processor. If that verification fails, return ERROR\_INVALID\_DATATYPE.
* If *pPrintProcessor* is not NULL, verify that it points to a string that identifies a print processor. If that verification fails, return ERROR\_UNKNOWN\_PRINTPROCESSOR.
* If *pSepFile* is not NULL, verify that it points to a string that names an existing file on the server. If that verification fails, return ERROR\_INVALID\_SEPARATOR\_FILE.
* Verify that *pPortName* is not NULL and points to a string that identifies an existing port on the server. If that verification fails, return ERROR\_UNKNOWN\_PORT.
* Verify that *pDriverName* is not NULL and points to a string that identifies an existing printer driver on the server. If that verification fails, return ERROR\_UNKNOWN\_PRINTER\_DRIVER.
* Verify that, if the *Attributes* parameter has the flag PRINTER\_ATTRIBUTE\_SHARED (section [2.2.3.12](#Section_1625e9d929e448f4b83d3bd0fdaea787)) set, the printer driver identified by *pDriverName* does not have the attribute that printers using that printer driver cannot be shared (section [3.1.1](#Section_1bd61448db4b4284a4b18346732402d4)). If that verification fails, return ERROR\_PRINTER\_NOT\_SHAREABLE.
* Verify that the **Priority** member of PRINTER\_INFO\_2 is within the range specified in section 2.2.1.10.3. If that verification fails, return ERROR\_INVALID\_PRIORITY.
* Verify that all members of the PRINTER\_CONTAINER structure comply with the constraints defined in section 2.2.1.10, with the exception of **pServerName**, which SHOULD be ignored.

Unless noted otherwise, if any of the preceding validation steps fail, return ERROR\_INVALID\_PARAMETER.

The individual method sections further include the following parameter processing steps by reference:

* If the value of the **Level** member specifies a PRINTER\_INFO structure that contains a **pDevMode** member (section 2.2.1.3.6), replace the value of that member with the **pDevMode** value of the [DEVMODE\_CONTAINER](#Section_c2b14fe764794460a7ba633a845dd61a) structure (section 2.2.1.2.1) that is pointed to by the *pDevModeContainer* parameter of the method.
* If the value of the **Level** member specifies a PRINTER\_INFO structure that contains a **pSecurityDescriptor** member (section 2.2.1.3.6), replace the value of that member with the **pSecurity** value of the [SECURITY\_CONTAINER](#Section_6cf2b4ff1b2b453c87f985243823cea8) structure (2.2.1.2.13) that is pointed to by the *pSecurityContainer* parameter of the method.

###### SECURITY\_CONTAINER Parameters

*pSecurityContainer*: This parameter is a pointer to a [SECURITY\_CONTAINER (section 2.2.1.2.13)](#Section_6cf2b4ff1b2b453c87f985243823cea8) structure that specifies security information and components. The created printer MUST allow security access based on this information.[<304>](#Appendix_A_304)

The individual method sections include the following parameter validation steps by reference:

* If the *pSecurityContainer* parameter is declared with the "unique" IDL attribute, and its value is NULL, skip the validation steps and assume validation success.
* Verify that the **pSecurity** member of the SECURITY\_CONTAINER structure is NULL, or that it points to a well-formed SECURITY\_DESCRIPTOR in self-relative form ([[MS-DTYP]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-DTYP%5d.pdf) section 2.4.6).

Unless noted otherwise, if any of the preceding validation steps fail, return ERROR\_INVALID\_PARAMETER.

###### SPLCLIENT\_CONTAINER Parameters

*pClientInfo*: This parameter is a pointer to an [SPLCLIENT\_CONTAINER (section 2.2.1.2.14)](#Section_cca15c7257d342fc91e749442b623c11) structure that specifies client information. The **Level** member of the SPLCLIENT\_CONTAINER structure MUST be 0x00000001.[<305>](#Appendix_A_305)

The individual method sections include the following parameter validation steps by reference:

* If the *pClientInfo* parameter is declared with the "unique" IDL attribute, and its value is NULL, skip the validation steps and assume validation success.
* Verify that *pClientInfo* points to an SPLCLIENT\_CONTAINER that contains a pointer to a structure, and that all members of that structure comply with the constraints defined in section 2.2.1.2.14.

Unless noted otherwise, if any of the preceding validation steps fail, return ERROR\_INVALID\_PARAMETER.

###### MONITOR\_CONTAINER Parameters

*pMonitorContainer*: This parameter is a pointer to a [MONITOR\_CONTAINER (section 2.2.1.2.6)](#Section_b23382165e3e4379a722af2230d06df2) structure that specifies monitor information. The individual method sections include the following parameter validation steps by reference:

* If the *pMonitorContainer* parameter is declared with the "unique" IDL attribute, and its value is NULL, skip the validation steps and assume validation success.
* Verify that *pMonitorContainer* points to a MONITOR\_CONTAINER that specifies an appropriate level as defined in the referring method definition. If that verification fails, return ERROR\_INVALID\_LEVEL.
* If the value of the **Level** member specifies a [MONITOR\_INFO](#Section_822a6b4d37814a579eefd572dcf4d86f) structure that contains a **pEnvironment** member, verify that the string that is referenced by **pEnvironment** identifies an environment name that is supported on this server; and if that verification fails, return ERROR\_INVALID\_ENVIRONMENT.
* If the value of the **Level** member specifies a **MONITOR\_INFO** structure that contains a **pEnvironment** member, the environment name referenced by **pEnvironment** is "Windows ARM", and this server doesn’t support that environment name (see section [2.2.4.4](#Section_ad7758938a194786ab4634731f52d1af)), return ERROR\_NOT\_SUPPORTED. Print servers can optionally be configured to not perform this validation step.[<306>](#Appendix_A_306)
* Verify that all members of the MONITOR\_CONTAINER structure comply with the constraints defined in 2.2.1.8.

Unless noted otherwise, if any of the preceding validation steps fail, return ERROR\_INVALID\_PARAMETER.

##### INFO Structures Query Parameters

Unless noted otherwise, methods that return one or more of the INFO structures (see sections [2.2.1.3](#Section_964b517f67304b4a99edee9648e7f1f7) to [2.2.1.11](#Section_d23fee1a3f624cbb8a06d01180fe2ff9)) use a common API pattern, in which the caller passes in the following parameters:

* *Level*: The desired information level of the INFO structures, if this parameter is present in the method signature.
* *BUFFER*: A buffer into which the server copies the requested INFO structures. The term "BUFFER" is used here as a placeholder. Each method section defines which of its parameters contains the pointer to the buffer.
* *cbBuf*: The size, in bytes, of the buffer. The value of *cbBuf* can be larger than the required size for the requested INFO structures.
* *pcbNeeded*: A pointer to a variable into which the server copies the number of bytes between the start of *BUFFER* and the last byte written by the server into *BUFFER*, both inclusive; or the required size of the buffer, in bytes, if the value of *cbBuf* is smaller than the size of the data to return to the caller.

For methods capable of returning more than one INFO structure, the caller also passes in:

* *pcReturned*: This parameter is a pointer to a variable into which the server copies the number of INFO structures written to the buffer, if the buffer was large enough to hold them.

The individual method sections use the following documentation conventions:

* **BUFFER TYPE**: The type of INFO structures returned, which is one of the following:
  + DATATYPES\_INFO\_1
  + \_DRIVER\_INFO
  + \_FORM\_INFO
  + \_JOB\_INFO
  + \_MONITOR\_INFO
  + \_PORT\_INFO
  + \_PRINTER\_INFO
  + PRINTPROCESSOR\_INFO\_1
* *Level*: The valid levels of INFO structures.

The individual method sections include the following parameter validation steps by reference:

* The server MUST verify that *Level* is valid, and if that verification fails, the server MUST return ERROR\_INVALID\_LEVEL.
* The server MUST verify that the value of *cbBuf* is not smaller than the number of bytes required to hold the INFO structures to be written to the buffer, and if that verification fails, the server MUST write the number of bytes required into the variable pointed to by *pcbNeeded* and return ERROR\_INSUFFICIENT\_BUFFER.
* If the value of *cbBuf* is zero, the server MUST ignore the pointer to the buffer that was passed in. If the value of *cbBuf* is not zero, the server MUST verify that the pointer to the buffer that was passed in is a non-NULL pointer, and if that verification fails, the server MUST return ERROR\_INVALID\_USER\_BUFFER.

The individual method sections include the following processing and response steps by reference:

* The server MUST populate *BUFFER* with INFO structures of a type specified by **TYPE** that describe the objects enumerated according to method-specific enumeration steps.
* For methods capable of returning more than one INFO structure, the server MUST store the number of INFO structures that it writes to *BUFFER* in the variable pointed to by *pcReturned*.
* The server MUST return zero for success or a nonzero error code if the method was not successful.

Except for diagnostic purposes, the server state as visible to the client through this protocol MUST NOT change as a result of processing the method call.

##### PRINTER\_ENUM\_VALUES Structures Query Parameters

Unless noted otherwise, methods returning one or more [PRINTER\_ENUM\_VALUES](#Section_1daff8b90e2546d586a2a965ce67f74e) structures (section 2.2.2.11) use a common API pattern, in which the caller passes in the following parameters:

* *BUFFER*: A buffer into which the server copies the requested PRINTER\_ENUM\_VALUES structures. The term "BUFFER" is used here as a placeholder; each method section defines which of its parameters contains a pointer to the buffer.
* *cbEnumValues*: The size, in bytes, of *BUFFER*. The value of *cbEnumValues* can be larger than the required size for the requested PRINTER\_ENUM\_VALUES structures.
* *pcbEnumValues*: A pointer to a variable into which the server copies the number of bytes between the start of *BUFFER* and the last byte written by the server into *BUFFER*, both inclusive; or the required size of the buffer, in bytes, if the value of *cbEnumValues* is smaller than the size of the data to return to the caller.

For methods capable of returning more than one PRINTER\_ENUM\_VALUES structure, the caller also passes in:

* *pnEnumValues*: A pointer to a variable into which the server copies the number of PRINTER\_ENUM\_VALUES structures written to the buffer, if the buffer was large enough to hold them.

The individual method sections include the following parameter validation steps by reference:

* The server MUST verify that the value of *cbEnumValues* is not smaller than the number of bytes required by the PRINTER\_ENUM\_VALUES structures to be written to the buffer. If that verification fails, the server MUST write the number of bytes required to the variable that is pointed to by *pcbEnumValues* and return ERROR\_MORE\_DATA.
* If the value of *cbEnumValues* is not zero, the server MUST verify that a non-NULL pointer to the buffer was passed in. If that verification fails, the server MUST return ERROR\_INVALID\_USER\_BUFFER.
* For a printer object with a printer driver version (see *cVersion* in section [2.2.1.3.1](#Section_4464eaf0f34f40d5b970736437a21913)) of 0x00000004, the server SHOULD verify that the size in bytes of the data to be returned to the client (the value to be returned via the *pcbEnumValues* parameter) does not exceed 0xFFFFFFFF bytes. If this verification fails, the server SHOULD return ERROR\_NOT\_ENOUGH\_MEMORY. [<307>](#Appendix_A_307)

The individual method sections include the following processing and response steps by reference:

* The server MUST populate *BUFFER* with PRINTER\_ENUM\_VALUES structures that describe the enumerated objects according to method-specific enumeration steps.
* For methods capable of returning more than one PRINTER\_ENUM\_VALUES structure, the server MUST write the number of PRINTER\_ENUM\_VALUES structures that were written to *BUFFER* into the variable pointed to by the *pnEnumValues* parameter.
* The server MUST return zero for success or a nonzero error code if the method was not successful.

Except for diagnostic purposes, the server state as visible to the client through this protocol MUST NOT change as a result of processing the method call.

##### PRINTER\_HANDLE Parameters

*hPrinter*: An RPC context handle ([[C706]](http://go.microsoft.com/fwlink/?LinkId=89824)) to an object managed by the server. The individual method sections describe which methods are used to obtain the handle and which types of object (printer object, server object, port object, or job object) are acceptable.

The individual method sections include the following parameter validation steps by reference:

* Verify that *hPrinter* is an RPC context handle to an object managed by the server.
* Verify that *hPrinter* is an RPC context handle to an appropriate object as defined in the referring method definition.
* Unless noted otherwise, if the preceding validation steps fail, return ERROR\_INVALID\_PARAMETER as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

#### Printer Management and Discovery Methods

This section specifies methods for discovering and obtaining access to supported printers.

| Method | Description |
| --- | --- |
| [RpcEnumPrinters](#Section_0d83c60dec7546718a3b99517651ff1b) | RpcEnumPrinters enumerates available printers, print servers, domains, or print providers.  Opnum 0 |
| [RpcOpenPrinter](#Section_989357e2446e4872bb381dce21e1313f) | RpcOpenPrinter retrieves a handle for a printer, port, port monitor, print job, or print server.  Opnum 1 |
| [RpcAddPrinter](#Section_bd0bb8d526fd470581f5614d32b9f38f) | RpcAddPrinter adds a printer to the list of supported printers for a specified server.  Opnum 5 |
| [RpcDeletePrinter](#Section_e46c727493db443e8787ad49e5a08466) | RpcDeletePrinter deletes the specified printer object.  Opnum 6 |
| [RpcSetPrinter](#Section_382997b5b02a4b6a8c6849a64b8c45f8) | RpcSetPrinter sets the data for a specified printer or sets the state of the specified printer by pausing or resuming printing or clearing all print jobs.  Opnum 7 |
| [RpcGetPrinter](#Section_ccbf288a0768416dbede0ef1987fc286) | RpcGetPrinter retrieves information about a specified printer.  Opnum 8 |
| [RpcGetPrinterData](#Section_9c04a0f2adc443938df47c36a14183f5) | RpcGetPrinterData retrieves printer configuration data for a printer or print server.  Opnum 26 |
| [RpcSetPrinterData](#Section_e0d3e41e097c459da7fbacaf1343ea27) | RpcSetPrinterData sets the configuration data for a printer or print server.  Opnum 27 |
| [RpcClosePrinter](#Section_8482c18677734fb58034e9bae7a91e52) | RpcClosePrinter closes a handle to a printer object, server object, job object, or port object.  Opnum 29 |
| [RpcCreatePrinterIC](#Section_b0686ba23a6b4a6c943fb26600dbd88c) | RpcCreatePrinterIC called by the Graphics Device Interface (GDI) to create an information context for a printer.  Opnum 40 |
| [RpcPlayGdiScriptOnPrinterIC](#Section_eff167adf4004c6eb0e2c48b62ac0826) | RpcPlayGdiScriptOnPrinterIC returns identifying information for fonts available for printing to a printer object.  Opnum 41 |
| [RpcDeletePrinterIC](#Section_d9792de2d19c4c9bb7a4034523b7784b) | RpcDeletePrinterIC deletes a printer information context.  Opnum 42 |
| [RpcResetPrinter](#Section_6d979ef765e848af94c76bbb9d2be06a) | RpcResetPrinter resets the data type and device mode values to use for printing documents submitted by the [RpcStartDocPrinter](#Section_90f95ff03ae04ddea83d35f457c710a6) method (see section 3.1.4.9.1).  Opnum 52 |
| [RpcOpenPrinterEx](#Section_0e81ce1872b146c38584a205393b04ff) | RpcOpenPrinterEx retrieves a handle for a printer, port, port monitor, print job, or print server.  Opnum 69 |
| [RpcAddPrinterEx](#Section_3af45717308f4b4f8428430d12856110) | RpcAddPrinterEx installs a printer on the print server.  Opnum 70 |
| [RpcEnumPrinterData](#Section_51bfa051e7e44fc4b79eea9d4b08cb84) | RpcEnumPrinterData enumerates configuration data for a specified printer.  Opnum 72 |
| [RpcDeletePrinterData](#Section_a67521b7c92d4ebf900836fb321196ff) | RpcDeletePrinterData deletes specified configuration data for a printer.  Opnum 73 |
| [RpcSetPrinterDataEx](#Section_58520b8b44ce46fc90d30362457098e0) | RpcSetPrinterDataEx sets the configuration data for a printer or print server. This method extends RpcSetPrinterData (see section 3.1.4.2.8), but by additionally allowing the caller to specify the registry key under which to store the data.  Opnum 77 |
| [RpcGetPrinterDataEx](#Section_74a9f79910bf4ef6a2a40dcd9c43fb5c) | RpcGetPrinterDataEx retrieves configuration data for the specified printer or print server. This method extends RpcGetPrinterData (see section 3.1.4.2.7) and can retrieve values sorted under a specified key by RpcSetPrinterDataEx (see section 3.1.4.2.18).  Opnum 78 |
| [RpcEnumPrinterDataEx](#Section_9574b3e74a294bfaba67c35dab674162) | RpcEnumPrinterDataEx enumerates all value names and data for a specified printer and key. This method extends RpcEnumPrinterData (see section 3.1.4.2.16) by retrieving several values in a single call.  Opnum 79 |
| [RpcEnumPrinterKey](#Section_128ff57da4a44a23b2ee36f71ca5c0d5) | RpcEnumPrinterKey enumerates the subkeys of a specified key for a specified printer. Printer data is stored in the registry.  Opnum 80 |
| [RpcDeletePrinterDataEx](#Section_19c2e0e5b7ec4d068b211488f2275505) | RpcDeletePrinterDataEx deletes a specified value from a printer's configuration data, which consists of a set of named and typed values stored in a hierarchy of registry keys.  Opnum 81 |
| [RpcDeletePrinterKey](#Section_cde0dcc363264509ad3fe170977931ba) | RpcDeletePrinterKey deletes a specified key and all of its subkeys for a specified printer.  Opnum 82 |
| [RpcAddPerMachineConnection](#Section_3648c131b1554cc69a2d79b95b422a5b) | RpcAddPerMachineConnection stores the print server name and the name of the binary executable used as a provider for a particular connection.  Opnum 85 |
| [RpcDeletePerMachineConnection](#Section_8861fe40b2764c778ef7cc6a28063e0e) | RpcDeletePerMachineConnection deletes information about a server and connection provider.  Opnum 86 |
| [RpcEnumPerMachineConnections](#Section_3b1e7385d1ca48dfb372461c524f78d9) | RpcEnumPerMachineConnections enumerates each connection and copies [PRINTER\_INFO\_4](#Section_ac391864cd9945099ff3ea0366dd9b2b) structures (see section 2.2.1.10.5) for all the per-machine connections into the buffer *pPrinterEnum*.  Opnum 87 |
| [RpcSendRecvBidiData](#Section_5aec83472ebf464eb6d1540aa60b5c71) | RpcSendRecvBidiData sends and receives bidirectional data. This method is used to communicate with port monitors that support such data.  Opnum 97 |

##### RpcEnumPrinters (Opnum 0)

RpcEnumPrinters enumerates available printers, print servers, domains, or print providers.

1. DWORD RpcEnumPrinters(
2. [in] DWORD Flags,
3. [in, string, unique] STRING\_HANDLE Name,
4. [in] DWORD Level,
5. [in, out, unique, size\_is(cbBuf), disable\_consistency\_check]
6. BYTE\* pPrinterEnum,
7. [in] DWORD cbBuf,
8. [out] DWORD\* pcbNeeded,
9. [out] DWORD\* pcReturned
10. );

**Flags:** The types of print objects that this method enumerates. The value of this parameter MUST be the result of a bitwise OR of one or more of the [Printer Enumeration Flags (section 2.2.3.7)](#Section_2a1fe8a4e8be4cf38b378d19f9a2edcd), with the following additional specifications:

* **PRINTER\_ENUM\_NAME**: If the *Name* parameter is NULL or points to an empty string, and the *Level* parameter value is 0x00000001, available print providers SHOULD be enumerated. If this flag is not set, the server SHOULD ignore the *Name* parameter.
* **PRINTER\_ENUM\_REMOTE**: The *Level* parameter value MUST be 0x00000001.
* **PRINTER\_ENUM\_NETWORK**: The *Level* parameter value MUST be 0x00000001.

**Name:** NULL or a server name parameter as specified in [Printer Server Name Parameters (section 3.1.4.1.4)](#Section_acb355a4913b4abb8cb5415aea160015). If the *Flags* parameter contains the **PRINTER\_ENUM\_NAME** flag, the *Name* parameter value controls where the server SHOULD enumerate. The server SHOULD enumerate locally if the *Name* parameter is either NULL or an empty string; otherwise, remotely on the server whose name is specified by the *Name* string.

**Level:** The level of printer information structure, as follows.

| Value | Meaning |
| --- | --- |
| 0x00000000 | [\_PRINTER\_INFO\_STRESS (section 2.2.2.9.1)](#Section_b026cf5c680f464a8e08675228ddd42c) |
| 0x00000001 | [\_PRINTER\_INFO\_1 (section 2.2.2.9.2)](#Section_39aa2f28ffe04f918bddc0fa5dee40c5) |
| 0x00000002 | [\_PRINTER\_INFO\_2 (section 2.2.2.9.3)](#Section_9e3273757ac9419fa61061b88bbbd918) |
| 0x00000004 | [\_PRINTER\_INFO\_4 (section 2.2.2.9.5)](#Section_b739452348b74e16bc9d3c87d8a5f0c0) |
| 0x00000005 | [\_PRINTER\_INFO\_5 (section 2.2.2.9.6)](#Section_68a546b0d7484a3da5d737f713e037d9) |

**pPrinterEnum:** A pointer to a [BUFFER](#Section_f29e9057df2940d49c51c615c0da62a1) defined in INFO Structures Query Parameters (section 3.1.4.1.9).

**BUFFER TYPE**: \_PRINTER\_INFO.

This parameter can be NULL if the value of *cbBuf* is zero.

**cbBuf:** An input parameter that MUST adhere to the specification in INFO Structures Query Parameters.

**pcbNeeded:** An output parameter that MUST adhere to the specification in INFO Structures Query Parameters.

**pcReturned:** An output parameter that MUST adhere to the specification in INFO Structures Query Parameters.

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server SHOULD[<308>](#Appendix_A_308) validate parameters as follows:

* Perform validation steps as specified in Print Server Name Parameters (section 3.1.4.1.4).
* Perform validation steps as specified in INFO Structures Query Parameters (section 3.1.4.1.9).
* If the **PRINTER\_ENUM\_NETWORK** or **PRINTER\_ENUM\_REMOTE** flag is set, verify that the value of the *Level* parameter is 0x00000001. Otherwise, the server SHOULD[<309>](#Appendix_A_309) return ERROR\_INVALID\_LEVEL ([MS-ERREF] section 2.2).
* Additional validation MAY[<310>](#Appendix_A_310) be performed.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* If the value of the *Level* parameter is 0x00000001 and the **PRINTER\_ENUM\_NETWORK** bit is set in the *Flags* parameter, the server SHOULD enumerate all printers from the "List of Known Printers" (section [3.1.1](#Section_1bd61448db4b4284a4b18346732402d4)). [<311>](#Appendix_A_311)

If the server does not maintain a list of known printers, or if the list has not contained at least one entry for an implementation-specific period of time, the server MAY[<312>](#Appendix_A_312) return ERROR\_CAN\_NOT\_COMPLETE ([MS-ERREF] section 2.2).[<313>](#Appendix_A_313)

* For any other validated values for the *Level* and *Flags* parameters, the server SHOULD enumerate all printers in the "List of Printers" (section 3.1.1) on the print server or print provider that comply with the value of the *Flags* parameter. This information SHOULD be restricted for security reasons.[<314>](#Appendix_A_314)
* Using the enumerated objects, perform the processing and response steps specified in INFO Structures Query Parameters (section 3.1.4.1.9).
* Return the status of the operation.

##### RpcOpenPrinter (Opnum 1)

RpcOpenPrinter retrieves a handle for a printer, port, port monitor, print job, or print server.

1. DWORD RpcOpenPrinter(
2. [in, string, unique] STRING\_HANDLE pPrinterName,
3. [out] PRINTER\_HANDLE\* pHandle,
4. [in, string, unique] wchar\_t\* pDatatype,
5. [in] DEVMODE\_CONTAINER\* pDevModeContainer,
6. [in] DWORD AccessRequired
7. );

**pPrinterName:** A [STRING\_HANDLE (section 2.2.1.1.7)](#Section_59883eb85ce64ad4bdd2599434cfdcf2) for a printer connection, printer object, server object, job object, port object, or port monitor object. For opening a server object, this parameter MUST adhere to the specification in [Print Server Name Parameters (section 3.1.4.1.4)](#Section_acb355a4913b4abb8cb5415aea160015); for opening all other objects, it MUST adhere to the specification in [Printer Name Parameters (section 3.1.4.1.5)](#Section_3f2d41eb30e147e193bb534ea50cf8eb).

**pHandle:** A pointer to a [PRINTER\_HANDLE (section 2.2.1.1.4)](#Section_9c5d49b8692047b0946687ba910e6071) that MUST receive the RPC context handle [[C706]](http://go.microsoft.com/fwlink/?LinkId=89824) to the object identified by the *pPrinterName* parameter.

**pDatatype:** A pointer to a string that specifies the data type to be associated with the printer handle. This parameter MUST adhere to the specification in [Datatype Name Parameters (section 3.1.4.1.1)](#Section_c31e2daba809474c98640cbb61068903).

**pDevModeContainer:** A pointer to a [DEVMODE\_CONTAINER](#Section_c2b14fe764794460a7ba633a845dd61a) structure. This parameter MUST adhere to the specification in [DEVMODE\_CONTAINER Parameters (section 3.1.4.1.8.1)](#Section_bf31abc0da864ac68ebc8be665e11503).

**AccessRequired:** The access level that the client requires for interacting with the object to which a handle is being opened. The value of this parameter MUST be one of those specified in [Access Values (section 2.2.3.1)](#Section_945c4df5496949c5b1ce59c1b3f35024) or 0. For rules governing access values, see section [2.2.4.1](#Section_f5b6c93c04534c28a11ca9f966c4e3aa).

If AccessRequired is set to 0 (if no specific access level is requested), the server MUST assume a GENERIC\_READ (see section 2.2.3.1) access level.

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate parameters as follows:

* For opening a server object, perform the validation steps specified in Print Server Name Parameters; for opening all other objects, perform the validation steps specified in Printer Name Parameters.
* Perform the validation steps that are specified in Datatype Name Parameters (section 3.1.4.1.1).
* Perform the validation steps that are specified in DEVMODE\_CONTAINER Parameters (section 3.1.4.1.8.1).
* Verify that the client issuing the call has authorization equivalent to the value of the *AccessRequired* parameter.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Locate one of the following objects that corresponds to the request:
  + The printer in the "List of Printers", or a job queued for that printer
  + The server in the "List of Print Server Names"
  + The port monitor in the "List of Port Monitors"
  + The port in the "List of Ports".

These lists are specified in section [3.1.1](#Section_1bd61448db4b4284a4b18346732402d4).

* Create an implementation-specific representation of the printer, server, job, port monitor, or port ("the object") that MUST include:
  + A remote procedure call (RPC) handle, which is a snapshot of the printer, server, job, port monitor, or port data that is specific to this instance of the invocation.
  + The data from *pDataType* and *pDevModeContainer*, if they were not NULL.
  + All other relevant, implementation-specific data required to process all other protocol methods passing in a PRINTER\_HANDLE.
* Store the RPC handle for the object in the variable pointed to by *pHandle*.
* Increment the reference count of the object to prevent deletion.
* Return the status of the operation.

##### RpcAddPrinter (Opnum 5)

RpcAddPrinter adds a printer to the list of supported printers for a specified server.

1. DWORD RpcAddPrinter(
2. [in, string, unique] STRING\_HANDLE pName,
3. [in] PRINTER\_CONTAINER\* pPrinterContainer,
4. [in] DEVMODE\_CONTAINER\* pDevModeContainer,
5. [in] SECURITY\_CONTAINER\* pSecurityContainer,
6. [out] PRINTER\_HANDLE\* pHandle
7. );

**pName:** A parameter specified in [Print Server Name Parameters (section 3.1.4.1.4)](#Section_acb355a4913b4abb8cb5415aea160015).

**pPrinterContainer:** A parameter specified in [PRINTER\_CONTAINER Parameters (section 3.1.4.1.8.6)](#Section_07b78bd9247c4c58981e9a77538238ae). The **Level** member of the [PRINTER\_CONTAINER](#Section_ad7d4d4e0feb45e093edb3b2b9a313d1) MUST be 0x00000001 or 0x00000002.

**pDevModeContainer:** A parameter specified in [DEVMODE\_CONTAINER Parameters (section 3.1.4.1.8.1)](#Section_bf31abc0da864ac68ebc8be665e11503).

**pSecurityContainer:** A parameter specified in [SECURITY\_CONTAINER Parameters (section 3.1.4.1.8.7)](#Section_6c6051441c004365a6c75499676dcb52).

**pHandle:** A pointer to a variable that receives the printer RPC context handle to the printer object added. RPC context handles are specified in [[C706]](http://go.microsoft.com/fwlink/?LinkId=89824).

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate parameters as follows:

* Perform validation steps as specified in Print Server Name Parameters (section 3.1.4.1.4).
* Perform validation steps as specified in PRINTER\_CONTAINER Parameters (section 3.1.4.1.8.6).
* Perform validation steps as specified in DEVMODE\_CONTAINER Parameters (section 3.1.4.1.8.1).
* Perform validation steps as specified in SECURITY\_CONTAINER Parameters (section 3.1.4.1.8.7).
* If the **Level** member of the PRINTER\_CONTAINER is 0x00000002:
  + Verify that the printer driver specified in the [PRINTER\_INFO](#Section_125fb29b3c5141fca333b1e0e8794a91) that is pointed to by the **pointer** member of the PRINTER\_CONTAINER pointed to by *pPrinterContainer* already exists in the system, and if that verification fails, return ERROR\_UNKNOWN\_PRINTER\_DRIVER, as specified in [MS-ERREF].
  + Verify that the port specified in the PRINTER\_INFO that is pointed to by the **pointer** member of the PRINTER\_CONTAINER pointed to by *pPrinterContainer* already exists in the system, and if that verification fails, return ERROR\_UNKNOWN\_PORT, as specified in [MS-ERREF].
  + Verify that the print processor specified in the PRINTER\_INFO that is pointed to by the **pointer** member of the PRINTER\_CONTAINER pointed to by *pPrinterContainer* already exists in the system, and if that verification fails, return ERROR\_UNKNOWN\_PRINTPROCESSOR, as specified in [MS-ERREF].
  + Verify that the printer with the name specified in the PRINTER\_INFO that is pointed to by the **pointer** member of the PRINTER\_CONTAINER pointed to by *pPrinterContainer* does not already exist in the system, and if that verification fails, return ERROR\_PRINTER\_ALREADY\_EXISTS, as specified in [MS-ERREF].
* Additional validation MAY[<315>](#Appendix_A_315) be performed.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Perform PRINTER\_CONTAINER parameter processing steps as specified in PRINTER\_CONTAINER Parameters (section 3.1.4.1.8.6).
* If the value of the **Level** member of the PRINTER\_CONTAINER is 0x00000001, and if the server does not maintain a "List of Known Printers", the server MUST return ERROR\_PRINTER\_ALREADY\_EXISTS, as specified in [MS-ERREF]. Otherwise, the server MUST continue to process the message and compose a response to the client as follows:
  + If the [PRINTER\_ATTRIBUTE\_SHARED](#Section_1625e9d929e448f4b83d3bd0fdaea787) bit is set in the **Flags** member of the [PRINTER\_INFO\_1](#Section_0d1a296b7027441c83938aa5c3764b27) structure, add the printer[<316>](#Appendix_A_316) to the "List of Known Printers" as specified in section [3.1.1](#Section_1bd61448db4b4284a4b18346732402d4).
  + If PRINTER\_ATTRIBUTE\_SHARED bit is not set in the **Flags** member of the PRINTER\_INFO\_1 structure, remove the printer from the "List of Known Printers".
  + Store NULL in the output parameter pointed to by *pHandle*.
  + Increment the number of network printers added to this server.
  + Return ERROR\_PRINTER\_ALREADY\_EXISTS, as specified in [MS-ERREF].

**Note:** An error return code is required by remote procedure call (RPC) because NULL was stored to the output parameter pointed to by *pHandle*.

* If the **Level** member of the PRINTER\_CONTAINER is 0x00000002:
  + Create the printer object and assign to it the security descriptor from the [SECURITY\_CONTAINER](#Section_6cf2b4ff1b2b453c87f985243823cea8) that is pointed to by *pSecurityContainer* parameter.
  + Add the printer to the "List of Printers" (section 3.1.1).
  + Create a session that includes:
    - An RPC handle
    - A snapshot of the printer data specific to this instance of the printer invocation.
    - The data from the [DEVMODE](#Section_ccc2a501794e4d2bb312f69c75131c2e) that is contained in the [DEVMODE\_CONTAINER](#Section_c2b14fe764794460a7ba633a845dd61a) pointed to by the *pDevModeContainer* parameter if it is not NULL.
  + Store the RPC handle for the session in the output parameter pointed to by *pHandle*. The handle returned from this method MUST be granted [PRINTER\_ALL\_ACCESS](#Section_945c4df5496949c5b1ce59c1b3f35024) permission.
  + Increment the printer's reference count to prevent deletion.
  + If there are any clients that are registered for notifications on the server object change, a notification MUST be sent to those clients.
* Return the status of the operation.

##### RpcDeletePrinter (Opnum 6)

RpcDeletePrinter is a method that deletes the specified printer object.

1. DWORD RpcDeletePrinter(
2. [in] PRINTER\_HANDLE hPrinter
3. );

**hPrinter:** This parameter MUST specify a handle to a printer object that MUST have been opened by using the [RpcAddPrinter](#Section_bd0bb8d526fd470581f5614d32b9f38f), [RpcAddPrinterEx](#Section_3af45717308f4b4f8428430d12856110), [RpcOpenPrinter](#Section_989357e2446e4872bb381dce21e1313f), or [RpcOpenPrinterEx](#Section_0e81ce1872b146c38584a205393b04ff) methods.

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate parameters as follows:

* Perform the validation steps as specified in [PRINTER\_HANDLE Parameters (section 3.1.4.1.11)](#Section_10bfa8c2d4564ad0912de674766a36c8).
* If any jobs are pending on the printer, use the implementation-specific policy[<317>](#Appendix_A_317) to determine if a delete operation can be made pending or if an error should be returned.

Additional validation MAY[<318>](#Appendix_A_318) be performed.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Mark the printer object as "Delete Pending".
* Modify the list of printers in the system to exclude the deleted printer for any subsequent calls to [RpcEnumPrinters](#Section_0d83c60dec7546718a3b99517651ff1b), RpcOpenPrinter, RpcOpenPrinterEx, and [RpcStartDocPrinter](#Section_90f95ff03ae04ddea83d35f457c710a6). Clients that already have a valid handle to the same printer object MAY continue using the printer object for all operations except RpcStartDocPrinter.
* If the deleted printer has been published to the directory service, delete the print queue object from the directory as described in section [2.3.3.2](#Section_1c49f0840f634ac49e566be4c47915e6).[<319>](#Appendix_A_319) If the directory service operation fails, the print server MUST continue processing the printer deletion operation and MUST NOT return the status of the directory service operation to the client.
* If any clients have registered for notifications of the server object change, a notification MUST be broadcast to them.
* Return the status of the operation.

##### RpcSetPrinter (Opnum 7)

RpcSetPrinter sets the data or state of a specified printer by pausing or resuming printing or by clearing all print jobs.

1. DWORD RpcSetPrinter(
2. [in] PRINTER\_HANDLE hPrinter,
3. [in] PRINTER\_CONTAINER\* pPrinterContainer,
4. [in] DEVMODE\_CONTAINER\* pDevModeContainer,
5. [in] SECURITY\_CONTAINER\* pSecurityContainer,
6. [in] DWORD Command
7. );

**hPrinter:** A handle to a printer object or server object that was opened by [RpcAddPrinter (section 3.1.4.2.3)](#Section_bd0bb8d526fd470581f5614d32b9f38f), [RpcAddPrinterEx (section 3.1.4.2.15)](#Section_3af45717308f4b4f8428430d12856110), [RpcOpenPrinter (section 3.1.4.2.2)](#Section_989357e2446e4872bb381dce21e1313f), or [RpcOpenPrinterEx (section 3.1.4.2.14)](#Section_0e81ce1872b146c38584a205393b04ff).

**pPrinterContainer:** A parameter specified in [PRINTER\_CONTAINER Parameters (section 3.1.4.1.8.6)](#Section_07b78bd9247c4c58981e9a77538238ae). If the *Command* is 0, the **Level** member of the [PRINTER\_CONTAINER](#Section_ad7d4d4e0feb45e093edb3b2b9a313d1) MUST be 0x00000000 or a number from 0x00000002 to 0x00000007. If the *Command* parameter is 1, 2, or 3, the **Level** member of the PRINTER\_CONTAINER MUST be 0x00000000.

**pDevModeContainer:** A parameter specified in [DEVMODE\_CONTAINER Parameters (section 3.1.4.1.8.1)](#Section_bf31abc0da864ac68ebc8be665e11503).

**pSecurityContainer:** A parameter specified in [SECURITY\_CONTAINER Parameters (section 3.1.4.1.8.7)](#Section_6c6051441c004365a6c75499676dcb52).

**Command:** A value that specifies an action to perform. If this value is non-zero, it MUST be one of the values in the following table, and the **Level** member of the PRINTER\_CONTAINER (section 2.2.1.2.9) structure that is pointed to by the *pPrinterContainer* parameter MUST be 0x00000000.

| Printer Control Name/Value | Meaning |
| --- | --- |
| PRINTER\_CONTROL\_PAUSE  0x00000001 | Pauses the printer object. |
| PRINTER\_CONTROL\_RESUME  0x00000002 | Resumes a paused printer object. |
| PRINTER\_CONTROL\_PURGE  0x00000003 | Deletes all print jobs queued for the printer object. |

If this value is zero, the **PrinterInfo** member of the PRINTER\_CONTAINER structure that is pointed to by the *pPrinterContainer* parameter MUST contain a pointer to a [PRINTER\_INFO (section 2.2.2.9)](#Section_fc749a2738ec4063b0c1dc477c120ffd) structure that this method can use. See section [2.2.1.10.1](#Section_9e94b8f4a84b4c96b9ae4b4acb3469b8) for details.

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate parameters as follows:

* Perform the validation steps specified in [PRINTER\_HANDLE Parameters (section 3.1.4.1.11)](#Section_10bfa8c2d4564ad0912de674766a36c8).
* Perform the validation steps specified in PRINTER\_CONTAINER Parameters.
* Verify that the information provided in the PRINTER\_CONTAINER that is pointed to by the *pPrinterContainer* parameter is consistent with the value in *Command*, according to the following table:

| Command | Level in PRINTER\_CONTAINER |
| --- | --- |
| 0x00000000 | MUST be 0x00000000 or a number from 0x00000002 to 0x00000007, inclusive. |
| 0x00000001 | MUST be 0x00000000. |
| 0x00000002 | MUST be 0x00000000. |
| 0x00000003 | MUST be 0x00000000. |

If this verification fails, return ERROR\_INVALID\_LEVEL.

* Perform the validation steps specified in DEVMODE\_CONTAINER Parameters (section 3.1.4.1.8.1).
* Perform the validation steps specified in SECURITY\_CONTAINER Parameters (section 3.1.4.1.8.7).
* If the **Level** member of the PRINTER\_CONTAINER is 0x00000007 and the **dwAction** field in the **PRINTER\_INFO\_7** structure that is pointed to by the **pPrinterInfo7** member of the PRINTER\_CONTAINER pointed to by **pPrinterContainer** is DSPRINT\_UPDATE (0x00000002):
  + Verify that *hPrinter* specifies a printer object that is already published in the directory service. If *hPrinter* represents a printer that is not already published in the directory service, the server MUST fail the call by returning an HRESULT (see [MS-ERREF] section 2.1) code of 0x80070002 with the same meaning as the ERROR\_FILE\_NOT\_FOUND error specified in [MS-ERREF].
* Additional validation SHOULD[<320>](#Appendix_A_320) be performed.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* If *hPrinter* specifies a server object, the server MUST only apply the [SECURITY\_CONTAINER](#Section_6cf2b4ff1b2b453c87f985243823cea8) parameter to set the print server's security descriptor, and MUST not perform the remaining processing steps that follow.
* Perform PRINTER\_CONTAINER parameter processing steps that are specified in PRINTER\_CONTAINER Parameters.
* Perform the operation from the following table that corresponds to the value of the *Command* parameter.

| Command parameter values | Operation that MUST be performed |
| --- | --- |
| 0x0000000  No command | Update the printer configuration using the settings in *pPrinterContainer*.[<321>](#Appendix_A_321) |
| 0x00000001  Pause the printer | Temporarily suspend sending data to the printer without changing the state of any jobs associated with the printer. Clients are allowed to continue adding data to the job. |
| 0x00000002  Resume the printer | Resume sending data to the printer without changing the state of any jobs associated with the printer. |
| 0x00000003  Purge the printer | Remove all jobs that are currently associated with the printer and mark them as having failed to print. |

* If *hPrinter* specifies a printer object that the print server has published in the Active Directory, and as a result of this call the print server has changed the printer driver associated with the printer or changed printer configuration information that the print server has published in the Active Directory, update the print queue object in the directory as described in section [2.3.3.2](#Section_1c49f0840f634ac49e566be4c47915e6) with attributes obtained from the print queue configuration and the printer driver as described in section [2.3.3.1](#Section_3f4f0a52e417495aad460466a9d0ef9d).[<322>](#Appendix_A_322) If the directory service operation fails, the print server MUST continue processing the current call and MUST NOT return the status of the directory service operation to the client.
* If any clients registered for notifications of the printer object change, a notification MUST be broadcast to them.
* Return the status of the operation.

##### RpcGetPrinter (Opnum 8)

RpcGetPrinter retrieves information about a specified printer.

1. DWORD RpcGetPrinter(
2. [in] PRINTER\_HANDLE hPrinter,
3. [in] DWORD Level,
4. [in, out, unique, size\_is(cbBuf), disable\_consistency\_check]
5. BYTE\* pPrinter,
6. [in] DWORD cbBuf,
7. [out] DWORD\* pcbNeeded
8. );

**hPrinter:** A handle to a printer object that was opened by [RpcAddPrinter (section 3.1.4.2.3)](#Section_bd0bb8d526fd470581f5614d32b9f38f), [RpcAddPrinterEx (section 3.1.4.2.15)](#Section_3af45717308f4b4f8428430d12856110), [RpcOpenPrinter (section 3.1.4.2.2)](#Section_989357e2446e4872bb381dce21e1313f), or [RpcOpenPrinterEx (section 3.1.4.2.14)](#Section_0e81ce1872b146c38584a205393b04ff). This value MAY be a handle to a print server object.[<323>](#Appendix_A_323)

**Level:** The level of printer information structure, as follows.

| Value | Meaning |
| --- | --- |
| 0x00000000 | Corresponds to [\_PRINTER\_INFO\_STRESS (section 2.2.2.9.1)](#Section_b026cf5c680f464a8e08675228ddd42c). |
| 0x00000001 | Corresponds to [\_PRINTER\_INFO\_1 (section 2.2.2.9.2)](#Section_39aa2f28ffe04f918bddc0fa5dee40c5). |
| 0x00000002 | Corresponds to [\_PRINTER\_INFO\_2 (section 2.2.2.9.3)](#Section_9e3273757ac9419fa61061b88bbbd918). |
| 0x00000003 | Corresponds to [\_PRINTER\_INFO\_3 (section 2.2.2.9.4)](#Section_0cc3c48367144682aee454f588cab949). |
| 0x00000004 | Corresponds to [\_PRINTER\_INFO\_4 (section 2.2.2.9.5)](#Section_b739452348b74e16bc9d3c87d8a5f0c0). |
| 0x00000005 | Corresponds to [\_PRINTER\_INFO\_5 (section 2.2.2.9.6)](#Section_68a546b0d7484a3da5d737f713e037d9). |
| 0x00000006 | Corresponds to [\_PRINTER\_INFO\_6 (section 2.2.2.9.7)](#Section_f784565a2b814cc0a1e31c77212f7e28). |
| 0x00000007 | Corresponds to [\_PRINTER\_INFO\_7 (section 2.2.2.9.8)](#Section_01d350511b3945f8ad3964391d8fb445). |
| 0x00000008 | Corresponds to [\_PRINTER\_INFO\_8 (section 2.2.2.9.9)](#Section_55f96f18a1344778b1fa556a801997b7). |
| 0x00000009 | Not valid remotely; the server MUST respond by returning ERROR\_NOT\_SUPPORTED. |

**pPrinter:** A pointer to a [BUFFER](#Section_f29e9057df2940d49c51c615c0da62a1) (INFO Structures Query Parameters (section 3.1.4.1.9)).

**BUFFER TYPE:** \_PRINTER\_INFO.

**cbBuf:** A parameter specified in INFO Structures Query Parameters.

**pcbNeeded:** A parameter specified in INFO Structures Query Parameters.

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate parameters as follows:

* Perform the validation steps specified in [PRINTER\_HANDLE Parameters (section 3.1.4.1.11)](#Section_10bfa8c2d4564ad0912de674766a36c8).
* Perform the validation steps specified in INFO Structures Query Parameters.
* Additional validation MAY[<324>](#Appendix_A_324) be performed.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Using information about the printer, perform the processing and response steps specified in INFO Structures Query Parameters.
* Return the status of the operation.

##### RpcGetPrinterData (Opnum 26)

RpcGetPrinterData retrieves printer configuration data for a printer or print server.

1. DWORD RpcGetPrinterData(
2. [in] PRINTER\_HANDLE hPrinter,
3. [in, string] wchar\_t\* pValueName,
4. [out] DWORD\* pType,
5. [out, size\_is(nSize)] BYTE\* pData,
6. [in] DWORD nSize,
7. [out] DWORD\* pcbNeeded
8. );

**hPrinter:** A handle to a printer object or server object that was opened by [RpcAddPrinter (section 3.1.4.2.3)](#Section_bd0bb8d526fd470581f5614d32b9f38f), [RpcAddPrinterEx (section 3.1.4.2.15)](#Section_3af45717308f4b4f8428430d12856110), [RpcOpenPrinter (section 3.1.4.2.2)](#Section_989357e2446e4872bb381dce21e1313f), or [RpcOpenPrinterEx (section 3.1.4.2.14)](#Section_0e81ce1872b146c38584a205393b04ff).

**pValueName:** A pointer to a string that identifies the configuration data to get. For rules governing value names, see section [2.2.4.18](#Section_cd9850cf1e6c48759b65e159cf2d83d6).

For print servers, the value name MUST be one of the predefined strings listed in [Server Handle Key Values (section 2.2.3.10)](#Section_1393a76f585149a5bf147188ca6763b2).

For printer objects, the value name MAY be one of the predefined strings listed in [Printer Data Values (section 2.2.3.11)](#Section_4a81aa1c469e49418c4db8c4093b7fc2). Also, the value name "ChangeID"[<325>](#Appendix_A_325) is reserved by the protocol and has a special meaning. It identifies a read-only value that specifies that a change identifier MUST be returned in the buffer pointed to by *pData*. This identifier is a **DWORD** that MUST be set by the print server to a new, unique value each time printer information changes. The client SHOULD use the change identifier to decide if it has stale information about a printer object, in which case it SHOULD call this method or [RpcGetPrinter (section 3.1.4.2.6)](#Section_ccbf288a0768416dbede0ef1987fc286) to update its view of the printer object. Only the fact that the *pData* buffer value changes is significant; the change identifier value itself is arbitrary. If the value name is not one of these predefined strings, it is an arbitrary string defined by the printer driver associated with the printer object or by client applications.

**pType:** A parameter specified in [Dynamically Typed Query Parameters (section 3.1.4.1.2)](#Section_183564fd8a694fa2a01e51cebc7fb475).

**pData:** A pointer to BUFFER as specified in Dynamically Typed Query Parameters.

This parameter can be NULL if *nSize* equals zero

**nSize:** A parameter specified in Dynamically Typed Query Parameters.

**pcbNeeded:** A parameter specified in Dynamically Typed Query Parameters.

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server validates parameters as follows:

* The server MUST perform the validation steps that are specified in [PRINTER\_HANDLE Parameters (section 3.1.4.1.11)](#Section_10bfa8c2d4564ad0912de674766a36c8).
* The server MUST perform the validation steps that are specified in Dynamically Typed Query Parameters.
* For server objects, the server MUST verify that the *pValueName* parameter points to a string that is one of the predefined value names listed in Server Handle Key Values (section 2.2.3.10). If this verification fails, return ERROR\_INVALID\_PARAMETER.
* For printer objects, the server MUST verify that, if the *pValueName* parameter points to a string that is one of the predefined value names listed in Printer Data Values (section 2.2.3.11), the print server supports retrieving the value for this printer according to the rules in section 2.2.3.11. If this verification fails, return ERROR\_NOT\_SUPPORTED.
* For printer objects, the server MAY verify that the *pValueName* parameter points to a string that complies with the rules specified in section 2.2.4.18.
* The server MUST NOT perform access checks on the *hPrinter* object.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* With the data identified by *pValueName*, perform the processing and response steps[<326>](#Appendix_A_326) specified in Dynamically Typed Query Parameters.
* Return the status of the operation.

##### RpcSetPrinterData (Opnum 27)

RpcSetPrinterData sets the configuration data for a printer or print server.

1. DWORD RpcSetPrinterData(
2. [in] PRINTER\_HANDLE hPrinter,
3. [in, string] wchar\_t\* pValueName,
4. [in] DWORD Type,
5. [in, size\_is(cbData)] BYTE\* pData,
6. [in] DWORD cbData
7. );

**hPrinter:** A handle to a printer object or server object that was opened by [RpcAddPrinter (section 3.1.4.2.3)](#Section_bd0bb8d526fd470581f5614d32b9f38f), [RpcAddPrinterEx (section 3.1.4.2.15)](#Section_3af45717308f4b4f8428430d12856110), [RpcOpenPrinter (section 3.1.4.2.2)](#Section_989357e2446e4872bb381dce21e1313f), or [RpcOpenPrinterEx (section 3.1.4.2.14)](#Section_0e81ce1872b146c38584a205393b04ff).

**pValueName:** A pointer to a string that identifies the configuration data to set. For rules governing value names, see section [2.2.4.18](#Section_cd9850cf1e6c48759b65e159cf2d83d6).

For print servers, the value name MUST be one of the predefined strings listed in [Server Handle Key Values (section 2.2.3.10)](#Section_1393a76f585149a5bf147188ca6763b2).

For printer objects, the value name is an arbitrary string defined by the printer driver associated with the printer object. The value name "ChangeID"[<327>](#Appendix_A_327) is reserved by the protocol and MAY NOT be used in a call to RpcSetPrinterData.

**Type:** The type value for data pointed to by the *pData* parameter. This value SHOULD be one of the type codes defined in section [2.2.3.9](#Section_25cce7007fcf4bb6a2f30f6d08430a55). For rules governing registry type values, see section [2.2.4.15](#Section_f3ded7900d0a462b9bd173051033cb8a).

**pData:** A pointer to an array of bytes that contain the printer configuration data. The type of the data in the buffer is specified by the *Type* parameter.

**cbData:** The size, in bytes, of the *pData* array. This value SHOULD NOT be zero.

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server validates parameters as follows:

* The server MUST perform the validation steps that are specified in [PRINTER\_HANDLE Parameters (section 3.1.4.1.11)](#Section_10bfa8c2d4564ad0912de674766a36c8),.
* For server objects, the server MUST verify that the *pValueName* parameter points to a string that is one of the predefined value names listed in Server Handle Key Values with the "read-write" column selected. If this verification fails, return ERROR\_INVALID\_PARAMETER.
* For printer objects, the server MAY verify that the *pValueName* parameter points to a string that complies with the rules specified in section 2.2.4.18.
* Additional validation SHOULD[<328>](#Appendix_A_328) be performed.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Set the printer data[<329>](#Appendix_A_329) associated with *pValueName* to the data pointed to by *pData*.
* Return the status of the operation.

##### RpcClosePrinter (Opnum 29)

RpcClosePrinter closes a handle to a printer object, server object, job object, or port object.

1. DWORD RpcClosePrinter(
2. [in, out] PRINTER\_HANDLE\* phPrinter
3. );

**phPrinter:** A pointer to a handle to a printer object, server object, job object, or port object that was opened by [RpcAddPrinter (section 3.1.4.2.3)](#Section_bd0bb8d526fd470581f5614d32b9f38f), [RpcAddPrinterEx (section 3.1.4.2.15)](#Section_3af45717308f4b4f8428430d12856110), [RpcOpenPrinter (section 3.1.4.2.2)](#Section_989357e2446e4872bb381dce21e1313f), or [RpcOpenPrinterEx (section 3.1.4.2.14)](#Section_0e81ce1872b146c38584a205393b04ff).

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate parameters as follows:

* Perform the validation steps that are specified in [PRINTER\_HANDLE Parameters (section 3.1.4.1.11)](#Section_10bfa8c2d4564ad0912de674766a36c8). This method SHOULD assume that the handle to the printer object, server object, job object, or port object can be used without further access checks.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client.

Otherwise, the server MUST process the message and compose a response to the client as follows:

* If the object is a printer object, and [RpcStartDocPrinter](#Section_90f95ff03ae04ddea83d35f457c710a6) has been called without a matching [RpcEndDocPrinter](#Section_209fcf3205b444aba3a4779cfae181ca), the same processing as for RpcEndDocPrinter MUST occur.

If there is an active notification context associated with the object, as a result of the client not calling [RpcFindClosePrinterChangeNotification](#Section_f50c8a861b2d4124b2205b840aacd2c7), the server MUST close the notification context now by calling the client's [RpcReplyClosePrinter](#Section_d6c73ecbc02b4150a01bad73a642990a) method.

* Free any internal state that is associated with the handle that is pointed to by *phPrinter*.
* Store NULL in the variable that is pointed to by *phPrinter*.
* Decrement the reference count on that object.
* If the object is a printer object marked as "Delete Pending", and the usage count is zero, the following steps MUST be performed:
  + Handle any pending jobs in an implementation-specific manner.
  + Clear references to this printer from any other data structures.
  + Delete the printer object.
* Return the status of the operation.

##### RpcCreatePrinterIC (Opnum 40)

RpcCreatePrinterIC is called by the Graphics Device Interface (GDI) to create an information context for a specified printer.

1. DWORD RpcCreatePrinterIC(
2. [in] PRINTER\_HANDLE hPrinter,
3. [out] GDI\_HANDLE\* pHandle,
4. [in] DEVMODE\_CONTAINER\* pDevModeContainer
5. );

**hPrinter:** A handle to a printer object (section [2.2.1.1.4](#Section_9c5d49b8692047b0946687ba910e6071)) that was opened by [RpcAddPrinter (section 3.1.4.2.3)](#Section_bd0bb8d526fd470581f5614d32b9f38f), [RpcAddPrinterEx (section 3.1.4.2.15)](#Section_3af45717308f4b4f8428430d12856110), [RpcOpenPrinter (section 3.1.4.2.2)](#Section_989357e2446e4872bb381dce21e1313f), or [RpcOpenPrinterEx (section 3.1.4.2.14)](#Section_0e81ce1872b146c38584a205393b04ff).

**pHandle:** A pointer to a printer information context handle (section [2.2.1.1.2](#Section_ec9126fabc33423fadfc1bf52bd7890e)).

**pDevModeContainer:** A parameter specified in [DEVMODE\_CONTAINER Parameters (section 3.1.4.1.8.1)](#Section_bf31abc0da864ac68ebc8be665e11503).

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate parameters as follows:

* Perform the validation steps specified in [PRINTER\_HANDLE Parameters (section 3.1.4.1.11)](#Section_10bfa8c2d4564ad0912de674766a36c8). This method SHOULD assume that the handle to the printer object can be used without further access checks.
* Perform the validation steps specified in DEVMODE\_CONTAINER Parameters.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Perform implementation-specific steps to create the appropriate printer information context.
* Store an RPC context handle associated with the information context in *pHandle*.
* Return the status of the operation.

Except for diagnostic purposes, the server state, as visible to the client through this or any other protocol, MUST NOT change as a result of processing this call.

##### RpcPlayGdiScriptOnPrinterIC (Opnum 41)

RpcPlayGdiScriptOnPrinterIC returns font information for a printer connection. [UNIVERSAL\_FONT\_ID (section 2.2.2.12)](#Section_97cd8dd84c4b4a7aa160fae63f3e59bf) structures are used to identify the fonts.

1. DWORD RpcPlayGdiScriptOnPrinterIC(
2. [in] GDI\_HANDLE hPrinterIC,
3. [in, size\_is(cIn)] BYTE\* pIn,
4. [in] DWORD cIn,
5. [out, size\_is(cOut)] BYTE\* pOut,
6. [in] DWORD cOut,
7. [in] DWORD ul
8. );

**hPrinterIC:** A printer information context handle (section [2.2.1.1.2](#Section_ec9126fabc33423fadfc1bf52bd7890e)) that was returned by [RpcCreatePrinterIC (section 3.1.4.2.10)](#Section_b0686ba23a6b4a6c943fb26600dbd88c).

**pIn:** A pointer that SHOULD be set to NULL when sent and MUST be ignored on receipt.

**cIn:** A value that SHOULD be set to zero when sent and MUST be ignored on receipt.

**pOut:** A pointer to a buffer, the size and contents of which are determined by the value of the *cOut* parameter.

**cOut:** The size, in bytes, of the buffer pointed to by *pOut*.

**ul:** A value that SHOULD be set to zero when sent and MUST be ignored on receipt.

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate the following:

* Verify that *hPrinterIC* is a valid printer information context handle. This method SHOULD assume that this handle can be used without further access checks.[<330>](#Appendix_A_330)
* Verify that the value of the *pOut* parameter is not NULL.[<331>](#Appendix_A_331)
* Verify the value of the *cOut* parameter as follows:
  + If *cOut* is less than 0x00000004, ERROR\_NOT\_ENOUGH\_MEMORY SHOULD be returned, as specified in [MS-ERREF].
  + If *cOut* is equal to 0x00000004, proceed.
  + If *cOut* is greater than 0x00000004, it specifies the size of the buffer pointed to by the *pOut* parameter. In this case, the minimum value of *cOut* is computed as follows:

1. ((\*pOut) \* (size of(UNIVERSAL\_FONT\_ID))) + 0x00000004

The buffer pointed to by *pOut* MUST be large enough to contain all the font information plus a DWORD for the number of fonts. If the value of *cOut* is less than this minimum, ERROR\_NOT\_ENOUGH\_MEMORY SHOULD be returned.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* If *cOut* is equal to 0x00000004, the value of *pOut* is a pointer to a DWORD that specifies the number of UNIVERSAL\_FONT\_ID structures to be returned by the next call to this method.
* If *cOut* is greater than 0x00000004, font information MUST be returned as follows:
  1. Query the fonts that are available on the server.
  2. Write the DWORD number of fonts to the buffer location that is pointed to by the value of the *pOut* parameter.
  3. Write UNIVERSAL\_FONT\_ID structures for the fonts to the buffer location that is pointed to by the value of *pOut* plus 0x00000004.

A print client MAY assume that the fonts identified by the UNIVERSAL\_FONT\_ID structures are available on the print server for use in spooled print jobs.[<332>](#Appendix_A_332)

* Return the status of the operation.

Except for diagnostic purposes, the server state, as visible to the client through this or any other protocol, MUST NOT change as a result of processing this call.

##### RpcDeletePrinterIC (Opnum 42)

RpcDeletePrinterIC deletes a printer information context.

1. DWORD RpcDeletePrinterIC(
2. [in, out] GDI\_HANDLE\* phPrinterIC
3. );

**phPrinterIC:** A pointer to a printer information context handle (section [2.2.1.1.2](#Section_ec9126fabc33423fadfc1bf52bd7890e)) that was returned by [RpcCreatePrinterIC (section 3.1.4.2.10)](#Section_b0686ba23a6b4a6c943fb26600dbd88c).

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST verify that the handle pointed to by the *phPrinterIC* parameter is associated with an information context. This method SHOULD assume that this handle can be used without further access checks.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Delete the printer information context.
* Store NULL in the variable pointed to by *phPrinterIC*.
* Return the status of the operation.

Except for diagnostic purposes, the server state, as visible to the client through this or any other protocol, MUST NOT change as a result of processing this call.

##### RpcResetPrinter (Opnum 52)

RpcResetPrinter resets the data type and device mode (For more information, see [[DEVMODE]](http://go.microsoft.com/fwlink/?LinkId=89844)) values to use for printing documents submitted by the [RpcStartDocPrinter (section 3.1.4.9.1)](#Section_90f95ff03ae04ddea83d35f457c710a6) method.

1. DWORD RpcResetPrinter(
2. [in] PRINTER\_HANDLE hPrinter,
3. [in, string, unique] wchar\_t\* pDatatype,
4. [in] DEVMODE\_CONTAINER\* pDevModeContainer
5. );

**hPrinter:** A handle to a printer object that was opened by [RpcAddPrinter (section 3.1.4.2.3)](#Section_bd0bb8d526fd470581f5614d32b9f38f), [RpcAddPrinterEx (section 3.1.4.2.15)](#Section_3af45717308f4b4f8428430d12856110), [RpcOpenPrinter (section 3.1.4.2.2)](#Section_989357e2446e4872bb381dce21e1313f), or [RpcOpenPrinterEx (section 3.1.4.2.14)](#Section_0e81ce1872b146c38584a205393b04ff).

**pDatatype:** A parameter specified in [Datatype Name Parameters (section 3.1.4.1.1)](#Section_c31e2daba809474c98640cbb61068903).

**pDevModeContainer:** A parameter specified in [DEVMODE\_CONTAINER Parameters](#Section_bf31abc0da864ac68ebc8be665e11503).

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate parameters as follows:

* Perform the validation steps specified in [PRINTER\_HANDLE Parameters (section 3.1.4.1.11)](#Section_10bfa8c2d4564ad0912de674766a36c8). This method SHOULD assume that the handle to the printer object can be used without further access checks.
* Perform the validation steps specified in Datatype Name Parameters.
* Perform the validation steps specified in DEVMODE\_CONTAINER Parameters.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Update the default data type that is associated with the context for *hPrinter*.
* Update the default [DEVMODE](#Section_ccc2a501794e4d2bb312f69c75131c2e) structure that is associated with the context for *hPrinter*.
* Return the status of the operation.

##### RpcOpenPrinterEx (Opnum 69)

RpcOpenPrinterEx retrieves a handle for a printer, port, port monitor, print job, or print server.[<333>](#Appendix_A_333)

1. DWORD RpcOpenPrinterEx(
2. [in, string, unique] STRING\_HANDLE pPrinterName,
3. [out] PRINTER\_HANDLE\* pHandle,
4. [in, string, unique] wchar\_t\* pDatatype,
5. [in] DEVMODE\_CONTAINER\* pDevModeContainer,
6. [in] DWORD AccessRequired,
7. [in] SPLCLIENT\_CONTAINER\* pClientInfo
8. );

**pPrinterName:** A [STRING\_HANDLE (section 2.2.1.1.7)](#Section_59883eb85ce64ad4bdd2599434cfdcf2) for a printer connection, printer object, server object, job object, port object, or port monitor object. For opening a server object, this parameter MUST adhere to the specification in [Print Server Name Parameters (section 3.1.4.1.4)](#Section_acb355a4913b4abb8cb5415aea160015); for opening all other objects, it MUST adhere to the specification in [Printer Name Parameters (section 3.1.4.1.5)](#Section_3f2d41eb30e147e193bb534ea50cf8eb).

**pHandle:** A pointer to a [PRINTER\_HANDLE (section 2.2.1.1.4)](#Section_9c5d49b8692047b0946687ba910e6071) that MUST receive the RPC context handle [[C706]](http://go.microsoft.com/fwlink/?LinkId=89824) to the object identified by the *pPrinterName* parameter.

**pDatatype:** A pointer to a string that specifies the data type to be associated with the printer handle. This parameter MUST adhere to the specification in [Datatype Name Parameters (section 3.1.4.1.1)](#Section_c31e2daba809474c98640cbb61068903).

**pDevModeContainer:** A pointer to a [DEVMODE\_CONTAINER](#Section_c2b14fe764794460a7ba633a845dd61a) structure. This parameter MUST adhere to the specification in [DEVMODE\_CONTAINER Parameters (section 3.1.4.1.8.1)](#Section_bf31abc0da864ac68ebc8be665e11503).

**AccessRequired:** The access level that the client requires for interacting with the object to which a handle is being opened. The value of this parameter MUST be one of those specified in [Access Values (section 2.2.3.1)](#Section_945c4df5496949c5b1ce59c1b3f35024). For rules governing access values, see section [2.2.4.1](#Section_f5b6c93c04534c28a11ca9f966c4e3aa).

If no specific access level is requested, the server MUST assume a generic read access level.

**pClientInfo:** A pointer to a [SPLCLIENT\_CONTAINER (section 2.2.1.2.14)](#Section_cca15c7257d342fc91e749442b623c11) structure. This parameter MUST adhere to the specification in [SPLCLIENT\_CONTAINER Parameters (section 3.1.4.1.8.8)](#Section_e6a9e216bc2c43209f67ac3866fb2bc5).

The value of the **Level** member of the container that is pointed to by *pClientInfo* MUST be 0x00000001.

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate parameters as follows:

* For opening a server object, perform the validation steps specified in Print Server Name Parameters; for opening all other objects, perform the validation steps specified in Printer Name Parameters.
* Perform the validation steps that are specified in Datatype Name Parameters.
* Perform the validation steps that are specified in DEVMODE\_CONTAINER Parameters (section 3.1.4.1.8.1).
* Perform the validation steps that are specified in SPLCLIENT\_CONTAINER Parameters (section 3.1.4.1.8.8).
* Verify that the client issuing the call has authorization equivalent to the value of the *AccessRequired* parameter. If verification fails, return ERROR\_ACCESS\_DENIED.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Locate one of the following objects that correspond to the request:
  + The printer in the "List of Printers", or a job queued for that printer;
  + The server in the "List of Print Server Names"; or
  + The port monitor in the "List of Port Monitors"; or
  + The port in the "List of Ports".

These lists are specified in section [3.1.1](#Section_1bd61448db4b4284a4b18346732402d4).

* Create an implementation-specific representation of the printer, server, job, port monitor, or port ("the object") that MUST include:
  + An **RPC handle**, which is a snapshot of the printer, server, job, port monitor, or port data that is specific to this instance of the invocation.
  + The data from *pDatatype* and *pDevModeContainer*, if they are not NULL.
  + All other relevant, implementation-specific data that is required to process all other protocol methods that pass in a PRINTER\_HANDLE.
  + The data from *pClientInfo*, if it is not NULL.
* Increment the object's reference count to prevent deletion.
* Store the **RPC handle** for the session in the output parameter *pHandle*.
* Return the status of the operation.

##### RpcAddPrinterEx (Opnum 70)

RpcAddPrinterEx installs a printer on the print server.[<334>](#Appendix_A_334)

1. DWORD RpcAddPrinterEx(
2. [in, string, unique] STRING\_HANDLE pName,
3. [in] PRINTER\_CONTAINER\* pPrinterContainer,
4. [in] DEVMODE\_CONTAINER\* pDevModeContainer,
5. [in] SECURITY\_CONTAINER\* pSecurityContainer,
6. [in] SPLCLIENT\_CONTAINER\* pClientInfo,
7. [out] PRINTER\_HANDLE\* pHandle
8. );

**pName:** A parameter specified in [Print Server Name Parameters (section 3.1.4.1.4)](#Section_acb355a4913b4abb8cb5415aea160015).

**pPrinterContainer:** A parameter specified in [PRINTER\_CONTAINER Parameters (section 3.1.4.1.8.6)](#Section_07b78bd9247c4c58981e9a77538238ae). The **Level** member of the [PRINTER\_CONTAINER](#Section_ad7d4d4e0feb45e093edb3b2b9a313d1) MUST be 0x00000001 or 0x00000002.

**pDevModeContainer:** A parameter specified in [DEVMODE\_CONTAINER Parameters (section 3.1.4.1.8.1)](#Section_bf31abc0da864ac68ebc8be665e11503).

**pSecurityContainer:** A parameter specified in [SECURITY\_CONTAINER Parameters (section 3.1.4.1.8.7)](#Section_6c6051441c004365a6c75499676dcb52).

**pClientInfo:** A parameter specified in [SPLCLIENT\_CONTAINER Parameters (section 3.1.4.1.8.8)](#Section_e6a9e216bc2c43209f67ac3866fb2bc5).

**pHandle:** A pointer to a variable that MUST receive the printer remote procedure call (RPC) context handle to the printer object added. RPC context handles are specified in [[C706]](http://go.microsoft.com/fwlink/?LinkId=89824).

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate parameters as follows:

* Perform the validation steps specified in Print Server Name Parameters.
* Perform the validation steps specified in PRINTER\_CONTAINER Parameters.
* Perform the validation steps specified in DEVMODE\_CONTAINER Parameters.
* Perform the validation steps specified in SECURITY\_CONTAINER Parameters.
* Perform the validation steps specified in SPLCLIENT\_CONTAINER Parameters.
* If the value of the **Level** member of the PRINTER\_CONTAINER that is pointed to by the *pPrinterContainer* parameter is 0x00000002:
  + Verify that the printer driver specified in the [PRINTER\_INFO](#Section_125fb29b3c5141fca333b1e0e8794a91) that is pointed to by the **pointer** member of the PRINTER\_CONTAINER pointed to by the *pPrinterContainer* parameter already exists in the system; if that verification fails, return ERROR\_UNKNOWN\_PRINTER\_DRIVER, as specified in [MS-ERREF].
  + Verify that the port specified in the PRINTER\_INFO that is pointed to by the **pointer** member of the PRINTER\_CONTAINER pointed to by the *pPrinterContainer* parameter already exists in the system; if that verification fails, return ERROR\_UNKNOWN\_PORT, as specified in [MS-ERREF].
  + Verify that the print processor specified in the PRINTER\_INFO that is pointed to by the **pointer** member of the PRINTER\_CONTAINER pointed to by the *pPrinterContainer* parameter already exists in the system; if that verification fails, return ERROR\_UNKNOWN\_PRINTPROCESSOR, as specified in [MS-ERREF].
  + Verify that the printer with the name specified in the PRINTER\_INFO that is pointed to by the **pointer** member of the PRINTER\_CONTAINER pointed to by the *pPrinterContainer* parameter does not already exist in the system; if that verification fails, return ERROR\_PRINTER\_ALREADY\_EXISTS, as specified in [MS-ERREF].
* Additional validation MAY[<335>](#Appendix_A_335) be performed.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Perform PRINTER\_CONTAINER parameter processing steps as specified in PRINTER\_CONTAINER Parameters, section 3.1.4.1.8.6.
* If the value of the **Level** member of the PRINTER\_CONTAINER that is pointed to by the *pPrinterContainer* parameter is 0x00000001, and if the server does not maintain a "List of Known Printers", the server MUST return ERROR\_PRINTER\_ALREADY\_EXISTS, as specified in [MS-ERREF]. Otherwise, the server MUST continue to process the message and compose a response to the client as follows:
  + If the [PRINTER\_ATTRIBUTE\_SHARED](#Section_1625e9d929e448f4b83d3bd0fdaea787) bit is set in the **Flags** member of the PRINTER\_INFO structure pointed to by the **pPrinterInfo1** member of the PRINTER\_CONTAINER that is pointed to by the *pPrinterContainer* parameter, add the printer to the "List of Known Printers" as specified in [Abstract Data Model (section 3.1.1)](#Section_1bd61448db4b4284a4b18346732402d4).[<336>](#Appendix_A_336)
  + If PRINTER\_ATTRIBUTE\_SHARED is not set in the **Flags** member of the PRINTER\_INFO structure pointed to by the **pPrinterInfo1** member of the PRINTER\_CONTAINER that is pointed to by the *pPrinterContainer* parameter, remove the printer from the "List of Known Printers".
  + Store NULL in the output parameter that is pointed to by *pHandle*.
  + Return ERROR\_PRINTER\_ALREADY\_EXISTS, as specified in [MS-ERREF].

**Note:** An error return code is required by remote procedure call (RPC) because NULL was stored to the output parameter pointed to by *pHandle*.

* If the value of the **Level** member of the PRINTER\_CONTAINER that is pointed to by the *pPrinterContainer* parameter is 0x00000002:
  + Instead of failing the validation steps for missing printer driver, port, and print processor, the server MAY create the required printer driver, port, and print processor if they do not exist in the system.[<337>](#Appendix_A_337)
  + Create the printer object and assign to it the security descriptor from the [SECURITY\_CONTAINER](#Section_6cf2b4ff1b2b453c87f985243823cea8) that is pointed to by the *pSecurityContainer* parameter.
  + Add the printer to the "List of Printers" specified in section 3.1.1).
  + Create a session that includes:
    - An **RPC handle**.
    - A snapshot of the printer data specific to this instance of the printer invocation.
    - The data from [DEVMODE](#Section_ccc2a501794e4d2bb312f69c75131c2e) that is contained in the [DEVMODE\_CONTAINER](#Section_c2b14fe764794460a7ba633a845dd61a) pointed to by the *pDevModeContainer* parameter if it is not NULL.
    - The data from the [SPLCLIENT\_CONTAINER](#Section_cca15c7257d342fc91e749442b623c11) that is pointed to by the *pClientInfo* parameter if it is not NULL.
  + Store the **RPC handle** for the session in the output parameter *pHandle*. The handle returned from this method MUST be granted [PRINTER\_ALL\_ACCESS](#Section_945c4df5496949c5b1ce59c1b3f35024) permission.
  + Increment the printer's reference count to prevent deletion.
  + If any clients are registered for notifications of the server object change, a notification MUST be broadcast to them.
* Return the status of the operation.

##### RpcEnumPrinterData (Opnum 72)

RpcEnumPrinterData enumerates configuration data for a specified printer.[<338>](#Appendix_A_338)

1. DWORD RpcEnumPrinterData(
2. [in] PRINTER\_HANDLE hPrinter,
3. [in] DWORD dwIndex,
4. [out, size\_is(cbValueName/sizeof(wchar\_t))]
5. wchar\_t\* pValueName,
6. [in] DWORD cbValueName,
7. [out] DWORD\* pcbValueName,
8. [out] DWORD\* pType,
9. [out, size\_is(cbData)] BYTE\* pData,
10. [in] DWORD cbData,
11. [out] DWORD\* pcbData
12. );

**hPrinter:** A handle to a printer object that was opened by [RpcAddPrinter (section 3.1.4.2.3)](#Section_bd0bb8d526fd470581f5614d32b9f38f), [RpcAddPrinterEx (section 3.1.4.2.15)](#Section_3af45717308f4b4f8428430d12856110), [RpcOpenPrinter (section 3.1.4.2.2)](#Section_989357e2446e4872bb381dce21e1313f), or [RpcOpenPrinterEx (section 3.1.4.2.14)](#Section_0e81ce1872b146c38584a205393b04ff).

**dwIndex:** The index of the configuration data value to retrieve. The value MUST be greater than or equal to zero and less than the total number of configuration data values for the printer. The client SHOULD use **RpcEnumPrinterKeys** to determine the total number of configuration data values for the printer.

**pValueName:** A pointer to a buffer that receives a string specifying the name of the configuration data value. For rules governing value names, see section [2.2.4.18](#Section_cd9850cf1e6c48759b65e159cf2d83d6).

This parameter can be NULL if *cbValueName* equals zero.

**cbValueName:** The size, in bytes, of the buffer that is pointed to by the *pValueName* parameter.

**pcbValueName:** A pointer to a variable that receives the number of bytes stored in the buffer that is pointed to by the *pValueName* parameter.

**pType:** A parameter specified in [Dynamically Typed Query Parameters (section 3.1.4.1.2)](#Section_183564fd8a694fa2a01e51cebc7fb475).

**pData:** A pointer to BUFFER as specified in Dynamically Typed Query Parameters. This parameter can be NULL if *cbData* equals zero.

**cbData:** A parameter specified in Dynamically Typed Query Parameters.

**pcbData:** A parameter specified in Dynamically Typed Query Parameters.

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate parameters as follows:

* Perform the validation steps specified in [PRINTER\_HANDLE Parameters (section 3.1.4.1.11)](#Section_10bfa8c2d4564ad0912de674766a36c8). This method SHOULD assume that the handle to the printer object can be used without further access checks.
* Verify that the value of the *cbValueName* parameter is not smaller than the number of bytes required to hold the string that specifies the name of the value. If that verification fails, the server MUST update the variable that is pointed to by the *pcbValueName* parameter with the number of bytes required and return ERROR\_MORE\_DATA, as specified in [MS-ERREF].
* Verify that the value of the *dwIndex* parameter is greater than or equal to zero, and smaller than the total number of values for the printer. If *dwIndex* is greater than or equal to the number of values available for the printer, the server MUST return ERROR\_NO\_MORE\_ITEMS, as specified in [MS-ERREF]
* Perform the validation steps specified in Dynamically Typed Query Parameters.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Store the name of the printer property in the string buffer that is pointed to by the *pValueName* parameter and store the length of the name stored in the variable that is pointed to by the *pcbValueName* parameter.
* Using the data identified by *pValueName*, [<339>](#Appendix_A_339) perform the processing and response steps specified in Dynamically Typed Query Parameters.
* Return the status of the operation.

##### RpcDeletePrinterData (Opnum 73)

RpcDeletePrinterData deletes specified configuration data for a printer.[<340>](#Appendix_A_340)

1. DWORD RpcDeletePrinterData(
2. [in] PRINTER\_HANDLE hPrinter,
3. [in, string] wchar\_t\* pValueName
4. );

**hPrinter:** A handle to a printer object that was opened by [RpcAddPrinter (section 3.1.4.2.3)](#Section_bd0bb8d526fd470581f5614d32b9f38f), [RpcAddPrinterEx (section 3.1.4.2.15)](#Section_3af45717308f4b4f8428430d12856110), [RpcOpenPrinter (section 3.1.4.2.2)](#Section_989357e2446e4872bb381dce21e1313f), or [RpcOpenPrinterEx (section 3.1.4.2.14)](#Section_0e81ce1872b146c38584a205393b04ff).

**pValueName:** A pointer to a string that identifies the configuration data to delete. For rules governing value names, see section [2.2.4.18](#Section_cd9850cf1e6c48759b65e159cf2d83d6).

The value name is an arbitrary string defined by the printer driver associated with the printer object. The value name "ChangeID"[<341>](#Appendix_A_341) is reserved by the protocol and MAY NOT be used in a call to RpcDeletePrinterData.

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server validates parameters as follows:

* The server MUST perform the validation steps that are specified in [PRINTER\_HANDLE Parameters (section 3.1.4.1.11)](#Section_10bfa8c2d4564ad0912de674766a36c8).
* The server MAY verify that the value name is not "ChangeID" and that it complies with the rules specified in section 2.2.4.18.
* Additional validation SHOULD[<342>](#Appendix_A_342) be performed.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Delete the printer data specified by *pValueName*.[<343>](#Appendix_A_343)
* Return the status of the operation.

##### RpcSetPrinterDataEx (Opnum 77)

RpcSetPrinterDataEx sets the configuration data for a printer or print server.[<344>](#Appendix_A_344) This method is similar to [RpcSetPrinterData (section 3.1.4.2.8)](#Section_e0d3e41e097c459da7fbacaf1343ea27) but additionally allows the caller to specify the registry key under which to store the data.

1. DWORD RpcSetPrinterDataEx(
2. [in] PRINTER\_HANDLE hPrinter,
3. [in, string] const wchar\_t\* pKeyName,
4. [in, string] const wchar\_t\* pValueName,
5. [in] DWORD Type,
6. [in, size\_is(cbData)] BYTE\* pData,
7. [in] DWORD cbData
8. );

**hPrinter:** A handle to a printer object or server object that was opened by [RpcAddPrinter (section 3.1.4.2.3)](#Section_bd0bb8d526fd470581f5614d32b9f38f), [RpcAddPrinterEx (section 3.1.4.2.15)](#Section_3af45717308f4b4f8428430d12856110), [RpcOpenPrinter (section 3.1.4.2.2)](#Section_989357e2446e4872bb381dce21e1313f), or [RpcOpenPrinterEx (section 3.1.4.2.14)](#Section_0e81ce1872b146c38584a205393b04ff).

**pKeyName:** A pointer to a string that specifies the key under which the value is to be set. A key name is an arbitrary string defined by the printer driver associated with the printer object. For rules governing key names, see section [2.2.4.7](#Section_c35aab9057fc4099b3886ec323ded6ec).

If *hPrinter* is a handle to a server object, the key name can be NULL, and the server MUST ignore this parameter.

**pValueName:** A pointer to a string that identifies the data to set. For rules governing value names, see section [2.2.4.18](#Section_cd9850cf1e6c48759b65e159cf2d83d6).

For print servers, a value name MUST be one of the predefined strings listed in [Server Handle Key Values (section 2.2.3.10)](#Section_1393a76f585149a5bf147188ca6763b2).

For printer objects, a value name is an arbitrary string defined by the printer driver associated with the printer object. The value name "ChangeID"[<345>](#Appendix_A_345) is reserved by the protocol and MAY NOT be used in a call to RpcSetPrinterDataEx.

**Type:** A code that indicates the type of data that is pointed to by the *pData* parameter. The value SHOULD be one of the possible type codes defined by type values in section [2.2.3.9](#Section_25cce7007fcf4bb6a2f30f6d08430a55). For rules governing registry type values, see section [2.2.4.15](#Section_f3ded7900d0a462b9bd173051033cb8a).

**pData:** A pointer to an array of bytes that contain the printer configuration data. The type of the data in the buffer is specified by the *Type* parameter.

**cbData:** The size, in bytes, of the *pData* array. This value SHOULD NOT be zero.

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server validates parameters as follows:

* The server MUST perform the validation steps that are specified in [PRINTER\_HANDLE Parameters (section 3.1.4.1.11)](#Section_10bfa8c2d4564ad0912de674766a36c8).
* If the *hPrinter* parameter is a handle to a printer object, the server MUST verify that the *pKeyName* parameter points to a string that complies with the rules for key names specified in section 2.2.4.7.
* For server objects, the server MUST verify that the *pValueName* parameter points to a string that is one of the predefined value names listed in Server Handle Key Values (section 2.2.3.10) with the "read-write" column selected. If this verification fails, return ERROR\_INVALID\_PARAMETER.
* For printer objects, the server MAY verify that the *pValueName* parameter points to a string that complies with the rules specified in section 2.2.4.18.
* Additional validation SHOULD[<346>](#Appendix_A_346) be performed.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* If the *hPrinter* parameter is a handle to a printer object, store the data that is provided by *pData* with the type that is supplied by *Type* in the printer data value that is identified by *pKeyName* and *pValueName*.
* If *hPrinter* is a handle to a server object, store the data that is provided by *pData* with the type that is supplied by *Type* in the server data value that is identified by the *pValueName* parameter.
* Return the status of the operation.

##### RpcGetPrinterDataEx (Opnum 78)

RpcGetPrinterDataEx retrieves configuration data for the specified printer or print server.[<347>](#Appendix_A_347) This method is similar to [RpcGetPrinterData (section 3.1.4.2.7)](#Section_9c04a0f2adc443938df47c36a14183f5), but it also allows the caller to specify the registry key from which to retrieve the data.

1. DWORD RpcGetPrinterDataEx(
2. [in] PRINTER\_HANDLE hPrinter,
3. [in, string] const wchar\_t\* pKeyName,
4. [in, string] const wchar\_t\* pValueName,
5. [out] DWORD\* pType,
6. [out, size\_is(nSize)] BYTE\* pData,
7. [in] DWORD nSize,
8. [out] DWORD\* pcbNeeded
9. );

**hPrinter:** A handle to a printer object or server object that was opened by [RpcAddPrinter (section 3.1.4.2.3)](#Section_bd0bb8d526fd470581f5614d32b9f38f), [RpcAddPrinterEx (section 3.1.4.2.15)](#Section_3af45717308f4b4f8428430d12856110), [RpcOpenPrinter (section 3.1.4.2.2)](#Section_989357e2446e4872bb381dce21e1313f), or [RpcOpenPrinterEx (section 3.1.4.2.14)](#Section_0e81ce1872b146c38584a205393b04ff).

**pKeyName:** A pointer to a string that specifies the key under which the value is to be queried. A key name is an arbitrary string defined by the printer driver associated with the printer object. For rules governing key names, see section [2.2.4.7](#Section_c35aab9057fc4099b3886ec323ded6ec).

If *hPrinter* is a handle to a server object, the key name can be NULL.

**pValueName:** A pointer to a string that identifies the data to get. For rules governing value names, see section [2.2.4.18](#Section_cd9850cf1e6c48759b65e159cf2d83d6).

For print servers, the value name MUST be one of the predefined strings listed in [Server Handle Key Values (section 2.2.3.10)](#Section_1393a76f585149a5bf147188ca6763b2).

For printer objects, the value name MAY be one of the predefined strings listed in [Printer Data Values (section 2.2.3.11)](#Section_4a81aa1c469e49418c4db8c4093b7fc2). If the value name is not one of the predefined strings, it is an arbitrary string defined by the printer driver associated with the printer object. See RpcGetPrinterData for further details on the interpretation of this value.

**pType:** A parameter specified in [Dynamically Typed Query Parameters (section 3.1.4.1.2)](#Section_183564fd8a694fa2a01e51cebc7fb475).

**pData:** A pointer to BUFFER, as specified in Dynamically Typed Query Parameters. This parameter can be NULL if *nSize* equals zero

**nSize:** A parameter specified in Dynamically Typed Query Parameters.

**pcbNeeded:** A parameter specified in Dynamically Typed Query Parameters.

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server validates parameters as follows:

* The server MUST perform the validation steps that are specified in [PRINTER\_HANDLE Parameters (section 3.1.4.1.11)](#Section_10bfa8c2d4564ad0912de674766a36c8).
* If *hPrinter* is a handle to a printer object, the server MUST verify that the *pKeyName* parameter points to a string that complies with the rules for key names specified in section 2.2.4.7.
* For server objects, the server MUST verify that the *pValueName* parameter points to a string that is one of the predefined value names listed in Server Handle Key Values (section 2.2.3.10). If this verification fails, return ERROR\_INVALID\_PARAMETER.
* For printer objects, the server MUST verify that, if the *pValueName* parameter points to a string that is one of the predefined value names listed in Printer Data Values (section 2.2.3.11), the print server supports retrieving the value for this printer according to the rules in section 2.2.3.11. If this verification fails, return ERROR\_NOT\_SUPPORTED.
* For printer objects, the server MAY verify that the *pValueName* parameter points to a string that complies with the rules specified in section 2.2.4.18.
* The server MUST perform the validation steps that are specified in section 3.1.4.1.2.
* The server MUST NOT perform access checks on the *hPrinter* object.
* If *hPrinter* is a handle to a printer object with a printer driver version (see *cVersion* in section [2.2.1.3.1](#Section_4464eaf0f34f40d5b970736437a21913)) of 0x00000004, the server SHOULD verify that the size in bytes of the data to be returned to the client (the value to be returned via the *pcbNeeded* parameter) does not exceed 0xFFFFFFFF bytes. If this verification fails, the server SHOULD return ERROR\_NOT\_ENOUGH\_MEMORY.[<348>](#Appendix_A_348)
* If *hPrinter* is a handle to a printer object with a printer driver version of 0x00000004, the server SHOULD verify that that the *pValueName* parameter points to a string value that is supported for the printer object. If this verification fails, the server SHOULD return ERROR\_NOT\_SUPPORTED. [<349>](#Appendix_A_349)

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* If the *hPrinter* parameter is a handle to a printer object, with the data identified by *pKeyName* and *pValueName*, perform the processing and response steps that are specified in section 3.1.4.1.2.
* If *hPrinter* is a handle to a server object, with the data that is identified by *pValueName*, perform the processing and response steps that are specified in section 3.1.4.1.2.
* Return the status of the operation.

##### RpcEnumPrinterDataEx (Opnum 79)

RpcEnumPrinterDataEx enumerates all value names and data for a specified printer and key.[<350>](#Appendix_A_350) This method is similar to [RpcEnumPrinterData](#Section_51bfa051e7e44fc4b79eea9d4b08cb84) (see section 3.1.4.2.16) but also allows the caller to specify the registry key from which to enumerate the data, and allows retrieving several values in a single call.

1. DWORD RpcEnumPrinterDataEx(
2. [in] PRINTER\_HANDLE hPrinter,
3. [in, string] const wchar\_t\* pKeyName,
4. [out, size\_is(cbEnumValues)] BYTE\* pEnumValues,
5. [in] DWORD cbEnumValues,
6. [out] DWORD\* pcbEnumValues,
7. [out] DWORD\* pnEnumValues
8. );

**hPrinter:** A handle to a printer object that was opened by [RpcAddPrinter (section 3.1.4.2.3)](#Section_bd0bb8d526fd470581f5614d32b9f38f), [RpcAddPrinterEx (section 3.1.4.2.15)](#Section_3af45717308f4b4f8428430d12856110), [RpcOpenPrinter (section 3.1.4.2.2)](#Section_989357e2446e4872bb381dce21e1313f), or [RpcOpenPrinterEx (section 3.1.4.2.14)](#Section_0e81ce1872b146c38584a205393b04ff).

**pKeyName:** A pointer to a string that specifies the key containing the values to enumerate. A key name is an arbitrary string defined by the printer driver associated with the printer object. For rules governing key names, see section [2.2.4.7](#Section_c35aab9057fc4099b3886ec323ded6ec).

**pEnumValues:** A pointer to [BUFFER](#Section_6f49353314324438bd5940f97a02e688) as specified in PRINTER\_ENUM\_VALUES Structures Query Parameters (section 3.1.4.1.10).

This parameter can be NULL if *cbEnumValues* equals zero.

**cbEnumValues:** A parameter specified in PRINTER\_ENUM\_VALUES Structures Query Parameters.

**pcbEnumValues:** A parameter specified in PRINTER\_ENUM\_VALUES Structures Query Parameters.

**pnEnumValues:** A parameter specified in PRINTER\_ENUM\_VALUES Structures Query Parameters.

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate parameters as follows:

* Perform the validation steps specified in [PRINTER\_HANDLE Parameters (section 3.1.4.1.11)](#Section_10bfa8c2d4564ad0912de674766a36c8). This method SHOULD assume that the handle to the printer object can be used without further access checks.
* Verify that the *pKeyName* parameter points to a string that complies with the rules for key names specified in section 2.2.4.7. If this verification fails, return ERROR\_INVALID\_PARAMETER.
* Perform the validation steps specified in PRINTER\_ENUM\_VALUES Structures Query Parameters.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Enumerate all the values referenced by the specified printer data key.
* Using the enumerated objects, perform the processing and response steps specified in PRINTER\_ENUM\_VALUES Structures Query Parameters.
* Return the status of the operation.

##### RpcEnumPrinterKey (Opnum 80)

RpcEnumPrinterKey enumerates the subkeys of a specified key for a specified printer.[<351>](#Appendix_A_351)

1. DWORD RpcEnumPrinterKey(
2. [in] PRINTER\_HANDLE hPrinter,
3. [in, string] const wchar\_t\* pKeyName,
4. [out, size\_is(cbSubkey/sizeof(wchar\_t))]
5. wchar\_t\* pSubkey,
6. [in] DWORD cbSubkey,
7. [out] DWORD\* pcbSubkey
8. );

**hPrinter:** A handle to a printer object that was opened by [RpcAddPrinter (section 3.1.4.2.3)](#Section_bd0bb8d526fd470581f5614d32b9f38f), [RpcAddPrinterEx (section 3.1.4.2.15)](#Section_3af45717308f4b4f8428430d12856110), [RpcOpenPrinter (section 3.1.4.2.2)](#Section_989357e2446e4872bb381dce21e1313f), or [RpcOpenPrinterEx (section 3.1.4.2.14)](#Section_0e81ce1872b146c38584a205393b04ff).

**pKeyName:** A pointer to a string that specifies the key containing the subkeys to enumerate. A key name is an arbitrary string defined by the printer driver associated with the printer object. For rules governing key names, see section [2.2.4.7](#Section_c35aab9057fc4099b3886ec323ded6ec).

**pSubkey:** A pointer to [BUFFER](#Section_6bae59aff92341b9b39513872aaf2ce0) as specified in String Query Parameters (section 3.1.4.1.7).

This parameter can be NULL if *cbSubkey* equals zero.

**cbSubkey:** A value that is synonymous with the *cbBuf* parameter specified in String Query Parameters.

**pcbSubkey:** A value that is synonymous with the *pcbNeeded* parameter specified in String Query Parameters.

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate parameters as follows:

* Perform the validation steps specified in [PRINTER\_HANDLE Parameters (section 3.1.4.1.11)](#Section_10bfa8c2d4564ad0912de674766a36c8). This method SHOULD assume that the handle to the printer object can be used without further access checks.
* Perform the validation steps specified in String Query Parameters, substituting ERROR\_MORE\_DATA (as specified in [MS-ERREF]) for ERROR\_INSUFFICIENT\_BUFFER.
* Verify that the key specified in the string that is pointed to by the *pKeyName* parameter exists on the server. If it does not exist, the server MUST return ERROR\_FILE\_NOT\_FOUND, as specified in [MS-ERREF].

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Enumerate the key names that have the key specified in the string that is pointed to by the *pKeyName* parameter as an immediate parent. This method returns zero or more key names by storing them as multisz values in the *BUFFER* pointed to by *pSubkey*.
* Using the enumerated objects, perform the processing and response steps that are specified in String Query Parameters.
* Return the status of the operation.

##### RpcDeletePrinterDataEx (Opnum 81)

RpcDeletePrinterDataEx deletes a specified value from a printer's configuration data, which consists of a set of named and typed values stored in a hierarchy of registry keys.[<352>](#Appendix_A_352)

1. DWORD RpcDeletePrinterDataEx(
2. [in] PRINTER\_HANDLE hPrinter,
3. [in, string] const wchar\_t\* pKeyName,
4. [in, string] const wchar\_t\* pValueName
5. );

**hPrinter:** A handle to a printer object that was opened by [RpcAddPrinter (section 3.1.4.2.3)](#Section_bd0bb8d526fd470581f5614d32b9f38f), [RpcAddPrinterEx (section 3.1.4.2.15)](#Section_3af45717308f4b4f8428430d12856110), [RpcOpenPrinter (section 3.1.4.2.2)](#Section_989357e2446e4872bb381dce21e1313f), or [RpcOpenPrinterEx (section 3.1.4.2.14)](#Section_0e81ce1872b146c38584a205393b04ff).

**pKeyName:** A pointer to a string that specifies the key containing the value to delete. A key name is an arbitrary string defined by the printer driver associated with the printer object. For rules governing key names, see section [2.2.4.7](#Section_c35aab9057fc4099b3886ec323ded6ec).

**pValueName:** A pointer to a string that identifies the configuration data to delete. For rules governing value names, see section [2.2.4.18](#Section_cd9850cf1e6c48759b65e159cf2d83d6).

The value name is an arbitrary string defined by the printer driver associated with the printer object. The value name "ChangeID"[<353>](#Appendix_A_353) is reserved by the protocol and SHOULD NOT be used in a call to RpcDeletePrinterDataEx.

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server validates parameters as follows:

* The server MUST perform the validation steps that are specified in [PRINTER\_HANDLE Parameters (section 3.1.4.1.11)](#Section_10bfa8c2d4564ad0912de674766a36c8),.
* The server MUST verify that the *pKeyName* parameter points to a string that complies with the rules for key names specified in section 2.2.4.7. If this verification fails, return ERROR\_INVALID\_PARAMETER.
* The server MAY verify that the *pValueName* parameter points to a string that complies with the rules specified in section 2.2.4.18.
* Additional validation SHOULD[<354>](#Appendix_A_354) be performed.

If parameter validation fails, the server MUST fail the operation immediately, returning a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Delete the printer data value indicated by the *pKeyName* and *pValueName* parameters.
* Return the status of the operation.

##### RpcDeletePrinterKey (Opnum 82)

RpcDeletePrinterKey deletes a specified key and all of its subkeys for a specified printer.[<355>](#Appendix_A_355)

1. DWORD RpcDeletePrinterKey(
2. [in] PRINTER\_HANDLE hPrinter,
3. [in, string] const wchar\_t\* pKeyName
4. );

**hPrinter:** A handle to a printer object that was opened by [RpcAddPrinter (section 3.1.4.2.3)](#Section_bd0bb8d526fd470581f5614d32b9f38f), [RpcAddPrinterEx (section 3.1.4.2.15)](#Section_3af45717308f4b4f8428430d12856110), [RpcOpenPrinter (section 3.1.4.2.2)](#Section_989357e2446e4872bb381dce21e1313f), or [RpcOpenPrinterEx (section 3.1.4.2.14)](#Section_0e81ce1872b146c38584a205393b04ff) methods.

**pKeyName:** A pointer to a string that specifies the key to delete. A key name is an arbitrary string defined by the printer driver associated with the printer object. For rules governing key names, see section [2.2.4.7](#Section_c35aab9057fc4099b3886ec323ded6ec).

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate parameters as follows:

* Perform the validation steps that are specified in [PRINTER\_HANDLE Parameters (section 3.1.4.1.11)](#Section_10bfa8c2d4564ad0912de674766a36c8).
* Verify that the *pKeyName* parameter points to a string that complies with the rules for key names specified in section 2.2.4.7. If this verification fails, return ERROR\_INVALID\_PARAMETER.
* Additional validation MAY[<356>](#Appendix_A_356) be performed.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Delete the printer data key indicated by the *pKeyName* parameter and all the subkeys of that key.
* Return the status of the operation.

##### RpcAddPerMachineConnection (Opnum 85)

RpcAddPerMachineConnection adds a remote printer name to the list of supported printer connections for every user who locally logs onto the computer running the print server.[<357>](#Appendix_A_357)

This method is used for remote administration of client computers running the print system.

1. DWORD RpcAddPerMachineConnection(
2. [in, string, unique] STRING\_HANDLE pServer,
3. [in, string] const wchar\_t\* pPrinterName,
4. [in, string] const wchar\_t\* pPrintServer,
5. [in, string] const wchar\_t\* pProvider
6. );

**pServer:** A value that MUST adhere to the specification in [Print Server Name Parameters (section 3.1.4.1.4)](#Section_acb355a4913b4abb8cb5415aea160015).

**pPrinterName:** A value that MUST adhere to the specification in [Printer Name Parameters (section 3.1.4.1.5)](#Section_3f2d41eb30e147e193bb534ea50cf8eb). This parameter MUST specify a printer connection, and it MUST be of the form:

1. SERVER\_NAME LOCAL\_PRINTER\_NAME [with a non-empty SERVER\_NAME.]

**pPrintServer:** A pointer to a string that specifies the name of the print server that is hosting the printer to which the connection is established. For rules governing server names, see section [2.2.4.16](#Section_9b3f813570224b72accbaefcc360c83b).

**pProvider:** A pointer to a string that specifies the name of a print provider. If the string is the empty string "", an implementation-specific default print provider name MUST be used.[<358>](#Appendix_A_358) For rules governing print provider names, see section [2.2.4.12](#Section_ffe40469bbc04cc2b7c6bebd036a928d).

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate parameters as follows:

* Perform the validation steps that are specified in Printer Name Parameters. Perform validations only of the correctness of the syntax of the name; do not perform existence checks for the actual printer object. The existence checks are deferred until the actual creation of the printer object upon user login, and no status feedback is possible. The administration client, therefore, is expected to pass only existing names.
* Perform the validation steps that are specified in Print Server Name Parameters. Perform validations only of the correctness of the syntax of the name; do not perform existence checks for the actual server object. The existence checks are deferred until the actual creation of the printer object upon user login, and no status feedback is possible. The administration client, therefore, is expected to pass only existing names.
* Verify that a per-machine printer connection with the same name does not already exist; and if that verification fails, return ERROR\_PRINTER\_ALREADY\_EXISTS, as specified in [MS-ERREF].
* Additional validation MAY[<359>](#Appendix_A_359) be performed.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Add the printer to the **list of per-machine connections** on the print server.
* Defer creation of the actual printer object for each user until the user logs on, and then create a printer object for the printer connection to the printer specified in *pPrinterName*. The created printer object cannot be shared, and can be used only when locally logged onto the computer.
* Return the status of the operation.

##### RpcDeletePerMachineConnection (Opnum 86)

RpcDeletePerMachineConnection deletes information about a printer connection.[<360>](#Appendix_A_360)

This method is used for remote administration of client computers running the print system.

1. DWORD RpcDeletePerMachineConnection(
2. [in, string, unique] STRING\_HANDLE pServer,
3. [in, string] const wchar\_t\* pPrinterName
4. );

**pServer:** A parameter specified in [Print Server Name Parameters (section 3.1.4.1.4)](#Section_acb355a4913b4abb8cb5415aea160015).

**pPrinterName:** A parameter specified in [Printer Name Parameters (section 3.1.4.1.5)](#Section_3f2d41eb30e147e193bb534ea50cf8eb). This parameter MUST specify a printer connection.

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate parameters as follows:

* Perform the validation steps specified in Print Server Name Parameters.
* Perform the validation steps specified in Printer Name Parameters.
* Verify that the per-machine connection exists, and if that fails, return ERROR\_INVALID\_PRINTER\_NAME.
* Additional validation MAY[<361>](#Appendix_A_361) be performed.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Delete the per-machine printer connection from the **list of per-machine connections** that is identified by the string that is pointed to by the *pPrinterName* parameter. Defer deletion of the actual printer object for each user until the user logs on, and then delete the printer object for the printer connection to the printer specified in *pPrinterName*.
* Return the status of the operation.

##### RpcEnumPerMachineConnections (Opnum 87)

Enumerates each of the connections and copies [PRINTER\_INFO\_4 (section 2.2.1.10.5)](#Section_ac391864cd9945099ff3ea0366dd9b2b) structures for all the per-machine connections into the buffer *pPrinterEnum*.[<362>](#Appendix_A_362)

This method is used for remote administration of client computers running the print system.

1. DWORD RpcEnumPerMachineConnections(
2. [in, string, unique] STRING\_HANDLE pServer,
3. [in, out, unique, size\_is(cbBuf), disable\_consistency\_check]
4. BYTE\* pPrinterEnum,
5. [in] DWORD cbBuf,
6. [out] DWORD\* pcbNeeded,
7. [out] DWORD\* pcReturned
8. );

**pServer:** A parameter specified in [Print Server Name Parameters (section 3.1.4.1.4)](#Section_acb355a4913b4abb8cb5415aea160015).

**pPrinterEnum:** A pointer to the [BUFFER](#Section_f29e9057df2940d49c51c615c0da62a1), as specified in INFO Structures Query Parameters (section 3.1.4.1.9).

**BUFFER TYPE:** \_PRINTER\_INFO\_4

This parameter can be NULL if *cbBuf* equals zero.

**cbBuf:** A parameter specified in section 3.1.4.1.9.

**pcbNeeded:** A parameter specified in section 3.1.4.1.9.

**pcReturned:** A parameter specified in section 3.1.4.1.9.

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate parameters as follows:

* Perform the validation steps specified in section 3.1.4.1.4.
* Perform the validation steps specified in section 3.1.4.1.9.
* This method SHOULD NOT perform any access checks.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Enumerate all printers in the **list of per-machine connections** on the server that is identified by the *pServer* parameter (regardless of whether the user has logged on and the printer object has actually been created).
* Using the enumerated objects, perform the processing and response steps specified in section 3.1.4.1.9.
* Return the status of the operation.

The server MUST NOT change the list of printer objects representing pushed printers as part of processing this method.

##### RpcSendRecvBidiData (Opnum 97)

RpcSendRecvBidiData sends and receives bidirectional data. This method is used to communicate with port monitors that support such data.[<363>](#Appendix_A_363)

1. DWORD RpcSendRecvBidiData(
2. [in] PRINTER\_HANDLE hPrinter,
3. [in, string, unique] const wchar\_t\* pAction,
4. [in] RPC\_BIDI\_REQUEST\_CONTAINER\* pReqData,
5. [out] RPC\_BIDI\_RESPONSE\_CONTAINER\*\* ppRespData
6. );

**hPrinter:** A handle to a printer object or port object that was opened by [RpcAddPrinter (section 3.1.4.2.3)](#Section_bd0bb8d526fd470581f5614d32b9f38f), [RpcAddPrinterEx (section 3.1.4.2.15)](#Section_3af45717308f4b4f8428430d12856110), [RpcOpenPrinter (section 3.1.4.2.2)](#Section_989357e2446e4872bb381dce21e1313f), or [RpcOpenPrinterEx (section 3.1.4.2.14)](#Section_0e81ce1872b146c38584a205393b04ff).

**pAction:** A pointer to a string that specifies an action to take. The following actions SHOULD be supported.[<364>](#Appendix_A_364)[<365>](#Appendix_A_365) Port monitors MAY support additional, implementation-specific action strings.[<366>](#Appendix_A_366)

| Name/value | Meaning |
| --- | --- |
| BIDI\_ACTION\_ENUM\_SCHEMA  "EnumSchema" | The method MUST enumerate the supported schemas. The *pReqData* parameter MUST be ignored. The method MUST store one or more values that correspond to supported schema entries in the buffer that is pointed to by the *ppRespData* parameter. |
| BIDI\_ACTION\_GET  "Get" | The method MUST return the specific value item requested. The *pReqData* parameter MUST specify a single value entry in the schema. The method MUST store the value of that entry in the buffer that is pointed to by the *ppRespData* parameter. |
| BIDI\_ACTION\_SET  "Set" | The method MUST store the supplied data in a single value item in the schema. The *pReqData* parameter MUST specify a single value entry for the schema and the new value to be stored there. This action MUST NOT change the contents of the buffer that is pointed to by the *ppRespData* parameter. |
| BIDI\_ACTION\_GET\_ALL  "GetAll" | The method MUST return one or more value items that are reachable from the requested schema item. The *pReqData* parameter MUST specify an entry in the schema, which is either a value item or an inner schema entry. The action MUST store one or more value entries, and their associated values, in the buffer that is pointed to by the *ppRespData* parameter. |
| BIDI\_ACTION\_GET\_WITH\_ARGUMENT  "GetWithArgument" | The method MUST return one or more value items that are reachable from the requested schema item. The *pReqData* parameter MUST specify an entry in the schema, which is either a value item or an inner schema entry, and a data value to be used when processing the request. The action MUST store one or more value entries, and their associated values, in the buffer that is pointed to by the *ppRespData* parameter. |

**pReqData:** A pointer to an [RPC\_BIDI\_REQUEST\_CONTAINER (section 2.2.1.2.10)](#Section_4ea856c44c374485868ae7ac8e331e5e) structure that contains the requested binary data.

**ppRespData:** A pointer to a variable that receives a pointer to a [RPC\_BIDI\_RESPONSE\_CONTAINER (section 2.2.1.2.11)](#Section_bdfdec23d5034f86a8486c3386d609d6) structure that contains the response binary data.

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate parameters as follows:

* Perform the validation steps specified in [PRINTER\_HANDLE Parameters (section 3.1.4.1.11)](#Section_10bfa8c2d4564ad0912de674766a36c8). This method SHOULD assume that the handle to the printer or port object can be used without further access checks.
* Verify that the port monitor supports this method, and if that verification fails, return ERROR\_NOT\_SUPPPORTED.
* Verify that the string that is pointed to by the *pAction* parameter specifies a valid command and is supported by the port monitor, and if that verification fails, return any of the following error codes to indicate the request cannot be supported: ERROR\_NOT\_SUPPORTED, ERROR\_INVALID\_PARAMETER.
* Verify that the *pReqData* is a non-NULL pointer, and if that verification fails, return ERROR\_INVALID\_PARAMETER.
* Additional validation MAY be performed[<367>](#Appendix_A_367).

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* If the *hPrinter* parameter is a handle to a printer object, load the executable object of the monitor supporting the port associated with the printer identified by *hPrinter*.
* If the *hPrinter* parameter is a handle to a port object, load the executable object of the monitor supporting the port identified by *hPrinter*.
* Invoke the method in that library that is identified by the value of the *pAction* parameter and pass *pReqData* to that method.
* Copy the data that is sent from the action method in the buffer that is pointed to by the *ppRespData* parameter; the number of response items MUST match the number of request items.
* Return the status of the operation.

#### Job Management Methods

This section specifies methods for discovering, defining, and scheduling print jobs.

| Method | Description |
| --- | --- |
| [RpcSetJob](#Section_f978bd2ce99a43ffba0802e892948dac) | RpcSetJob pauses, resumes, cancels, or restarts a print job. It also sets print job parameters, such as the job priority and the document name.  Opnum 2 |
| [RpcGetJob](#Section_f7090127287b4aa7ba8b2b6b7247dfc6) | RpcGetJob retrieves information about a specified print job.  Opnum 3 |
| [RpcEnumJobs](#Section_0c77f1ff08274f4da06d5a3d802ffc0c) | RpcEnumJobs retrieves information about a specified set of print jobs for a specified printer.  Opnum 4 |
| [RpcAddJob](#Section_7ba3267737294b5f96b5c8a9f99d34a5) | RpcAddJob returns ERROR\_INVALID\_PARAMETER.  Opnum 24 |
| [RpcScheduleJob](#Section_4d6454a7f7d44d308ba18644cb6b2186) | RpcScheduleJob returns ERROR\_SPL\_NO\_ADDJOB.  Opnum 25 |

##### RpcSetJob (Opnum 2)

RpcSetJob pauses, resumes, cancels, or restarts a print job. It also sets print job parameters, such as the job priority and the document name.

1. DWORD RpcSetJob(
2. [in] PRINTER\_HANDLE hPrinter,
3. [in] DWORD JobId,
4. [in, unique] JOB\_CONTAINER\* pJobContainer,
5. [in] DWORD Command
6. );

**hPrinter:** A [PRINTER\_HANDLE (section 2.2.1.1.4)](#Section_9c5d49b8692047b0946687ba910e6071) to a printer object, job object, or server object that was opened by [RpcAddPrinter (section 3.1.4.2.3)](#Section_bd0bb8d526fd470581f5614d32b9f38f), [RpcAddPrinterEx (section 3.1.4.2.15)](#Section_3af45717308f4b4f8428430d12856110), [RpcOpenPrinter (section 3.1.4.2.2)](#Section_989357e2446e4872bb381dce21e1313f), or [RpcOpenPrinterEx (section 3.1.4.2.14)](#Section_0e81ce1872b146c38584a205393b04ff).

**JobId:** The identifier of the print job. This value MUST NOT be zero.

**pJobContainer:** An optional pointer to a [JOB\_CONTAINER (section 2.2.1.2.5)](#Section_0cd5f47b54214b199ed05a29d63f3c2a) that specifies the parameters to set on the job object.

If the value of the *Command* parameter is zero, this pointer MUST be specified.

**Command:** A [Job Control Value (section 2.2.4.6)](#Section_c22b303eb62c4d368336e9b42322f125) that specifies an action. This value MUST be one of the following:

| Job control action name/value | Meaning |
| --- | --- |
| 0x00000000 | Perform no additional action. |
| JOB\_CONTROL\_PAUSE  0x00000001 | Pause the print job. |
| JOB\_CONTROL\_RESUME  0x00000002 | Resume a paused print job. |
| JOB\_CONTROL\_CANCEL  0x00000003 | Delete a print job.[<368>](#Appendix_A_368) |
| JOB\_CONTROL\_RESTART  0x00000004 | Restart a print job. |
| JOB\_CONTROL\_DELETE  0x00000005 | Delete a print job.[<369>](#Appendix_A_369) |
| JOB\_CONTROL\_SENT\_TO\_PRINTER  0x00000006 | Used by port monitors to signal that a print job has been sent to the printer. This value SHOULD NOT be used remotely. |
| JOB\_CONTROL\_LAST\_PAGE\_EJECTED  0x00000007 | Used by language monitors to signal that the last page of a print job has been ejected from the printer. This value SHOULD NOT be used remotely. |
| JOB\_CONTROL\_RETAIN  0x00000008 | Keep the print job in the print queue after it prints. |
| JOB\_CONTROL\_RELEASE  0x00000009 | Release the print job, undoing the effect of a **JOB\_CONTROL\_RETAIN** action. |

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate parameters as follows:

* Perform the validation steps for [PRINTER\_HANDLE parameters (section 3.1.4.1.11)](#Section_10bfa8c2d4564ad0912de674766a36c8).
* Verify that the value of the *JobId* parameter corresponds to a print job in the list of jobs. If this method is called with a job object handle, *JobId* MUST match the job identifier specified in the call to RpcOpenPrinter or RpcOpenPrinterEx.
* If the *pJobContainer* parameter is specified, verify that it points to a valid JOB\_CONTAINER as follows:
  + The **Level** member MUST be a value between 0x00000001 and 0x00000004, inclusive;
  + The **JobInfo** member MUST point to a [JOB\_INFO structure (section 2.2.1.7)](#Section_87ea3e7722fb4914b803ef3b2619195a);
  + If the **Level** value specified in JOB\_CONTAINER is 0x00000003, the **JobId** member of the contained [JOB\_INFO\_3 structure (section 2.2.1.7.3)](#Section_1385aef9d3234e8d9d285f2f011bd39c) MUST match the *JobId* input parameter; otherwise, the *JobId* parameter MUST be ignored.
  + The members of the JOB\_INFO structure MUST comply with the constraints specified for [JOB\_INFO members (section 2.2.1.3.3)](#Section_94d1857ec44449b18fb483eac6696060) with the exception of **JobId** for all levels except 0x00000003, **PrinterName**, **ServerName**, **PrinterDriverName**, **Size**, **Submitted**, **Time**, **TotalPages**, **pDevMode**, and **pSecurityDescriptor**, which MUST be ignored, and **pMachineName**, which SHOULD be ignored.
  + If the **Level** value specified in JOB\_CONTAINER is 0x00000001, 0x00000002, or 0x00000004, perform the validation steps that are specified in [Datatype Name Parameters (section 3.1.4.1.1)](#Section_c31e2daba809474c98640cbb61068903) on the *pDataType* member of the JOB\_INFO structure.
  + If the **Level** value specified in JOB\_CONTAINER is 0x00000002 or 0x00000004, and the print processor specified in the *pPrintProcessor* member of the JOB\_INFO structure does not already exist in the system, the server SHOULD return ERROR\_UNKNOWN\_PRINTPROCESSOR to the client.
* Verify that the *Command* parameter is a supported command.
* Additional validation SHOULD[<370>](#Appendix_A_370) be performed.

If parameter validation fails, the server MUST fail the operation immediately and return to the client ERROR\_INVALID\_PARAMETER or another nonzero error described in the preceding JOB\_CONTAINER validation steps; otherwise, the server MUST process the message and respond to the client as follows:

* Modify the job with a **JobId** that matches the *JobId* input parameter to reflect the required changes based on the value of *Command*:
  + (0x00000000): No additional action will be performed.
  + Pausing the print job (0x00000001): Pause the current job specified by *JobId* and allow any succeeding job to print.
  + Resuming the print job (0x00000002): Resume the job specified by *JobId*.
  + Canceling the print job (0x00000003): Cancel the job specified by *JobId*.
  + Restarting the print job (0x00000004): Reinitialize the internal state of the job specified by *JobId* and re-schedule the job for printing.
  + Deleting the print job (0x00000005): Delete the job specified by *JobId* and any internal structures representing that job.
  + Sent the print job to the printer (0x00000006): MAY be set by port monitors associated with a port to signal the job has been sent completely to the device, but it is not sent over the wire.
  + Last Page Ejected (0x00000007): MAY be set by language monitors associated with a port to signal that the physical printer ejected the last page of the job, but it is not sent over the wire.
  + Retain Job (0x00000008): Keep the print job in the print queue after printing is finished. It MAY then be restarted.
  + Release Job (0x00000009): Release a job previously retained. Remove the print job from the queue if it has finished printing and has not been restarted.
* If any clients have registered for notification of a job object change, those clients SHOULD be sent notifications about the changes that the server performs.
* Modify the print job with a **JobId** that matches the *JobId* input parameter by applying the information in the JOB\_INFO structure that is contained in the JOB\_CONTAINER specified by the *pJobContainer* parameter. The following modifications SHOULD be performed:
  + If the **Level** value specified in JOB\_CONTAINER is 0x00000003, the contained JOB\_INFO\_3 structure specifies the order of print jobs in the job queue. The server SHOULD change the order of jobs so that the job with the identifier specified by the **NextJobId** member of JOB\_INFO\_3 follows immediately after the job with the identifier specified by the **JobId** member. In addition, the server SHOULD link the two jobs together, so they form an atomic entity and are scheduled such that no other job can intervene between them.
  + If the **Level** value specified in JOB\_CONTAINER is 0x00000001, 0x00000002, or 0x00000004, the contained [JOB\_INFO\_1 (section 2.2.1.7.1)](#Section_794c1ba9ba0a4498ba9c2635563460dd), [JOB\_INFO\_2 (section 2.2.1.7.2)](#Section_5a48c6c9fc5f42ffa5a7f111fee58d1a), or [JOB\_INFO\_4 (section 2.2.1.7.4)](#Section_99086f6e3db14034ac6ccd4a5a77f22c) structure, respectively, specifies the new position of the print job in the job queue. The server SHOULD set the position of the job to the value specified by the **Position** member of the JOB\_INFO structure, and the server SHOULD reorder the list of jobs to reflect the new positions.
* Return the status of the operation.

##### RpcGetJob (Opnum 3)

RpcGetJob retrieves information about a specified print job.

1. DWORD RpcGetJob(
2. [in] PRINTER\_HANDLE hPrinter,
3. [in] DWORD JobId,
4. [in] DWORD Level,
5. [in, out, unique, size\_is(cbBuf), disable\_consistency\_check]
6. BYTE\* pJob,
7. [in] DWORD cbBuf,
8. [out] DWORD\* pcbNeeded
9. );

**hPrinter:** A handle to a printer object, job object, or server object that was opened by [RpcAddPrinter (section 3.1.4.2.3)](#Section_bd0bb8d526fd470581f5614d32b9f38f), [RpcAddPrinterEx (section 3.1.4.2.15)](#Section_3af45717308f4b4f8428430d12856110), [RpcOpenPrinter (section 3.1.4.2.2)](#Section_989357e2446e4872bb381dce21e1313f), or [RpcOpenPrinterEx (section 3.1.4.2.14)](#Section_0e81ce1872b146c38584a205393b04ff).

**JobId:** The identifier of the print job. This value MUST NOT be zero.

**Level:** The job information level. This value MUST be 0x00000001, 0x00000002, 0x00000003, or 0x00000004.

**pJob:** A pointer to [BUFFER](#Section_f29e9057df2940d49c51c615c0da62a1) as specified in INFO Structures Query Parameters (section 3.1.4.1.9).

**BUFFER TYPE**: \_JOB\_INFO.

This parameter can be NULL if *cbBuf* equals zero.

**cbBuf:** A parameter specified in section 3.1.4.1.9.

**pcbNeeded:** A parameter specified in section 3.1.4.1.9.

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate parameters as follows:

* Perform the validation steps specified in [PRINTER\_HANDLE Parameters (section 3.1.4.1.11)](#Section_10bfa8c2d4564ad0912de674766a36c8). This method SHOULD assume that the handle to the printer or server object can be used without further access checks.
* Verify that the value of the *JobId* parameter corresponds to a job in the list of jobs. If this verification fails, return ERROR\_INVALID\_PARAMETER.
* Perform the validation steps that are specified in section 3.1.4.1.9.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Using information about the job, perform the processing and response steps specified in section 3.1.4.1.9.
* Return the status of the operation.

##### RpcEnumJobs (Opnum 4)

RpcEnumJobs retrieves information about a specified set of print jobs for a specified printer or port.

1. DWORD RpcEnumJobs(
2. [in] PRINTER\_HANDLE hPrinter,
3. [in] DWORD FirstJob,
4. [in] DWORD NoJobs,
5. [in] DWORD Level,
6. [in, out, unique, size\_is(cbBuf), disable\_consistency\_check]
7. BYTE\* pJob,
8. [in] DWORD cbBuf,
9. [out] DWORD\* pcbNeeded,
10. [out] DWORD\* pcReturned
11. );

**hPrinter:** A handle to a printer object or port object that was opened by [RpcAddPrinter (section 3.1.4.2.3)](#Section_bd0bb8d526fd470581f5614d32b9f38f), [RpcAddPrinterEx (section 3.1.4.2.15)](#Section_3af45717308f4b4f8428430d12856110), [RpcOpenPrinter (section 3.1.4.2.2)](#Section_989357e2446e4872bb381dce21e1313f), or [RpcOpenPrinterEx (section 3.1.4.2.14)](#Section_0e81ce1872b146c38584a205393b04ff).

**FirstJob:** The zero-based position within the print queue of the first print job to enumerate.

**NoJobs:** The total number of print jobs to enumerate.

**Level:** The job information level.

This value MUST be 0x00000001, 0x00000002, 0x00000003, or 0x00000004.

**pJob:** A pointer to the [BUFFER](#Section_f29e9057df2940d49c51c615c0da62a1) structure specified in INFO Structures Query Parameters (section 3.1.4.1.9).

**BUFFER TYPE**: \_JOB\_INFO.

This parameter can be NULL if *cbBuf* equals zero.

**cbBuf:** Specified in INFO Structures Query Parameters (section 3.1.4.1.9) .

**pcbNeeded:** Specified in INFO Structures Query Parameters (section 3.1.4.1.9).

**pcReturned:** Specified in INFO Structures Query Parameters (section 3.1.4.1.9).

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate parameters as follows:

* Perform the validation steps specified in [PRINTER\_HANDLE Parameters (section 3.1.4.1.11)](#Section_10bfa8c2d4564ad0912de674766a36c8). This method SHOULD assume that the handle to the printer or port object can be used without further access checks.
* Perform the validation steps specified in INFO Structures Query Parameters (section 3.1.4.1.9).

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* If *hPrinter* specifies a printer object, enumerate jobs on the job queue of the printer, up to the number specified by the *NoJobs* parameter, starting with the job whose index is specified by the *FirstJob* parameter.
* If *hPrinter* specifies a port object, enumerate jobs on the job queue of an arbitrary printer associated with that port, up to the number specified by the *NoJobs* parameter, starting with the job whose index is as specified by the *FirstJob* parameter. The method for selecting an arbitrary printer SHOULD match the method used when [RpcStartDocPrinter (section 3.1.4.9.1)](#Section_90f95ff03ae04ddea83d35f457c710a6) is called with a port object.
* Using the enumerated objects, perform the processing and response steps specified in section 3.1.4.1.9.
* Return the status of the operation.

##### RpcAddJob (Opnum 24)

RpcAddJob does not perform any function but returns a nonzero Windows error code to indicate failure.

1. DWORD RpcAddJob(
2. [in] PRINTER\_HANDLE hPrinter,
3. [in] DWORD Level,
4. [in, out, unique, size\_is(cbBuf), disable\_consistency\_check]
5. BYTE\* pAddJob,
6. [in] DWORD cbBuf,
7. [out] DWORD\* pcbNeeded
8. );

**hPrinter:** A handle to a printer object that was opened using [RpcAddPrinter (section 3.1.4.2.3)](#Section_bd0bb8d526fd470581f5614d32b9f38f), [RpcAddPrinterEx (section 3.1.4.2.15)](#Section_3af45717308f4b4f8428430d12856110), [RpcOpenPrinter (section 3.1.4.2.2)](#Section_989357e2446e4872bb381dce21e1313f), or [RpcOpenPrinterEx (section 3.1.4.2.14)](#Section_0e81ce1872b146c38584a205393b04ff).

**Level:** A value that MUST be 0x00000001, 0x00000002 or 0x00000003.

**pAddJob:** A pointer to a buffer of undefined values. This value can be NULL if *cbBuf* is zero and *Level* is 0x00000001.

**cbBuf:** The size, in bytes, of the buffer pointed to by *pAddJob*. If *Level* is 0x00000002 or 0x00000003, this value SHOULD be greater than or equal to 10 bytes.

**pcbNeeded:** A pointer to a variable that SHOULD receive zero.

**Return Values:** This method MUST return a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server SHOULD validate parameters as follows:

* Verify that *Level* is valid, and if this verification fails, return ERROR\_INVALID\_LEVEL.
* If *Level* is 0x0000002 or 0x0000003, verify that *cbBuf* is greater than or equal to 10 bytes on 32-bit and 18 bytes on 64-bit, and if this verification fails, return ERROR\_INVALID\_DATATYPE.
* If *Level* is 0x0000002 or 0x0000003, verify that at offset 0 from the beginning of the *pAddJob* buffer, there is a 32-bit value on 32-bit implementations and a 64-bit value on 64-bit implementations, between 0 and *cbBuf*, inclusive. If this verification fails, return ERROR\_INVALID\_LEVEL.

If parameter validation fails, the server SHOULD fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST return ERROR\_INVALID\_PARAMETER.

This method MUST be implemented to ensure compatibility with protocol clients.

##### RpcScheduleJob (Opnum 25)

RpcScheduleJob does not perform any function, but returns a nonzero Windows error code to indicate failure.

1. DWORD RpcScheduleJob(
2. [in] PRINTER\_HANDLE hPrinter,
3. [in] DWORD JobId
4. );

**hPrinter:** A handle to a printer object that was opened using [RpcAddPrinter](#Section_bd0bb8d526fd470581f5614d32b9f38f), [RpcAddPrinterEx](#Section_3af45717308f4b4f8428430d12856110), [RpcOpenPrinter](#Section_989357e2446e4872bb381dce21e1313f), or [RpcOpenPrinterEx](#Section_0e81ce1872b146c38584a205393b04ff).

**JobId:** The identifier of the print job.

**Return Values:** This method MUST return a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server SHOULD validate parameters as follows:

* Perform the validation steps specified in [PRINTER\_HANDLE Parameters (section 3.1.4.1.11)](#Section_10bfa8c2d4564ad0912de674766a36c8).

If parameter validation fails, the server SHOULD fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST return ERROR\_SPL\_NO\_ADDJOB.

This method MUST be implemented to ensure compatibility with protocol clients.

#### Printer Driver Management Methods

This section specifies methods for discovering and installing printer drivers.

| Method | Description |
| --- | --- |
| [RpcAddPrinterDriver](#Section_f23a75191c7740699acea6d8eae47c22) | RpcAddPrinterDriver installs a printer driver on the print server and links the configuration, data, and printer driver files.  Opnum 9 |
| [RpcEnumPrinterDrivers](#Section_857d00ac36824a0d86ca3d3c372e5e4a) | RpcEnumPrinterDrivers enumerates the printer drivers installed on a specified print server.  Opnum 10 |
| [RpcGetPrinterDriver](#Section_00d91852785d42e4a774b1d96361b0ce) | RpcGetPrinterDriver retrieves printer driver data for the specified printer.  Opnum 11 |
| [RpcGetPrinterDriverDirectory](#Section_9df11cf440984852ad72d1f75a82bffe) | RpcGetPrinterDriverDirectory retrieves the path of the printer driver directory.  Opnum 12 |
| [RpcDeletePrinterDriver](#Section_e03b14adabe84158ab0ce0033d820116) | RpcDeletePrinterDriver removes the specified printer driver from the list of supported drivers for a server.  Opnum 13 |
| [RpcGetPrinterDriver2](#Section_a73e77d07b2448cfa6d38b4061749495) | RpcGetPrinterDriver2 retrieves printer driver data for the specified printer.  Opnum 53 |
| [RpcDeletePrinterDriverEx](#Section_21d247cd66e24373a187b858c907e13f) | RpcDeletePrinterDriverEx removes the specified printer driver from the list of supported drivers for a server and deletes the files associated with the printer driver. This method also can delete specific versions of the printer driver.  Opnum 84 |
| [RpcAddPrinterDriverEx](#Section_b96cc49759e54510ab045484993b259b) | RpcAddPrinterDriverEx installs a printer driver on the print server. This method performs a similar function as RpcAddPrinterDriver (see section 3.1.4.4.1) and is also used to specify options that permit printer driver upgrade, printer driver downgrade, copying of newer files only, and copying of all files (regardless of their time stamps).  Opnum 89 |
| [RpcGetCorePrinterDrivers](#Section_3d6282322aa944f484e01168d3208f27) | RpcGetCorePrinterDrivers gets the GUIDs, versions, and publish dates of the specified core printer drivers, and the paths to their packages.  Opnum 102 |
| [RpcGetPrinterDriverPackagePath](#Section_e453b11350c64a129ca6e327b5accca6) | RpcGetPrinterDriverPackagePath gets the path to the specified printer driver package.  Opnum 104 |

##### RpcAddPrinterDriver (Opnum 9)

RpcAddPrinterDriver installs a printer driver on the print server and links the configuration, data, and printer driver files.

1. DWORD RpcAddPrinterDriver(
2. [in, string, unique] STRING\_HANDLE pName,
3. [in] DRIVER\_CONTAINER\* pDriverContainer
4. );

**pName:** Specified in [Print Server Name Parameters (section 3.1.4.1.4)](#Section_acb355a4913b4abb8cb5415aea160015).

**pDriverContainer:** Specified in [DRIVER\_CONTAINER Parameters (section 3.1.4.1.8.3)](#Section_353ff7966fb341cf8b350022dd53d886). The **Level** member of the [DRIVER\_CONTAINER](#Section_3a3f9cf78ec44921b1f686cf8d139bc2) MUST be 0x00000002, 0x00000003, or 0x00000004.

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST perform the validation steps specified in:

* Print Server Name Parameters (section 3.1.4.1.4).
* DRIVER\_CONTAINER Parameters.

Additional validation MAY[<371>](#Appendix_A_371) be performed.

In addition, print servers SHOULD[<372>](#Appendix_A_372) validate parameters as follows:

* Validate that the **cVersion** member of the **DRIVER\_INFO** structure contained in the DRIVER\_CONTAINER pointed to by the *pDriverContainer* is strictly less than 0x00000004. If this validation fails, return ERROR\_PRINTER\_DRIVER\_BLOCKED.
* Validate that the **pEnvironment** member of the **DRIVER\_INFO** structure contained in the DRIVER\_CONTAINER pointed to by the *pDriverContainer* parameter is not "Windows ARM". If this validation fails, return ERROR\_NOT\_SUPPORTED.

If the installation requested by the print client is a printer driver upgrade, print servers SHOULD perform the following additional validation steps:

* Validate that the currently installed printer driver is not a class printer driver.
* Validate that if the currently installed printer driver has a driver version of 0x00000004, the currently installed printer driver does not have a newer driver date, or if the driver dates are the same, that the currently installed printer driver does not have a newer manufacturer-provided driver version number.
* Validate that if the currently installed printer driver has a driver version of 0x00000004, there are no printers on the print server that are shared and also use the currently installed printer driver.

If this validation fails, the print server MUST return ERROR\_PRINTER\_DRIVER\_BLOCKED. [<373>](#Appendix_A_373)

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Copy the printer driver files to their destination. If the copy operation fails, the server MUST fail the call immediately and return a nonzero error response to the client.
* Create the printer driver object, using an implementation-specific mechanism to determine the Boolean values of each of the attributes of the printer driver object. [<374>](#Appendix_A_374)
* If any clients have registered for notification of server object changes, a notification MUST be broadcast to them.
* Return the status of the operation.

##### RpcEnumPrinterDrivers (Opnum 10)

RpcEnumPrinterDrivers enumerates the printer drivers installed on a specified print server.

1. DWORD RpcEnumPrinterDrivers(
2. [in, string, unique] STRING\_HANDLE pName,
3. [in, string, unique] wchar\_t\* pEnvironment,
4. [in] DWORD Level,
5. [in, out, unique, size\_is(cbBuf), disable\_consistency\_check]
6. BYTE\* pDrivers,
7. [in] DWORD cbBuf,
8. [out] DWORD\* pcbNeeded,
9. [out] DWORD\* pcReturned
10. );

**pName:** Specified in [Print Server Name Parameters (section 3.1.4.1.4)](#Section_acb355a4913b4abb8cb5415aea160015).

**pEnvironment:** Specified in [Environment Name Parameters (section 3.1.4.1.3)](#Section_5b2bf7d5ac77464fbe5ef3a7c1782246).[<375>](#Appendix_A_375)

**Level:** The driver information level.

| Value | Meaning |
| --- | --- |
| 0x00000001 | Corresponds to [\_DRIVER\_INFO\_1 (section 2.2.2.4.1)](#Section_59eab35482fd4299884e9a1226073a4a). |
| 0x00000002 | Corresponds to [\_DRIVER\_INFO\_2 (section 2.2.2.4.2)](#Section_2825d22ec5a547cda2163e903fd6e030). |
| 0x00000003 | Corresponds to [\_DRIVER\_INFO\_3 (section 2.2.2.4.3)](#Section_b9a54cc1e79f45c8a1261acd1f0f8663). |
| 0x00000004 | Corresponds to [\_DRIVER\_INFO\_4 (section 2.2.2.4.4)](#Section_2a53e91004c34aba905a26741dcd59dd). |
| 0x00000005 | Corresponds to [\_DRIVER\_INFO\_5 (section 2.2.2.4.5)](#Section_e20b5b5573494f50a9f2bcc02d958cfc). |
| 0x00000006 | Corresponds to [\_DRIVER\_INFO\_6 (section 2.2.2.4.6)](#Section_9a048522474644c2ac92d881ac1f36a8). |
| 0x00000008 | Corresponds to [\_DRIVER\_INFO\_8 (section 2.2.2.4.8)](#Section_f9b749d608fc48208d7bfe0ee33e42c5). |

**pDrivers:** A pointer to the [BUFFER](#Section_f29e9057df2940d49c51c615c0da62a1), as specified in INFO Structures Query Parameters (section 3.1.4.1.9).

**BUFFER TYPE**: \_DRIVER\_INFO.

This parameter can be NULL if *cbBuf* equals zero.

**cbBuf:** Specified in INFO Structures Query Parameters (section 3.1.4.1.9).

**pcbNeeded:** Specified in INFO Structures Query Parameters (section 3.1.4.1.9).

**pcReturned:** Specified in INFO Structures Query Parameters (section 3.1.4.1.9).

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST perform the validation steps specified in the following sections:

* Print Server Name Parameters (section 3.1.4.1.4).
* Environment Name Parameters (section 3.1.4.1.3).
* INFO Structures Query Parameters (section 3.1.4.1.9).

Additional validation MAY[<376>](#Appendix_A_376) be performed.

In addition, the print server SHOULD validate that, if the *Level* parameter is 0x00000065, every printer driver installed on the print server has a driver version strictly less than 0x00000004. If that validation fails, this method MUST return ERROR\_CAN\_NOT\_COMPLETE.[<377>](#Appendix_A_377)

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Enumerate all drivers on the specified print server that match the requested environment.
* Using the enumerated objects, perform the processing and response steps specified in INFO Structures Query Parameters (section 3.1.4.1.9).
* If at any point during the operation the print server failed to calculate the size of the **INFO** structure for any printer driver, the print server SHOULD fail the operation immediately and return ERROR\_CAN\_NOT\_COMPLETE.[<378>](#Appendix_A_378)
* Return the status of the operation.

##### RpcGetPrinterDriver (Opnum 11)

RpcGetPrinterDriver retrieves printer driver data for the specified printer.

1. DWORD RpcGetPrinterDriver(
2. [in] PRINTER\_HANDLE hPrinter,
3. [in, string, unique] wchar\_t\* pEnvironment,
4. [in] DWORD Level,
5. [in, out, unique, size\_is(cbBuf), disable\_consistency\_check]
6. BYTE\* pDriver,
7. [in] DWORD cbBuf,
8. [out] DWORD\* pcbNeeded
9. );

**hPrinter:** A handle to a printer object that was opened by [RpcAddPrinter (section 3.1.4.2.3)](#Section_bd0bb8d526fd470581f5614d32b9f38f), [RpcAddPrinterEx (section 3.1.4.2.15)](#Section_3af45717308f4b4f8428430d12856110), [RpcOpenPrinter (section 3.1.4.2.2)](#Section_989357e2446e4872bb381dce21e1313f), or [RpcOpenPrinterEx (section 3.1.4.2.14)](#Section_0e81ce1872b146c38584a205393b04ff).

**pEnvironment:** A parameter specified in [Environment Name Parameters (section 3.1.4.1.3)](#Section_5b2bf7d5ac77464fbe5ef3a7c1782246).

**Level:** The driver information level.

| Value | Meaning |
| --- | --- |
| 0x00000001 | Corresponds to [\_DRIVER\_INFO\_1 (section 2.2.2.4.1)](#Section_59eab35482fd4299884e9a1226073a4a). |
| 0x00000002 | Corresponds to [\_DRIVER\_INFO\_2 (section 2.2.2.4.2)](#Section_2825d22ec5a547cda2163e903fd6e030). |
| 0x00000003 | Corresponds to [\_DRIVER\_INFO\_3 (section 2.2.2.4.3)](#Section_b9a54cc1e79f45c8a1261acd1f0f8663). |
| 0x00000004 | Corresponds to [\_DRIVER\_INFO\_4 (section 2.2.2.4.4)](#Section_2a53e91004c34aba905a26741dcd59dd). |
| 0x00000006 | Corresponds to [\_DRIVER\_INFO\_6 (section 2.2.2.4.6)](#Section_9a048522474644c2ac92d881ac1f36a8). |
| 0x00000008 | Corresponds to [\_DRIVER\_INFO\_8 (section 2.2.2.4.8)](#Section_f9b749d608fc48208d7bfe0ee33e42c5). |

**pDriver:** An optional pointer to *BUFFER*, as specified in [INFO Structures Query Parameters (section 3.1.4.1.9)](#Section_f29e9057df2940d49c51c615c0da62a1).

**BUFFER TYPE**: \_DRIVER\_INFO.

This parameter SHOULD be NULL if *cbBuf* is zero.

**cbBuf:** Specified in INFO Structures Query Parameters.

**pcbNeeded:** Specified in INFO Structures Query Parameters.

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate parameters by performing the validation steps specified in:

* [PRINTER\_HANDLE Parameters (section 3.1.4.1.11)](#Section_10bfa8c2d4564ad0912de674766a36c8)). This method SHOULD assume that the handle to the printer object can be used without further access checks.
* Environment Name Parameters (section 3.1.4.1.3).
* INFO Structures Query Parameters (section 3.1.4.1.9).

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* The server SHOULD select a printer driver that meets the following specifications:
  + Compatible with the environment specified by the *pEnvironment* parameter.
  + Compatible with one of the printer drivers listed in the **pszzPreviousNames** member of the [DRIVER\_INFO](#Section_17943579a5144d0b8ded422265b23f45) of the printer driver that is associated with the printer.

If such a printer driver cannot be located, the server SHOULD return ERROR\_UNKNOWN\_PRINTER\_DRIVER.[<379>](#Appendix_A_379)

* Using the information about the printer driver, perform the processing and response steps specified in INFO Structures Query Parameters.
* Return the status of the operation.

##### RpcGetPrinterDriverDirectory (Opnum 12)

RpcGetPrinterDriverDirectory retrieves the path of the printer driver directory.

1. DWORD RpcGetPrinterDriverDirectory(
2. [in, string, unique] STRING\_HANDLE pName,
3. [in, string, unique] wchar\_t\* pEnvironment,
4. [in] DWORD Level,
5. [in, out, unique, size\_is(cbBuf), disable\_consistency\_check]
6. BYTE\* pDriverDirectory,
7. [in] DWORD cbBuf,
8. [out] DWORD\* pcbNeeded
9. );

**pName:** Specified in [Print Server Name Parameters (section 3.1.4.1.4)](#Section_acb355a4913b4abb8cb5415aea160015).

**pEnvironment:** Specified in [Environment Name Parameters (section 3.1.4.1.3)](#Section_5b2bf7d5ac77464fbe5ef3a7c1782246).

**Level:** A value that MUST be 0x00000001.

**pDriverDirectory:** An optional pointer to *BUFFER*, as specified in [String Query Parameters (section 3.1.4.1.7)](#Section_6bae59aff92341b9b39513872aaf2ce0). If *cbBuf* is zero, this parameter SHOULD be NULL.

**cbBuf:** Specified in String Query Parameters (section 3.1.4.1.7).

**pcbNeeded:** Specified in String Query Parameters (section 3.1.4.1.7).

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate parameters by performing the validation steps specified in:

* Print Server Name Parameters (section 3.1.4.1.4).
* Environment Name Parameters (section 3.1.4.1.3).
* String Query Parameters (section 3.1.4.1.7).

Additional validation MAY[<380>](#Appendix_A_380) be performed.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* With the path of the printer driver directory on the print server, perform the processing and response steps specified in String Query Parameters.
* Return the status of the operation.

##### RpcDeletePrinterDriver (Opnum 13)

RpcDeletePrinterDriver removes the specified printer driver from the list of supported drivers for a server.

1. DWORD RpcDeletePrinterDriver(
2. [in, string, unique] STRING\_HANDLE pName,
3. [in, string] wchar\_t\* pEnvironment,
4. [in, string] wchar\_t\* pDriverName
5. );

**pName:** Specified in [Print Server Name Parameters (section 3.1.4.1.4)](#Section_acb355a4913b4abb8cb5415aea160015).

**pEnvironment:** Specified in [Environment Name Parameters (section 3.1.4.1.3)](#Section_5b2bf7d5ac77464fbe5ef3a7c1782246).

**pDriverName:** This parameter MUST be a non-NULL pointer to a string that MUST specify the name of the printer driver to delete. For rules governing printer driver names, see section [2.2.4.3](#Section_a4c7686fa88c48f996ebbf4dfc3fe519).

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate parameters as follows:

* Perform the validation steps that are specified in Print Server Name Parameters.
* Perform the validation steps that are specified in Environment Name Parameters.
* Verify that the string pointed to by the *pDriverName* parameter contains the name of a driver that is part of the list of drivers that are installed on the server for the environment specified by the string pointed to by the *pEnvironment* parameter; if that verification fails, return ERROR\_UNKNOWN\_PRINTER\_DRIVER.
* Verify that the printer driver is not used by any printer in the system, and if that verification fails, return ERROR\_PRINTER\_DRIVER\_IN\_USE.
* Additional validation MAY[<381>](#Appendix_A_381) be performed.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Clear all references to this printer driver from any other data structures.
* Delete the printer driver object.
* If any clients have registered for notifications of the server object change, a notification MUST be broadcast to them.
* Return the status of the operation.

##### RpcGetPrinterDriver2 (Opnum 53)

RpcGetPrinterDriver2 retrieves printer driver data for the specified printer.[<382>](#Appendix_A_382)

1. DWORD RpcGetPrinterDriver2(
2. [in] PRINTER\_HANDLE hPrinter,
3. [in, string, unique] wchar\_t\* pEnvironment,
4. [in] DWORD Level,
5. [in, out, unique, size\_is(cbBuf), disable\_consistency\_check]
6. BYTE\* pDriver,
7. [in] DWORD cbBuf,
8. [out] DWORD\* pcbNeeded,
9. [in] DWORD dwClientMajorVersion,
10. [in] DWORD dwClientMinorVersion,
11. [out] DWORD\* pdwServerMaxVersion,
12. [out] DWORD\* pdwServerMinVersion
13. );

**hPrinter:** A handle to a printer object that was opened by [RpcAddPrinter (section 3.1.4.2.3)](#Section_bd0bb8d526fd470581f5614d32b9f38f), [RpcAddPrinterEx (section 3.1.4.2.15)](#Section_3af45717308f4b4f8428430d12856110), [RpcOpenPrinter (section 3.1.4.2.2)](#Section_989357e2446e4872bb381dce21e1313f), or [RpcOpenPrinterEx (section 3.1.4.2.14)](#Section_0e81ce1872b146c38584a205393b04ff).

**pEnvironment:** A parameter specified in [Environment Name Parameters (section 3.1.4.1.3)](#Section_5b2bf7d5ac77464fbe5ef3a7c1782246).

**Level:** The driver information level.

| Value | Meaning |
| --- | --- |
| 0x00000001 | Corresponds to [\_DRIVER\_INFO\_1 (section 2.2.2.4.1)](#Section_59eab35482fd4299884e9a1226073a4a). |
| 0x00000002 | Corresponds to [\_DRIVER\_INFO\_2 (section 2.2.2.4.2)](#Section_2825d22ec5a547cda2163e903fd6e030). |
| 0x00000003 | Corresponds to [\_DRIVER\_INFO\_3 (section 2.2.2.4.3)](#Section_b9a54cc1e79f45c8a1261acd1f0f8663). |
| 0x00000004 | Corresponds to [\_DRIVER\_INFO\_4 (section 2.2.2.4.4)](#Section_2a53e91004c34aba905a26741dcd59dd). |
| 0x00000005 | Corresponds to [\_DRIVER\_INFO\_5 (section 2.2.2.4.5)](#Section_e20b5b5573494f50a9f2bcc02d958cfc). |
| 0x00000006 | Corresponds to [\_DRIVER\_INFO\_6 (section 2.2.2.4.6)](#Section_9a048522474644c2ac92d881ac1f36a8). |
| 0x00000008 | Corresponds to [\_DRIVER\_INFO\_8 (section 2.2.2.4.8)](#Section_f9b749d608fc48208d7bfe0ee33e42c5). |
| 0x00000065 | Corresponds to [\_DRIVER\_INFO\_101 (section 2.2.2.4.9)](#Section_821d675e39444c77b4ec3b519883cfa6). |

**pDriver:** A pointer to the [BUFFER](#Section_f29e9057df2940d49c51c615c0da62a1), as specified in INFO Structures Query Parameters (section 3.1.4.1.9).

**BUFFER TYPE**: \_DRIVER\_INFO.

This parameter can be NULL if *cbBuf* is zero.

**cbBuf:** A parameter specified in INFO Structures Query Parameters.

**pcbNeeded:** A parameter specified in INFO Structures Query Parameters.

**dwClientMajorVersion:** The implementation-specific major printer driver version of the client operating system.[<383>](#Appendix_A_383)

**dwClientMinorVersion:** The implementation-specific minor printer driver version of the client operating system.[<384>](#Appendix_A_384)

**pdwServerMaxVersion:** A pointer to a DWORD that receives the implementation-specific major version that the operating system supports for that printer driver.

**pdwServerMinVersion:** A pointer to a DWORD that receives the implementation-specific minimum version that the operating system supports for that printer driver.[<385>](#Appendix_A_385)

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate parameters as follows:

* Perform the validation steps specified in [PRINTER\_HANDLE Parameters (section 3.1.4.1.11)](#Section_10bfa8c2d4564ad0912de674766a36c8). This method SHOULD assume that the handle to the printer object can be used without further access checks.
* Perform the validation steps specified in Environment Name Parameters.
* Perform the validation steps specified in INFO Structures Query Parameters.

In addition, the print server SHOULD validate that, if the *Level* parameter is 0x00000065, the printer driver associated with the printer object has a driver version strictly less than 0x00000004. If that validation fails, this method MUST return ERROR\_CAN\_NOT\_COMPLETE.[<386>](#Appendix_A_386)

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Find a printer driver that is compatible with the OS version on the specified print server that is specified by the value of the *dwClientMajorVersion* and *dwClientMinorVersion* parameters.
* Using the identified printer driver, perform the processing and response steps specified in INFO Structures Query Parameters.

If parameter validation was successful, the server SHOULD also perform the following steps:

* Store the actual compatible operating system version for the printer driver in the variables pointed to by the *pdwServerMaxVersion* and *pdwServerMinVersion* parameters.
* Return the status of the operation.

##### RpcDeletePrinterDriverEx (Opnum 84)

RpcDeletePrinterDriverEx removes the specified printer driver from the list of supported drivers for a server and deletes the files associated with it.[<387>](#Appendix_A_387) This method can also be used to delete specific versions of a driver.

1. DWORD RpcDeletePrinterDriverEx(
2. [in, string, unique] STRING\_HANDLE pName,
3. [in, string] wchar\_t\* pEnvironment,
4. [in, string] wchar\_t\* pDriverName,
5. [in] DWORD dwDeleteFlag,
6. [in] DWORD dwVersionNum
7. );

**pName:** A [STRING\_HANDLE structure (section 2.2.1.1.7)](#Section_59883eb85ce64ad4bdd2599434cfdcf2) that conforms to the parameter specification in [Print Server Name Parameters (section 3.1.4.1.4)](#Section_acb355a4913b4abb8cb5415aea160015).

**pEnvironment:** A string that conforms to the parameter specification in [Environment Name Parameters (section 3.1.4.1.3)](#Section_5b2bf7d5ac77464fbe5ef3a7c1782246).

**pDriverName:** A pointer to a string that specifies the name of the printer driver to delete. For rules governing printer driver names, see section [2.2.4.3](#Section_a4c7686fa88c48f996ebbf4dfc3fe519).

**dwDeleteFlag:** A bit field that specifies options for deleting files and versions of the printer driver. If the value of this parameter is zero, the driver MUST be removed from the list of supported drivers, and the driver files MUST remain on the print server.

These flags can be combined to specify multiple options.

| Value | Meaning |
| --- | --- |
| DPD\_DELETE\_UNUSED\_FILES  0x00000001 | Remove unused printer driver files. In this case, an error MUST NOT be returned if some of the files are being used by another installed driver. |
| DPD\_DELETE\_SPECIFIC\_VERSION  0x00000002 | Delete the version specified by the value of the *dwVersionNum* parameter. Because more than one version of a printer driver can be installed on a print server, setting this flag does not guarantee that the driver will be removed from the list of supported drivers on the server. |
| DPD\_DELETE\_ALL\_FILES  0x00000004 | Delete the printer driver only if all its associated files can be removed. In this case, an error MUST be returned if some of the files are being used by another installed driver. |

**dwVersionNum:** The version of the printer driver to delete.

The value of this parameter is implementation-specific and identifies the driver version and the operating system for which the driver was written (the driver version contained by each printer driver object in the list of printer drivers described in section [3.1.1](#Section_1bd61448db4b4284a4b18346732402d4)). It has the same format and meaning as the **cVersion** members in [RPC\_DRIVER\_INFO structures (section 2.2.1.5)](#Section_17943579a5144d0b8ded422265b23f45).[<388>](#Appendix_A_388)

This parameter MUST be ignored if the **DPD\_DELETE\_SPECIFIC\_VERSION** flag in the *dwDeleteFlag* parameter is not set.

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate the following:

* Perform the validation steps for Print Server Name Parameters.
* Perform the validation steps for Environment Name Parameters.
* Verify that the string pointed to by the *pDriverName* parameter contains the name of a driver that is part of the list of drivers that are installed on the server for the environment specified by the string pointed to by the *pEnvironment* parameter. If that verification fails, ERROR\_UNKNOWN\_PRINTER\_DRIVER MUST be returned.
* Verify that the printer driver identified by *pDriverName* is not being used by any printer in the system. If that verification fails, ERROR\_PRINTER\_DRIVER\_IN\_USE MUST be returned.
* Verify that the value of the *dwDeleteFlag* parameter contains the result of the bitwise OR of zero or more of the **DPD\_DELETE** defined constants and that all other bits are zero. If that verification fails, ERROR\_INVALID\_PARAMETER MUST be returned.
* Additional validation MAY[<389>](#Appendix_A_389) be performed.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Clear references to this version of the printer driver in any other data structures.
* Delete the printer driver object and any associated driver files in compliance with the settings in the *dwDeleteFlag* parameter.
* If the **DPD\_DELETE\_SPECIFIC\_VERSION** bit is set in *dwDeleteFlag*, delete only printer drivers with a version number that matches the value of *dwVersionNum*.
* If any clients have registered for notification of server object changes, a notification MUST be broadcast to them.
* Return the status of the operation.

##### RpcAddPrinterDriverEx (Opnum 89)

RpcAddPrinterDriverEx installs a printer driver on the print server.[<390>](#Appendix_A_390) This method performs a function similar to [RpcAddPrinterDriver (section 3.1.4.4.1)](#Section_f23a75191c7740699acea6d8eae47c22) and is also used to specify options that permit printer driver upgrade, printer driver downgrade, copying of newer files only, and copying of all files regardless of their time stamps.

1. DWORD RpcAddPrinterDriverEx(
2. [in, string, unique] STRING\_HANDLE pName,
3. [in] DRIVER\_CONTAINER\* pDriverContainer,
4. [in] DWORD dwFileCopyFlags
5. );

**pName:** A string that conforms to the parameter specification in [Print Server Name Parameters (section 3.1.4.1.4)](#Section_acb355a4913b4abb8cb5415aea160015).

**pDriverContainer:** A pointer to a [DRIVER\_CONTAINER structure (section 2.2.1.2.3)](#Section_3a3f9cf78ec44921b1f686cf8d139bc2) that MUST conform to the specification in [DRIVER\_CONTAINER parameters (section 3.1.4.1.8.3)](#Section_353ff7966fb341cf8b350022dd53d886).

The **Level** member of the DRIVER\_CONTAINER refers to the level of driver information structure, as follows:

| Value | Meaning |
| --- | --- |
| 0x00000002 | DRIVER\_INFO\_2, specified in section [2.2.1.5.2](#Section_39bbfc3087684cd49930434857e2c2a2). |
| 0x00000003 | RPC\_DRIVER\_INFO\_3, specified in section [2.2.1.5.3](#Section_b4d3b5f3c6314fc393484c983590a0d8). |
| 0x00000004 | RPC\_DRIVER\_INFO\_4, specified in section [2.2.1.5.4](#Section_459d24e40b754fc7b246fe40a1981eed). |
| 0x00000006 | RPC\_DRIVER\_INFO\_6, specified in section [2.2.1.5.5](#Section_ae12474015ef482c824e56e5436ed44b). |
| 0x00000008 | RPC\_DRIVER\_INFO\_8, specified in section [2.2.1.5.6](#Section_937d5fc867f145f8ab878b56a8fca558). |

**dwFileCopyFlags:** A bit field that specifies options for copying replacement printer driver files. The value of this parameter is a combination of flags from the following tables.

Exactly one of the following flags MUST be specified:

| Value | Meaning |
| --- | --- |
| APD\_STRICT\_UPGRADE  0x00000001 | Add the replacement printer driver only if none of the files of the replacement driver are older than any corresponding files of the currently installed driver. |
| APD\_STRICT\_DOWNGRADE  0x00000002 | Add the replacement printer driver only if none of the files of the currently installed driver are older than any corresponding files of the replacement driver. |
| APD\_COPY\_ALL\_FILES  0x00000004 | Add the printer driver and copy all the files in the driver directory. File time stamps MUST be ignored. |
| APD\_COPY\_NEW\_FILES  0x00000008 | Add the printer driver and copy the files in the driver directory that are newer than any of the corresponding files that are currently in use. |

Zero or more of the following flags can be specified.

| Value | Meaning |
| --- | --- |
| APD\_COPY\_FROM\_DIRECTORY  0x00000010 | Add the printer driver by using the fully qualified file names that are specified in the [\_DRIVER\_INFO\_6](#Section_9a048522474644c2ac92d881ac1f36a8) structure. If this flag is specified, one of the other copy flags in this bit field MUST be specified. |
| APD\_DONT\_COPY\_FILES\_TO\_CLUSTER  0x00001000 | When adding a printer driver to a print server cluster, do not copy the driver files to the shared cluster disk. |
| APD\_COPY\_TO\_ALL\_SPOOLERS  0x00002000 | Add the printer driver to cluster spooler servers. |
| APD\_INSTALL\_WARNED\_DRIVER  0x00008000 | Add the printer driver even if it is in the server's warned list.[<391>](#Appendix_A_391) |
| APD\_RETURN\_BLOCKING\_STATUS\_CODE  0x00010000 | Specifies the implementation-specific error code to return if the printer driver is blocked from installation by server policy. [<392>](#Appendix_A_392) |

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate parameters as follows:

* Perform the validation steps that are specified in Print Server Name Parameters.
* Perform the validation steps that are specified in DRIVER\_CONTAINER Parameters.
* Verify that *dwFileCopyFlags* contains a valid flag value or set of values as described in the *dwFileCopyFlags* parameter definition above. If this verification fails, return ERROR\_INVALID\_PARAMETER.
* Additional validation MAY [<393>](#Appendix_A_393) be performed.

In addition, print servers SHOULD[<394>](#Appendix_A_394) validate parameters as follows:

* Validate that the **cVersion** member of the **DRIVER\_INFO** structure contained in the DRIVER\_CONTAINER pointed to by the pDriverContainer is strictly less than 0x00000004. If this validation fails, return ERROR\_PRINTER\_DRIVER\_BLOCKED.
* Validate that the **pEnvironment** member of the **DRIVER\_INFO** structure contained in the DRIVER\_CONTAINER pointed to by the *pDriverContainer* parameter is not "Windows ARM". If this validation fails, return ERROR\_NOT\_SUPPORTED.

If the installation requested by the print client is a printer driver upgrade, print servers SHOULD perform the following additional validation steps:

* Validate that the currently installed printer driver is not a class printer driver.
* Validate that if the currently installed printer driver has a driver version of 0x00000004, the currently installed printer driver does not have a newer driver date, or if the driver dates are the same, that the currently installed printer driver does not have a newer manufacturer-provided driver version number.
* Validate that if the currently installed printer driver has a driver version of 0x00000004, there are no printers on the print server that are shared and also use the currently installed printer driver.

If this validation fails, the print server MUST return ERROR\_PRINTER\_DRIVER\_BLOCKED. [<395>](#Appendix_A_395)

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Copy the printer driver files to their destinations, in compliance with the settings described by the value of the *dwFileCopyFlags* parameter.
* Create the printer driver object, using an implementation-specific mechanism to determine the Boolean values of each of the attributes of the printer driver.[<396>](#Appendix_A_396)
* If any clients have registered for notification of server object changes, a notification MUST be broadcast to them.
* Return the status of the operation.

##### RpcGetCorePrinterDrivers (Opnum 102)

RpcGetCorePrinterDrivers gets the GUIDs, versions, and publish dates of the specified core printer drivers, and the paths to their packages.[<397>](#Appendix_A_397)

1. HRESULT RpcGetCorePrinterDrivers(
2. [in, string, unique] STRING\_HANDLE pszServer,
3. [in, string] const wchar\_t\* pszEnvironment,
4. [in] DWORD cchCoreDrivers,
5. [in, size\_is(cchCoreDrivers)] const wchar\_t\* pszzCoreDriverDependencies,
6. [in] DWORD cCorePrinterDrivers,
7. [out, size\_is(cCorePrinterDrivers)]
8. CORE\_PRINTER\_DRIVER\* pCorePrinterDrivers
9. );

**pszServer:** A [STRING\_HANDLE (section 2.2.1.1.7)](#Section_59883eb85ce64ad4bdd2599434cfdcf2) for a server object. This parameter MUST adhere to the specification in [Print Server Name Parameters (section 3.1.4.1.4)](#Section_acb355a4913b4abb8cb5415aea160015).

**pszEnvironment:** A pointer to a **string** that specifies the environment name for which the core printer driver information will be returned. For rules governing environment names and product behaviors, see [2.2.4.4](#Section_ad7758938a194786ab4634731f52d1af).

**cchCoreDrivers:** The size, in bytes, of the buffer that is referenced by the *pszzCoreDriverDependencies* parameter.

**pszzCoreDriverDependencies:** A pointer to a multisz that contains a list of IDs[<398>](#Appendix_A_398) of the core printer drivers to be retrieved.

A print client SHOULD obtain this list of IDs as follows:

1. Call [RpcGetPrinterDriver (section 3.1.4.4.3)](#Section_00d91852785d42e4a774b1d96361b0ce) with a *Level* parameter value of 0x00000008.
2. A [\_DRIVER\_INFO\_8](#Section_f9b749d608fc48208d7bfe0ee33e42c5) custom-marshaled structure is returned in the *pDriver* parameter.
3. In the \_DRIVER\_INFO\_8 structure, the **szzCoreDependenciesOffset** member contains an offset to a multisz that contains the list of IDs.

**cCorePrinterDrivers:** The count of [CORE\_PRINTER\_DRIVER (section 2.2.2.13)](#Section_7cdc8cf152854420bfa53233ed20125b) structures that are contained in the buffer that is pointed to by the *pCorePrinterDrivers* parameter. It MUST equal the number of IDs that are specified in the multisz that is pointed to by the *pszzCoreDriverDependencies* parameter.

**pCorePrinterDrivers:** A pointer to a buffer that receives an array of CORE\_PRINTER\_DRIVER structures.

**Return Values:** This method MUST return zero or an **HRESULT** success value ([MS-ERREF] section 2.1) to indicate successful completion or an ***HRESULT*** error value to indicate failure.

**Exceptions Thrown:** This method MUST NOT throw any exceptions other than those that are thrown by the underlying RPC protocol [[MS-RPCE]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-RPCE%5d.pdf).

**Parameter Validation Requirements:** Upon receiving this method call, the server MUST validate parameters as follows:

* The string pointed to by the *pszEnvironment* parameter MUST specify one of the supported environment names on the server for that type of driver; otherwise the server MUST return ERROR\_INVALID\_ENVIRONMENT.
* *cCorePrinterDrivers* MUST be equal to or greater than 1; otherwise the server MUST return ERROR\_INVALID\_PARAMETER.
* *cCorePrinterDrivers* MUST be equal to the number of GUIDs present in *pszzCoreDriverDependencies*; otherwise the server MUST return ERROR\_INVALID\_PARAMETER.
* The *pCorePrinterDrivers* parameter MUST NOT be NULL.

If parameter validation fails, the server MUST fail the operation immediately, and return a nonzero error response to the client.

**Processing and Response Requirements:** If parameter validation succeeds, the server MUST process the method call by

* Enumerating the CORE\_PRINTER\_DRIVER structures in the system.
* Populating each CORE\_PRINTER\_DRIVER structure in the supplied buffer with information about the core printer driver.
* Returning a response that MUST contain the output parameters mentioned above and the status of the operation.

The server MUST NOT change the **List of Core Printer Drivers** as part of processing this method call.

##### RpcGetPrinterDriverPackagePath (Opnum 104)

RpcGetPrinterDriverPackagePath gets the path to the specified printer driver package.[<399>](#Appendix_A_399)

1. HRESULT RpcGetPrinterDriverPackagePath(
2. [in, string, unique] STRING\_HANDLE pszServer,
3. [in, string] const wchar\_t\* pszEnvironment,
4. [in, string, unique] const wchar\_t\* pszLanguage,
5. [in, string] const wchar\_t\* pszPackageID,
6. [in, out, unique, size\_is(cchDriverPackageCab)]
7. wchar\_t\* pszDriverPackageCab,
8. [in] DWORD cchDriverPackageCab,
9. [out] DWORD\* pcchRequiredSize
10. );

**pszServer:** A [STRING\_HANDLE (section 2.2.1.1.7)](#Section_59883eb85ce64ad4bdd2599434cfdcf2) for a server object. This parameter MUST adhere to the specification in [Print Server Name Parameters (section 3.1.4.1.4)](#Section_acb355a4913b4abb8cb5415aea160015).

**pszEnvironment:** This parameter MUST be a non-NULL pointer to a string that MUST specify the environment name for which the [**driver package**](#gt_1874576f-4b40-4a78-a1f3-9e47b7ce21c9) path MUST be returned. For rules governing environment names, see section [2.2.4.4](#Section_ad7758938a194786ab4634731f52d1af).

**pszLanguage:** This parameter MUST be NULL, or MUST be a non-NULL pointer to a string that MUST specify the language for which the driver package path MUST[<400>](#Appendix_A_400) be returned.

**pszPackageID:** This parameter MUST be a non-NULL pointer to a string that MUST specify the package name. The package name MUST be obtained by calling [RpcGetCorePrinterDrivers](#Section_3d6282322aa944f484e01168d3208f27).

**pszDriverPackageCab:** This parameter MUST be a pointer to a buffer that MUST[<401>](#Appendix_A_401) receive a string that specifies the path name of the driver package file. For rules governing path names, see section [2.2.4.9](#Section_1d75f622d07c45ccbf1ea6191e408516). *pszDriverPackageCab* MUST NOT be NULL unless *cchDriverPackageCab* is zero.

**cchDriverPackageCab:** This parameter MUST specify the size, in characters, of the buffer that is referenced by the *pszDriverPackageCab* parameter. The value of this parameter MAY[<402>](#Appendix_A_402) be zero.

**pcchRequiredSize:** This parameter MUST be a non-NULL pointer to a variable that MUST receive the required size of the buffer that is pointed to by the *pszDriverPackageCab* parameter.

**Return Values:** This method MUST return zero or an **HRESULT** success value ([[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf) section 2.1) to indicate successful completion or an **HRESULT** error value to indicate failure.

**Exceptions Thrown:** This method MUST NOT throw any exceptions other than those that are thrown by the underlying RPC protocol [[MS-RPCE]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-RPCE%5d.pdf).

**Parameter Validation Requirements:** Upon receiving this method call, the server MUST validate parameters as follows:

* The string pointed to by the *pszEnvironment* parameter MUST specify one of the supported environment names on the server for that type of driver; otherwise the server MUST return ERROR\_INVALID\_ENVIRONMENT.
* The value of the *pszPackageID* parameter MUST NOT be NULL; otherwise the server MUST return ERROR\_INVALID\_PARAMETER.
* The value of the *pcchRequiredSize* parameter MUST NOT be NULL; otherwise the server MUST return ERROR\_INVALID\_PARAMETER.
* The size specified by *cchDriverPackageCab* MUST be sufficient to hold the path name of the driver package file; otherwise the server MUST calculate the required number of characters and write this number to the variable pointed to by the *pcchRequiredSize* output parameter, and return ERROR\_INSUFFICIENT\_BUFFER.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client.

**Processing and Response Requirements:** If parameter validation succeeds, the server MUST process the method call by:

* Searching for the driver package cab file for the specified *pszPackageID*.
* Returning the driver package cab path for package ID in the output parameter pszDriverPackageCab.
* Setting the contents of the parameter *pcchRequiredSize* to the size of the string buffer required to hold the driver package cab.
* Returning a response that MUST contain the output parameters mentioned above and the status of the operation.

The server MUST NOT change the list of driver package cabs as part of processing this method call.

#### Form Management Methods

This section specifies methods for discovering and configuring printer forms.

| Method | Description |
| --- | --- |
| [RpcAddForm](#Section_d6eb54304a274e5ab6988140a12fa889) | RpcAddForm adds a form name to the list of supported forms.  Opnum 30 |
| [RpcDeleteForm](#Section_1b25a56720084ca1ac93feaeaec7ec54) | RpcDeleteForm removes a form name from the list of supported forms.  Opnum 31 |
| [RpcGetForm](#Section_be369d1d1ede49b181c209c69411c25d) | RpcGetForm retrieves information about a specified form.  Opnum 32 |
| [RpcSetForm](#Section_c17cbf6601a24ac0b57b1d2d8d1f73de) | RpcSetForm replaces the form information for the specified form.  Opnum 33 |
| [RpcEnumForms](#Section_0b9dde1473d14f1a9f748e82255a9612) | The RpcEnumForms method enumerates the forms that the specified printer supports.  Opnum 34 |

##### RpcAddForm (Opnum 30)

RpcAddForm adds a form name to the list of supported forms.

1. DWORD RpcAddForm(
2. [in] PRINTER\_HANDLE hPrinter,
3. [in] FORM\_CONTAINER\* pFormInfoContainer
4. );

**hPrinter:** This parameter MUST specify a handle to a printer object or server object that MUST have been opened using the [RpcAddPrinter](#Section_bd0bb8d526fd470581f5614d32b9f38f), [RpcAddPrinterEx](#Section_3af45717308f4b4f8428430d12856110), [RpcOpenPrinter](#Section_989357e2446e4872bb381dce21e1313f), or [RpcOpenPrinterEx](#Section_0e81ce1872b146c38584a205393b04ff) methods.

**pFormInfoContainer:** This parameter MUST adhere to the parameter specification in [FORM\_CONTAINER Parameters (section 3.1.4.1.8.4)](#Section_edeaada070ae47e3bcf50d0f476902ee).

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate the following:

* Perform the validation steps that are specified in [PRINTER\_HANDLE Parameters (section 3.1.4.1.11)](#Section_10bfa8c2d4564ad0912de674766a36c8).
* Perform the validation steps that are specified in FORM\_CONTAINER Parameters (section 3.1.4.1.8.4).
* Verify that the form does not already exist, and if that verification fails, return ERROR\_FILE\_EXISTS, as specified in [MS-ERREF].
* Additional validation MAY[<403>](#Appendix_A_403) be performed.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Create the form object.
* If any clients have registered for notification of server object changes, a notification MUST be broadcast to them.
* Return the status of the operation.

##### RpcDeleteForm (Opnum 31)

RpcDeleteForm removes a form name from the list of supported forms.

1. DWORD RpcDeleteForm(
2. [in] PRINTER\_HANDLE hPrinter,
3. [in, string] wchar\_t\* pFormName
4. );

**hPrinter:** This parameter MUST specify a handle to a printer object or server object that MUST have been opened by using the [RpcAddPrinter](#Section_bd0bb8d526fd470581f5614d32b9f38f), [RpcAddPrinterEx](#Section_3af45717308f4b4f8428430d12856110), [RpcOpenPrinter](#Section_989357e2446e4872bb381dce21e1313f), or [RpcOpenPrinterEx](#Section_0e81ce1872b146c38584a205393b04ff) methods.

**pFormName:** This parameter MUST be a non-NULL pointer to a string that MUST identify the form to delete. For rules governing form names, see section [2.2.4.5](#Section_cd213afb1b674479ac15316ea0fb537e).

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate the following:

* Perform the validation steps that are specified in [PRINTER\_HANDLE Parameters (section 3.1.4.1.11)](#Section_10bfa8c2d4564ad0912de674766a36c8).
* Verify that the *pFormName* parameter points to a string that identifies an existing form. If that verification fails, return ERROR\_INVALID\_FORM\_NAME, as specified in [MS-ERREF].
* Additional validation MAY[<404>](#Appendix_A_404) be performed.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Clear the references to this form from any other data structures.
* Delete the form object.
* If any clients have registered for notification of server object changes, a notification MUST be broadcast to them.
* Return the status of the operation.

##### RpcGetForm (Opnum 32)

RpcGetForm retrieves information about a specified form.

1. DWORD RpcGetForm(
2. [in] PRINTER\_HANDLE hPrinter,
3. [in, string] wchar\_t\* pFormName,
4. [in] DWORD Level,
5. [in, out, unique, size\_is(cbBuf), disable\_consistency\_check]
6. BYTE\* pForm,
7. [in] DWORD cbBuf,
8. [out] DWORD\* pcbNeeded
9. );

**hPrinter:** A handle to a printer object or server object that was opened by [RpcAddPrinter (section 3.1.4.2.3)](#Section_bd0bb8d526fd470581f5614d32b9f38f), [RpcAddPrinterEx (section 3.1.4.2.15)](#Section_3af45717308f4b4f8428430d12856110), [RpcOpenPrinter (section 3.1.4.2.2)](#Section_989357e2446e4872bb381dce21e1313f), or [RpcOpenPrinterEx (section 3.1.4.2.14)](#Section_0e81ce1872b146c38584a205393b04ff).

**pFormName:** A pointer to a string that specifies the form name for which data is required. For rules governing form names, see section [2.2.4.5](#Section_cd213afb1b674479ac15316ea0fb537e).

**Level:** The form information level.

| Value | Meaning |
| --- | --- |
| 0x00000001 | Corresponds to [\_FORM\_INFO\_1 (section 2.2.2.5.1)](#Section_95e918defbdb4d0aad4cd7005ab33305). |
| 0x00000002 | Corresponds to [\_FORM\_INFO\_2 (section 2.2.2.5.2)](#Section_b15260d2493c49ac9062720162e1aee1). |

**pForm:** A pointer to the [BUFFER](#Section_f29e9057df2940d49c51c615c0da62a1), as specified in INFO Structures Query Parameters (section 3.1.4.1.9).

**BUFFER TYPE**: \_FORM\_INFO.

This parameter can be NULL if *cbBuf* equals zero.

**cbBuf:** Specified in INFO Structures Query Parameters.

**pcbNeeded:** Specified in INFO Structures Query Parameters.

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate parameters as follows:

* Perform the validation steps that are specified in [PRINTER\_HANDLE Parameters (section 3.1.4.1.11)](#Section_10bfa8c2d4564ad0912de674766a36c8). This method SHOULD assume that the handle to the printer or server object can be used without further access checks.
* Verify that the *pFormName* parameter points to a string that identifies an existing form. If that verification fails, return ERROR\_INVALID\_FORM\_NAME, as specified in [MS-ERREF].
* Perform the validation steps specified in INFO Structures Query Parameters.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Using information about the form, perform the processing and response steps specified in INFO Structures Query Parameters.
* Return the status of the operation.

##### RpcSetForm (Opnum 33)

RpcSetForm replaces the form information for the specified form.

1. DWORD RpcSetForm(
2. [in] PRINTER\_HANDLE hPrinter,
3. [in, string] wchar\_t\* pFormName,
4. [in] FORM\_CONTAINER\* pFormInfoContainer
5. );

**hPrinter:** A handle to a printer object or server object that was opened by [RpcAddPrinter (section 3.1.4.2.3)](#Section_bd0bb8d526fd470581f5614d32b9f38f), [RpcAddPrinterEx (section 3.1.4.2.15)](#Section_3af45717308f4b4f8428430d12856110), [RpcOpenPrinter (section 3.1.4.2.2)](#Section_989357e2446e4872bb381dce21e1313f), or [RpcOpenPrinterEx (section 3.1.4.2.14)](#Section_0e81ce1872b146c38584a205393b04ff).

**pFormName:** A pointer to a string that specifies the form name on which the form information is set. For rules governing form names, see section [2.2.4.5](#Section_cd213afb1b674479ac15316ea0fb537e).

**pFormInfoContainer:** A parameter specified in [FORM\_CONTAINER Parameters](#Section_edeaada070ae47e3bcf50d0f476902ee), section 3.1.4.1.8.4.

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate parameters as follows:

* Perform the validation steps specified in [PRINTER\_HANDLE Parameters](#Section_10bfa8c2d4564ad0912de674766a36c8), section 3.1.4.1.11.
* Verify that the *pFormName* parameter points to a string that identifies an existing form. If that verification fails, return ERROR\_INVALID\_FORM\_NAME, as specified in [MS-ERREF].
* Perform the validation steps that are specified in FORM\_CONTAINER Parameters, section 3.1.4.1.8.4.
* Additional validation MAY[<405>](#Appendix_A_405) be performed.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Modify the object that corresponds to *pFormName* in order to reflect the new settings supplied in *pFormInfoContainer*.
* If any clients have registered for notification of server object changes, a notification MUST be broadcast to them.
* Return the status of the operation.

##### RpcEnumForms (Opnum 34)

The RpcEnumForms method enumerates the forms that the specified printer supports.

1. DWORD RpcEnumForms(
2. [in] PRINTER\_HANDLE hPrinter,
3. [in] DWORD Level,
4. [in, out, unique, size\_is(cbBuf), disable\_consistency\_check]
5. BYTE\* pForm,
6. [in] DWORD cbBuf,
7. [out] DWORD\* pcbNeeded,
8. [out] DWORD\* pcReturned
9. );

**hPrinter:** This parameter MUST specify a handle to a printer object or server object that MUST have been opened by using the [RpcAddPrinter](#Section_bd0bb8d526fd470581f5614d32b9f38f), [RpcAddPrinterEx](#Section_3af45717308f4b4f8428430d12856110), [RpcOpenPrinter](#Section_989357e2446e4872bb381dce21e1313f), or [RpcOpenPrinterEx](#Section_0e81ce1872b146c38584a205393b04ff) methods.

**Level:** This value refers to the level of form information structure, as follows.

| Value | Meaning |
| --- | --- |
| 0x00000001 | Corresponds to [\_FORM\_INFO\_1](#Section_95e918defbdb4d0aad4cd7005ab33305), specified in section 2.2.2.5.1. |
| 0x00000002 | Corresponds to [\_FORM\_INFO\_2](#Section_b15260d2493c49ac9062720162e1aee1), specified in section 2.2.2.5.2. |

**pForm:** This parameter MAY be NULL if *cbBuf* equals zero; otherwise, it MUST be a non-NULL pointer to the [BUFFER](#Section_f29e9057df2940d49c51c615c0da62a1), as specified in INFO Structures Query Parameters, section 3.1.4.1.9.

**BUFFER TYPE**: \_FORM\_INFO.

**cbBuf:** This parameter MUST adhere to the parameter specification in INFO Structures Query Parameters, section 3.1.4.1.9.

**pcbNeeded:** This parameter MUST adhere to the parameter specification in INFO Structures Query Parameters, section 3.1.4.1.9.

**pcReturned:** This parameter MUST adhere to the parameter specification in INFO Structures Query Parameters, section 3.1.4.1.9.

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate parameters as follows:

* Perform the validation steps that are specified in [PRINTER\_HANDLE Parameters](#Section_10bfa8c2d4564ad0912de674766a36c8), section 3.1.4.1.11.
* Perform the validation steps that are specified in INFO Structures Query Parameters, section 3.1.4.1.9.
* The server MUST NOT perform access checks on the *hPrinter* object.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Enumerate all forms on the printer or print server.
* Using the enumerated objects, perform the processing and response steps specified in INFO Structures Query Parameters, section 3.1.4.1.9.
* Return the status of the operation.

#### Port Management Methods

This section specifies methods for discovering and communicating with printer ports.

| Method | Description |
| --- | --- |
| [RpcEnumPorts](#Section_4f018733e78e4088ac48f6e19ee00863) | RpcEnumPorts enumerates the ports that are available for printing on a specified server. |
| [RpcDeletePort](#Section_cb13f514f95a4d00936bf934641b6a4b) | Removes a port added by the [RpcAddPortEx](#Section_1303a48b47724226a8b8c495d255304e) method (see section 3.1.4.6.3). |
| RpcAddPortEx | RpcAddPortEx adds a port name to the list of supported ports. |
| [RpcSetPort](#Section_b0a0984e6c724dc8b8c9db98276c94cd) | RpcSetPort sets the status associated with a printer port. |
| [RpcXcvData](#Section_bf9d2351aac0481e9d3705eb5cc961da) | RpcXcvData provides an extensible mechanism by which a client can control ports on the server and exchange port-specific commands and data with the server. For more information about language monitor methods, see section [3.1.4.11](#Section_07ce205123c64d508f55c25d7ac09cb3). |

##### RpcEnumPorts (Opnum 35)

RpcEnumPorts enumerates the ports that are available for printing on a specified server.

1. DWORD RpcEnumPorts(
2. [in, string, unique] STRING\_HANDLE pName,
3. [in] DWORD Level,
4. [in, out, unique, size\_is(cbBuf), disable\_consistency\_check]
5. BYTE\* pPort,
6. [in] DWORD cbBuf,
7. [out] DWORD\* pcbNeeded,
8. [out] DWORD\* pcReturned
9. );

**pName:** A parameter specified in [Print Server Name Parameters](#Section_acb355a4913b4abb8cb5415aea160015), section 3.1.4.1.4.

**Level:** The port information level.

| Value | Meaning |
| --- | --- |
| 0x00000001 | Corresponds to [\_PORT\_INFO\_1](#Section_086847b9a37041499ceda41ecb436d3b), specified in section 2.2.2.8.1. |
| 0x00000002 | Corresponds to [\_PORT\_INFO\_2](#Section_d09458b84d2b48f2b7500d07ea24f47a), specified in section [2.2.1.9.2](#Section_7a923364dffc40b3bca5bf2eed610f31). |

**pPort:** A pointer to the [BUFFER](#Section_f29e9057df2940d49c51c615c0da62a1), as specified in INFO Structures Query Parameters, section 3.1.4.1.9.

**BUFFER TYPE**: \_PORT\_INFO.

This parameter can be NULL if *cbBuf* equals zero.

**cbBuf:** A parameter specified in INFO Structures Query Parameters, section 3.1.4.1.9.

**pcbNeeded:** A parameter specified in INFO Structures Query Parameters, section 3.1.4.1.9.

**pcReturned:** A parameter specified in INFO Structures Query Parameters, section 3.1.4.1.9.

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate parameters as follows:

* Perform the validation steps specified in Print Server Name Parameters, section 3.1.4.1.4.
* Perform the validation steps specified in INFO Structures Query Parameters, section 3.1.4.1.9.
* Additional validation MAY[<406>](#Appendix_A_406) be performed.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Enumerate all ports on the print server.
* Using the enumerated objects, perform the processing and response steps specified in INFO Structures Query Parameters, section 3.1.4.1.9.
* Return the status of the operation.

##### RpcDeletePort (Opnum 39)

Removes a port.[<407>](#Appendix_A_407)

1. DWORD RpcDeletePort(
2. [in, string, unique] STRING\_HANDLE pName,
3. [in] ULONG\_PTR hWnd,
4. [in, string] wchar\_t\* pPortName
5. );

**pName:** A parameter specified in [Print Server Name Parameters (section 3.1.4.1.4)](#Section_acb355a4913b4abb8cb5415aea160015).

**hWnd:** The value of this parameter SHOULD be set to zero when sent and MUST be ignored on receipt.

**pPortName:** A pointer to a string that specifies the name of the port that will be deleted. For rules governing port names, see section [2.2.4.10](#Section_03c5feb74fbb4569b0649eb6ad833f50).

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate the following:

* Perform the validation steps specified in Print Server Name Parameters (section 3.1.4.1.4).
* Verify that the string that is referenced by *pPortName* represents a port installed on the server, and if that validation fails, return ERROR\_UNKNOWN\_PORT.
* Verify that the port monitor for the port supports the **DeletePort** [**monitor module**](#gt_971b7035-4384-4e85-9878-b88c49aef05e) method, and if that validation fails, return ERROR\_NOT\_SUPPORTED.
* Verify that the port is not being used by any printer or print job in the system, and if that validation fails, return ERROR\_BUSY.
* Additional validation MAY[<408>](#Appendix_A_408) be performed.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Clear the references to this port from any other data structures.
* Delete the port object.
* Modify the list of ports in the system to exclude the deleted port.
* If any clients have registered for notification of server object changes, a notification MUST be broadcast to them.
* Return the status of the operation.

##### RpcAddPortEx (Opnum 61)

RpcAddPortEx adds a port name to the list of supported ports.[<409>](#Appendix_A_409)

1. DWORD RpcAddPortEx(
2. [in, string, unique] STRING\_HANDLE pName,
3. [in] PORT\_CONTAINER\* pPortContainer,
4. [in] PORT\_VAR\_CONTAINER\* pPortVarContainer,
5. [in, string] wchar\_t\* pMonitorName
6. );

**pName:** A parameter specified in [Print Server Name Parameters (section 3.1.4.1.4)](#Section_acb355a4913b4abb8cb5415aea160015).

**pPortContainer:** A parameter specified in [PORT\_CONTAINER Parameters (section 3.1.4.1.8.5)](#Section_fff791a0fcab420ab1b1be6eca61d301). The value of the **Level** member in the [PORT\_CONTAINER](#Section_e05adf6ca04e4e398b34e7c118ce2bc8) that is referenced by this parameter MUST be 0x00000001 or 0xFFFFFFFF.

**pPortVarContainer:** A pointer to a [PORT\_VAR\_CONTAINER (section 2.2.1.2.8)](#Section_0cf200e2c00a4c538cb7c150c44cbdbd) information structure that contains information about the port.

**pMonitorName:** A pointer to a string that specifies the monitor associated with the port. For rules governing monitor names, see section [2.2.4.8](#Section_808e2e6c7f914bdd92da56fbca513404).

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate parameters as follows:

* Perform the validation steps specified in Print Server Name Parameters (section 3.1.4.1.4).
* Perform the validation steps specified in PORT\_CONTAINER Parameters (section 3.1.4.1.8.5).
* If the value of the **Level** member of the PORT\_CONTAINER that is referenced by the *pPortContainer* parameter is 0xFFFFFFFF, verify that the *pPortVarContainer* parameter is not NULL; if that verification fails, return ERROR\_INVALID\_PARAMETER, as specified in [MS-ERREF].
* Verify that the port does not already exist, and if that verification fails, return ERROR\_ALREADY\_EXISTS, as specified in [MS-ERREF].
* Verify that the port monitor identified by the *pMonitorName* parameter exists, and if that verification fails, the server MUST return ERROR\_INVALID\_NAME, as specified in [MS-ERREF].
* Verify that the port monitor for the port supports the **AddPort** monitor module method, and if that validation fails, return ERROR\_INVALID\_PARAMETER.
* Additional validation MAY[<410>](#Appendix_A_410) be performed.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Create a new port that is compatible with the port monitor identified by the string that is referenced by the *pMonitorName* parameter; if the **Level** member of the PORT\_CONTAINER is 0xFFFFFFFF, pass the data that is contained in the PORT\_VAR\_CONTAINER that is pointed to by the *pPortVarContainer* parameter to the port monitor.
* Associate the new port with the port monitor identified by the string that is referenced by the *pMonitorName* parameter.
* Modify the list of ports in the system to include the port created by this method.
* If any clients have registered for notification of server object changes, a notification MUST be broadcast to them.
* Return the status of the operation.

##### RpcSetPort (Opnum 71)

RpcSetPort sets the status associated with a printer port.[<411>](#Appendix_A_411)

1. DWORD RpcSetPort(
2. [in, string, unique] STRING\_HANDLE pName,
3. [in, string, unique] wchar\_t\* pPortName,
4. [in] PORT\_CONTAINER\* pPortContainer
5. );

**pName:** A parameter that adheres to the specification in [Print Server Name Parameters (section 3.1.4.1.4)](#Section_acb355a4913b4abb8cb5415aea160015).

**pPortName:** A pointer to a string that specifies the name of the printer port. For rules governing port names, see section [2.2.4.10](#Section_03c5feb74fbb4569b0649eb6ad833f50).

**pPortContainer:** A parameter that adheres to the specification in [PORT\_CONTAINER Parameters (section 3.1.4.1.8.5)](#Section_fff791a0fcab420ab1b1be6eca61d301). The level as specified in the **Level** member of the [PORT\_CONTAINER](#Section_e05adf6ca04e4e398b34e7c118ce2bc8) structure MUST be 0x00000003.

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate parameters as follows:

* Verify that the string that is referenced by *pPortName* represents a port installed on the server, and if validation fails, the server MUST fail the operation immediately and return ERROR\_UNKNOWN\_PORT.
* Perform the validation steps that are specified in Print Server Name Parameters.
* Perform the validation steps that are specified in PORT\_CONTAINER Parameters.
* Additional validation MAY[<412>](#Appendix_A_412) be performed.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* The *pPortContainer* parameter points to a PORT\_CONTAINER structure. This structure references a [PORT\_INFO](#Section_05c1a8d20bcb4292bc65c8cc2e7f90cf) structure that contains members **dwStatus**, **pszStatus**, and **dwSeverity**. Copy these members to the port object that is referenced by the *pPortName* parameter.
* Return the status of the operation.

##### RpcXcvData (Opnum 88)

RpcXcvData provides an extensible mechanism by which a client can control ports on the server and exchange port specific commands and data with the server.[<413>](#Appendix_A_413)

1. DWORD RpcXcvData(
2. [in] PRINTER\_HANDLE hXcv,
3. [in, string] const wchar\_t\* pszDataName,
4. [in, size\_is(cbInputData)] BYTE\* pInputData,
5. [in] DWORD cbInputData,
6. [out, size\_is(cbOutputData)] BYTE\* pOutputData,
7. [in] DWORD cbOutputData,
8. [out] DWORD\* pcbOutputNeeded,
9. [in, out] DWORD\* pdwStatus
10. );

**hXcv:** A handle to a port or port monitor object that was opened by [RpcOpenPrinter (section 3.1.4.2.2)](#Section_989357e2446e4872bb381dce21e1313f) or [RpcOpenPrinterEx (section 3.1.4.2.14)](#Section_0e81ce1872b146c38584a205393b04ff).

**pszDataName:** A pointer to a string representing the name of the requested data or action. The following table shows the actions that SHOULD be supported. Other port monitor–specific action strings MAY be supported.[<414>](#Appendix_A_414)

| Value | Meaning |
| --- | --- |
| "AddPort" | Add an instance of a specific port type controlled by the port monitor. |
| "DeletePort" | Delete an instance of a specific port type controlled by the port monitor. |
| "MonitorUI" | The action returns the name of the associated port monitor client-side executable configuration module in the buffer that is referenced by the *pOutputData* parameter. |

**pInputData:** A pointer to a buffer that contains input data. This parameter can be NULL if *cbInputData* equals zero.

**cbInputData:** The size, in bytes, of the buffer pointed to by the *pInputData* parameter.

**pOutputData:** A pointer to a buffer to receive output data. This parameter can be NULL if *cbOutputData* equals zero.

**cbOutputData:** The size, in bytes, of the buffer pointed to by the *pOutputData* parameter.

**pcbOutputNeeded:** A pointer to a location that receives the size, in bytes, required for the buffer pointed to by the *pOutputData* parameter.

**pdwStatus:** A pointer to a variable that receives the status value that is returned by the port monitor's XcvData method. The value MUST be zero to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate that the print server successfully called the port monitor's XcvData method, or a nonzero Windows error code to indicate failure, as specified in [MS-ERREF].

Upon receiving this message, the server MUST validate parameters as follows:

* Perform the validation steps specified in [PRINTER\_HANDLE Parameters](#Section_10bfa8c2d4564ad0912de674766a36c8), section 3.1.4.1.11, substituting *hXcv* for *hPrinter*. This method assumes that the handle to the port object can be used without further access checks.

The print server SHOULD[<415>](#Appendix_A_415) further validate parameters as follows:

* Verify that the string referenced by the *pszDataName* parameter is a valid command name, and if that verification fails, return ERROR\_INVALID\_PARAMETER.
* Verify that *pdwStatus* is not NULL, and if that verification fails, return ERROR\_INVALID\_PARAMETER.
* If the *pszDataName* parameter is set to "AddPort", verify that the *pInputData* parameter is not set to NULL, verify that the value of the *cbInputData* parameter is not zero and that the *pInputData* parameter contains a null terminated character string, and if those verifications fail, return ERROR\_INVALID\_DATA.
* If the action returns information in *pOutputData* and *cbOutputData* is not zero, verify that the value of the *pOutputData* parameter is not NULL, and if that verification fails, return ERROR\_INVALID\_PARAMETER.
* For actions that return data in the buffer that is pointed to by the *pOutputData* parameter, verify that the size of the buffer, as specified by the value of the *cbOutputData* parameter, is sufficient. If that verification fails, store the required buffer size in the variable that is pointed to by the *pcbOutputNeeded* parameter and return ERROR\_INSUFFICIENT\_BUFFER, as specified in [MS-ERREF].

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Load the executable object of the monitor that supports the port that is associated with the port object that is referenced by the *hXcv* parameter.
* Invoke the **XcvData** method in that library, passing in the *pszDataName*, *pInputData*, *cbInputData*, *pOutputData*, *cbOutputData*, *pcbOutputNeeded*, and *pdwStatus* parameters.
* Return the status from the **XcvData** method.

The port monitor's **XcvData** method processes the message by performing the operation indicated by the string pointed to by the *pszDataName* parameter, and then composes a response as follows:

* If the *pOutputData* parameter is not NULL, copy the output data from the method into *pOutputData*, up to the limit that is specified by the value of the *cbOutputData* parameter.
* Write the size of the data that was copied into *pcbOutputNeeded*.
* If the *pdwStatus* parameter is not NULL, store the status of the port in the variable that is referenced by the *pdwStatus* parameter.
* Return the status of the operation.

#### Port Monitor Management Methods

This section specifies methods for discovering and installing port monitors.

| Method | Description |
| --- | --- |
| [RpcEnumMonitors](#Section_6c5836fed9ea41a0ad7e01152f39e8a1) | The RpcEnumMonitors method retrieves information about the port monitors installed on the specified server.  Opnum 36 |
| [RpcAddMonitor](#Section_dee4feea274940bdab7cc1c0ca1fbcf8) | RpcAddMonitor installs a local port monitor and links the configuration, data, and monitor files.  Opnum 46 |
| [RpcDeleteMonitor](#Section_2483519c230c4e8bb52d29677d941ad8) | RpcDeleteMonitor removes a port monitor.  Opnum 47 |

##### RpcEnumMonitors (Opnum 36)

The RpcEnumMonitors method retrieves information about the port monitors installed on the specified server.

1. DWORD RpcEnumMonitors(
2. [in, string, unique] STRING\_HANDLE pName,
3. [in] DWORD Level,
4. [in, out, unique, size\_is(cbBuf), disable\_consistency\_check]
5. BYTE\* pMonitor,
6. [in] DWORD cbBuf,
7. [out] DWORD\* pcbNeeded,
8. [out] DWORD\* pcReturned
9. );

**pName:** This parameter MUST adhere to the parameter specification in [Print Server Name Parameters](#Section_acb355a4913b4abb8cb5415aea160015), section 3.1.4.1.4.

**Level:** This value refers to the level of port monitor information structure, as follows.

| Value | Meaning |
| --- | --- |
| 0x00000001 | Corresponds to [\_MONITOR\_INFO\_1](#Section_0496f2b4c42c4363bb032a5df819f7ba), specified in section 2.2.2.7.1. |
| 0x00000002 | Corresponds to [\_MONITOR\_INFO\_2](#Section_a69e678518be483a90f97a283df84fbf), specified in section 2.2.2.7.2. |

**pMonitor:** This parameter SHOULD be ignored if *cbBuf* equals zero; otherwise, it is a pointer to the [BUFFER](#Section_f29e9057df2940d49c51c615c0da62a1), as specified in INFO Structures Query Parameters, section 3.1.4.1.9.

**BUFFER TYPE**: \_MONITOR\_INFO.

**cbBuf:** This parameter MUST adhere to the parameter specification in INFO Structures Query Parameters, section 3.1.4.1.9.

**pcbNeeded:** This parameter MUST adhere to the parameter specification in INFO Structures Query Parameters, section 3.1.4.1.9.

**pcReturned:** This parameter MUST adhere to the parameter specification in INFO Structures Query Parameters, section 3.1.4.1.9.

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate parameters as follows:

* Perform the validation steps that are specified in Print Server Name Parameters, section 3.1.4.1.4.
* Perform the validation steps that are specified in INFO Structures Query Parameters, section 3.1.4.1.9.
* Additional validation MAY[<416>](#Appendix_A_416) be performed.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Enumerate all port monitors supporting the methods listed as required or optional in section [Monitor Module Methods (section 3.1.4.11)](#Section_07ce205123c64d508f55c25d7ac09cb3) [<417>](#Appendix_A_417) on the print server.
* Using the enumerated objects, perform the processing and response steps specified in INFO Structures Query Parameters, section 3.1.4.1.9.
* Return the status of the operation.

##### RpcAddMonitor (Opnum 46)

RpcAddMonitor installs a local port monitor and links the configuration, data, and monitor files.

1. DWORD RpcAddMonitor(
2. [in, string, unique] STRING\_HANDLE Name,
3. [in] MONITOR\_CONTAINER\* pMonitorContainer
4. );

**Name:** A parameter that adheres to the specification in [Print Server Name Parameters (section 3.1.4.1.4)](#Section_acb355a4913b4abb8cb5415aea160015).

**pMonitorContainer:** A parameter that adheres to the specification in [MONITOR\_CONTAINER Parameters (section 3.1.4.1.8.9)](#Section_81865d2ce10843eeae34145233434946). The **Level** member of the [MONITOR\_CONTAINER](#Section_b23382165e3e4379a722af2230d06df2) MUST be 0x00000002.

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate parameters as follows:

* Perform the validation steps that are specified in Print Server Name Parameters, section 3.1.4.1.4.
* Perform validation steps as specified in MONITOR\_CONTAINER Parameters (section 3.1.4.1.8.9).
* Verify that the port monitor does not already exist in the system, and if that verification fails, return *ERROR\_PRINT\_MONITOR\_ALREADY\_INSTALLED*, as specified in [MS-ERREF].
* Additional validation MAY[<418>](#Appendix_A_418) be performed.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Create the monitor object.
* Return the status of the operation.

##### RpcDeleteMonitor (Opnum 47)

RpcDeleteMonitor removes a port monitor.

1. DWORD RpcDeleteMonitor(
2. [in, string, unique] STRING\_HANDLE Name,
3. [in, string, unique] wchar\_t\* pEnvironment,
4. [in, string] wchar\_t\* pMonitorName
5. );

**Name:** This parameter MUST adhere to the parameter specification in [Print Server Name Parameters](#Section_acb355a4913b4abb8cb5415aea160015), section 3.1.4.1.4.

**pEnvironment:** This parameter MUST adhere to the parameter specification in [Environment Name Parameters](#Section_5b2bf7d5ac77464fbe5ef3a7c1782246), section 3.1.4.1.3.

**pMonitorName:** This parameter MUST be a non-NULL pointer to a string that MUST specify the name of the monitor to remove. For rules governing monitor names, see section [2.2.4.8](#Section_808e2e6c7f914bdd92da56fbca513404).

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate the following:

* Perform the validation steps that are specified in Print Server Name Parameters, section 3.1.4.1.4.
* Perform the validation steps that are specified in Environment Name Parameters, section 3.1.4.1.3.
* Verify that the *pMonitorName* parameter points to a string identifying a port monitor installed on the system and, if that verification fails, return ERROR\_UNKNOWN\_PRINT\_MONITOR as specified in MS-ERREF section 2.2.
* Verify that there are no ports (in use by any printer or print job) on the system that are controlled by this monitor at this time and, if that verification fails, return ERROR\_PRINT\_MONITOR\_IN\_USE as specified in MS-ERREF section 2.2.
* Additional validation MAY be performed.[<419>](#Appendix_A_419)

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Clear references to the port monitor from any other data structures.
* Delete the port monitor object.
* Return the status of the operation.

#### Print Processor Management Methods

This section specifies methods for discovering and manipulating print processor objects.

| Method | Description |
| --- | --- |
| [RpcAddPrintProcessor](#Section_d7e6fdb25256483592c127aa6f76758a) | RpcAddPrintProcessor installs a print processor on the specified server and adds its name to an internal list of supported print processors.  Opnum 14 |
| [RpcEnumPrintProcessors](#Section_8c3c2db185d84d8894c11e09cda435c7) | RpcEnumPrintProcessors enumerates the print processors installed on a specified server.  Opnum 15 |
| [RpcGetPrintProcessorDirectory](#Section_17eaee3be5754f82855ce60657d5b2c8) | RpcGetPrintProcessorDirectory retrieves the path for the print processor on the specified server.  Opnum 16 |
| [RpcDeletePrintProcessor](#Section_e08179d3d3c3421bab77a0ba50a4254e) | RpcDeletePrintProcessor removes a print processor.  Opnum 48 |
| [RpcEnumPrintProcessorDatatypes](#Section_c4a35c2040824ce6840102c52ee3158e) | RpcEnumPrintProcessorDatatypes enumerates the data types that a specified print processor supports.  Opnum 51 |

##### RpcAddPrintProcessor (Opnum 14)

RpcAddPrintProcessor installs a print processor on the specified server and adds its name to an internal list of supported print processors.

1. DWORD RpcAddPrintProcessor(
2. [in, string, unique] STRING\_HANDLE pName,
3. [in, string] wchar\_t\* pEnvironment,
4. [in, string] wchar\_t\* pPathName,
5. [in, string] wchar\_t\* pPrintProcessorName
6. );

**pName:** This parameter MUST adhere to the parameter specification in [Print Server Name Parameters (section 3.1.4.1.4)](#Section_acb355a4913b4abb8cb5415aea160015).

**pEnvironment:** This parameter MUST adhere to the parameter specification in [Environment Name Parameters (section 3.1.4.1.3)](#Section_5b2bf7d5ac77464fbe5ef3a7c1782246).

**pPathName:** This parameter MUST be a non-NULL pointer to a string that MUST specify the file name of the print processor. For rules governing path names, see section [2.2.4.9](#Section_1d75f622d07c45ccbf1ea6191e408516).

**pPrintProcessorName:** This parameter MUST be a non-NULL pointer to a string that MUST specify the name of the print processor. For rules governing print processor names, see section [2.2.4.11](#Section_807e0bdd21a54a729a6a1a7fb1b08a8e).

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate parameters as follows:

* Perform the validation steps that are specified in Print Server Name Parameters.
* Perform the validation steps that are specified in Environment Name Parameters.
* Verify that the path identified by the string that is referenced by the *pPathName* parameter contains the necessary file for installing the print processor.
* Verify that the print processor to be added does not have the name "winprint", and if that verification fails, return ERROR\_PRINT\_PROCESSOR\_ALREADY\_INSTALLED, as specified in [MS-ERREF].
* Verify that the environment name specified by the *pEnvironment* parameter is not "Windows ARM", and if that verification fails, return ERROR\_NOT\_SUPPORTED, as specified in [MS-ERREF].[<420>](#Appendix_A_420)
* Additional validation MAY[<421>](#Appendix_A_421) be performed.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Copy the print processor file as appropriate to its destination and overwrite an existing print processor with the same name, if necessary.
* Create the print processor object.
* If any clients have registered for notification of server object changes, a notification MUST be broadcast to them.
* Return the status of the operation.

##### RpcEnumPrintProcessors (Opnum 15)

RpcEnumPrintProcessors enumerates the print processors installed on a specified server.

1. DWORD RpcEnumPrintProcessors(
2. [in, string, unique] STRING\_HANDLE pName,
3. [in, string, unique] wchar\_t\* pEnvironment,
4. [in] DWORD Level,
5. [in, out, unique, size\_is(cbBuf), disable\_consistency\_check]
6. BYTE\* pPrintProcessorInfo,
7. [in] DWORD cbBuf,
8. [out] DWORD\* pcbNeeded,
9. [out] DWORD\* pcReturned
10. );

**pName:** A parameter specified in [Print Server Name Parameters](#Section_acb355a4913b4abb8cb5415aea160015), section 3.1.4.1.4.

**pEnvironment:** A parameter specified in [Environment Name Parameters](#Section_5b2bf7d5ac77464fbe5ef3a7c1782246), section 3.1.4.1.3.

**Level:** The information level. This value MUST be 0x00000001.

**pPrintProcessorInfo:** A pointer to [BUFFER](#Section_f29e9057df2940d49c51c615c0da62a1) as specified in INFO Structures Query Parameters, section 3.1.4.1.9

**BUFFER TYPE**: PRINTPROCESSOR\_INFO\_1.

This parameter can be NULL if *cbBuf* equals zero.

**cbBuf:** A parameter specified in INFO Structures Query Parameters, section 3.1.4.1.9.

**pcbNeeded:** A parameter specified in INFO Structures Query Parameters, section 3.1.4.1.9.

**pcReturned:** A parameter specified in INFO Structures Query Parameters, section 3.1.4.1.9.

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate parameters as follows:

* Perform the validation steps specified in Print Server Name Parameters, section 3.1.4.1.4.
* Perform the validation steps specified in Environment Name Parameters, section 3.1.4.1.3.
* Perform the validation steps specified in INFO Structures Query Parameters, section 3.1.4.1.9.
* Additional validation MAY[<422>](#Appendix_A_422) be performed.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Enumerate all print processors on the print server.
* Using the enumerated objects, perform the processing and response steps specified in INFO Structures Query Parameters, section 3.1.4.1.9.
* Return the status of the operation.

##### RpcGetPrintProcessorDirectory (Opnum 16)

RpcGetPrintProcessorDirectory retrieves the path for the print processor on the specified server.

1. DWORD RpcGetPrintProcessorDirectory(
2. [in, string, unique] STRING\_HANDLE pName,
3. [in, string, unique] wchar\_t\* pEnvironment,
4. [in] DWORD Level,
5. [in, out, unique, size\_is(cbBuf), disable\_consistency\_check]
6. BYTE\* pPrintProcessorDirectory,
7. [in] DWORD cbBuf,
8. [out] DWORD\* pcbNeeded
9. );

**pName:** This parameter MUST adhere to the parameter specification in [Print Server Name Parameters](#Section_acb355a4913b4abb8cb5415aea160015), section 3.1.4.1.4.

**pEnvironment:** This parameter MUST adhere to the parameter specification in [Environment Name Parameters](#Section_5b2bf7d5ac77464fbe5ef3a7c1782246), section 3.1.4.1.3.

**Level:** The value of this parameter MUST be 0x00000001.

**pPrintProcessorDirectory:** This parameter MAY be NULL if *cbBuf* equals zero; otherwise, it MUST be a non-NULL pointer to [BUFFER](#Section_6bae59aff92341b9b39513872aaf2ce0) as specified in String Query Parameters, section 3.1.4.1.7.

**cbBuf:** This parameter MUST adhere to the parameter specification in String Query Parameters, section 3.1.4.1.7.

**pcbNeeded:** This parameter MUST adhere to the parameter specification in String Query Parameters, section 3.1.4.1.7.

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate parameters as follows:

* Perform the validation steps that are specified in Print Server Name Parameters, section 3.1.4.1.4.
* Perform the validation steps that are specified in Environment Name Parameters, section 3.1.4.1.3.
* Perform the validation steps that are specified in String Query Parameters, section 3.1.4.1.7.
* Additional validation MAY[<423>](#Appendix_A_423) be performed.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Using the path of the print processor directory on the print server, perform the processing and response steps specified in String Query Parameters, section 3.1.4.1.7.
* Return the status of the operation.

##### RpcDeletePrintProcessor (Opnum 48)

RpcDeletePrintProcessor removes a print processor.

1. DWORD RpcDeletePrintProcessor(
2. [in, string, unique] STRING\_HANDLE Name,
3. [in, string, unique] wchar\_t\* pEnvironment,
4. [in, string] wchar\_t\* pPrintProcessorName
5. );

**Name:** A parameter specified in [Print Server Name Parameters (section 3.1.4.1.4)](#Section_acb355a4913b4abb8cb5415aea160015).

**pEnvironment:** A parameter specified in [Environment Name Parameters (section 3.1.4.1.3)](#Section_5b2bf7d5ac77464fbe5ef3a7c1782246).

**pPrintProcessorName:** A pointer to a string that specifies the name of the print processor that will be removed. For rules governing print processor names, see [Print Processor Names (section 2.2.4.11)](#Section_807e0bdd21a54a729a6a1a7fb1b08a8e).

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate the following:

* Perform the validation steps specified in Print Server Name Parameters.
* Perform the validation steps specified in Environment Name Parameters.
* The string that is referenced by the *pPrintProcessorName* parameter identifies a print processor installed on the server.
* Verify that there are no printers on the system that use the print processor at this time, and if that verification fails, return ERROR\_CAN\_NOT\_COMPLETE.
* Additional validation MAY[<424>](#Appendix_A_424) be performed.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Clear all references to the specified print processor from any other data structures.
* Delete the print processor object.
* If any clients have registered for notification of server object changes, a notification MUST be broadcast to them.
* Return the status of the operation.

##### RpcEnumPrintProcessorDatatypes (Opnum 51)

RpcEnumPrintProcessorDatatypes enumerates the data types that a specified print processor supports.

1. DWORD RpcEnumPrintProcessorDatatypes(
2. [in, string, unique] STRING\_HANDLE pName,
3. [in, string, unique] wchar\_t\* pPrintProcessorName,
4. [in] DWORD Level,
5. [in, out, unique, size\_is(cbBuf), disable\_consistency\_check]
6. BYTE\* pDatatypes,
7. [in] DWORD cbBuf,
8. [out] DWORD\* pcbNeeded,
9. [out] DWORD\* pcReturned
10. );

**pName:** This parameter MUST adhere to the parameter specification in [Print Server Name Parameters](#Section_acb355a4913b4abb8cb5415aea160015), section 3.1.4.1.4.

**pPrintProcessorName:** This parameter MUST be a non-NULL pointer to a string that MUST specify the name of the print processor whose data types MUST be enumerated. For rules governing print processor names, see section [2.2.4.11](#Section_807e0bdd21a54a729a6a1a7fb1b08a8e).

**Level:** The value of this parameter MUST be 0x00000001.

**pDatatypes:** This parameter MAY be NULL if *cbBuf* equals zero; otherwise, it MUST be a non-NULL pointer to [BUFFER](#Section_f29e9057df2940d49c51c615c0da62a1) as specified in INFO Structures Query Parameters, section 3.1.4.1.9.

**BUFFER TYPE:** \_DATATYPES\_INFO\_1

**cbBuf:** This parameter MUST adhere to the parameter specification in INFO Structures Query Parameters, section 3.1.4.1.9.

**pcbNeeded:** This parameter MUST adhere to the parameter specification in INFO Structures Query Parameters, section 3.1.4.1.9.

**pcReturned:** This parameter MUST adhere to the parameter specification in INFO Structures Query Parameters, section 3.1.4.1.9.

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate parameters as follows:

* Perform the validation steps that are specified in Print Server Name Parameters, section 3.1.4.1.4.
* Verify that the print processor that is identified by the string that is referenced by the *pPrintProcessorName* parameter is in the list of print processors.
* Perform the validation steps that are specified in INFO Structures Query Parameters, section 3.1.4.1.9.
* Additional validation MAY[<425>](#Appendix_A_425) be performed.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Enumerate all data types that are supported by the specified print processor.
* Using the enumerated objects, perform the processing and response steps specified in INFO Structures Query Parameters, section 3.1.4.1.9.
* Return the status of the operation.

#### Document Printing Methods

This section specifies methods for printing documents, pages, and data.

| Method | Description |
| --- | --- |
| [RpcStartDocPrinter](#Section_90f95ff03ae04ddea83d35f457c710a6) | RpcStartDocPrinter notifies the print spooler that a document is being spooled for printing.  Opnum 17 |
| [RpcStartPagePrinter](#Section_76261f544c634fc2a2dabaee584e076b) | RpcStartPagePrinter notifies the spooler that a page is about to be printed on the specified printer.  Opnum 18 |
| [RpcWritePrinter](#Section_6dd263aac8b547748d211daa0d0d9920) | RpcWritePrinter sends data to the print spooler.  Opnum 19 |
| [RpcEndPagePrinter](#Section_c4a17128584d402e9bac6a050e9da6d0) | RpcEndPagePrinter notifies the print spooler that the application is at the end of a page in a print job.  Opnum 20 |
| [RpcAbortPrinter](#Section_924cd448b8b74dbaaa56d9d3244cd386) | RpcAbortPrinter aborts the currently spooling print document.  Opnum 21 |
| [RpcReadPrinter](#Section_58fc87857f004f00bc784c3b438acbb7) | RpcReadPrinter retrieves data from the specified printer.  Opnum 22 |
| [RpcEndDocPrinter](#Section_209fcf3205b444aba3a4779cfae181ca) | RpcEndDocPrinter notifies the print spooler that the application is at the end of the current print job.  Opnum 23 |
| [RpcFlushPrinter](#Section_be7ec83d52eb42e9bc5d7244e5b44b94) | RpcFlushPrinter is used by the printer driver to send a buffer of bytes to the port to cleanly abort a print job. It also allows delaying the I/O line to the printer.  Opnum 96 |

##### RpcStartDocPrinter (Opnum 17)

RpcStartDocPrinter notifies the print server that a document is being spooled for printing.

1. DWORD RpcStartDocPrinter(
2. [in] PRINTER\_HANDLE hPrinter,
3. [in] DOC\_INFO\_CONTAINER\* pDocInfoContainer,
4. [out] DWORD\* pJobId
5. );

**hPrinter:** A handle to a printer object or port object that was opened by [RpcAddPrinter (section 3.1.4.2.3)](#Section_bd0bb8d526fd470581f5614d32b9f38f), [RpcAddPrinterEx (section 3.1.4.2.15)](#Section_3af45717308f4b4f8428430d12856110), [RpcOpenPrinter (section 3.1.4.2.2)](#Section_989357e2446e4872bb381dce21e1313f), or [RpcOpenPrinterEx (section 3.1.4.2.14)](#Section_0e81ce1872b146c38584a205393b04ff). The printer handle MUST NOT be in use for printing another document at the time of this call.

**pDocInfoContainer:** A parameter specified in [DOC\_INFO\_CONTAINER Parameters (section 3.1.4.1.8.2)](#Section_35a96c739f4f4280ae578e4f72834929).

**pJobId:** A pointer to a variable that receives a nonzero print job identifier. The job MUST be created with an identifier that is unique for this printer.

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate the following:

* Perform the validation steps that are specified in [PRINTER\_HANDLE Parameters (section 3.1.4.1.11)](#Section_10bfa8c2d4564ad0912de674766a36c8). This method SHOULD assume that the handle to the printer or port object can be used without further access checks.
* Perform the validation steps that are specified in DOC\_INFO\_CONTAINER Parameters.
* The server MUST verify that RpcStartDocPrinter does not get called twice for a given printer or port object without an intervening call to [RpcEndDocPrinter (section 3.1.4.9.7)](#Section_209fcf3205b444aba3a4779cfae181ca). If that verification fails, return ERROR\_INVALID\_HANDLE.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Create the job object and associate it with a job queue.
  + If *hPrinter* is a printer object handle, associate the job with the job queue of that printer object.
  + If *hPrinter* is a port object handle, select an arbitrary printer object associated with that port object and associate the job with the job queue of that printer object.
* Associate a data type with the print job.
  + If the **pDatatype** member of the [DOC\_INFO\_1](#Section_20f7b510e6a04f6c97a05630d1e70879) structure that is pointed to by the **pDocInfo1** pointer in the [DOC\_INFO\_CONTAINER](#Section_c4f39f63301745bf8b8a81ab9458f052) and pointed to by *pDocInfoContainer* is non-NULL, then use the data type specified in **pDatatype** as the data type for the print job.
  + If the **pDatatype** member of the DOC\_INFO\_1 structure is NULL, *hPrinter* is a printer object handle, and the data type contained in the context for *hPrinter* is non-NULL, then use the data type specified by the context for *hPrinter* for the print job.
  + If the **pDatatype** member of the DOC\_INFO\_1 structure is NULL and the context for *hPrinter* does not contain a data type, then use the printer's default data type for the print job.
* Write the ID of the created job in the variable that is pointed to by the *pJobId* parameter.
* If any clients that have registered for notification of the job object creation, a notification MUST be broadcast to them.
* Return the status of the operation.

##### RpcStartPagePrinter (Opnum 18)

RpcStartPagePrinter notifies the spooler that a page is about to be printed on the specified printer.

1. DWORD RpcStartPagePrinter(
2. [in] PRINTER\_HANDLE hPrinter
3. );

**hPrinter:** A handle to a printer object or port object that was opened by [RpcAddPrinter (section 3.1.4.2.3)](#Section_bd0bb8d526fd470581f5614d32b9f38f), [RpcAddPrinterEx (section 3.1.4.2.15)](#Section_3af45717308f4b4f8428430d12856110), [RpcOpenPrinter (section 3.1.4.2.2)](#Section_989357e2446e4872bb381dce21e1313f), or [RpcOpenPrinterEx (section 3.1.4.2.14)](#Section_0e81ce1872b146c38584a205393b04ff).

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate the following:

* Perform the validation steps that are specified in [PRINTER\_HANDLE Parameters (section 3.1.4.1.11)](#Section_10bfa8c2d4564ad0912de674766a36c8). This method SHOULD assume that the handle to the printer or port object can be used without further access checks.
* Verify that a job has been associated with *hPrinter* using [RpcStartDocPrinter](#Section_90f95ff03ae04ddea83d35f457c710a6), and if that verification fails, return ERROR\_SPL\_NO\_STARTDOC, as specified in [MS-ERREF].
* Verify that printing of the job has not been canceled, and if that verification fails, return ERROR\_PRINT\_CANCELLED, as specified in [MS-ERREF].

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Update print job statistics to reflect incremented page count.
* Return the status of the operation.

For each page of a print job, a print client SHOULD perform the following sequence of actions:

* call RpcStartPagePrinter.
* call [RpcWritePrinter (section 3.1.4.9.3)](#Section_6dd263aac8b547748d211daa0d0d9920) zero or more times.
* call [RpcEndPagePrinter (section 3.1.4.9.4)](#Section_c4a17128584d402e9bac6a050e9da6d0).

The print server SHOULD treat the calls to RpcStartPagePrinter and RpcEndPagePrinter as informational only, with the only visible result being updating the page count of the print job. The server MUST NOT make any assumptions or perform any validation steps regarding the relative order or frequency of calls to these three methods.

##### RpcWritePrinter (Opnum 19)

RpcWritePrinter sends data to the print server.

1. DWORD RpcWritePrinter(
2. [in] PRINTER\_HANDLE hPrinter,
3. [in, size\_is(cbBuf)] BYTE\* pBuf,
4. [in] DWORD cbBuf,
5. [out] DWORD\* pcWritten
6. );

**hPrinter:** A handle to a printer object or port object that was opened by [RpcAddPrinter (section 3.1.4.2.3)](#Section_bd0bb8d526fd470581f5614d32b9f38f), [RpcAddPrinterEx (section 3.1.4.2.15)](#Section_3af45717308f4b4f8428430d12856110), [RpcOpenPrinter (section 3.1.4.2.2)](#Section_989357e2446e4872bb381dce21e1313f), or [RpcOpenPrinterEx (section 3.1.4.2.14)](#Section_0e81ce1872b146c38584a205393b04ff).

**pBuf:** A pointer to a buffer that contains the data to be written. This parameter can be NULL if the value of the *cbBuf* parameter is zero.

**cbBuf:** The number of bytes of data to be written.

**pcWritten:** A pointer to a value that receives the number of bytes of data that were written.

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate parameters as follows:

* Perform the validation steps that are specified in [PRINTER\_HANDLE Parameters](#Section_10bfa8c2d4564ad0912de674766a36c8), section 3.1.4.1.11. This method SHOULD assume that the handle to the printer or port object can be used without further access checks.
* Verify that a job has been associated with *hPrinter* using [RpcStartDocPrinter](#Section_90f95ff03ae04ddea83d35f457c710a6), and if that verification fails, return ERROR\_SPL\_NO\_STARTDOC, as specified in [MS-ERREF].
* Verify that printing of the job has not been canceled, and if that verification fails, return ERROR\_PRINT\_CANCELLED, as specified in [MS-ERREF].
* If *cbBuf* is not zero, verify that *pBuf* is not NULL.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* If the *hPrinter* parameter is a printer object handle, copy *cbBuf* bytes of data pointed to by *pBuf* to the job; depending on server policy and settings, the data is added to temporary storage of the job (for example, a spool file), or sent directly to the port.
* If the *hPrinter* parameter is a port object handle, copy *cbBuf* bytes of data pointed to by *pBuf* directly to the port.
* Write the number of bytes that were written to the variable that is pointed to by the *pcWritten* parameter.
* Return the status of the operation.
* If the operation is successful, the server MUST modify the job object to indicate the number of bytes written so far to that job.

##### RpcEndPagePrinter (Opnum 20)

RpcEndPagePrinter notifies the print server that the application is at the end of a page in a print job.

1. DWORD RpcEndPagePrinter(
2. [in] PRINTER\_HANDLE hPrinter
3. );

**hPrinter:** A handle to a printer object or port object that was opened by [RpcAddPrinter (section 3.1.4.2.3)](#Section_bd0bb8d526fd470581f5614d32b9f38f), [RpcAddPrinterEx (section 3.1.4.2.15)](#Section_3af45717308f4b4f8428430d12856110), [RpcOpenPrinter (section 3.1.4.2.2)](#Section_989357e2446e4872bb381dce21e1313f), or [RpcOpenPrinterEx (section 3.1.4.2.14)](#Section_0e81ce1872b146c38584a205393b04ff).

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate parameters as follows:

* Perform the validation steps that are specified in [PRINTER\_HANDLE Parameters (section 3.1.4.1.11)](#Section_10bfa8c2d4564ad0912de674766a36c8). This method SHOULD assume that the handle to the printer or port object can be used without further access checks.
* Verify that printing of the job has not been canceled, and if that verification fails, return ERROR\_PRINT\_CANCELLED, as specified in [MS-ERREF].

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* This method MAY trigger scheduling the job to the device, depending on server settings and policy, such as "Print while spooling".
* Update the count of the pages processed and printed so far in the job object.
* If any clients have registered for notification of job object changes, a notification MUST be broadcast to them.
* Return the status of the operation.

##### RpcAbortPrinter (Opnum 21)

RpcAbortPrinter aborts the currently spooling print document.

1. DWORD RpcAbortPrinter(
2. [in] PRINTER\_HANDLE hPrinter
3. );

**hPrinter:** A handle to a printer object or port object that was opened by [RpcAddPrinter (section 3.1.4.2.3)](#Section_bd0bb8d526fd470581f5614d32b9f38f), [RpcAddPrinterEx (section 3.1.4.2.15)](#Section_3af45717308f4b4f8428430d12856110), [RpcOpenPrinter (section 3.1.4.2.2)](#Section_989357e2446e4872bb381dce21e1313f), or [RpcOpenPrinterEx (section 3.1.4.2.14)](#Section_0e81ce1872b146c38584a205393b04ff).

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate parameters as follows:

* Perform the validation steps that are specified in [PRINTER\_HANDLE Parameters (section 3.1.4.1.11)](#Section_10bfa8c2d4564ad0912de674766a36c8). This method SHOULD assume that the handle to the printer or port object can be used without further access checks.
* Verify that a job has been associated with *hPrinter* by using [RpcStartDocPrinter](#Section_90f95ff03ae04ddea83d35f457c710a6), and if that verification fails, return ERROR\_SPL\_NO\_STARTDOC, as specified in [MS-ERREF].

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* The current job is aborted. If it is in spool stage, spooling MUST stop. If it is in printing stage, printing MUST stop.
* Modify the job object to indicate that the job has been aborted.
* Delete the spool file, if one exists.
* Delete the job object.
* Modify the list of jobs to exclude this deleted job.
* Return the status of the operation.

##### RpcReadPrinter (Opnum 22)

RpcReadPrinter retrieves data from the specified job or port.

1. DWORD RpcReadPrinter(
2. [in] PRINTER\_HANDLE hPrinter,
3. [out, size\_is(cbBuf)] BYTE\* pBuf,
4. [in] DWORD cbBuf,
5. [out] DWORD\* pcNoBytesRead
6. );

**hPrinter:** A handle to a job object or port object that was opened by [RpcOpenPrinter (section 3.1.4.2.2)](#Section_989357e2446e4872bb381dce21e1313f) or [RpcOpenPrinterEx (section 3.1.4.2.14)](#Section_0e81ce1872b146c38584a205393b04ff).

**pBuf:** A pointer to a buffer that receives the printer data. If the *hPrinter* parameter is the handle to a port object, this method returns the data that is returned by the port monitor.

This parameter can be NULL if the value of the *cbBuf* parameter is zero.

**cbBuf:** The size, in bytes, of data to be read into the buffer that is pointed to by the *pBuf* parameter.

**pcNoBytesRead:** A pointer to a variable that receives the number of bytes of data copied into the array to which *pBuf* points.

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate parameters as follows:

* Perform the validation steps that are specified in [PRINTER\_HANDLE Parameters (section 3.1.4.1.11)](#Section_10bfa8c2d4564ad0912de674766a36c8).
* Verify that printing of the job has not been canceled and if that verification fails, return ERROR\_PRINT\_CANCELLED, as specified in [MS-ERREF].
* If the value of the *cbBuf* parameter is not zero, verify that the *pBuf* parameter is not NULL.
* Additional validation MAY[<426>](#Appendix_A_426) be performed.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* If the *hPrinter* parameter is a job object handle, copy data from the temporary storage of the job object to the buffer pointed to by *pBuf*, up to the number of bytes indicated in *cbBuf*, or to the end of the temporary storage's data, whichever comes first.
* If the *hPrinter* parameter is a port object handle, read directly from the port and copy the read data to the buffer pointed to by *pBuf*, up to the number of bytes indicated in *cbBuf* or until no more data can be read, whichever comes first. This requires that the port monitor for the port supports reading data from the port; otherwise, return ERROR\_INVALID\_HANDLE. [<427>](#Appendix_A_427)
* Write the number of bytes that were copied to the variable that is pointed to by *pcNoBytesRead*.
* If reading from a job object, update the read pointer, so a subsequent read will continue at the correct location.
* Return the status of the operation.

##### RpcEndDocPrinter (Opnum 23)

RpcEndDocPrinter notifies the print server that the application is at the end of the current print job.

1. DWORD RpcEndDocPrinter(
2. [in] PRINTER\_HANDLE hPrinter
3. );

**hPrinter:** A handle to a printer object or port object that was opened by [RpcAddPrinter (section 3.1.4.2.3)](#Section_bd0bb8d526fd470581f5614d32b9f38f), [RpcAddPrinterEx (section 3.1.4.2.15)](#Section_3af45717308f4b4f8428430d12856110), [RpcOpenPrinter (section 3.1.4.2.2)](#Section_989357e2446e4872bb381dce21e1313f), or [RpcOpenPrinterEx (section 3.1.4.2.14)](#Section_0e81ce1872b146c38584a205393b04ff).

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate parameters as follows:

* Perform the validation steps that are specified in [PRINTER\_HANDLE Parameters](#Section_10bfa8c2d4564ad0912de674766a36c8), section 3.1.4.1.11. This method SHOULD assume that the handle to the printer or port object can be used without further access checks.
* Verify that a print job has been associated with *hPrinter* using [RpcStartDocPrinter (section 3.1.4.9.1)](#Section_90f95ff03ae04ddea83d35f457c710a6), and if that verification fails, return ERROR\_SPL\_NO\_STARTDOC, as specified in [MS-ERREF].

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* If any clients have registered for notification of job object changes, a notification MUST be broadcast to them.
* Modify the state of the job object to indicate the job has completed spooling if *hPrinter* is a printer object handle; or has completed printing if *hPrinter* is a port object handle.
* If the *hPrinter* parameter is a printer object handle, schedule the job for printing—subject to the configuration of the server and implementation-specific policies—and modify the state of the job object to indicate the job is printing.
* Return the status of the operation.

##### RpcFlushPrinter (Opnum 96)

RpcFlushPrinter is used by printer drivers to send a buffer of bytes to a specified port to cleanly abort a print job.[<428>](#Appendix_A_428) It also allows delaying the I/O line to the printer.

1. DWORD RpcFlushPrinter(
2. [in] PRINTER\_HANDLE hPrinter,
3. [in, size\_is(cbBuf)] BYTE\* pBuf,
4. [in] DWORD cbBuf,
5. [out] DWORD\* pcWritten,
6. [in] DWORD cSleep
7. );

**hPrinter:** A handle to a port object that was opened by [RpcOpenPrinter (section 3.1.4.2.2)](#Section_989357e2446e4872bb381dce21e1313f) or [RpcOpenPrinterEx (section 3.1.4.2.14)](#Section_0e81ce1872b146c38584a205393b04ff).

**pBuf:** A pointer to the array of bytes containing the data to be written to the printer. This parameter can be NULL if the value of the *cbBuf* parameter is zero.

**cbBuf:** The size, in bytes, of the array pointed to by the *pBuf* parameter.

**pcWritten:** A pointer to a variable that receives the number of bytes of data that were written to the printer.

**cSleep:** The time, in milliseconds, to delay the I/O line to the printer port. A value of zero indicates no delay.

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate parameters as follows:

* Perform the validation steps that are specified in [PRINTER\_HANDLE Parameters (section 3.1.4.1.11)](#Section_10bfa8c2d4564ad0912de674766a36c8). This method SHOULD assume that the handle to the port object can be used without further access checks.
* Verify that a previous [RpcWritePrinter (section 3.1.4.9.3)](#Section_6dd263aac8b547748d211daa0d0d9920) on the same port object has failed due to job cancellation, and if that verification fails, return ERROR\_INVALID\_HANDLE as specified in [MS-ERREF].
* If the value of the *cbBuf* parameter is not zero, verify that the *pBuf* parameter is not NULL.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Send the contents of the buffer that is pointed to by the *pBuf* parameter to the port.
* If the value of the *cSleep* parameter is not zero, the server MUST halt operations to the port for the number of milliseconds specified by the value of the *cSleep* parameter.
* Write the number of bytes that were written to the port, to the variable that is pointed to by the *pcWritten* parameter.
* Return the status of the operation.

#### Notification Methods

This section specifies methods for obtaining notifications of printing events.

| Method | Description |
| --- | --- |
| [RpcWaitForPrinterChange](#Section_e40cb68f679e4b84a33e86a94a141d83) | RpcWaitForPrinterChange retrieves information about the most recent change notification associated with a printer or print server.  Opnum 28 |
| [RpcFindClosePrinterChangeNotification](#Section_f50c8a861b2d4124b2205b840aacd2c7) | RpcFindClosePrinterChangeNotification closes a change notification object created by calling either the [RpcRemoteFindFirstPrinterChangeNotification](#Section_b8b414d9f1cd4191bb6b87d09ab2fd83) or [RpcRemoteFindFirstPrinterChangeNotificationEx](#Section_eb66b2211c1f4249b8bcc5befec2314d) function. The printer or print server associated with the change notification object will no longer be monitored by that object.  Opnum 56 |
| RpcRemoteFindFirstPrinterChangeNotification | RpcRemoteFindFirstPrinterChangeNotification creates a remote change notification object that monitors changes to printer objects, and sends change notifications to the client using the method [RpcRouterReplyPrinter](#Section_38bf5d2ce4a744b2b16f91ff8226f836) (see section 3.2.4.1.2).  Opnum 62 |
| RpcRemoteFindFirstPrinterChangeNotificationEx | RpcRemoteFindFirstPrinterChangeNotificationEx creates a remote change notification object that monitors changes to printer objects, and sends change notifications to the client using the method [RpcRouterReplyPrinterEx](#Section_1b44cb8295b5485c9ec90360c3a3ecb0) (see section 3.2.4.1.4).  Opnum 65 |
| [RpcRouterRefreshPrinterChangeNotification](#Section_cb5d22ed0352430383d5b5afd32fb461) | RpcRouterRefreshPrinterChangeNotification returns change notification information.  Opnum 67 |

##### RpcWaitForPrinterChange (Opnum 28)

RpcWaitForPrinterChange retrieves information about the most recent change notification that is associated with a printer or print server.

1. DWORD RpcWaitForPrinterChange(
2. [in] PRINTER\_HANDLE hPrinter,
3. [in] DWORD Flags,
4. [out] DWORD\* pFlags
5. );

**hPrinter:** A handle to a printer object or server object that was opened by [RpcAddPrinter (section 3.1.4.2.3)](#Section_bd0bb8d526fd470581f5614d32b9f38f), [RpcAddPrinterEx (section 3.1.4.2.15)](#Section_3af45717308f4b4f8428430d12856110), [RpcOpenPrinter (section 3.1.4.2.2)](#Section_989357e2446e4872bb381dce21e1313f), or [RpcOpenPrinterEx (section 3.1.4.2.14)](#Section_0e81ce1872b146c38584a205393b04ff).

**Flags:** The change notifications to wait for. The value of this parameter is a bitwise OR of one or more printer change values, defined in sections [2.2.3.6.1](#Section_1D63F9EFA4CD41E99152004FD2A8855A) and [2.2.3.6.2](#Section_BADFE1EBA5214686B2ACCAC29DFF311A). For rules governing printer change values, see section [2.2.4.13](#Section_f7e72f6c8ba345738a7edd12f2cff50e).

**pFlags:** A pointer to a variable that receives the bitwise OR combination of one or more printer change values.

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate parameters as follows:

* Perform the validation steps that are specified in [PRINTER\_HANDLE Parameters](#Section_10bfa8c2d4564ad0912de674766a36c8), section 3.1.4.1.11. This method SHOULD assume that the handle to the printer or server object can be used without further access checks.

If parameter validation fails, the server MUST fail the operation immediately, returning a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Add the client and the requested change notifications to the list of notification clients for the printer or server object.
* Wait for an implementation-specific period of time,[<429>](#Appendix_A_429) or until one of the changes specified by the value of the *Flags* parameter occurs.
* Remove the client from the list of notification clients for the printer or server object.
* If the *hPrinter* handle is closed with [RpcClosePrinter (section 3.1.4.2.9)](#Section_8482c18677734fb58034e9bae7a91e52) during this wait period, return ERROR\_INVALID\_HANDLE.
* If one or more of the specified changes occurred within the time-out period, write a bitwise OR combination of the changes to the variable that is pointed to by *pFlags* and return zero.
* If the time-out period has expired without any of the specified changes, return **PRINTER\_CHANGE\_TIMEOUT**, as specified in [MS-ERREF].

**Note:** Because this method waits for an implementation-specific, potentially long, period of time, it can cause the client system to stop responding. Therefore, this method is deprecated and SHOULD NOT be used. The implementer of a protocol client SHOULD consider using [RpcRemoteFindFirstPrinterChangeNotificationEx](#Section_eb66b2211c1f4249b8bcc5befec2314d) instead.

##### RpcFindClosePrinterChangeNotification (Opnum 56)

The RpcFindClosePrinterChangeNotification method closes a change notification object created by [RpcRemoteFindFirstPrinterChangeNotification (section 3.1.4.10.3)](#Section_b8b414d9f1cd4191bb6b87d09ab2fd83) or [RpcRemoteFindFirstPrinterChangeNotificationEx (section 3.1.4.10.4)](#Section_eb66b2211c1f4249b8bcc5befec2314d).[<430>](#Appendix_A_430) The printer or print server associated with the change notification object will no longer be monitored by that object.

1. DWORD RpcFindClosePrinterChangeNotification(
2. [in] PRINTER\_HANDLE hPrinter
3. );

**hPrinter:** A handle to a printer object or server object that was opened by [RpcAddPrinter (section 3.1.4.2.3)](#Section_bd0bb8d526fd470581f5614d32b9f38f), [RpcAddPrinterEx (section 3.1.4.2.15)](#Section_3af45717308f4b4f8428430d12856110), [RpcOpenPrinter (section 3.1.4.2.2)](#Section_989357e2446e4872bb381dce21e1313f), or [RpcOpenPrinterEx (section 3.1.4.2.14)](#Section_0e81ce1872b146c38584a205393b04ff).

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate parameters as follows:

* Perform validation steps as specified in [PRINTER\_HANDLE Parameters](#Section_10bfa8c2d4564ad0912de674766a36c8), section 3.1.4.1.11. This method SHOULD assume that the handle to the printer or server object can be used without further access checks.
* Verify that there is a change notification object associated with the printer object handle.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Clear all internal change notification objects associated with the *hPrinter*.
* Remove the client from the list of notification clients for the server or printer object.
* Return the status of the operation.

##### RpcRemoteFindFirstPrinterChangeNotification (Opnum 62)

RpcRemoteFindFirstPrinterChangeNotification creates a remote change notification object that monitors changes to printer objects and sends change notifications to a print client using either [RpcRouterReplyPrinter (section 3.2.4.1.2)](#Section_38bf5d2ce4a744b2b16f91ff8226f836) [<431>](#Appendix_A_431) or [RpcRouterReplyPrinterEx (section 3.2.4.1.4)](#Section_1b44cb8295b5485c9ec90360c3a3ecb0).[<432>](#Appendix_A_432)

1. DWORD RpcRemoteFindFirstPrinterChangeNotification(
2. [in] PRINTER\_HANDLE hPrinter,
3. [in] DWORD fdwFlags,
4. [in] DWORD fdwOptions,
5. [in, string, unique] wchar\_t\* pszLocalMachine,
6. [in] DWORD dwPrinterLocal,
7. [in, range(0,512)] DWORD cbBuffer,
8. [in, out, unique, size\_is(cbBuffer), disable\_consistency\_check]
9. BYTE\* pBuffer
10. );

**hPrinter:** A handle to a printer or server object that was opened by [RpcAddPrinter (section 3.1.4.2.3)](#Section_bd0bb8d526fd470581f5614d32b9f38f), [RpcAddPrinterEx (section 3.1.4.2.15)](#Section_3af45717308f4b4f8428430d12856110), [RpcOpenPrinter (section 3.1.4.2.2)](#Section_989357e2446e4872bb381dce21e1313f), or [RpcOpenPrinterEx (section 3.1.4.2.14)](#Section_0e81ce1872b146c38584a205393b04ff).

**fdwFlags:** Flags that specify the conditions that are required for a change notification object to enter a signaled state. A change notification MUST occur when one or more of the specified conditions are met.

This parameter specifies a bitwise OR of zero or more [Printer Change Values (section 2.2.4.13)](#Section_f7e72f6c8ba345738a7edd12f2cff50e). The rules governing printer change values are specified in section 2.2.4.13.

**fdwOptions:** The category of printers for which change notifications are returned. This parameter MUST be one of the supported values specified in Printer Notification Values (section [2.2.3.8](#Section_41d5c622ec7749ff95e369b325ce4e77)).

**pszLocalMachine:** A pointer to a string that represents the name of the client computer. The rules governing server names are specified in section [2.2.4.16](#Section_9b3f813570224b72accbaefcc360c83b).

**dwPrinterLocal:** An implementation-specific unique value that MUST be sufficient for the client to determine whether a call to [RpcReplyOpenPrinter (section 3.2.4.1.1)](#Section_7fcd3036d45a4ec7b081f2b860e66676) by the server is associated with the *hPrinter* parameter in this call. [<433>](#Appendix_A_433)

**cbBuffer:** A value that SHOULD be set to zero when sent and MUST be ignored on receipt.

**pBuffer:** A pointer that MUST be set to NULL when sent and MUST be ignored on receipt.

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the print server MUST validate parameters as follows:

* Perform the validation steps specified in [PRINTER\_HANDLE Parameters (section 3.1.4.1.11)](#Section_10bfa8c2d4564ad0912de674766a36c8). This method SHOULD assume that the handle to the printer or server object can be used without further access checks.
* Verify that a notification object is not already associated with the current handle.

If parameter validation fails, the server MUST fail the operation immediately returning a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Create and initialize a notification object that captures the notification settings requested by the user.
* Create and initialize a notification channel back to the client, over which the server can communicate the change notifications. This MUST be done by calling RpcReplyOpenPrinter on the client specified by the name pointed to by *pszLocalMachine*.
* Associate the notification object with the context for *hPrinter*.
* After the preceding steps have been performed, the server SHOULD add the client to the list of notification clients for the printer object or server object, and it SHOULD notify the client by using RpcRouterReplyPrinter or RpcRouterReplyPrinterEx when the object changes.

The choice of notification methods does not depend on whether notifications have been requested using RpcRemoteFindFirstPrinterChangeNotification or [RpcRemoteFindFirstPrinterChangeNotificationEx](#Section_eb66b2211c1f4249b8bcc5befec2314d). It is based on whether notifications can be expressed in the *fdwFlags* parameter of RpcRouterReplyPrinter alone, or if additional information is required to be provided using the additional parameters of RpcRouterReplyPrinterEx.

* Return the status of the operation.

##### RpcRemoteFindFirstPrinterChangeNotificationEx (Opnum 65)

RpcRemoteFindFirstPrinterChangeNotificationEx creates a remote change notification object that monitors changes to printer objects and sends change notifications to a print client using either [RpcRouterReplyPrinter (section 3.2.4.1.2)](#Section_38bf5d2ce4a744b2b16f91ff8226f836) [<434>](#Appendix_A_434) or [RpcRouterReplyPrinterEx (section 3.2.4.1.4)](#Section_1b44cb8295b5485c9ec90360c3a3ecb0).[<435>](#Appendix_A_435)

1. DWORD RpcRemoteFindFirstPrinterChangeNotificationEx(
2. [in] PRINTER\_HANDLE hPrinter,
3. [in] DWORD fdwFlags,
4. [in] DWORD fdwOptions,
5. [in, string, unique] wchar\_t\* pszLocalMachine,
6. [in] DWORD dwPrinterLocal,
7. [in, unique] RPC\_V2\_NOTIFY\_OPTIONS\* pOptions
8. );

**hPrinter:** A handle to a printer or server object that was opened by [RpcAddPrinter (section 3.1.4.2.3)](#Section_bd0bb8d526fd470581f5614d32b9f38f), [RpcAddPrinterEx (section 3.1.4.2.15)](#Section_3af45717308f4b4f8428430d12856110), [RpcOpenPrinter (section 3.1.4.2.2)](#Section_989357e2446e4872bb381dce21e1313f), or [RpcOpenPrinterEx (section 3.1.4.2.14)](#Section_0e81ce1872b146c38584a205393b04ff).

**fdwFlags:** Flags that specify the conditions that are required for a change notification object to enter a signaled state. A change notification MUST occur when one or more of the specified conditions are met.

This parameter specifies a bitwise OR of zero or more [Printer Change Values (section 2.2.4.13)](#Section_f7e72f6c8ba345738a7edd12f2cff50e). The rules governing printer change values are specified in section 2.2.4.13.

**fdwOptions:** The category of printers for which change notifications are returned. This parameter MUST be one of the supported values specified in Printer Notification Values (section [2.2.3.8](#Section_41d5c622ec7749ff95e369b325ce4e77)).

**pszLocalMachine:** A pointer to a string that represents the name of the client computer. The rules governing server names are specified in section [2.2.4.16](#Section_9b3f813570224b72accbaefcc360c83b).

**dwPrinterLocal:** An implementation-specific unique value that MUST be sufficient for the client to determine whether a call to [RpcReplyOpenPrinter (section 3.2.4.1.1)](#Section_7fcd3036d45a4ec7b081f2b860e66676) by the server is associated with the *hPrinter* parameter in this call. [<436>](#Appendix_A_436)

**pOptions:** A pointer to an [RPC\_V2\_NOTIFY\_OPTIONS (section 2.2.1.13.1)](#Section_4599c052e81441749bd500473329a9f8) structure that specifies printer or job members that the client listens to for notifications. For lists of members that can be monitored, see Printer Notification Values (section 2.2.3.8) and [Job Notification Values (section 2.2.3.3)](#Section_76acb7fbb63f4cad88dd6920096a2f13).

The value of this parameter can be NULL if the value of *fdwFlags* is nonzero.

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate parameters as follows:

* Perform the validation steps specified in [PRINTER\_HANDLE Parameters (section 3.1.4.1.11)](#Section_10bfa8c2d4564ad0912de674766a36c8). This method SHOULD assume that the handle to the printer or server object can be used without further access checks.
* Verify that a notification object is not already associated with the current handle.
* Verify that either *pOptions* is not NULL or that the value *fdwFlags* is valid and not zero.

If parameter validation fails, the server MUST fail the operation immediately returning a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Create and initialize a notification object that captures the notification settings requested by the user.
* Create and initialize a notification channel back to the client over which the server MUST communicate the change notifications. This MUST be done by calling RpcReplyOpenPrinter on the client specified by the name pointed to by *pszLocalMachine*.
* Associate the notification object with the context for *hPrinter*.
* After the preceding steps have been performed, the server SHOULD add the client to the list of notification clients for the printer object or server object, and it SHOULD notify the client by using RpcRouterReplyPrinter or RpcRouterReplyPrinterEx when the object changes.

The choice of notification methods does not depend on whether notifications have been requested using [RpcRemoteFindFirstPrinterChangeNotification](#Section_b8b414d9f1cd4191bb6b87d09ab2fd83) or RpcRemoteFindFirstPrinterChangeNotificationEx. It is based on whether notifications can be expressed in the *fdwFlags* parameter of RpcRouterReplyPrinter alone, or if additional information is required to be provided using the additional parameters of RpcRouterReplyPrinterEx.

* Return the status of the operation.

##### RpcRouterRefreshPrinterChangeNotification (Opnum 67)

RpcRouterRefreshPrinterChangeNotification returns change notification information.[<437>](#Appendix_A_437)

1. DWORD RpcRouterRefreshPrinterChangeNotification(
2. [in] PRINTER\_HANDLE hPrinter,
3. [in] DWORD dwColor,
4. [in, unique] RPC\_V2\_NOTIFY\_OPTIONS\* pOptions,
5. [out] RPC\_V2\_NOTIFY\_INFO\*\* ppInfo
6. );

**hPrinter:** A handle to a printer object or server object that was opened by [RpcAddPrinter (section 3.1.4.2.3)](#Section_bd0bb8d526fd470581f5614d32b9f38f), [RpcAddPrinterEx (section 3.1.4.2.15)](#Section_3af45717308f4b4f8428430d12856110), [RpcOpenPrinter (section 3.1.4.2.2)](#Section_989357e2446e4872bb381dce21e1313f), or [RpcOpenPrinterEx (section 3.1.4.2.14)](#Section_0e81ce1872b146c38584a205393b04ff).

This handle MUST have been previously used successfully by the client in a call to [RpcRemoteFindFirstPrinterChangeNotification (section 3.1.4.10.3)](#Section_b8b414d9f1cd4191bb6b87d09ab2fd83) or [RpcRemoteFindFirstPrinterChangeNotificationEx (section 3.1.4.10.4)](#Section_eb66b2211c1f4249b8bcc5befec2314d), and it MUST NOT have been closed by calling [RpcFindClosePrinterChangeNotification (section 3.1.4.10.2)](#Section_f50c8a861b2d4124b2205b840aacd2c7).

**dwColor:** An implementation-specific value that MAY be used by print clients to get an indication of the order of notifications.[<438>](#Appendix_A_438)

**pOptions:** A pointer to an [RPC\_V2\_NOTIFY\_OPTIONS (section 2.2.1.13.1)](#Section_4599c052e81441749bd500473329a9f8) structure that specifies printer or job members that the client listens to for notifications. For lists of members that can be monitored, see [Printer Notification Values (section 2.2.3.8)](#Section_41d5c622ec7749ff95e369b325ce4e77) and [Job Notification Values (section 2.2.3.3)](#Section_76acb7fbb63f4cad88dd6920096a2f13).

**ppInfo:** A pointer to a variable that receives a pointer to an [RPC\_V2\_NOTIFY\_INFO (section 2.2.1.13.3)](#Section_79db335febe74a6f8533efdda32cc5d8) structure that contains notification information.

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate parameters as follows:

* Perform the validation steps that are specified in [PRINTER\_HANDLE Parameters (section 3.1.4.1.11)](#Section_10bfa8c2d4564ad0912de674766a36c8). This method SHOULD assume that the handle to the printer or server object can be used without further access checks.
* Verify that the client is in the list of notification clients for the printer object or server object.
* Verify that a notification back channel to the client has been established and is still open.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Collect all the notification data requested for the printer objects.
* Allocate a buffer and write the collected notification data in the buffer.
* Associate the buffer with *ppInfo* output parameter.
* Return the status of the operation.

This method MUST be called when the client receives an RPC\_V2\_NOTIFY\_INFO structure with the **PRINTER\_NOTIFY\_INFO\_DISCARDED** bit set in its **Flags** member. This indicates that an overflow or other error has occurred and that notifications might have been lost, which sets the notification object to the discarded state. The server MUST NOT send any additional notifications until the client makes this method call. If the operation is successful, the server MUST modify the notification object to clear the discarded state.

#### Monitor Module Methods

A monitor module is a server-side executable object that provides a communication path between a print server and the drivers that access hardware on a machine. A port monitor module manages access to I/O port hardware.

Port monitor modules are implementation specific for a given port type.

A port monitor module provides the actual implementation used by the print spooler when one of the [Port Management Methods (section 3.1.4.6)](#Section_1cf0e23868da43bdbaba44b35b115cc0) is called. The print spooler also uses methods provided by the port monitor module to communicate with the physical print device.

Port monitor modules MUST support the following methods:

* Either: OpenPort or OpenPortEx
* ClosePort
* StartDocPort
* WritePort
* ReadPort
* EndDocPort

Port monitor modules MAY[<439>](#Appendix_A_439) support the following optional methods:

* AddPort
* AddPortEx
* ConfigurePort
* DeletePort

Port monitor modules SHOULD support an additional set of methods, all of which MUST either be implemented together or not be present at all:

* XcvOpenPort
* XcvClosePort

**XcvData** SHOULD support the following actions (see section [3.1.4.6.5](#Section_bf9d2351aac0481e9d3705eb5cc961da)):

* AddPort
* DeletePort
* MonitorUI

Actions MUST be specified by the client in a string pointed to by the *pszDataName* parameter of RpcXcvData.

Additional actions MAY be supported in a given implementation.[<440>](#Appendix_A_440) The following sections, [LOCALMON](#Section_85d1c3126bba4b01bea52ec153334c1b) and [LPRMON](#Section_faf227a5b8a84a0eadcfa206059a91a9), describe the implementation of the **XcvData** method, its supported actions, and corresponding behaviors in the [**LOCALMON**](#gt_57d430d9-cd24-4371-b022-0fb502e41817) and [**LPRMON**](#gt_0f4b31da-52ab-4a6f-b093-c0913574dc1f) monitor modules. All method descriptions assume the standard buffer size validation pattern, as specified for the RpcXcvData method. Unless otherwise specified, for actions not using *pInputData*, *pInputData* MUST be NULL and *cbInputData* MUST be zero. Unless otherwise specified, for actions not using *pOutputData*, *pOutputData* MUST be NULL, *cbOutputData* MUST be zero, and *pcbOutputNeeded* MUST be NULL. For historical reasons, the names of some of the actions supported by **XcvData** are identical to some of the other port monitor module methods. The server method RpcXcvData routes calls to a port monitor's **XcvData** method, and the parameter lists of RpcXcvData and the port monitor's **XcvData** are identical.

##### LOCALMON

This section describes the implementation of **XcvData** methods in LOCALMON. This monitor module is used to control parallel and serial ports that could have a printer connected to them.[<441>](#Appendix_A_441)

For parallel and serial port naming, see section [2.2.4.10](#Section_03c5feb74fbb4569b0649eb6ad833f50). For more information about the values listed in the left column in the following table, see section [3.1.4.6.5](#Section_bf9d2351aac0481e9d3705eb5cc961da).

| Value | Meaning |
| --- | --- |
| "MonitorUI" | This action returns the name of the client-side interface monitor. The *pOutputData* parameter MUST be a non-NULL pointer to the buffer that receives the string representing the name of the interface monitor. |
| "AddPort" | This action adds a local port. The *pInputData* parameter MUST be a non-NULL pointer to a string representing the name of the port. |
| "DeletePort" | This action deletes a local port. The *pInputData* parameter MUST be a non-NULL pointer to a string representing the name of the port. |
| "ConfigureLPTPortCommandOK" | This action determines configuration of the LPT port and sets the transmission retry timeout. The *pInputData* parameter MUST be a non-NULL pointer to a string representing a number from 1 through 3, inclusive. Those numbers map to the port to configure; that is, whether it is "LPT1" OR "LPT2" OR "LPT3". The *pOutputData* parameter is not used. |
| "PortExists" | This action checks whether a local port exists in the print server's list of port objects. The *pInputData* parameter MUST be a non-NULL pointer to a string representing the name of the port. The *pOutputData* parameter MUST be a non-NULL pointer to a DWORD variable that MUST receive the value 0 if the port does not exist and a nonzero value if the port exists. |
| "PortIsValid" | This action determines whether a given name is a valid port name accepted by the port monitor. The *pInputData* parameter MUST be a non-NULL pointer to a string representing the name of the port. If the port identifies a valid port name, the method MUST return ERROR\_SUCCESS; otherwise, it MUST return a nonzero error code. |

##### LPRMON

This section describes the implementation of **XcvData** methods in LPRMON. This monitor module is used to control printers over a network on machines that have implemented Unix print server functions and expose them through the **Line Printer (LPR) Protocol**, as defined in [[RFC1179]](http://go.microsoft.com/fwlink/?LinkId=90269).[<442>](#Appendix_A_442)

For network port naming, see section [2.2.4.10](#Section_03c5feb74fbb4569b0649eb6ad833f50). For more information about the values listed in the Value column in the following table, see section [3.1.4.6.5](#Section_bf9d2351aac0481e9d3705eb5cc961da).

| Value | Meaning |
| --- | --- |
| "MonitorUI" | This action returns the name of the client-side interface monitor. The *pOutputData* parameter MUST be a non-NULL pointer to a buffer that receives the string representing the name of the interface monitor. |
| "AddPort" | This action adds an LPR printer port. The *pInputData* parameter MUST be a non-NULL pointer to a string representing the name of the port. The *pOutputData* parameter is not used. |
| "DeletePort" | This action deletes an LPR printer port. The *pInputData* parameter MUST be a non-NULL pointer to a string representing the name of the port. The *pOutputData* parameter is not used. |
| "CheckPrinter" | This action checks on the LPR printer port. The *pInputData* parameter MUST be a non-NULL pointer to a string representing the name of the port connected to the printer. |

##### TCPMON

Information about the implementation of the TCPMON monitor module can be found in the command value following table and in the subsections that follow.

The TCPMON monitor module is used to control printers directly connected to a **TCP/IP** network.[<443>](#Appendix_A_443)

For network port naming, see section [2.2.4.10](#Section_03c5feb74fbb4569b0649eb6ad833f50).

For structures used with the TCPMON monitor module, see section [2.2.2.14](#Section_c6430e00b96a4e58b0f467d17c4adeb5)

The following table discusses command values used by the TCPMON monitor module.

| Value | Meaning |
| --- | --- |
| "AddPort" | This action adds a printer port. The *pInputData* parameter is a non-NULL pointer to a [PORT\_DATA\_1](#Section_bd1e99bb8a2747c3b276e38fda1f9d34) or [PORT\_DATA\_2](#Section_2d256a4b819c4b848e8fb1edb1dcf0c2) structure. |
| "DeletePort" | This action deletes a printer port. The *pInputData* parameter is a non-NULL pointer to a [DELETE\_PORT\_DATA\_1](#Section_98b804cc08694f1f8f91dacf63769d7e) structure. |
| "MonitorUI" | This action returns the name of the client-side monitor interface module. The *pOutputData* parameter is a non-NULL pointer to a buffer that receives the string representing the name of the interface module. |
| "ConfigPort" | This action configures a printer port. The *pInputData* parameter is a non-NULL pointer to a PORT\_DATA\_1 or PORT\_DATA\_2 structure. |
| "GetConfigInfo" | This action gets configuration information for a printer port. The *pInputData* parameter is a non-NULL pointer to a [CONFIG\_INFO\_DATA\_1](#Section_9764f82547a1440c973a32328d204296) structure. The *pOutputData* parameter points to a buffer that receives a PORT\_DATA\_1 or PORT\_DATA\_2 structure describing the port. |
| "HostAddress" | This action gets the printer's host name. *pOutputData* is a non-NULL pointer to a buffer that receives a string containing the printer's host name. |
| "IPAddress" | This action gets the printer's IP address. *pOutputData* is a non-NULL pointer to a buffer that receives a string containing the printer's IP address. |
| "SNMPCommunity" | This action gets the printer's Simple Network Management Protocol (SNMP) [[RFC1157]](http://go.microsoft.com/fwlink/?LinkId=121934) community name. *pOutputData* is a non-NULL pointer to a buffer that receives a string containing the printer's SNMP community name. |
| "SNMPDeviceIndex" | This action gets the printer's SNMP device index. *pOutputData* is a non-NULL pointer to a variable that receives a DWORD value containing the printer's SNMP device index. |
| "SNMPEnabled" | This action determines whether SNMP is enabled for the printer. *pOutputData* is a non-NULL pointer to a variable that receives the DWORD value 0x00000000 if SNMP is disabled and a nonzero value otherwise. |
| "GetIdlePollingState" | This action determines whether TCPMON is set to poll automatically for the printer ("idle polling"). *pOutputData* is a non-NULL pointer to a variable that receives the DWORD value 0x00000000 if idle polling is disabled, and a nonzero value otherwise. |
| "SetIdlePollingState" | This action sets the idle polling state for the printer. *pInputData* is a non-NULL pointer to a DWORD which is set to 0x00000001 to enable idle polling, and to 0x00000000 to disable it. |
| "SetDeviceIDOid" | This action sets the object identifier (OID) used to query the IEEE 1284 device ID from the printer.[<444>](#Appendix_A_444) (For details, see [[IEEE1284]](http://go.microsoft.com/fwlink/?LinkId=89898).) *pInputData* is a non-NULL pointer to a string specifying the OID. If this OID is not set, TCPMON uses a default of "1.3.6.1.4.1.2699.1.2.1.2.1.1.3.<1-based port index>". |
| "DeviceID" | This action initiates a query for the IEEE 1284 device ID.[<445>](#Appendix_A_445) (For details, see [IEEE1284].) *pOutputData* is a non-NULL pointer to a buffer that receives a string containing the IEEE 1284 device ID, as specified in [IEEE1284]. |
| "GetPortList" | This action requests the list of supported ports on a device.[<446>](#Appendix_A_446)*pInputData* is a non-NULL pointer to a string containing the IP address of hostname of the device. *pOutputData* is a non-NULL pointer to a buffer that receives a [PORT\_DATA\_LIST\_1](#Section_345c0cf7ae98467a8df5849ce3eb7e69) structure. |
| "CleanupPort" | This action attempts to remove the TCP/IP port associated with the **hXcv** handle.[<447>](#Appendix_A_447) If printers are still using the port, it will not be removed. |

##### WSDMON

This section describes the implementation of the [XcvData](#Section_bf9d2351aac0481e9d3705eb5cc961da) method in [**WSDMON**](#gt_98ae481c-f1c3-4b23-b605-d23616879180). This monitor module is used to control Web Services for Devices (WSD) printers.[<448>](#Appendix_A_448) WSDMON does not have a corresponding user interface module.

For network port naming, see section [2.2.4.10](#Section_03c5feb74fbb4569b0649eb6ad833f50).

For structures used with the WSDMON monitor module, see section [2.2.2.15](#Section_6e2abe47b5c24c43a2328fca3897287f).

The following table discusses command values used by the WSDMON monitor module.

| Value | Meaning |
| --- | --- |
| "CleanupPort" | Attempts to remove the Web Services for Devices (WSD) port associated with the **hXcv** handle. If printers are still using the port, it will not be removed. |
| "DeviceID" | Initiates a query for the WSD **DeviceID**. The required value for the *pOutputData* parameter is a non-NULL pointer to a buffer that is required to receive a string containing the WSD **DeviceID** (PKEY\_PNPX\_GlobalIdentity). |
| "PnPXID" | Initiates a query for the WSD **PnPXID**. The required value for the *pOutputData* parameter is a non-NULL pointer to a buffer that is required to receive a string containing the WSD **PnPXID** (PKEY\_PNPX\_ID). |
| "ResetCommunication" | Attempts to make sure that the communication between the printer and the operating system is working properly. |
| "ServiceID" | Initiates a query for the WSD **ServiceID**. The required value for the *pOutputData* parameter is a non-NULL pointer to a buffer that is required to receive a string containing the WSD **ServiceID** (PKEY\_PNPX\_ServiceID). |
| "CheckCluster" | Determines whether the queried server is a stand-alone server or a cluster node.[<449>](#Appendix_A_449) The required value for the *pOutputData* parameter is a non-NULL pointer to a variable that is required to receive the DWORD value 0x00000000 if the queried server is not a cluster node and a nonzero value otherwise. |
| "DiscoverDevice" | Uses WS-discovery directed [**unicast**](#gt_e73c7149-240a-4fad-8a27-5c6b7fdc956a) search to try to find a WSD–enabled device at the supplied [**URI**](#gt_e18af8e8-01d7-4f91-8a1e-0fb21b191f95) of the endpoint.[<450>](#Appendix_A_450) The required value for the *pInputData* parameter is a non-NULL pointer to a string specifying the URI of the WSD endpoint. If a WSD device is found, and it supports the WSD print service definition, ERROR\_SUCCESS is returned; otherwise, ERROR\_PRINTER\_NOT\_FOUND is returned, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf). |
| "DriverAvailable" | Determines whether a printer driver for the queried device is available in the server's driver store.[<451>](#Appendix_A_451) The required value for the *pInputData* parameter is a non-NULL pointer to a string specifying the URI of the WSD endpoint. The required value for the *pOutputData* parameter is a non-NULL pointer to a buffer that is required to receive a [WSD\_DRIVER\_DATA](#Section_3a887ca485f94dd3b83f6794a0b55353) structure if the specified endpoint supports a WSD Printer Service and a driver is available. If the endpoint does not support the WSD Printer Service, ERROR\_PRINTER\_NOT\_FOUND is returned, as specified in [MS-ERREF]. If no driver can be found, ERROR\_CANNOT\_DETECT\_DRIVER\_FAILURE is returned. |
| "AssocDevice" | Searches for a WSD Printer Service at the supplied URI of the endpoint and installs the printer if found.[<452>](#Appendix_A_452) This command is only supported on stand-alone servers. The required value for the *pInputData* parameter is a non-NULL pointer to a string specifying the URI of the WSD endpoint. If a WSD Printer Service is found, a **PnPX** installation of the printer is initiated; otherwise, ERROR\_PRINTER\_NOT\_FOUND is returned, as specified in [MS-ERREF]. |
| "AddPrinterPort" | Searches for a WSD Printer Service at the supplied URI of the endpoint, and if one is found, creates a new WSD port connected to the discovered device.[<453>](#Appendix_A_453) This command is only supported on cluster servers. The required value for the *pInputData* parameter is a non-NULL pointer to a string specifying the URI of the WSD endpoint. The required value for the *pOutputData* parameter is a non-NULL pointer to a buffer that is required to receive a string identifying the new port name if a WSD Printer Service is found; otherwise, ERROR\_PRINTER\_NOT\_FOUND is returned, as specified in [MS-ERREF]. |
| "BackupPort" | Initiates a query for the WSD port backup data.[<454>](#Appendix_A_454) The required value for the *pOutputData* parameter is a non-NULL pointer to a buffer that is required to receive a [WSD\_BACKUP\_PORT\_DATA](#Section_45f18ad3e1a44c44a7800dc2f0422914) structure. |
| "AssocDeviceMulticast" | Searches for a WSD Printer Service at the device endpoint that is specified by the **GlobalID**, which in turn is specified by the string pointed to by *pInputData*, and installs the printer if found.[<455>](#Appendix_A_455) This command is only supported on stand-alone servers. The required value for the *pInputData* parameter is a non-NULL pointer to a string specifying the **GlobalID** of the WSD endpoint. If a WSD Printer Service is found, a **PnPX** installation of the printer is initiated; otherwise, ERROR\_PRINTER\_NOT\_FOUND is returned, as specified in [MS-ERREF]. |
| "RestorePort" | Searches for a WSD printer service that is identified by the **ServiceID** at the device endpoint specified by the **GlobalID** and optionally **RemoteURL**, depending on the value of the **DiscoveryMethod**.[<456>](#Appendix_A_456)If the value of **DiscoveryMethod** is *kMulticast*, only the **ServiceID** and **GlobalID** values MUST be used for the search. If the value of **DiscoveryMethod** is *kDirected*, the **ServiceID**, **GlobalID**, and **RemoteURL** values MUST be used for the search. The required value for the *pInputData* parameter is a [WSD\_BACKUP\_PORT\_DATA\_EX](#Section_226099c983cb426097e1f796444b4502) structure (section 2.2.2.15.3) that contains the values relevant to the printer port to restore. If a matching WSD printer service is found, a **PnPX** installation of the printer is initiated; otherwise a new port with the specified information is created with a status of PORT\_STATUS\_OFFLINE (section [2.2.1.9.3](#Section_411e442325e44f6184b745336e40094e)). The required value for the *pOutputData* parameter is a non-NULL pointer to a buffer that receives a string identifying the new port by name. |
| "AddMulticastPort" | Searches for a WSD Printer Service at the device endpoint that is specified by the **GlobalID**, which in turn is specified by the string pointed to by *pInputData*, and if one is found, creates a new WSD port connected to the discovered device.[<457>](#Appendix_A_457) [<458>](#Appendix_A_458) |

#### Job Named Property Management Methods

This section specifies methods for creating, updating, deleting, and enumerating **Job Named Properties** (section [3.1.1](#Section_1bd61448db4b4284a4b18346732402d4)) for a specified print job.

| Method | Description |
| --- | --- |
| [RpcGetJobNamedPropertyValue (section 3.1.4.12.1)](#Section_64d3b0c08eef4a2b859960904de7c823) | RpcGetJobNamedPropertyValue retrieves the value of the specified **Job Named Property** for the print job.  Opnum 110 |
| [RpcSetJobNamedProperty (section 3.1.4.12.2)](#Section_a08c060ad4e24a88a4874f6aaecfcf09) | RpcSetJobNamedProperty creates a new **Job Named Property** or changes the value of an existing **Job Named Property** for the print job.  Opnum 111 |
| [RpcDeleteJobNamedProperty (section 3.1.4.12.3)](#Section_02b59e33f1534870af67199267dc0b64) | RpcDeleteJobNamedProperty deletes a **Job Named Property** for the print job.  Opnum 112 |
| [RpcEnumJobNamedProperties (section 3.1.4.12.4)](#Section_6b1559f993494d90b8cf24292ab736ad) | RpcEnumJobNamedProperties enumerates the **Job Named Properties** for the print job.  Opnum 113 |

##### RpcGetJobNamedPropertyValue (Opnum 110)

**RpcGetJobNamedPropertyValue** retrieves the current value of the specified **Job Named Property** (section [3.1.1](#Section_1bd61448db4b4284a4b18346732402d4)).[<459>](#Appendix_A_459)

1. DWORD RpcGetJobNamedPropertyValue(
2. [in] PRINTER\_HANDLE hPrinter,
3. [in] DWORD JobId,
4. [in, string] const wchar\_t\* pszName,
5. [out] RPC\_PrintPropertyValue\* pValue
6. );

**hPrinter:** A [PRINTER\_HANDLE (section 2.2.1.1.4)](#Section_9c5d49b8692047b0946687ba910e6071) to a printer object, job object, or server object that was opened by [RpcAddPrinter (section 3.1.4.2.3)](#Section_bd0bb8d526fd470581f5614d32b9f38f), [RpcAddPrinterEx (section 3.1.4.2.15)](#Section_3af45717308f4b4f8428430d12856110), [RpcOpenPrinter (section 3.1.4.2.2)](#Section_989357e2446e4872bb381dce21e1313f), or [RpcOpenPrinterEx (section 3.1.4.2.14)](#Section_0e81ce1872b146c38584a205393b04ff).

**JobId:** The identifier of a print job. This value MUST NOT be zero.

**pszName:** A pointer to a string that specifies the **Job Named Property** to be queried. This pointer MUST NOT be NULL.

**pValue:** A pointer to an [RPC\_PrintPropertyValue (section 2.2.1.14.1)](#Section_2ed8404520264696b2e263dbffdc6d21) structure that on return from this call contains the value of the **Job Named Property** specified by the *pszName* argument.

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

On receiving this message, the server MUST validate the following:

* Perform the validation steps that are specified in [PRINTER\_HANDLE Parameters (section 3.1.4.1.11)](#Section_10bfa8c2d4564ad0912de674766a36c8).
* Verify that the value of the *JobId* parameter corresponds to a job in the list of jobs. If the object specified by the *hPrinter* parameter is a server object, search for a print job in each printer in the list of printers on the print server. If the object specified by the *hPrinter* parameter is a printer object, search for a print job only in the list of print jobs for the specified printer. If the object specified by the *hPrinter* parameter is a job object, compare the identifier of this print job with the specified *JobId*. If this verification fails, return ERROR\_INVALID\_PARAMETER.
* Verify that the value of the *pValue* parameter is a non-NULL pointer to a string. If this verification fails, return ERROR\_INVALID\_PARAMETER.
* Verify that the value of the *pszName* parameter corresponds to an existent **Job Named Property** for the print job specified with the *JobId* parameter. If this verification fails, return ERROR\_NOT\_FOUND.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the print client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Allocate and initialize a buffer with the **PrintPropertyValue** structure to be returned to the client via the *pValue* parameter.
* Return the status of the operation.

On successful completion of this call, the client SHOULD free the buffer specified by the *pValue* parameter.

##### RpcSetJobNamedProperty (Opnum 111)

**RpcSetJobNamedProperty** creates a new **Job Named Property** (section [3.1.1](#Section_1bd61448db4b4284a4b18346732402d4)), or changes the value of an existing **Job Named Property** for the specified print job.[<460>](#Appendix_A_460)

1. DWORD RpcSetJobNamedProperty(
2. [in] PRINTER\_HANDLE hPrinter,
3. [in] DWORD JobId,
4. [in] RPC\_PrintNamedProperty\* pProperty
5. );

**hPrinter:** A [PRINTER\_HANDLE (section 2.2.1.1.4)](#Section_9c5d49b8692047b0946687ba910e6071) to a printer object, job object, or server object that was opened by [RpcAddPrinter (section 3.1.4.2.3)](#Section_bd0bb8d526fd470581f5614d32b9f38f), [RpcAddPrinterEx (section 3.1.4.2.15)](#Section_3af45717308f4b4f8428430d12856110), [RpcOpenPrinter (section 3.1.4.2.2)](#Section_989357e2446e4872bb381dce21e1313f), or [RpcOpenPrinterEx (section 3.1.4.2.14)](#Section_0e81ce1872b146c38584a205393b04ff).

**JobId:** The identifier of a print job. This value MUST NOT be zero.

**pProperty:** A pointer to an [RPC\_PrintNamedProperty (section 2.2.1.14.2)](#Section_31bdf5289b8947f9823ba45bbc877716) structure specifies the property to be created if it not exists for the print job specified by the *JobId* parameter, or update an existing property with a new value. This pointer MUST NOT be NULL.

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate the following:

* Perform the validation steps that are specified in [PRINTER\_HANDLE Parameters (section 3.1.4.1.11)](#Section_10bfa8c2d4564ad0912de674766a36c8).
* Verify that the value of the *JobId* parameter corresponds to a job in the list of jobs. If the object specified by the *hPrinter* parameter is a server object, search for a print job in each printer in the list of printers on the print server. If the object specified by the *hPrinter* parameter is a printer object, search for a print job only in the list of print jobs for the specified printer. If the object specified by the *hPrinter* parameter is a job object, compare the identifier of this print job with the specified *JobId*. If this verification fails, return ERROR\_INVALID\_PARAMETER.
* Verify that the RPC\_PrintNamedProperty structure specified by the *pProperty* parameter contains an **ePropertyType** member set to a valid value as specified in section [2.2.1.14.3](#Section_0360160c35514c5bbcee9164122d34e2). If this verification fails, return ERROR\_INVALID\_FLAGS.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* If the property specified by the *pProperty* parameter does not exist for the print job specified by the *JobId* parameter, create a new property with the name and value specified by the *pProperty* parameter.
* If the property specified by the *pProperty* parameter does exist for the print job specified by the *JobId* parameter, update the property with the value specified by the *pProperty* parameter.
* Return the status of the operation.

##### RpcDeleteJobNamedProperty (Opnum 112)

**RpcDeleteJobNamedProperty** deletes an existing **Job Named Property** (section [3.1.1](#Section_1bd61448db4b4284a4b18346732402d4)) for the specified print job.[<461>](#Appendix_A_461)

1. DWORD RpcDeleteJobNamedProperty(
2. [in] PRINTER\_HANDLE hPrinter,
3. [in] DWORD JobId,
4. [in, string] const wchar\_t\* pszName
5. );

**hPrinter:** A [PRINTER\_HANDLE (section 2.2.1.1.4)](#Section_9c5d49b8692047b0946687ba910e6071) to a printer object, job object, or server object that was opened by [RpcAddPrinter (section 3.1.4.2.3)](#Section_bd0bb8d526fd470581f5614d32b9f38f), [RpcAddPrinterEx (section 3.1.4.2.15)](#Section_3af45717308f4b4f8428430d12856110), [RpcOpenPrinter (section 3.1.4.2.2)](#Section_989357e2446e4872bb381dce21e1313f), or [RpcOpenPrinterEx (section 3.1.4.2.14)](#Section_0e81ce1872b146c38584a205393b04ff).

**JobId:** The identifier of a print job. This value MUST NOT be zero.

**pszName:** A pointer to a string that specifies the **Job Named Property** to be deleted. This pointer MUST NOT be NULL.

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate the following:

* Perform the validation steps that are specified in [PRINTER\_HANDLE Parameters (section 3.1.4.1.11)](#Section_10bfa8c2d4564ad0912de674766a36c8).
* Verify that the value of the *JobId* parameter corresponds to a job in the list of jobs. If the object specified by the *hPrinter* parameter is a server object, search for a print job in each printer in the list of printers on the print server. If the object specified by the *hPrinter* parameter is a printer object, search for a print job only in the list of print jobs for the specified printer. If the object specified by the *hPrinter* parameter is a job object, compare the identifier of this print job with the specified *JobId*. If this verification fails, return ERROR\_INVALID\_PARAMETER.
* Verify that the value of the *pszName* parameter is a non-NULL pointer to a string. If this verification fails, return ERROR\_INVALID\_PARAMETER.
* Verify that the value of the *pszName* parameter corresponds to an existent **Job Named Property** for the print job specified with the *JobId* parameter. If this verification fails, return ERROR\_NOT\_FOUND.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Delete the property specified by the *pszName* parameter for the print job specified by the *JobId* parameter.
* Return the status of the operation.

##### RpcEnumJobNamedProperties (Opnum 113)

**RpcEnumJobNamedProperties** enumerates the **Job Named Properties** (section [3.1.1](#Section_1bd61448db4b4284a4b18346732402d4)) for the specified print job.[<462>](#Appendix_A_462)

1. DWORD RpcEnumJobNamedProperties(
2. [in] PRINTER\_HANDLE hPrinter,
3. [in] DWORD JobId,
4. [out] DWORD\* pcProperties,
5. [out, size\_is(,\*pcProperties)] RPC\_PrintNamedProperty\*\* ppProperties
6. );

**hPrinter:** A [PRINTER\_HANDLE (section 2.2.1.1.4)](#Section_9c5d49b8692047b0946687ba910e6071) to a printer object, job object, or server object that was opened by [RpcAddPrinter (section 3.1.4.2.3)](#Section_bd0bb8d526fd470581f5614d32b9f38f), [RpcAddPrinterEx (section 3.1.4.2.15)](#Section_3af45717308f4b4f8428430d12856110), [RpcOpenPrinter (section 3.1.4.2.2)](#Section_989357e2446e4872bb381dce21e1313f), or [RpcOpenPrinterEx (section 3.1.4.2.14)](#Section_0e81ce1872b146c38584a205393b04ff).

**JobId:** The identifier of a print job. This value MUST NOT be zero.

**pcProperties:** On successful return from this call, this parameter is a pointer to the address of an array of [RPC\_PrintNamedProperty (section 2.2.1.14.2)](#Section_31bdf5289b8947f9823ba45bbc877716) structures returned. This pointer MUST NOT be NULL.

**ppProperties:** On successful return from this call, this parameter is a pointer to the address of an array of RPC\_PrintNamedProperty structures returned. This pointer MUST NOT be NULL.

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the server MUST validate the following:

* Perform the validation steps that are specified in [PRINTER\_HANDLE Parameters (section 3.1.4.1.11)](#Section_10bfa8c2d4564ad0912de674766a36c8).
* Verify that the value of the *JobId* parameter corresponds to a job in the list of jobs. If the object specified by the *hPrinter* parameter is a server object, search for a print job in each printer in the list of printers on the print server. If the object specified by the *hPrinter* parameter is a printer object, search for a print job only in the list of print jobs for the specified printer. If the object specified by the *hPrinter* parameter is a job object, compare the identifier of this print job with the specified *JobId*. If this verification fails, return ERROR\_INVALID\_PARAMETER.
* Verify that the *pcProperties* and *ppProperties* pointers are not NULL. If this verification fails, return ERROR\_INVALID\_PARAMETER.
* Verify that the buffer to contain the array of **RPC\_PrintNamedProperty** structures can be successfully allocated. If this verification fails, return ERROR\_NOT\_ENOUGH\_MEMORY.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Allocate and initialize a buffer containing the array of **RPC\_PrintNamedProperty** structures to be returned to the client via the *ppProperties* parameter.
* Return the number of enumerated properties (the number of returned **RPC\_PrintNamedProperty** structures) in the *pcProperties* parameter.
* Return the status of the operation.

Upon a successful completion of this call the client SHOULD free the buffer specified by the *ppProperties* parameter.

#### Branch Office Print Remote Logging Methods

This section specifies methods for processing **Branch Office Print Remote Log Entries** (section [3.1.1](#Section_1bd61448db4b4284a4b18346732402d4)) for a specified printer.

| Method | Description |
| --- | --- |
| [RpcLogJobInfoForBranchOffice (section 3.1.4.13.1)](#Section_65a54d4df30b41438299d61f086ac7fe) | RpcLogJobInfoForBranchOffice processes one or more **Branch Office Print Remote Log Entries** by writing them to the Microsoft-Windows-PrintService/Admin and Microsoft-Windows-PrintService/Operations event channels.  Opnum: 116 |

##### RpcLogJobInfoForBranchOffice (Opnum 116)

RpcLogJobInfoForBranchOffice processes one or more **Branch Office Print Remote Log Entries** (section [3.1.1](#Section_1bd61448db4b4284a4b18346732402d4)).[<463>](#Appendix_A_463)

1. DWORD RpcLogJobInfoForBranchOffice(
2. [in] PRINTER\_HANDLE hPrinter,
3. [in, ref] RPC\_BranchOfficeJobDataContainer\* pBranchOfficeJobDataContainer
4. );

**hPrinter:** A [PRINTER\_HANDLE (section 2.2.1.1.4)](#Section_9c5d49b8692047b0946687ba910e6071) to a printer object that was opened by [RpcAddPrinter (section 3.1.4.2.3)](#Section_bd0bb8d526fd470581f5614d32b9f38f), [RpcAddPrinterEx (section 3.1.4.2.15)](#Section_3af45717308f4b4f8428430d12856110), [RpcOpenPrinter (section 3.1.4.2.2)](#Section_989357e2446e4872bb381dce21e1313f), or [RpcOpenPrinterEx (section 3.1.4.2.14)](#Section_0e81ce1872b146c38584a205393b04ff).

**pBranchOfficeJobDataContainer:** A pointer to an [RPC\_BranchOfficeJobDataContainer (section 2.2.1.2.17)](#Section_8bed3226cdef4b90b762b35f1a3ad362) structure that contains one or more [RPC\_BranchOfficeJobData (section 2.2.1.15.2)](#Section_6e442be62c54497593c685895289ba0d) structures, each of which holds a single **Branch Office Print Remote Log Entry**.

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

On receiving this message, the server MUST validate the following:

* Perform the validation steps that are specified in [PRINTER\_HANDLE Parameters (section 3.1.4.1.11)](#Section_10bfa8c2d4564ad0912de674766a36c8). This method SHOULD assume that the handle to the printer object can be used without further access checks.
* Verify that the value of the *pBranchOfficeJobDataContainer* parameter is a non-NULL pointer, and if that verification fails, return ERROR\_INVALID\_PARAMETER.

If parameter validation fails, the server MUST fail the operation immediately and return a nonzero error response to the client. Otherwise, the server MUST process the message and compose a response to the client as follows:

* Create event channel entries that correspond to the data in the array of RPC\_BranchOfficeJobData structures in the RPC\_BranchOfficeJobDataContainer structure specified by the *pBranchOfficeJobDataContainer* parameter. The number of structures is specified by the **cJobDataEntries** member of the container structure.
* Return the status of the operation.

### Timer Events

No protocol timer events are required on the server beyond the timers required in the underlying remote procedure call (RPC) protocol.

### Other Local Events

No additional local events are used on the client beyond the events maintained in the underlying remote procedure call (RPC) protocol.

## Client Details

### Abstract Data Model

This section describes a conceptual model of a possible data organization that a print client implementation might need to maintain in order to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This specification does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this specification.

The following abstract data model is used to support a print client operating in branch office print mode.[<464>](#Appendix_A_464)

**Branch Office Print Remote Log Offline Archive:** When branch office print remote logging is enabled (section [2.2.3.11](#Section_4a81aa1c469e49418c4db8c4093b7fc2)), a print client can create certain event channel entries on the print server in response to Windows Events while processing a print job. Those entries are in the form of **Branch Office Print Remote Log Entries** (section [3.1.1](#Section_1bd61448db4b4284a4b18346732402d4)) in [Branch Office Print Remote Logging Structures (section 2.2.1.15)](#Section_b331caea43e24a00b5031969e7a0ab92), and they are sent to the server in a [RPC\_BranchOfficeJobDataContainer (section 2.2.1.2.17)](#Section_8bed3226cdef4b90b762b35f1a3ad362) by using the [RpcLogJobInfoForBranchOffice (section 3.1.4.13.1)](#Section_65a54d4df30b41438299d61f086ac7fe) method.

If a print client is unable to contact the print server when processing a branch office print job, printing-related Windows Events are written to the **Branch Office Print Remote Log Offline Archive**. When contact with the print server is restored, those entries can be retransmitted.

**Branch Office Print Remote Log Offline Archive Overflow Flag:** If the Branch Office Print Remote Log Offline Archive exceeds its maximum size (section 2.2.3.11), the Branch Office Print Remote Log Offline Archive Overflow Flag is set to TRUE.[<465>](#Appendix_A_465) It is initialized to FALSE, and it is reset to FALSE after the connection with the print server has been restored and the server has been notified of the overflow.

To notify the print server of an overflow of the offline archive, the print client creates a **Branch Office Print Remote Log Entry** in a [RPC\_BranchOfficeLogOfflineFileFull (section 2.2.1.15.7)](#Section_55fcc7c1d63e448888cd8ce5d9dd945c) structure and sends it to the server.

### Timers

No protocol timers are required beyond those used internally by remote procedure call (RPC) ([[MS-RPCE]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-RPCE%5d.pdf) section 3.2.3.2) to implement resiliency to network outages.

### Initialization

The client MUST perform initialization according to the following rules when calling an RPC method:

* Either create an RPC binding handle to the server or use an RPC context handle. Details concerning binding handles are as specified in [[C706]](http://go.microsoft.com/fwlink/?LinkId=89824).
* Use context handles across multiple calls to the server for methods taking a [PRINTER\_HANDLE](#Section_9c5d49b8692047b0946687ba910e6071).
* Use handles bound to a single call to the server for name-based methods taking a [STRING\_HANDLE](#Section_59883eb85ce64ad4bdd2599434cfdcf2). A STRING\_HANDLE\_BIND method MUST be implemented by the client.
* Reuse a context handle in multiple invocations when creating a print job, such as in a call to [RpcOpenPrinter](#Section_989357e2446e4872bb381dce21e1313f) followed by multiple calls to [RpcStartPagePrinter](#Section_76261f544c634fc2a2dabaee584e076b) and [RpcWritePrinter](#Section_6dd263aac8b547748d211daa0d0d9920). For an example of this sequence of calls, see section [3.2.4.2.1](#Section_e542c06ce139499ea500d001f317bf5a).
* A context handle SHOULD be reused in multiple invocations when getting or setting information on a printer, such as in a call to RpcOpenPrinter followed by multiple calls to [RpcGetPrinter](#Section_ccbf288a0768416dbede0ef1987fc286), [RpcGetPrinterData](#Section_9c04a0f2adc443938df47c36a14183f5), [RpcSetPrinter](#Section_382997b5b02a4b6a8c6849a64b8c45f8), or other methods taking a PRINTER\_HANDLE or [GDI\_HANDLE](#Section_ec9126fabc33423fadfc1bf52bd7890e).
* When creating the RPC binding handle on the named pipe **\pipe\spoolss**, the client MUST specify an **ImpersonationLevel** of 2 (**Impersonation** [[MS-SMB2]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-SMB2%5d.pdf) (section 2.2.13).

### Message Processing Events and Sequencing Rules

This protocol MUST indicate to the remote procedure call (RPC) runtime ([[MS-RPCE]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-RPCE%5d.pdf) section 3) that:

* It is to perform a strict network data representation (NDR) data consistency check at target level 6.0.
* It is to reject a NULL unique or full pointer with a nonzero conformant value.

The print client SHOULD ignore errors returned from the RPC server and SHOULD notify the application invoker of the error received in the higher layer. Unless otherwise specified, no special message processing is required on the client beyond that required in the underlying RPC protocol. [<466>](#Appendix_A_466)

#### Client-Side Notification Processing Methods

This section specifies processing rules for the notification-processing methods that a print client system MUST implement in its *Local Server* component to handle notifications from a print server.

| Method | Description |
| --- | --- |
| [RpcReplyOpenPrinter](#Section_7fcd3036d45a4ec7b081f2b860e66676) | Establishes a context handle from a print server to a print client.  Opnum 58 |
| [RpcRouterReplyPrinter](#Section_38bf5d2ce4a744b2b16f91ff8226f836) | Handles a notification from a print server.  Opnum 59 |
| [RpcReplyClosePrinter](#Section_d6c73ecbc02b4150a01bad73a642990a) | Closes the notification channel between a print server and a print client.  Opnum 60 |
| [RpcRouterReplyPrinterEx](#Section_1b44cb8295b5485c9ec90360c3a3ecb0) | Handles a notification from a print server.  Opnum 66 |

All these methods are request/response remote procedure call (RPC) methods. They MUST return zero to indicate successful completion and nonzero values to indicate failure, except where specifically described.

##### RpcReplyOpenPrinter (Opnum 58)

RpcReplyOpenPrinter establishes a context handle from a print server to a print client.[<467>](#Appendix_A_467) The server uses the RPC context handle returned by this method to send notification data to the client machine.

1. DWORD RpcReplyOpenPrinter(
2. [in, string] STRING\_HANDLE pMachine,
3. [out] PRINTER\_HANDLE\* phPrinterNotify,
4. [in] DWORD dwPrinterRemote,
5. [in] DWORD dwType,
6. [in, range(0,512)] DWORD cbBuffer,
7. [in, unique, size\_is(cbBuffer), disable\_consistency\_check]
8. BYTE\* pBuffer
9. );

**pMachine:** A string that specifies the print client computer name. It is synonymous with *pName*, as specified in [Print Server Name Parameters (section 3.1.4.1.4)](#Section_acb355a4913b4abb8cb5415aea160015).

**phPrinterNotify:** A pointer to a remote printer RPC context handle that will be used by a print server to send notifications to a print client. RPC context handles are specified in [[C706]](http://go.microsoft.com/fwlink/?LinkId=89824).

**dwPrinterRemote:** A value that is supplied to the server by the *dwPrinterLocal* parameter of a corresponding call to [RpcRemoteFindFirstPrinterChangeNotification (section 3.1.4.10.3)](#Section_b8b414d9f1cd4191bb6b87d09ab2fd83) or [RpcRemoteFindFirstPrinterChangeNotificationEx (section 3.1.4.10.4)](#Section_eb66b2211c1f4249b8bcc5befec2314d). This value MUST NOT be zero.

**dwType:** A value that MUST be 0x00000001.

**cbBuffer:** A value that SHOULD be set to zero when sent and MUST be ignored on receipt.

**pBuffer:** A pointer that SHOULD be set to NULL when sent and MUST be ignored on receipt.

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the client MUST validate parameters by verifying that the *pMachine* parameter corresponds to the current machine.

This method SHOULD execute without further access checks.

If parameter validation fails, the client MUST fail the operation immediately and return a nonzero error response to the server. Otherwise, the client MUST process the message as follows:

* Locate the notification state that is identified by the *dwPrinterRemote* parameter.
* Create a back channel RPC context handle and associate it with this notification state.
* Store the back channel RPC context handle in the handle pointed to by *phPrinterNotify*.
* Return the status of the operation.[<468>](#Appendix_A_468)

##### RpcRouterReplyPrinter (Opnum 59)

RpcRouterReplyPrinter handles a notification from a print server.[<469>](#Appendix_A_469)

1. DWORD RpcRouterReplyPrinter(
2. [in] PRINTER\_HANDLE hNotify,
3. [in] DWORD fdwFlags,
4. [in, range(0,512)] DWORD cbBuffer,
5. [in, unique, size\_is(cbBuffer), disable\_consistency\_check]
6. BYTE\* pBuffer
7. );

**hNotify:** A notification handle that was opened by the server using [RpcReplyOpenPrinter (section 3.2.4.1.1)](#Section_7fcd3036d45a4ec7b081f2b860e66676).

**fdwFlags:** A value that contains [Printer Change Flags (section 2.2.3.6)](#Section_ef51a734e033467eb45d4b5c06f440e7), which indicate changes in printer configuration values.

**cbBuffer:** A value that SHOULD be set to zero when sent and MUST be ignored on receipt.

**pBuffer:** A pointer that SHOULD be set to NULL when sent and MUST be ignored on receipt.

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the print client MUST validate parameters as follows:

* Verify that the *hNotify* parameter is an RPC context handle to a notification object opened by calling RpcReplyOpenPrinter. If that verification fails, ERROR\_INVALID\_HANDLE MUST be returned. This method SHOULD assume that this handle can be used without further access checks.

If parameter validation fails, the client MUST fail the operation immediately and return a nonzero error response to the server. Otherwise, the client MUST process the message as follows:

* Capture the *fdwFlags* in the notification state it maintains.
* If the operation is successful, the client MUST send the received data to the caller that registered for the notifications, by calling [RpcRemoteFindFirstPrinterChangeNotification (section 3.1.4.10.3)](#Section_b8b414d9f1cd4191bb6b87d09ab2fd83) or [RpcRemoteFindFirstPrinterChangeNotificationEx (section 3.1.4.10.4)](#Section_eb66b2211c1f4249b8bcc5befec2314d).

##### RpcReplyClosePrinter (Opnum 60)

RpcReplyClosePrinter closes the notification channel between a print server and a print client.[<470>](#Appendix_A_470)

1. DWORD RpcReplyClosePrinter(
2. [in, out] PRINTER\_HANDLE\* phNotify
3. );

**phNotify:** A pointer to the notification context handle to close that was opened by [RpcReplyOpenPrinter (section 3.2.4.1.1)](#Section_7fcd3036d45a4ec7b081f2b860e66676).

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the client MUST validate parameters as follows:

* Verify that the *phNotify* parameter is an RPC context handle to a notification object that was opened using RpcReplyOpenPrinter. If that verification fails, ERROR\_INVALID\_HANDLE MUST be returned. This method SHOULD assume that this handle can be used without further access checks.

If parameter validation fails, the client MUST fail the operation immediately and return a nonzero error response to the server. Otherwise, the client MUST process the message as follows:

* Free the context handle associated with the notification state.
* Return a response to the client containing the output parameters and the status of the operation.
* If the operation is successful, the client MUST modify the notification state by removing the back channel context handle associated with it.

##### RpcRouterReplyPrinterEx (Opnum 66)

RpcRouterReplyPrinterEx handles a notification from a print server.[<471>](#Appendix_A_471)

1. DWORD RpcRouterReplyPrinterEx(
2. [in] PRINTER\_HANDLE hNotify,
3. [in] DWORD dwColor,
4. [in] DWORD fdwFlags,
5. [out] DWORD\* pdwResult,
6. [in] DWORD dwReplyType,
7. [in, switch\_is(dwReplyType)] RPC\_V2\_UREPLY\_PRINTER Reply
8. );

**hNotify:** A notification RPC context handle that was opened by [RpcReplyOpenPrinter (section 3.2.4.1.1)](#Section_7fcd3036d45a4ec7b081f2b860e66676).

**dwColor:** The value that was most recently specified by the client in the *dwColor* parameter of a call to [RpcRouterRefreshPrinterChangeNotification (section 3.1.4.10.5)](#Section_cb5d22ed0352430383d5b5afd32fb461).

**fdwFlags:** A value that contains [Printer Change Flags (section 2.2.3.6)](#Section_ef51a734e033467eb45d4b5c06f440e7), which indicate changes in printer configuration values.

**pdwResult:** A pointer to a value that contains [Change Notification Flags (section 2.2.3.2)](#Section_033938dbcdc74541819510b7b217d96a), which indicate how the client processed the notification.

**dwReplyType:** A value that MUST be zero.

**Reply:** A pointer to an [RPC\_V2\_UREPLY\_PRINTER](#Section_1ccdac5b0b2a4bd38afdd2c2589130fc) union, which contains a pointer to an [RPC\_V2\_NOTIFY\_INFO](#Section_79db335febe74a6f8533efdda32cc5d8) structure, which contains available notification data that matched the set of notifications that the client previously requested.

**Return Values:** This method MUST return zero (ERROR\_SUCCESS) to indicate successful completion or a nonzero Windows error code to indicate failure, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf).

Upon receiving this message, the print client MUST validate parameters as follows:

* Verify that the *hNotify* parameter is an RPC context handle to a notification object that was opened using RpcReplyOpenPrinter, and if that verification fails, return ERROR\_INVALID\_HANDLE, as specified in [MS-ERREF]. This method SHOULD assume that this handle can be used without further access checks.
* Verify that the value of the *dwColor* parameter matches the last value that was passed in the *dwColor* parameter in the call to RpcRouterRefreshPrinterChangeNotification; if that verification fails, set the **PRINTER\_NOTIFY\_INFO\_COLORMISMATCH** bit in the variable pointed to by *pdwResult* and return 0.

If parameter validation fails, the client MUST fail the operation immediately and return a nonzero error response to the server. Otherwise, the client MUST process the message as follows:

* Capture the *fdwFlags* in the notification state it maintains.
* Capture the notification data provided in the *Reply* parameter in the notification state.
* Store the result of processing the notification (Change Notification Flags) to the variable pointed to by *pdwResult*.

#### Client Interaction with the Print Server

This section contains sequence specifications that the client MUST follow to perform specific tasks on the print server. See [Protocol Examples (section 4)](#Section_42308ed8a001402fb40dd68e41e2608f) for additional information and sequence diagrams.

##### Printing a Document Using RpcStartDocPrinter

To print a document using [RpcStartDocPrinter (section 3.1.4.9.1)](#Section_90f95ff03ae04ddea83d35f457c710a6), the client MUST perform the following steps:

1. Invoke [RpcOpenPrinter (section 3.1.4.2.2)](#Section_989357e2446e4872bb381dce21e1313f), supplying the name of the target printer in the *pPrinterName* parameter and an *AccessRequired* [Access Value (section 2.2.3.1)](#Section_945c4df5496949c5b1ce59c1b3f35024) that includes **PRINTER\_ACCESS\_USE**.
2. Using the printer handle obtained from RpcOpenPrinter:
   1. Invoke RpcStartDocPrinter to initiate the print job.
   2. For each page in the print job:
      1. Optionally invoke [RpcStartPagePrinter](#Section_76261f544c634fc2a2dabaee584e076b) to begin the page.
      2. Invoke [RpcWritePrinter](#Section_6dd263aac8b547748d211daa0d0d9920) to send the client's print data to the printer.
      3. Optionally invoke [RpcEndPagePrinter](#Section_c4a17128584d402e9bac6a050e9da6d0) to end the page.
   3. Optionally invoke [RpcEndDocPrinter](#Section_209fcf3205b444aba3a4779cfae181ca) to end the print job.
3. If the client called RpcEndDocPrinter in step 2.3, the client can perform the following:
   1. Step 2 can be repeated for additional print jobs.
   2. Optionally invoke [RpcClosePrinter](#Section_8482c18677734fb58034e9bae7a91e52) when finished.
4. If the client did not call RpcEndDocPrinter in step 2.3, the client MUST invoke RpcClosePrinter, and the server MUST process the call, assuming an implicit call to RpcEndDocPrinter.

##### Enumerating Printers on a Print Server

To enumerate the printers on a print server, the client MUST perform the following steps:

1. Invoke [RpcEnumPrinters](#Section_0d83c60dec7546718a3b99517651ff1b), supplying the name of the target print server in the *Name* parameter, the types of printers to enumerate in the *Flags* parameter, an information level value in *Level*, zero in *cbBuf*, and a pointer to a variable to store the required buffer size in *pcbNeeded*.
2. While RpcEnumPrinters returns with ERROR\_INSUFFICIENT\_BUFFER, the client MUST:
   * Allocate new printer information buffer space with a size from the returned value for *pcbNeeded*.
   * Invoke RpcEnumPrinters, supplying the name of the target print server in the *Name* parameter, the types of printers to enumerate in the *Flags* parameter, an information level value in *Level*, a pointer to the printer information buffer in *pPrinterEnum*, the allocated size of the printer information buffer in *cbBuf*, a pointer to a variable to store the required buffer size in *pcbNeeded*, and a location to store the number of printer information items returned in *pcReturned*.

**Note**  Because the number of printers can change at any time, the client SHOULD be prepared to receive ERROR\_INSUFFICIENT\_BUFFER even after allocating the correct buffer size the first time.

##### Enumerating Jobs on a Printer

To enumerate the jobs that are currently queued to a printer, the client MUST perform the following steps:

1. Invoke [RpcOpenPrinter](#Section_989357e2446e4872bb381dce21e1313f), supplying the name of the target printer in the *pPrinterName* parameter and an *AccessRequired* value that includes [PRINTER\_ACCESS\_USE](#Section_945c4df5496949c5b1ce59c1b3f35024).
2. Using the printer handle that was obtained from RpcOpenPrinter, the client MUST:
   1. Set a local job position context to the desired starting index, typically zero.
   2. Set a local number of jobs to return in a single operation.[<472>](#Appendix_A_472)
   3. Until an [RpcEnumJobs](#Section_0c77f1ff08274f4da06d5a3d802ffc0c) call returns with a success status and a **pcReturned** pointing to a value of zero, or until the expected set of jobs has been returned, the client MUST:
      1. Invoke RpcEnumJobs, supplying the job position context in *FirstJob*, the number of jobs to return in a call in *NoJobs*, the desired information level in *Level*, a pointer to the job information buffer in *pJob*, the size of the job information buffer in *cbBuf*, a pointer to a variable to store the required buffer size in *pcbNeeded*, and a pointer to a variable to store the number of job information structures returned in *pcReturned*.
      2. While RpcEnumJobs returns with ERROR\_INSUFFICIENT\_BUFFER, as specified in [[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf), the client MUST:
         1. Allocate a new job information buffer with the size returned in *pcbNeeded*.
         2. Invoke RpcEnumJobs, supplying the job position context in *FirstJob*, the number of jobs to return in a call in *NoJobs*, the desired information level in *Level*, a pointer to the job information buffer in *pJob*, the size of the job information buffer in *cbBuf*, a pointer to a variable to store the required buffer size in *pcbNeeded*, and a pointer to a variable to store the number of job information structures in *pcReturned*.

**Note**  Because the number of jobs can change at any time, the client SHOULD be prepared to receive ERROR\_INSUFFICIENT\_BUFFER even after allocating the correct buffer size the first time.

* + 1. Increase the local job position context by the value supplied in *pcReturned*.

1. The client SHOULD invoke [RpcClosePrinter](#Section_8482c18677734fb58034e9bae7a91e52) with the printer handle obtained from RpcOpenPrinter or SHOULD repeat step 2 if there are further job enumeration requests to make.

##### Receiving Notifications from a Print Server

To receive notifications for a printing event, a print client MUST perform the following steps:

1. Invoke [RpcOpenPrinter (section 3.1.4.2.2)](#Section_989357e2446e4872bb381dce21e1313f), supplying the name of the target printer in the *pPrinterName* parameter and an *AccessRequired* value that includes **PRINTER\_ACCESS\_USE** from [Access Values (section 2.2.3.1)](#Section_945c4df5496949c5b1ce59c1b3f35024).
2. Using the printer handle that was obtained from RpcOpenPrinter, invoke [RpcRemoteFindFirstPrinterChangeNotificationEx (section 3.1.4.10.4)](#Section_eb66b2211c1f4249b8bcc5befec2314d), supplying the notification flags and the job and printer fields that notifications are to be delivered for. The call MUST also supply a value in *dwPrinterLocal* that the client will use to identify the source for the later notifications.

The print server opens a channel to the client as a result of processing this call by calling the client's [RpcReplyOpenPrinter (section 3.2.4.1.1)](#Section_7fcd3036d45a4ec7b081f2b860e66676) method. Therefore, the client MUST implement the Print System Remote Protocol interface [Client-Side Notification Processing Methods (section 3.2.4.1)](#Section_ee34aeeec9464991be9fc6d0049fb023).

1. The client MUST process notifications as follows:
   * Process an RpcReplyOpenPrinter call, using the value in *dwPrinterRemote* to determine the client context established by the *dwPrinterLocal* parameter in a previous RpcRemoteFindFirstPrinterChangeNotificationEx call.
   * This call MUST produce an remote procedure call (RPC) handle that the later notifications use.
   * Process [RpcRouterReplyPrinter (section 3.2.4.1.2)](#Section_38bf5d2ce4a744b2b16f91ff8226f836) and [RpcRouterReplyPrinterEx (section 3.2.4.1.4)](#Section_1b44cb8295b5485c9ec90360c3a3ecb0) method calls. The server calls RpcRouterReplyPrinter for notifications that do not have associated data and RpcRouterReplyPrinterEx for notifications that do have associated data. The value of *fdwFlags* MUST specify the notification type and any associated data.
2. To terminate the notifications, the client SHOULD invoke [RpcFindClosePrinterChangeNotification](#Section_f50c8a861b2d4124b2205b840aacd2c7), supplying the printer handle obtained from RpcOpenPrinter.

The server closes the channel to the client as a result of processing this call by calling [RpcReplyClosePrinter](#Section_d6c73ecbc02b4150a01bad73a642990a) on the client.

1. The client MUST process an RpcReplyClosePrinter to terminate the notification sequence. This provides the RPC handle from the associated RpcReplyOpenPrinter call.
2. The client SHOULD call [RpcClosePrinter](#Section_8482c18677734fb58034e9bae7a91e52). If the client has not called RpcFindClosePrinterChangeNotification in step 4, the server MUST implicitly close the notification channel, and in the processing of that, call RpcReplyClosePrinter, which MUST be processed by the client as specified in step 5.

##### Announcing Shared Printers to Print Servers

To announce its shared printers to print servers, the client MUST perform these steps:

1. Make a policy-specific determination whether shared printers should be enumerated to print servers.[<473>](#Appendix_A_473)
2. If shared printers should be enumerated, for each printer installed on the client that has the [PRINTER\_ATTRIBUTE\_SHARED](#Section_1625e9d929e448f4b83d3bd0fdaea787) set, create a [PRINTER\_CONTAINER](#Section_ad7d4d4e0feb45e093edb3b2b9a313d1) with **Level** set to 0x00000001, and populate it with a [PRINTER\_INFO\_1](#Section_0d1a296b7027441c83938aa5c3764b27) describing the printer, and then call the print server's [RpcAddPrinter](#Section_bd0bb8d526fd470581f5614d32b9f38f) or [RpcAddPrinterEx](#Section_3af45717308f4b4f8428430d12856110) method.[<474>](#Appendix_A_474)

##### Adding a Printer to a Print Server

To add a printer to a print server, the client performs these steps:

1. The client MAY use methods defined by this protocol to query the print server for information used to initialize other data structures.[<475>](#Appendix_A_475)
2. The client SHOULD call the print server's [RpcEnumPrinterDrivers](#Section_857d00ac36824a0d86ca3d3c372e5e4a) to determine whether a printer driver for the new printer is already installed on the server.
3. If a printer driver is not already installed, the client SHOULD call [RpcAddPrinterDriver](#Section_f23a75191c7740699acea6d8eae47c22) or [RpcAddPrinterDriverEx](#Section_b96cc49759e54510ab045484993b259b) to install a printer driver for the new printer.
4. The client MUST allocate a [PRINTER\_CONTAINER](#Section_ad7d4d4e0feb45e093edb3b2b9a313d1) structure and populate it with a [PRINTER\_INFO\_2](#Section_d40734e1ac9b4b2a862883a1bc7bbeed) structure describing the new printer.
5. The client MUST allocate a [DEVMODE\_CONTAINER](#Section_c2b14fe764794460a7ba633a845dd61a) and populate it with the default DEVMODE for the new printer.
6. The client MUST allocate a [SECURITY\_CONTAINER](#Section_6cf2b4ff1b2b453c87f985243823cea8) and populate it with a SECURITY\_DESCRIPTOR containing the security information for the new printer.
7. The client MUST call the print server's [RpcAddPrinter](#Section_bd0bb8d526fd470581f5614d32b9f38f) with the print server's name, and the CONTAINER parameters from steps 4, 5, and 6. Alternatively, the client can use the [RpcAddPrinterEx](#Section_3af45717308f4b4f8428430d12856110) and specify an additional [SPLCLIENT\_CONTAINER](#Section_cca15c7257d342fc91e749442b623c11) that describes the client in more detail. RpcAddPrinterEx returns a [PRINTER\_HANDLE](#Section_9c5d49b8692047b0946687ba910e6071) to the newly added printer in the variable pointed to by *pHandle*. The client SHOULD close that handle using [RpcClosePrinter](#Section_8482c18677734fb58034e9bae7a91e52) when it no longer requires it.

### Timer Events

No protocol timer events are required on the client beyond the timers required in the underlying RPC protocol.

### Other Local Events

A client's invocation of each method is typically the result of local application activity. The local application on the client computer specifies values for all input parameters. No other higher-layer triggered events are processed. The values specified for input parameters are described in section [2](#Section_2900a90635e04ac3b3c062272c0c5110).

No additional local events are used on the client beyond the events maintained in the underlying remote procedure call (RPC) protocol.

# Protocol Examples

## Adding a Printer to a Server

To add a printer ("HP LaserJet 4") to a print server ("CORPSERV"), a client ("TESTCLT") performs the following steps:

1. Enumerate existing printer drivers using [RpcEnumPrinterDrivers](#Section_857d00ac36824a0d86ca3d3c372e5e4a).
   * The client obtains the server platform by calling [RpcGetPrinterData](#Section_9c04a0f2adc443938df47c36a14183f5) on a server object with the "Architecture" key value.
   * The client calls RpcEnumPrinterDrivers, specifying the environment parameter to match the server platform.
2. RpcEnumPrinterDrivers( L"\\\\CORPSERV"., L"Windows NT x86", 1, NULL, 0, &countBytesNeeded, &driversFound )
   * The server returns ERROR\_INSUFFICIENT\_BUFFER and sets countBytesNeeded large enough to store [\_DRIVER\_INFO\_1](#Section_59eab35482fd4299884e9a1226073a4a) structures for all drivers matching the specified environment "Windows NT x86".
   * The client allocates memory, setting the size of driverInfo1[] to countBytesNeeded, and calls RpcEnumPrinterDrivers again.
3. RpcEnumPrinterDrivers ( L"\\\\CORPSERV", L"Windows NT x86", 1, driverInfo1, countBytesNeeded, &countBytesNeeded, &driversFound )
   * The server writes a \_DRIVER\_INFO\_1 structure for each driver matching the specified environment ("Windows NT x86") to the output buffer, writes the number of \_DRIVER\_INFO\_1 structures to driversFound, and returns 0 (success).

**Note**  If the number of drivers on the server has increased between the first and second call to RpcEnumPrinterDrivers, the server returns ERROR\_INSUFFICIENT\_BUFFER from the second call as well. In that case, the server updates countBytesNeeded and the client needs to allocate more memory and call RpcEnumPrinterDrivers again.

1. Select an existing printer driver or add a new printer driver using [RpcAddPrinterDriver](#Section_f23a75191c7740699acea6d8eae47c22).
   * Assume the server returned a driver named "HP LaserJet 4".
   * The client calls [RpcEnumPorts](#Section_4f018733e78e4088ac48f6e19ee00863) to enumerate the available ports. This process is analogous to the previous step, which enumerated printer drivers using RpcEnumPrinterDrivers.
   * The client displays a dialog box so the end user can pick the driver, enter a desired port, and optionally enter a share name for the new printer.
   * If the driver does not already exist, or the client requests to update the driver, use RpcAddPrinterDriver to add the driver to the print server, as shown in [4.2](#Section_bc4d18308e1f428a9073d655916494cf).
2. Populate a [PRINTER\_INFO\_2](#Section_d40734e1ac9b4b2a862883a1bc7bbeed) structure with information about the new printer and call [RpcAddPrinter](#Section_bd0bb8d526fd470581f5614d32b9f38f).
   * The client allocates and zero-initializes a PRINTER\_INFO\_2 structure.
   * The client sets the following members of the structure:
3. pPrinterName = L"HP LaserJet 4" /\* Typically set to the driver name \*/
4. pShareName = L"My Printer" /\* Any name the user selects \*/
5. pPortName = L"172.10.10.10" /\* A port that exists on the server \*/
6. pDriverName = L"HP LaserJet 4" /\* Driver selected in previous step \*/

For pPortName, the client may have previously enumerated the server's ports using RpcEnumPorts or the user may know a valid port name.

The client initializes all other members of the structure to zero or NULL, as appropriate. Or, the client specifies higher DRIVER\_INFO levels in the call to RpcEnumPrinterDrivers to obtain more details to initialize these structure members. Or, the client prompts the user to specify values such as priorities or times at which the printer is available and initializes these members with the values specified by the user.

* + The client allocates a [PRINTER\_CONTAINER (section 2.2.1.2.9)](#Section_ad7d4d4e0feb45e093edb3b2b9a313d1) structure and initializes it to contain the prepared PRINTER\_INFO\_2.
  + The client allocates a [DEVMODE\_CONTAINER](#Section_c2b14fe764794460a7ba633a845dd61a) devmodeContainer structure, and optionally initializes it with a [DEVMODE](#Section_0136fad4f02c44dd857f820477b19c1d) structure.
  + The client allocates a [SECURITY\_CONTAINER](#Section_6cf2b4ff1b2b453c87f985243823cea8) securityContainer structure, and optionally initializes it with a SECURITY\_DESCRIPTOR.
  + The client calls RpcAddPrinter to add the printer.

1. RpcAddPrinter( L"\\\\CORPSERV", &printerContainer, &devmodeContainer, &securityContainer, &hPrinter )
   * The server creates the print queue, writes the handle to hPrinter, and returns 0 (success).
2. Close the returned [PRINTER\_HANDLE](#Section_9c5d49b8692047b0946687ba910e6071) using [RpcClosePrinter](#Section_8482c18677734fb58034e9bae7a91e52).
   * When the client is done using the print queue, the client closes it.
3. RpcClosePrinter( &hPrinter )
   * The server frees the memory associated with the print queue handle, sets hPrinter to NULL, and returns 0 (success).

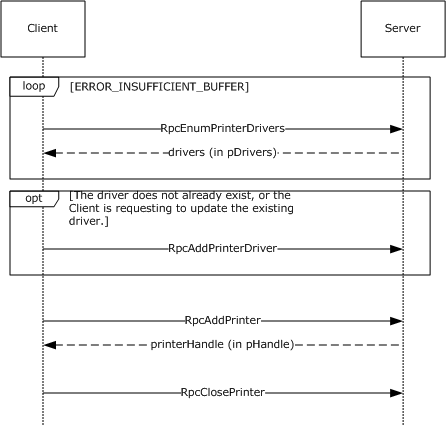


Figure : Adding a printer to a server

## Adding a Printer Driver to a Server

To add or update a printer driver ("OEM Printer Driver") to a print server ("CORPSERV"), a client ("TESTCLT") performs the following steps.

1. Enumerate existing printer drivers using [RpcEnumPrinterDrivers](#Section_857d00ac36824a0d86ca3d3c372e5e4a).

See [4.1](#Section_a33ebcedd8d544dba7287a77d0e40231) for an example using RpcEnumPrinterDrivers.

1. If the printer driver does not already exist or the client requests to update the printer driver, use [RpcAddPrinterDriver](#Section_f23a75191c7740699acea6d8eae47c22) to add the driver to the print server.
   * The client must make sure the files for the printer driver are in a location accessible to the server. For that purpose, the client can share a local directory containing the files, or use [[MS-SMB]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-SMB%5d.pdf) to place the files into a directory on the server.
   * The client then allocates and populates a [DRIVER\_INFO\_2](#Section_39bbfc3087684cd49930434857e2c2a2) structure as follows:
2. pName = L"OEM Printer Driver";
3. pEnvironment = L"Windows NT x86"; /\* Environment the driver is compatible with \*/
4. pDriverPath = "\\\\CORPSERV\\C$\\DRIVERSTAGING\\OEMDRV.DLL";
5. pDataFile = "\\\\CORPSERV\\C$\\DRIVERSTAGING\\OEMDATA.DLL";
6. pConfigFile = "\\\\CORPSERV\\C$\\DRIVERSTAGING\\OEMUI.DLL";
   * The client allocates a [DRIVER\_CONTAINER](#Section_3a3f9cf78ec44921b1f686cf8d139bc2) driverContainer structure and initializes it to contain the DRIVER\_INFO\_2 structure.
   * The client calls RpcAddPrinterDriver.
7. RpcAddPrinterDriver( L"\\\\CORPSERV", &driverContainer );
   * The server adds the printer driver and returns 0 (success).

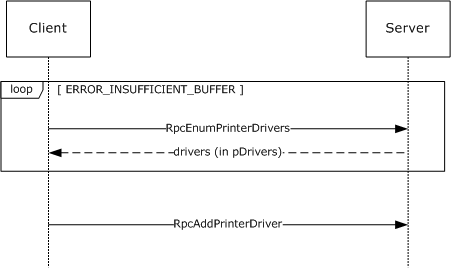


Figure : Adding a printer driver to a server

## Enumerating and Managing Printers

To manage printers on a print server ("CORPSERV"), a client ("TESTCLT") performs the following steps.

1. Enumerate existing printers using [RpcEnumPrinters](#Section_0d83c60dec7546718a3b99517651ff1b).
   * The client calls RpcEnumPrinters.
2. RpcEnumPrinters( PRINTER\_ENUM\_NAME, L"\\\\CORPSERV", 2, NULL, 0, &countBytesNeeded, &printersFound );
   * The server returns ERROR\_INSUFFICIENT\_BUFFER and sets countBytesNeeded to the size needed to store [\_PRINTER\_INFO\_2](#Section_9e3273757ac9419fa61061b88bbbd918) structures for all shared print queues.
   * The client allocates memory in printerInfo2[] with the size set to countBytesNeeded.
   * The client calls RpcEnumPrinters.
3. RpcEnumPrinters( PRINTER\_ENUM\_NAME, L"\\\\CORPSERV", 2, printerInfo2, countBytesNeeded, &countBytesNeeded, &printersFound );
   * The server writes a \_PRINTER\_INFO\_2 structure for each shared print queue to the output buffer, writes the number of \_PRINTER\_INFO\_2 structures to printersFound, and returns 0 (success).

**Note**  If the number of shared print queues on the server has increased between the first and second call to RpcEnumPrinters, the server returns ERROR\_INSUFFICIENT\_BUFFER from the second call as well. In that case, the server will update countBytesNeeded and the client needs to reallocate more memory and repeat the call to RpcEnumPrinters.

1. Open a handle to the print queue using [RpcOpenPrinter](#Section_989357e2446e4872bb381dce21e1313f).
   * The client selects a print queue from the \_PRINTER\_INFO\_2 structure and uses the **pPrinterName** or **pShareName** to open the print queue handle as follows:
     + The client allocates and initializes a [DEVMODE\_CONTAINER](#Section_c2b14fe764794460a7ba633a845dd61a) devmodeContainer structure.
     + The client calls RpcOpenPrinter.
2. RpcOpenPrinter( L"\\\\CORPSERV\\My Printer", &hPrinter, L"RAW", &devmodeContainer, PRINTER\_ACCESS\_USE);
   * The server allocates printer handle, writes it to hPrinter, and returns 0 (success).
3. Retrieve current information about a printer using [RpcGetPrinter](#Section_ccbf288a0768416dbede0ef1987fc286).
   * The client calls RpcGetPrinter.
4. RpcGetPrinter(hPrinter, 2, NULL, 0, &countBytesNeeded);
   * The server returns ERROR\_INSUFFICIENT\_BUFFER and sets countBytesNeeded to store a \_PRINTER\_INFO\_2 structure for the print queue.
   * The client allocates memory in printerInfo2[] with size set to countBytesNeeded.
   * The client calls RpcGetPrinter.
5. RpcGetPrinter( hPrinter, 2, printerInfo2, countBytesNeeded, &countBytesNeeded );
   * The server writes a \_PRINTER\_INFO\_2 structure for the print queue to the output buffer and returns 0 (success).

**Note**  If the size of data for the print queue on the server has increased between the first and second call to RpcGetPrinter, the server returns ERROR\_INSUFFICIENT\_BUFFER from the second call as well. That can happen under a race condition if another client changes the print queue data. In that case, the server will update countBytesNeeded and the client needs to reallocate more memory and repeat the call to RpcGetPrinter.

1. Use [RpcSetPrinter](#Section_382997b5b02a4b6a8c6849a64b8c45f8) to modify the state of the printer.
   * The client allocates a [PRINTER\_INFO\_2](#Section_d40734e1ac9b4b2a862883a1bc7bbeed) structure and populates it with members from the previously acquired \_PRINTER\_INFO\_2. The client changes those members that require change:
2. pLocation = L"Building 84, Room 1129";
   * The client allocates a [PRINTER\_CONTAINER](#Section_ad7d4d4e0feb45e093edb3b2b9a313d1) printerContainer structure and initializes it to contain the prepared PRINTER\_INFO\_2.
   * The client allocates a DEVMODE\_CONTAINER devmodeContainer structure, and optionally initializes it with a [DEVMODE](#Section_0136fad4f02c44dd857f820477b19c1d) structure.
   * The client allocates a [SECURITY\_CONTAINER](#Section_6cf2b4ff1b2b453c87f985243823cea8) securityContainer structure, and optionally initializes it with a SECURITY\_DESCRIPTOR.
   * The client calls RpcSetPrinter.
3. RpcSetPrinter( hPrinter, &printerContainer, &devmodeContainer, &securityContainer, 0 );
   * The server modifies the print queue and returns 0 (success).
4. Close the printer using [RpcClosePrinter](#Section_8482c18677734fb58034e9bae7a91e52).
   * The client calls RpcClosePrinter.
5. RpcClosePrinter( &hPrinter );
   * The server frees the memory associated with the print queue handle, sets hPrinter to NULL, and returns 0 (success).

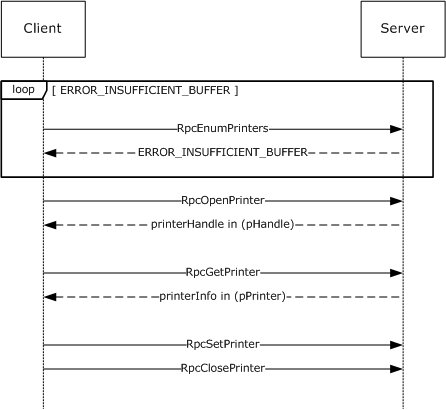


Figure : Enumerating and managing printers on a server

## Enumerating Jobs and Modifying Job Settings

To enumerate print jobs on a server ("CORPSERV"), modify job settings, or change job priorities, the client ("TESTCLT") performs the following steps.

1. Open the printer using [RpcOpenPrinter](#Section_989357e2446e4872bb381dce21e1313f).
   * The client allocates and initializes a *devmodeContainer* structure (see [DEVMODE\_CONTAINER (section 2.2.1.2.1)](#Section_c2b14fe764794460a7ba633a845dd61a).
   * The client calls RpcOpenPrinter.
2. RpcOpenPrinter( L"\\\\CORPSERV\\My Printer", &hPrinter, L"RAW", &devmodeContainer, PRINTER\_ACCESS\_USE );
   * The server allocates a printer handle, writes it to *hPrinter*, and returns 0 (success).
3. Enumerate jobs scheduled for printing on the printer using [RpcEnumJobs](#Section_0c77f1ff08274f4da06d5a3d802ffc0c).
   * The client calls RpcEnumJobs with *FirstJob* set to 0 and *NoJobs* set to the maximum unsigned integer to return all jobs.
4. RpcEnumJobs( hPrinter, 0, 0xFFFFFFFF, 1, NULL, 0, &countBytesNeeded, &jobsFound );
   * The server returns ERROR\_INSUFFICIENT\_BUFFER and sets *countBytesNeeded* to store [\_JOB\_INFO\_1](#Section_34f34c553d024991b5d3e24159a90936) structures for all shared print queues.
   * The client allocates memory in *jobInfo1[]* with size set to *countBytesNeeded*.
   * The client calls RpcEnumJobs.
5. RpcEnumJobs( hPrinter, 0, 0xFFFFFFFF, 1, jobInfo1, countBytesNeeded, &countBytesNeeded, &jobsFound );
   * The server writes \_JOB\_INFO\_1 for all jobs on the print queue to the output buffer, writes the number of \_JOB\_INFO\_1 structures to *jobsFound*, and returns 0 (success).

**Note**  If the number of jobs on the print queue on the server has increased between the first and second call to RpcEnumJobs, the server returns ERROR\_INSUFFICIENT\_BUFFER from the second call as well. In that case, the server will update *countBytesNeeded* and the client is required to reallocate more memory and repeat the call to RpcEnumJobs.

1. Modify job settings or job priority using [RpcSetJob](#Section_f978bd2ce99a43ffba0802e892948dac).
   * The client picks a job from the list of \_JOB\_INFO\_1 structures that it requests to modify. For this example, we assume the **JobId** is 12 and we want to cancel the job.
   * The client allocates and zero-initializes a *jobContainer* structure (see [JOB\_CONTAINER (section 2.2.1.2.5)](#Section_0cd5f47b54214b199ed05a29d63f3c2a).
   * To modify job settings, the client calls RpcSetJob with the *Command* parameter set to zero.
2. RpcSetJob( hPrinter, 12, &jobContainer, 0 );
   * To control the processing of a job, such as to cancel it, the client calls RpcSetJob with a non-zero *Command* parameter.
3. RpcSetJob( hPrinter, 12, &jobContainer, JOB\_CONTROL\_CANCEL );
   * The server modifies the specified print job and returns 0 (success).
4. The client closes the printer using [RpcClosePrinter](#Section_8482c18677734fb58034e9bae7a91e52).
5. RpcClosePrinter( &hPrinter );

The server frees the memory associated with the print queue handle, sets *hPrinter* to NULL, and returns 0 (success).

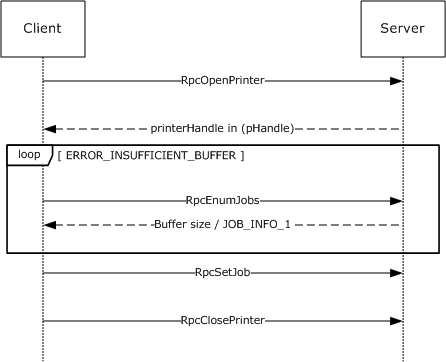


Figure : Enumerating jobs and modifying job settings

## Receiving Notifications on Printing Events

To receive notifications concerning state changes of print servers, printers, and print jobs, a client ("TESTCLT") can perform the following steps, as shown in the figure in [Notification of Print System Changes (section 1.3.3)](#Section_e947715e821c47fe97b8c0ed70eed5b6).

1. Open a handle to a printer using [RpcOpenPrinter (section 3.1.4.2.2)](#Section_989357e2446e4872bb381dce21e1313f).
   * The client calls RpcOpenPrinter.
2. RpcOpenPrinter( L"\\\\CORPSERV\\My Printer", &hPrinter, L"RAW", &devmodeContainer, PRINTER\_ACCESS\_USE );
   * The server allocates a printer handle, writes it to hPrinter, and returns 0 (success).
3. Register for change notifications using [RpcRemoteFindFirstPrinterChangeNotificationEx (section 3.1.4.10.4)](#Section_eb66b2211c1f4249b8bcc5befec2314d):
   * The client allocates and initializes a [RPC\_V2\_NOTIFY\_OPTIONS](#Section_79db335febe74a6f8533efdda32cc5d8) notifyOptions structure as follows:
4. WORD notifyFieldsJob[] = { 0x000A /\*JOB\_NOTIFY\_FIELD\_STATUS\*/, 0x000D /\*JOB\_NOTIFY\_FIELD\_DOCUMENT\*/ };
5. RPC\_V2\_NOTIFY\_OPTIONS\_TYPE notifyTypes[1] = {{1 /\*JOB\_NOTIFY\_TYPE\*/, 0, 0, 0, 2, notifyFieldsJob }};
6. RPC\_V2\_NOTIFY\_OPTIONS notifyOptions = {0x00000002,0x00000000,1,notifyTypes};
   * The client calls RpcRemoteFindFirstPrinterChangeNotificationEx.
7. RpcRemoteFindFirstPrinterChangeNotificationEx( hPrinter, 0x00000100 /\* PRINTER\_CHANGE\_ADD\_JOB \*/, 0, L"\\\\TESTCLT", 4711, &notifyOptions );
8. /\* The number 4711 is a unique number used as a cookie to match the server’s response. \*/
   * The server calls the client's [RpcReplyOpenPrinter (section 3.2.4.1.1)](#Section_7fcd3036d45a4ec7b081f2b860e66676) method to open a reverse channel, which will be used to send change notifications to the client. The client has to return a remote procedure call (RPC) binding handle that identifies the reverse channel.
     + The server calls RpcReplyOpenPrinter.
9. RpcReplyOpenPrinter( L"\\\\TESTCLT", &hPrinterNotify, 4711, 1, 0, NULL );
   * + The client opens a notification context and associates it with the open printer handle hPrinter. In order to do so, client matches the dwPrinterRemote value (4711 in this example).
     + The client writes the notification context handle to hPrinterNotify and returns 0 (success) to the server.
     + The server returns 0 (success) from processing RpcRemoteFindFirstPrinterChangeNotificationEx.
   * As long as the client stays registered for notifications, the server calls the client's [RpcRouterReplyPrinter (section 3.2.4.1.2)](#Section_38bf5d2ce4a744b2b16f91ff8226f836) or [RpcRouterReplyPrinterEx (section 3.2.4.1.4)](#Section_1b44cb8295b5485c9ec90360c3a3ecb0) method for each change of the requested type that occurs on the server.

Whenever the monitored print queue changes on the server, the server filters the change according to the filter options specified by the client in the registration call and:

* + - The server allocates and initializes an [RPC\_V2\_UREPLY\_PRINTER](#Section_1ccdac5b0b2a4bd38afdd2c2589130fc) reply structure:

1. RPC\_V2\_NOTIFY\_INFO notifyInfo; /\* Note: Pseudo-code only, assumes sufficient memory has been allocated for aData[] array at end of structure \*/
2. notifyInfo.Version = 2;
3. notifyInfo.Flags = 0;
4. notifyInfo.Count = 1;
5. notifyInfo.aData[0].Type = 1; /\* JOB\_NOTIFY\_TYPE \*/
6. notifyInfo.aData[0].Field = 0xD /\* JOB\_NOTIFY\_FIELD\_DOCUMENT \*/ notifyInfo.aData[0].String.pszString = L"My Test Print Job Name";
7. notifyInfo.aData[0].Id = 12; /\* This print job has ID 12 \*/
8. RPC\_V2\_UREPLY\_PRINTER reply;
9. Reply.pInfo = &notifyInfo;
   * + The server calls the client's RpcRouterReplyPrinterEx.
10. RpcRouterReplyPrinterEx( hPrinterNotify, 1, 0x00000100 /\* PRINTER\_CHANGE\_ADD\_JOB \*/, &result, 0, &reply );
    * + The client reflects the change in its internal state.
      + The client writes processing flags to the variable result and returns 0 (success).
11. Route the change notifications to applications or process it to reflect state changes.

The client makes local calls to applications or processes to notify them of the state change.

1. When state changes are no longer accepted, the client unregisters from notifications by calling [RpcFindClosePrinterChangeNotification (section 3.1.4.10.2)](#Section_f50c8a861b2d4124b2205b840aacd2c7) with the handle returned by the call to RpcOpenPrinter.
2. RpcFindClosePrinterChangeNotification( hPrinter );
   * The server calls the client's [RpcReplyClosePrinter (section 3.2.4.1.3)](#Section_d6c73ecbc02b4150a01bad73a642990a) with the handle previously obtained by RpcReplyOpenPrinter, notifying the client to close the binding handle for the reverse channel.
3. RpcReplyClosePrinter( &hPrinterNotify );
   * The client cleans up the notification context, writes NULL to hPrinterNotify, and returns 0 (success).
   * The server returns 0 (success) from RpcFindClosePrinterChangeNotification.
4. The client closes the handle to the printer or server object using [RpcClosePrinter (section 3.1.4.2.9)](#Section_8482c18677734fb58034e9bae7a91e52).
5. RpcClosePrinter( &hPrinter );

The server frees the memory associated with the print queue handle, sets hPrinter to NULL, and returns 0 (success).

# Security

## Security Considerations for Implementers

None.

## Index of Security Parameters

Security considerations for both authenticated and unauthenticated RPC are specified in [[C706]](http://go.microsoft.com/fwlink/?LinkId=89824) chapters Introduction to the RPC API and Security.[<476>](#Appendix_A_476)

# Appendix A: Full IDL

For ease of implementation, the full stand-alone Interface Definition Language (IDL) is provided. Some of the data types and structures used by this protocol are defined in other documents. In order for this IDL to stand alone, those types and structures, from [[MS-DTYP]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-DTYP%5d.pdf), are included below.

1. // [MS-RPRN] interface
2. [
3. uuid(12345678-1234-ABCD-EF00-0123456789AB),
4. version(1.0),
5. ms\_union,
6. endpoint("ncacn\_np:[\\pipe\\spoolss]"),
7. pointer\_default(unique)
8. ]
9. interface winspool {
10. import "ms-dtyp.idl";
11. #if \_\_midl < 700
12. #define disable\_consistency\_check
13. #endif
14. // [MS-RPRN] common constants
15. #define TABLE\_DWORD 0x1
16. #define TABLE\_STRING 0x2
17. #define TABLE\_DEVMODE 0x3
18. #define TABLE\_TIME 0x4
19. #define TABLE\_SECURITYDESCRIPTOR 0x5
20. #define SPLFILE\_CONTENT\_TYPE\_PROP\_NAME L"Spool File Contents"
21. // [MS-RPRN] common enumerations
22. typedef enum {
23. BIDI\_NULL = 0,
24. BIDI\_INT = 1,
25. BIDI\_FLOAT = 2,
26. BIDI\_BOOL = 3,
27. BIDI\_STRING = 4,
28. BIDI\_TEXT = 5,
29. BIDI\_ENUM = 6,
30. BIDI\_BLOB = 7
31. } BIDI\_TYPE;
32. typedef enum {
33. kRpcPropertyTypeString = 1,
34. kRpcPropertyTypeInt32,
35. kRpcPropertyTypeInt64,
36. kRpcPropertyTypeByte,
37. kRpcPropertyTypeBuffer
38. } RPC\_EPrintPropertyType;
39. // [MS-RPRN] common data types
40. typedef unsigned short LANGID;
41. typedef [context\_handle] void\* GDI\_HANDLE;
42. typedef [context\_handle] void\* PRINTER\_HANDLE;
43. typedef [handle] wchar\_t\* STRING\_HANDLE;
44. // [MS-RPRN] common utility structures
45. typedef struct {
46. long cx;
47. long cy;
48. } SIZE;
49. typedef struct {
50. long left;
51. long top;
52. long right;
53. long bottom;
54. } RECTL;
55. // [MS-RPRN] common device state structure
56. typedef struct \_devicemode {
57. wchar\_t dmDeviceName[32];
58. unsigned short dmSpecVersion;
59. unsigned short dmDriverVersion;
60. unsigned short dmSize;
61. unsigned short dmDriverExtra;
62. DWORD dmFields;
63. short dmOrientation;
64. short dmPaperSize;
65. short dmPaperLength;
66. short dmPaperWidth;
67. short dmScale;
68. short dmCopies;
69. short dmDefaultSource;
70. short dmPrintQuality;
71. short dmColor;
72. short dmDuplex;
73. short dmYResolution;
74. short dmTTOption;
75. short dmCollate;
76. wchar\_t dmFormName[32];
77. unsigned short reserved0;
78. DWORD reserved1;
79. DWORD reserved2;
80. DWORD reserved3;
81. DWORD dmNup;
82. DWORD reserved4;
83. DWORD dmICMMethod;
84. DWORD dmICMIntent;
85. DWORD dmMediaType;
86. DWORD dmDitherType;
87. DWORD reserved5;
88. DWORD reserved6;
89. DWORD reserved7;
90. DWORD reserved8;
91. } DEVMODE;
92. // [MS-RPRN] common info structures
93. typedef struct \_DOC\_INFO\_1 {
94. [string] wchar\_t\* pDocName;
95. [string] wchar\_t\* pOutputFile;
96. [string] wchar\_t\* pDatatype;
97. } DOC\_INFO\_1;
98. typedef struct \_DRIVER\_INFO\_1 {
99. [string] wchar\_t\* pName;
100. } DRIVER\_INFO\_1;
101. typedef struct \_DRIVER\_INFO\_2 {
102. DWORD cVersion;
103. [string] wchar\_t\* pName;
104. [string] wchar\_t\* pEnvironment;
105. [string] wchar\_t\* pDriverPath;
106. [string] wchar\_t\* pDataFile;
107. [string] wchar\_t\* pConfigFile;
108. } DRIVER\_INFO\_2;
109. typedef struct \_RPC\_DRIVER\_INFO\_3 {
110. DWORD cVersion;
111. [string] wchar\_t\* pName;
112. [string] wchar\_t\* pEnvironment;
113. [string] wchar\_t\* pDriverPath;
114. [string] wchar\_t\* pDataFile;
115. [string] wchar\_t\* pConfigFile;
116. [string] wchar\_t\* pHelpFile;
117. [string] wchar\_t\* pMonitorName;
118. [string] wchar\_t\* pDefaultDataType;
119. DWORD cchDependentFiles;
120. [size\_is(cchDependentFiles), unique]
121. wchar\_t\* pDependentFiles;
122. } RPC\_DRIVER\_INFO\_3;
123. typedef struct \_RPC\_DRIVER\_INFO\_4 {
124. DWORD cVersion;
125. [string] wchar\_t\* pName;
126. [string] wchar\_t\* pEnvironment;
127. [string] wchar\_t\* pDriverPath;
128. [string] wchar\_t\* pDataFile;
129. [string] wchar\_t\* pConfigFile;
130. [string] wchar\_t\* pHelpFile;
131. [string] wchar\_t\* pMonitorName;
132. [string] wchar\_t\* pDefaultDataType;
133. DWORD cchDependentFiles;
134. [size\_is(cchDependentFiles), unique]
135. wchar\_t\* pDependentFiles;
136. DWORD cchPreviousNames;
137. [size\_is(cchPreviousNames), unique]
138. wchar\_t\* pszzPreviousNames;
139. } RPC\_DRIVER\_INFO\_4;
140. typedef struct \_RPC\_DRIVER\_INFO\_6 {
141. DWORD cVersion;
142. [string] wchar\_t\* pName;
143. [string] wchar\_t\* pEnvironment;
144. [string] wchar\_t\* pDriverPath;
145. [string] wchar\_t\* pDataFile;
146. [string] wchar\_t\* pConfigFile;
147. [string] wchar\_t\* pHelpFile;
148. [string] wchar\_t\* pMonitorName;
149. [string] wchar\_t\* pDefaultDataType;
150. DWORD cchDependentFiles;
151. [size\_is(cchDependentFiles), unique]
152. wchar\_t\* pDependentFiles;
153. DWORD cchPreviousNames;
154. [size\_is(cchPreviousNames), unique]
155. wchar\_t\* pszzPreviousNames;
156. FILETIME ftDriverDate;
157. DWORDLONG dwlDriverVersion;
158. [string] wchar\_t\* pMfgName;
159. [string] wchar\_t\* pOEMUrl;
160. [string] wchar\_t\* pHardwareID;
161. [string] wchar\_t\* pProvider;
162. } RPC\_DRIVER\_INFO\_6;
163. typedef struct \_RPC\_DRIVER\_INFO\_8 {
164. DWORD cVersion;
165. [string] wchar\_t\* pName;
166. [string] wchar\_t\* pEnvironment;
167. [string] wchar\_t\* pDriverPath;
168. [string] wchar\_t\* pDataFile;
169. [string] wchar\_t\* pConfigFile;
170. [string] wchar\_t\* pHelpFile;
171. [string] wchar\_t\* pMonitorName;
172. [string] wchar\_t\* pDefaultDataType;
173. DWORD cchDependentFiles;
174. [size\_is(cchDependentFiles), unique]
175. wchar\_t\* pDependentFiles;
176. DWORD cchPreviousNames;
177. [size\_is(cchPreviousNames), unique]
178. wchar\_t\* pszzPreviousNames;
179. FILETIME ftDriverDate;
180. DWORDLONG dwlDriverVersion;
181. [string] wchar\_t\* pMfgName;
182. [string] wchar\_t\* pOEMUrl;
183. [string] wchar\_t\* pHardwareID;
184. [string] wchar\_t\* pProvider;
185. [string] wchar\_t\* pPrintProcessor;
186. [string] wchar\_t\* pVendorSetup;
187. DWORD cchColorProfiles;
188. [size\_is(cchColorProfiles), unique]
189. wchar\_t\* pszzColorProfiles;
190. [string] wchar\_t\* pInfPath;
191. DWORD dwPrinterDriverAttributes;
192. DWORD cchCoreDependencies;
193. [size\_is(cchCoreDependencies), unique]
194. wchar\_t\* pszzCoreDriverDependencies;
195. FILETIME ftMinInboxDriverVerDate;
196. DWORDLONG dwlMinInboxDriverVerVersion;
197. } RPC\_DRIVER\_INFO\_8;
198. typedef struct \_FORM\_INFO\_1 {
199. DWORD Flags;
200. [string] wchar\_t\* pName;
201. SIZE Size;
202. RECTL ImageableArea;
203. } FORM\_INFO\_1;
204. typedef struct \_RPC\_FORM\_INFO\_2 {
205. DWORD Flags;
206. [string, unique] const wchar\_t\* pName;
207. SIZE Size;
208. RECTL ImageableArea;
209. [string, unique] const char\* pKeyword;
210. DWORD StringType;
211. [string, unique] const wchar\_t\* pMuiDll;
212. DWORD dwResourceId;
213. [string, unique] const wchar\_t\* pDisplayName;
214. LANGID wLangID;
215. } RPC\_FORM\_INFO\_2;
216. typedef struct \_JOB\_INFO\_1 {
217. DWORD JobId;
218. [string] wchar\_t\* pPrinterName;
219. [string] wchar\_t\* pMachineName;
220. [string] wchar\_t\* pUserName;
221. [string] wchar\_t\* pDocument;
222. [string] wchar\_t\* pDatatype;
223. [string] wchar\_t\* pStatus;
224. DWORD Status;
225. DWORD Priority;
226. DWORD Position;
227. DWORD TotalPages;
228. DWORD PagesPrinted;
229. SYSTEMTIME Submitted;
230. } JOB\_INFO\_1;
231. typedef struct \_JOB\_INFO\_2 {
232. DWORD JobId;
233. [string] wchar\_t\* pPrinterName;
234. [string] wchar\_t\* pMachineName;
235. [string] wchar\_t\* pUserName;
236. [string] wchar\_t\* pDocument;
237. [string] wchar\_t\* pNotifyName;
238. [string] wchar\_t\* pDatatype;
239. [string] wchar\_t\* pPrintProcessor;
240. [string] wchar\_t\* pParameters;
241. [string] wchar\_t\* pDriverName;
242. ULONG\_PTR pDevMode;
243. [string] wchar\_t\* pStatus;
244. ULONG\_PTR pSecurityDescriptor;
245. DWORD Status;
246. DWORD Priority;
247. DWORD Position;
248. DWORD StartTime;
249. DWORD UntilTime;
250. DWORD TotalPages;
251. DWORD Size;
252. SYSTEMTIME Submitted;
253. DWORD Time;
254. DWORD PagesPrinted;
255. } JOB\_INFO\_2;
256. typedef struct \_JOB\_INFO\_3 {
257. DWORD JobId;
258. DWORD NextJobId;
259. DWORD Reserved;
260. } JOB\_INFO\_3;
261. typedef struct \_JOB\_INFO\_4 {
262. DWORD JobId;
263. [string] wchar\_t\* pPrinterName;
264. [string] wchar\_t\* pMachineName;
265. [string] wchar\_t\* pUserName;
266. [string] wchar\_t\* pDocument;
267. [string] wchar\_t\* pNotifyName;
268. [string] wchar\_t\* pDatatype;
269. [string] wchar\_t\* pPrintProcessor;
270. [string] wchar\_t\* pParameters;
271. [string] wchar\_t\* pDriverName;
272. ULONG\_PTR pDevMode;
273. [string] wchar\_t\* pStatus;
274. ULONG\_PTR pSecurityDescriptor;
275. DWORD Status;
276. DWORD Priority;
277. DWORD Position;
278. DWORD StartTime;
279. DWORD UntilTime;
280. DWORD TotalPages;
281. DWORD Size;
282. SYSTEMTIME Submitted;
283. DWORD Time;
284. DWORD PagesPrinted;
285. long SizeHigh;
286. } JOB\_INFO\_4;
287. typedef struct \_MONITOR\_INFO\_1 {
288. [string] wchar\_t\* pName;
289. } MONITOR\_INFO\_1;
290. typedef struct \_MONITOR\_INFO\_2 {
291. [string] wchar\_t\* pName;
292. [string] wchar\_t\* pEnvironment;
293. [string] wchar\_t\* pDLLName;
294. } MONITOR\_INFO\_2;
295. typedef struct \_PORT\_INFO\_1 {
296. [string] wchar\_t\* pPortName;
297. } PORT\_INFO\_1;
298. typedef struct \_PORT\_INFO\_2 {
299. [string] wchar\_t\* pPortName;
300. [string] wchar\_t\* pMonitorName;
301. [string] wchar\_t\* pDescription;
302. DWORD fPortType;
303. DWORD Reserved;
304. } PORT\_INFO\_2;
305. typedef struct \_PORT\_INFO\_3 {
306. DWORD dwStatus;
307. [string] wchar\_t\* pszStatus;
308. DWORD dwSeverity;
309. } PORT\_INFO\_3;
310. typedef struct \_PORT\_INFO\_FF {
311. [string] wchar\_t\* pPortName;
312. DWORD cbMonitorData;
313. BYTE\* pMonitorData;
314. } PORT\_INFO\_FF;
315. typedef struct \_PRINTER\_INFO\_STRESS {
316. [string] wchar\_t\* pPrinterName;
317. [string] wchar\_t\* pServerName;
318. DWORD cJobs;
319. DWORD cTotalJobs;
320. DWORD cTotalBytes;
321. SYSTEMTIME stUpTime;
322. DWORD MaxcRef;
323. DWORD cTotalPagesPrinted;
324. DWORD dwGetVersion;
325. DWORD fFreeBuild;
326. DWORD cSpooling;
327. DWORD cMaxSpooling;
328. DWORD cRef;
329. DWORD cErrorOutOfPaper;
330. DWORD cErrorNotReady;
331. DWORD cJobError;
332. DWORD dwNumberOfProcessors;
333. DWORD dwProcessorType;
334. DWORD dwHighPartTotalBytes;
335. DWORD cChangeID;
336. DWORD dwLastError;
337. DWORD Status;
338. DWORD cEnumerateNetworkPrinters;
339. DWORD cAddNetPrinters;
340. unsigned short wProcessorArchitecture;
341. unsigned short wProcessorLevel;
342. DWORD cRefIC;
343. DWORD dwReserved2;
344. DWORD dwReserved3;
345. } PRINTER\_INFO\_STRESS;
346. typedef struct \_PRINTER\_INFO\_1 {
347. DWORD Flags;
348. [string] wchar\_t\* pDescription;
349. [string] wchar\_t\* pName;
350. [string] wchar\_t\* pComment;
351. } PRINTER\_INFO\_1;
352. typedef struct \_PRINTER\_INFO\_2 {
353. [string] wchar\_t\* pServerName;
354. [string] wchar\_t\* pPrinterName;
355. [string] wchar\_t\* pShareName;
356. [string] wchar\_t\* pPortName;
357. [string] wchar\_t\* pDriverName;
358. [string] wchar\_t\* pComment;
359. [string] wchar\_t\* pLocation;
360. ULONG\_PTR pDevMode;
361. [string] wchar\_t\* pSepFile;
362. [string] wchar\_t\* pPrintProcessor;
363. [string] wchar\_t\* pDatatype;
364. [string] wchar\_t\* pParameters;
365. ULONG\_PTR pSecurityDescriptor;
366. DWORD Attributes;
367. DWORD Priority;
368. DWORD DefaultPriority;
369. DWORD StartTime;
370. DWORD UntilTime;
371. DWORD Status;
372. DWORD cJobs;
373. DWORD AveragePPM;
374. } PRINTER\_INFO\_2;
375. typedef struct \_PRINTER\_INFO\_3 {
376. ULONG\_PTR pSecurityDescriptor;
377. } PRINTER\_INFO\_3;
378. typedef struct \_PRINTER\_INFO\_4 {
379. [string] wchar\_t\* pPrinterName;
380. [string] wchar\_t\* pServerName;
381. DWORD Attributes;
382. } PRINTER\_INFO\_4;
383. typedef struct \_PRINTER\_INFO\_5 {
384. [string] wchar\_t\* pPrinterName;
385. [string] wchar\_t\* pPortName;
386. DWORD Attributes;
387. DWORD DeviceNotSelectedTimeout;
388. DWORD TransmissionRetryTimeout;
389. } PRINTER\_INFO\_5;
390. typedef struct \_PRINTER\_INFO\_6 {
391. DWORD dwStatus;
392. } PRINTER\_INFO\_6;
393. typedef struct \_PRINTER\_INFO\_7 {
394. [string] wchar\_t\* pszObjectGUID;
395. DWORD dwAction;
396. } PRINTER\_INFO\_7;
397. typedef struct \_PRINTER\_INFO\_8 {
398. ULONG\_PTR pDevMode;
399. } PRINTER\_INFO\_8;
400. typedef struct \_PRINTER\_INFO\_9 {
401. ULONG\_PTR pDevMode;
402. } PRINTER\_INFO\_9;
403. typedef struct \_SPLCLIENT\_INFO\_1 {
404. DWORD dwSize;
405. [string] wchar\_t\* pMachineName;
406. [string] wchar\_t\* pUserName;
407. DWORD dwBuildNum;
408. DWORD dwMajorVersion;
409. DWORD dwMinorVersion;
410. unsigned short wProcessorArchitecture;
411. } SPLCLIENT\_INFO\_1;
412. typedef struct \_SPLCLIENT\_INFO\_2 {
413. LONG\_PTR notUsed;
414. } SPLCLIENT\_INFO\_2;
415. typedef struct \_SPLCLIENT\_INFO\_3 {
416. unsigned int cbSize;
417. DWORD dwFlags;
418. DWORD dwSize;
419. [string] wchar\_t\* pMachineName;
420. [string] wchar\_t\* pUserName;
421. DWORD dwBuildNum;
422. DWORD dwMajorVersion;
423. DWORD dwMinorVersion;
424. unsigned short wProcessorArchitecture;
425. unsigned \_\_int64 hSplPrinter;
426. } SPLCLIENT\_INFO\_3;
427. // [MS-RPRN] common info container structures
428. typedef struct \_DEVMODE\_CONTAINER {
429. DWORD cbBuf;
430. [size\_is(cbBuf), unique] BYTE\* pDevMode;
431. } DEVMODE\_CONTAINER;
432. typedef struct \_DOC\_INFO\_CONTAINER {
433. DWORD Level;
434. [switch\_is(Level)] union {
435. [case(1)]
436. DOC\_INFO\_1\* pDocInfo1;
437. } DocInfo;
438. } DOC\_INFO\_CONTAINER;
439. typedef struct \_DRIVER\_CONTAINER {
440. DWORD Level;
441. [switch\_is(Level)] union {
442. [case(1)]
443. DRIVER\_INFO\_1\* pNotUsed;
444. [case(2)]
445. DRIVER\_INFO\_2\* Level2;
446. [case(3)]
447. RPC\_DRIVER\_INFO\_3\* Level3;
448. [case(4)]
449. RPC\_DRIVER\_INFO\_4\* Level4;
450. [case(6)]
451. RPC\_DRIVER\_INFO\_6\* Level6;
452. [case(8)]
453. RPC\_DRIVER\_INFO\_8\* Level8;
454. } DriverInfo;
455. } DRIVER\_CONTAINER;
456. typedef struct \_FORM\_CONTAINER {
457. DWORD Level;
458. [switch\_is(Level)] union {
459. [case(1)]
460. FORM\_INFO\_1\* pFormInfo1;
461. [case(2)]
462. RPC\_FORM\_INFO\_2\* pFormInfo2;
463. } FormInfo;
464. } FORM\_CONTAINER;
465. typedef struct \_JOB\_CONTAINER {
466. DWORD Level;
467. [switch\_is(Level)] union {
468. [case(1)]
469. JOB\_INFO\_1\* Level1;
470. [case(2)]
471. JOB\_INFO\_2\* Level2;
472. [case(3)]
473. JOB\_INFO\_3\* Level3;
474. [case(4)]
475. JOB\_INFO\_4\* Level4;
476. } JobInfo;
477. } JOB\_CONTAINER;
478. typedef struct \_MONITOR\_CONTAINER {
479. DWORD Level;
480. [switch\_is(Level)] union {
481. [case(1)]
482. MONITOR\_INFO\_1\* pMonitorInfo1;
483. [case(2)]
484. MONITOR\_INFO\_2\* pMonitorInfo2;
485. } MonitorInfo;
486. } MONITOR\_CONTAINER;
487. typedef struct \_PORT\_CONTAINER {
488. DWORD Level;
489. [switch\_is(0x00FFFFFF & Level)]
490. union {
491. [case(1)]
492. PORT\_INFO\_1\* pPortInfo1;
493. [case(2)]
494. PORT\_INFO\_2\* pPortInfo2;
495. [case(3)]
496. PORT\_INFO\_3\* pPortInfo3;
497. [case(0x00FFFFFF)]
498. PORT\_INFO\_FF\* pPortInfoFF;
499. } PortInfo;
500. } PORT\_CONTAINER;
501. typedef struct \_PORT\_VAR\_CONTAINER {
502. DWORD cbMonitorData;
503. [size\_is(cbMonitorData), unique, disable\_consistency\_check] BYTE\*
504. pMonitorData;
505. } PORT\_VAR\_CONTAINER;
506. typedef struct \_PRINTER\_CONTAINER {
507. DWORD Level;
508. [switch\_is(Level)] union {
509. [case(0)]
510. PRINTER\_INFO\_STRESS\* pPrinterInfoStress;
511. [case(1)]
512. PRINTER\_INFO\_1\* pPrinterInfo1;
513. [case(2)]
514. PRINTER\_INFO\_2\* pPrinterInfo2;
515. [case(3)]
516. PRINTER\_INFO\_3\* pPrinterInfo3;
517. [case(4)]
518. PRINTER\_INFO\_4\* pPrinterInfo4;
519. [case(5)]
520. PRINTER\_INFO\_5\* pPrinterInfo5;
521. [case(6)]
522. PRINTER\_INFO\_6\* pPrinterInfo6;
523. [case(7)]
524. PRINTER\_INFO\_7\* pPrinterInfo7;
525. [case(8)]
526. PRINTER\_INFO\_8\* pPrinterInfo8;
527. [case(9)]
528. PRINTER\_INFO\_9\* pPrinterInfo9;
529. } PrinterInfo;
530. } PRINTER\_CONTAINER;
531. typedef struct \_RPC\_BINARY\_CONTAINER {
532. DWORD cbBuf;
533. [size\_is(cbBuf), unique] BYTE\* pszString;
534. } RPC\_BINARY\_CONTAINER;
535. typedef struct \_RPC\_BIDI\_DATA {
536. DWORD dwBidiType;
537. [switch\_is(dwBidiType)] union {
538. [case(BIDI\_NULL, BIDI\_BOOL)]
539. int bData;
540. [case(BIDI\_INT)]
541. long iData;
542. [case(BIDI\_STRING, BIDI\_TEXT, BIDI\_ENUM)]
543. [string,unique] wchar\_t\* sData;
544. [case(BIDI\_FLOAT)]
545. float fData;
546. [case(BIDI\_BLOB)]
547. RPC\_BINARY\_CONTAINER biData;
548. } u;
549. } RPC\_BIDI\_DATA;
550. typedef struct \_RPC\_BIDI\_REQUEST\_DATA {
551. DWORD dwReqNumber;
552. [string, unique] wchar\_t\* pSchema;
553. RPC\_BIDI\_DATA data;
554. } RPC\_BIDI\_REQUEST\_DATA;
555. typedef struct \_RPC\_BIDI\_RESPONSE\_DATA {
556. DWORD dwResult;
557. DWORD dwReqNumber;
558. [string, unique] wchar\_t\* pSchema;
559. RPC\_BIDI\_DATA data;
560. } RPC\_BIDI\_RESPONSE\_DATA;
561. typedef struct \_RPC\_BIDI\_REQUEST\_CONTAINER {
562. DWORD Version;
563. DWORD Flags;
564. DWORD Count;
565. [size\_is(Count), unique] RPC\_BIDI\_REQUEST\_DATA aData[];
566. } RPC\_BIDI\_REQUEST\_CONTAINER;
567. typedef struct \_RPC\_BIDI\_RESPONSE\_CONTAINER {
568. DWORD Version;
569. DWORD Flags;
570. DWORD Count;
571. [size\_is(Count), unique] RPC\_BIDI\_RESPONSE\_DATA aData[];
572. } RPC\_BIDI\_RESPONSE\_CONTAINER;
573. typedef struct SECURITY\_CONTAINER {
574. DWORD cbBuf;
575. [size\_is(cbBuf), unique] BYTE\* pSecurity;
576. } SECURITY\_CONTAINER;
577. typedef struct \_SPLCLIENT\_CONTAINER {
578. DWORD Level;
579. [switch\_is(Level)] union {
580. [case(1)]
581. SPLCLIENT\_INFO\_1\* pClientInfo1;
582. [case(2)]
583. SPLCLIENT\_INFO\_2\* pNotUsed1;
584. [case(3)]
585. SPLCLIENT\_INFO\_3\* pNotUsed2;
586. } ClientInfo;
587. } SPLCLIENT\_CONTAINER;
588. typedef struct \_STRING\_CONTAINER {
589. DWORD cbBuf;
590. [size\_is(cbBuf/2), unique] WCHAR\* pszString;
591. } STRING\_CONTAINER;
592. typedef struct \_SYSTEMTIME\_CONTAINER {
593. DWORD cbBuf;
594. SYSTEMTIME\* pSystemTime;
595. } SYSTEMTIME\_CONTAINER;
596. typedef struct \_RPC\_V2\_NOTIFY\_OPTIONS\_TYPE {
597. unsigned short Type;
598. unsigned short Reserved0;
599. DWORD Reserved1;
600. DWORD Reserved2;
601. DWORD Count;
602. [size\_is(Count), unique] unsigned short\* pFields;
603. } RPC\_V2\_NOTIFY\_OPTIONS\_TYPE;
604. typedef struct \_RPC\_V2\_NOTIFY\_OPTIONS {
605. DWORD Version;
606. DWORD Reserved;
607. DWORD Count;
608. [size\_is(Count), unique] RPC\_V2\_NOTIFY\_OPTIONS\_TYPE\* pTypes;
609. } RPC\_V2\_NOTIFY\_OPTIONS;
610. typedef
611. [switch\_type (DWORD)]
612. union \_RPC\_V2\_NOTIFY\_INFO\_DATA\_DATA {
613. [case(TABLE\_STRING)]
614. STRING\_CONTAINER String;
615. [case(TABLE\_DWORD)]
616. DWORD dwData[2];
617. [case(TABLE\_TIME)]
618. SYSTEMTIME\_CONTAINER SystemTime;
619. [case(TABLE\_DEVMODE)]
620. DEVMODE\_CONTAINER DevMode;
621. [case(TABLE\_SECURITYDESCRIPTOR)]
622. SECURITY\_CONTAINER SecurityDescriptor;
623. } RPC\_V2\_NOTIFY\_INFO\_DATA\_DATA;
624. typedef struct \_RPC\_V2\_NOTIFY\_INFO\_DATA {
625. unsigned short Type;
626. unsigned short Field;
627. DWORD Reserved;
628. DWORD Id;
629. [switch\_is(Reserved & 0xffff)]
630. RPC\_V2\_NOTIFY\_INFO\_DATA\_DATA Data;
631. } RPC\_V2\_NOTIFY\_INFO\_DATA;
632. typedef struct \_RPC\_V2\_NOTIFY\_INFO {
633. DWORD Version;
634. DWORD Flags;
635. DWORD Count;
636. [size\_is(Count), unique] RPC\_V2\_NOTIFY\_INFO\_DATA aData[];
637. } RPC\_V2\_NOTIFY\_INFO;
638. typedef [switch\_type(DWORD)] union \_RPC\_V2\_UREPLY\_PRINTER {
639. [case (0)]
640. RPC\_V2\_NOTIFY\_INFO\* pInfo;
641. } RPC\_V2\_UREPLY\_PRINTER;
642. typedef struct \_CORE\_PRINTER\_DRIVER {
643. GUID CoreDriverGUID;
644. FILETIME ftDriverDate;
645. DWORDLONG dwlDriverVersion;
646. wchar\_t szPackageID[260];
647. } CORE\_PRINTER\_DRIVER;
648. typedef struct {
649. RPC\_EPrintPropertyType ePropertyType;
650. [switch\_is(ePropertyType)]
651. union {
652. [case(kRpcPropertyTypeString)]
653. [string] wchar\_t \*propertyString;
654. [case(kRpcPropertyTypeInt32)]
655. LONG propertyInt32;
656. [case(kRpcPropertyTypeInt64)]
657. LONGLONG propertyInt64;
658. [case(kRpcPropertyTypeByte)]
659. BYTE propertyByte;
660. [case(kRpcPropertyTypeBuffer)]
661. struct {
662. DWORD cbBuf;
663. [size\_is(cbBuf)] BYTE \*pBuf;
664. } propertyBlob;
665. } value;
666. } RPC\_PrintPropertyValue;
667. typedef struct {
668. [string] wchar\_t \*propertyName;
669. RPC\_PrintPropertyValue propertyValue;
670. } RPC\_PrintNamedProperty;
671. typedef enum {
672. kInvalidJobState = 0,
673. kLogJobPrinted,
674. kLogJobRendered,
675. kLogJobError,
676. kLogJobPipelineError,
677. kLogOfflineFileFull
678. } EBranchOfficeJobEventType;
679. typedef struct {
680. DWORD Status;
681. [string] wchar\_t\* pDocumentName;
682. [string] wchar\_t\* pUserName;
683. [string] wchar\_t\* pMachineName;
684. [string] wchar\_t\* pPrinterName;
685. [string] wchar\_t\* pPortName;
686. LONGLONG Size;
687. DWORD TotalPages;
688. } RPC\_BranchOfficeJobDataPrinted;
689. typedef struct {
690. LONGLONG Size;
691. DWORD ICMMethod;
692. short Color;
693. short PrintQuality;
694. short YResolution;
695. short Copies;
696. short TTOption;
697. } RPC\_BranchOfficeJobDataRendered;
698. typedef struct {
699. DWORD LastError;
700. [string] wchar\_t\* pDocumentName;
701. [string] wchar\_t\* pUserName;
702. [string] wchar\_t\* pPrinterName;
703. [string] wchar\_t\* pDataType;
704. LONGLONG TotalSize;
705. LONGLONG PrintedSize;
706. DWORD TotalPages;
707. DWORD PrintedPages;
708. [string] wchar\_t\* pMachineName;
709. [string] wchar\_t\* pJobError;
710. [string] wchar\_t\* pErrorDescription;
711. } RPC\_BranchOfficeJobDataError;
712. typedef struct {
713. [string] wchar\_t\* pDocumentName;
714. [string] wchar\_t\* pPrinterName;
715. [string] wchar\_t\* pExtraErrorInfo;
716. } RPC\_BranchOfficeJobDataPipelineFailed;
717. typedef struct {
718. [string] wchar\_t\* pMachineName;
719. } RPC\_BranchOfficeLogOfflineFileFull;
720. typedef struct {
721. EBranchOfficeJobEventType eEventType;
722. DWORD JobId;
723. [switch\_type(EBranchOfficeJobEventType), switch\_is(eEventType)]
724. union {
725. [case(kLogJobPrinted)]
726. RPC\_BranchOfficeJobDataPrinted LogJobPrinted;
727. [case(kLogJobRendered)]
728. RPC\_BranchOfficeJobDataRendered LogJobRendered;
729. [case(kLogJobError)]
730. RPC\_BranchOfficeJobDataError LogJobError;
731. [case(kLogJobPipelineError)]
732. RPC\_BranchOfficeJobDataPipelineFailed LogPipelineFailed;
733. [case(kLogOfflineFileFull)]
734. RPC\_BranchOfficeLogOfflineFileFull LogOfflineFileFull;
735. } JobInfo;
736. } RPC\_BranchOfficeJobData;
737. typedef struct {
738. DWORD cJobDataEntries;
739. [size\_is(cJobDataEntries), unique] RPC\_BranchOfficeJobData JobData[];
740. } RPC\_BranchOfficeJobDataContainer;
741. // [MS-RPRN] methods
742. DWORD
743. RpcEnumPrinters(
744. [in] DWORD Flags,
745. [in, string, unique] STRING\_HANDLE Name,
746. [in] DWORD Level,
747. [in, out, unique, size\_is(cbBuf), disable\_consistency\_check] BYTE\*
748. pPrinterEnum,
749. [in] DWORD cbBuf,
750. [out] DWORD\* pcbNeeded,
751. [out] DWORD\* pcReturned
752. );
753. DWORD
754. RpcOpenPrinter(
755. [in, string, unique] STRING\_HANDLE pPrinterName,
756. [out] PRINTER\_HANDLE\* pHandle,
757. [in, string, unique] wchar\_t\* pDatatype,
758. [in] DEVMODE\_CONTAINER\* pDevModeContainer,
759. [in] DWORD AccessRequired
760. );
761. DWORD
762. RpcSetJob(
763. [in] PRINTER\_HANDLE hPrinter,
764. [in] DWORD JobId,
765. [in, unique] JOB\_CONTAINER\* pJobContainer,
766. [in] DWORD Command
767. );
768. DWORD
769. RpcGetJob(
770. [in] PRINTER\_HANDLE hPrinter,
771. [in] DWORD JobId,
772. [in] DWORD Level,
773. [in, out, unique, size\_is(cbBuf), disable\_consistency\_check] BYTE\*
774. pJob,
775. [in] DWORD cbBuf,
776. [out] DWORD\* pcbNeeded
777. );
778. DWORD
779. RpcEnumJobs(
780. [in] PRINTER\_HANDLE hPrinter,
781. [in] DWORD FirstJob,
782. [in] DWORD NoJobs,
783. [in] DWORD Level,
784. [in, out, unique, size\_is(cbBuf), disable\_consistency\_check] BYTE\*
785. pJob,
786. [in] DWORD cbBuf,
787. [out] DWORD\* pcbNeeded,
788. [out] DWORD\* pcReturned
789. );
790. DWORD
791. RpcAddPrinter(
792. [in, string, unique] STRING\_HANDLE pName,
793. [in] PRINTER\_CONTAINER\* pPrinterContainer,
794. [in] DEVMODE\_CONTAINER\* pDevModeContainer,
795. [in] SECURITY\_CONTAINER\* pSecurityContainer,
796. [out] PRINTER\_HANDLE\* pHandle
797. );
798. DWORD
799. RpcDeletePrinter(
800. [in] PRINTER\_HANDLE hPrinter
801. );
802. DWORD
803. RpcSetPrinter(
804. [in] PRINTER\_HANDLE hPrinter,
805. [in] PRINTER\_CONTAINER\* pPrinterContainer,
806. [in] DEVMODE\_CONTAINER\* pDevModeContainer,
807. [in] SECURITY\_CONTAINER\* pSecurityContainer,
808. [in] DWORD Command
809. );
810. DWORD
811. RpcGetPrinter(
812. [in] PRINTER\_HANDLE hPrinter,
813. [in] DWORD Level,
814. [in, out, unique, size\_is(cbBuf), disable\_consistency\_check] BYTE\*
815. pPrinter,
816. [in] DWORD cbBuf,
817. [out] DWORD\* pcbNeeded
818. );
819. DWORD
820. RpcAddPrinterDriver(
821. [in, string, unique] STRING\_HANDLE pName,
822. [in] DRIVER\_CONTAINER\* pDriverContainer
823. );
824. DWORD
825. RpcEnumPrinterDrivers(
826. [in, string, unique] STRING\_HANDLE pName,
827. [in, string, unique] wchar\_t\* pEnvironment,
828. [in] DWORD Level,
829. [in, out, unique, size\_is(cbBuf), disable\_consistency\_check] BYTE\*
830. pDrivers,
831. [in] DWORD cbBuf,
832. [out] DWORD\* pcbNeeded,
833. [out] DWORD\* pcReturned
834. );
835. DWORD
836. RpcGetPrinterDriver(
837. [in] PRINTER\_HANDLE hPrinter,
838. [in, string, unique] wchar\_t\* pEnvironment,
839. [in] DWORD Level,
840. [in, out, unique, size\_is(cbBuf), disable\_consistency\_check] BYTE\*
841. pDriver,
842. [in] DWORD cbBuf,
843. [out] DWORD\* pcbNeeded
844. );
845. DWORD
846. RpcGetPrinterDriverDirectory(
847. [in, string, unique] STRING\_HANDLE pName,
848. [in, string, unique] wchar\_t\* pEnvironment,
849. [in] DWORD Level,
850. [in, out, unique, size\_is(cbBuf), disable\_consistency\_check] BYTE\*
851. pDriverDirectory,
852. [in] DWORD cbBuf,
853. [out] DWORD\* pcbNeeded
854. );
855. DWORD
856. RpcDeletePrinterDriver(
857. [in, string, unique] STRING\_HANDLE pName,
858. [in, string] wchar\_t\* pEnvironment,
859. [in, string] wchar\_t\* pDriverName
860. );
861. DWORD
862. RpcAddPrintProcessor(
863. [in, string, unique] STRING\_HANDLE pName,
864. [in, string] wchar\_t\* pEnvironment,
865. [in, string] wchar\_t\* pPathName,
866. [in, string] wchar\_t\* pPrintProcessorName
867. );
868. DWORD
869. RpcEnumPrintProcessors(
870. [in, string, unique] STRING\_HANDLE pName,
871. [in, string, unique] wchar\_t\* pEnvironment,
872. [in] DWORD Level,
873. [in, out, unique, size\_is(cbBuf), disable\_consistency\_check] BYTE\*
874. pPrintProcessorInfo,
875. [in] DWORD cbBuf,
876. [out] DWORD\* pcbNeeded,
877. [out] DWORD\* pcReturned
878. );
879. DWORD
880. RpcGetPrintProcessorDirectory(
881. [in, string, unique] STRING\_HANDLE pName,
882. [in, string, unique] wchar\_t\* pEnvironment,
883. [in] DWORD Level,
884. [in, out, unique, size\_is(cbBuf), disable\_consistency\_check] BYTE\*
885. pPrintProcessorDirectory,
886. [in] DWORD cbBuf,
887. [out] DWORD\* pcbNeeded
888. );
889. DWORD
890. RpcStartDocPrinter(
891. [in] PRINTER\_HANDLE hPrinter,
892. [in] DOC\_INFO\_CONTAINER\* pDocInfoContainer,
893. [out] DWORD\* pJobId
894. );
895. DWORD
896. RpcStartPagePrinter(
897. [in] PRINTER\_HANDLE hPrinter
898. );
899. DWORD
900. RpcWritePrinter(
901. [in] PRINTER\_HANDLE hPrinter,
902. [in, size\_is(cbBuf)] BYTE\* pBuf,
903. [in] DWORD cbBuf,
904. [out] DWORD\* pcWritten
905. );
906. DWORD
907. RpcEndPagePrinter(
908. [in] PRINTER\_HANDLE hPrinter
909. );
910. DWORD
911. RpcAbortPrinter(
912. [in] PRINTER\_HANDLE hPrinter
913. );
914. DWORD
915. RpcReadPrinter(
916. [in] PRINTER\_HANDLE hPrinter,
917. [out, size\_is(cbBuf)] BYTE\* pBuf,
918. [in] DWORD cbBuf,
919. [out] DWORD\* pcNoBytesRead
920. );
921. DWORD
922. RpcEndDocPrinter(
923. [in] PRINTER\_HANDLE hPrinter
924. );
925. DWORD
926. RpcAddJob(
927. [in] PRINTER\_HANDLE hPrinter,
928. [in] DWORD Level,
929. [in, out, unique, size\_is(cbBuf), disable\_consistency\_check] BYTE\*
930. pAddJob,
931. [in] DWORD cbBuf,
932. [out] DWORD\* pcbNeeded
933. );
934. DWORD
935. RpcScheduleJob(
936. [in] PRINTER\_HANDLE hPrinter,
937. [in] DWORD JobId
938. );
939. DWORD
940. RpcGetPrinterData(
941. [in] PRINTER\_HANDLE hPrinter,
942. [in, string] wchar\_t\* pValueName,
943. [out] DWORD\* pType,
944. [out, size\_is(nSize)] BYTE\* pData,
945. [in] DWORD nSize,
946. [out] DWORD\* pcbNeeded
947. );
948. DWORD
949. RpcSetPrinterData(
950. [in] PRINTER\_HANDLE hPrinter,
951. [in, string] wchar\_t\* pValueName,
952. [in] DWORD Type,
953. [in, size\_is(cbData)] BYTE\* pData,
954. [in] DWORD cbData
955. );
956. DWORD
957. RpcWaitForPrinterChange(
958. [in] PRINTER\_HANDLE hPrinter,
959. [in] DWORD Flags,
960. [out] DWORD\* pFlags
961. );
962. DWORD
963. RpcClosePrinter(
964. [in, out] PRINTER\_HANDLE\*phPrinter
965. );
966. DWORD
967. RpcAddForm(
968. [in] PRINTER\_HANDLE hPrinter,
969. [in] FORM\_CONTAINER\* pFormInfoContainer
970. );
971. DWORD
972. RpcDeleteForm(
973. [in] PRINTER\_HANDLE hPrinter,
974. [in, string] wchar\_t\* pFormName
975. );
976. DWORD
977. RpcGetForm(
978. [in] PRINTER\_HANDLE hPrinter,
979. [in, string] wchar\_t\* pFormName,
980. [in] DWORD Level,
981. [in, out, unique, size\_is(cbBuf), disable\_consistency\_check] BYTE\*
982. pForm,
983. [in] DWORD cbBuf,
984. [out] DWORD\* pcbNeeded
985. );
986. DWORD
987. RpcSetForm(
988. [in] PRINTER\_HANDLE hPrinter,
989. [in, string] wchar\_t\* pFormName,
990. [in] FORM\_CONTAINER\* pFormInfoContainer
991. );
992. DWORD
993. RpcEnumForms(
994. [in] PRINTER\_HANDLE hPrinter,
995. [in] DWORD Level,
996. [in, out, unique, size\_is(cbBuf), disable\_consistency\_check] BYTE\*
997. pForm,
998. [in] DWORD cbBuf,
999. [out] DWORD\* pcbNeeded,
1000. [out] DWORD\* pcReturned
1001. );
1002. DWORD
1003. RpcEnumPorts(
1004. [in, string, unique] STRING\_HANDLE pName,
1005. [in] DWORD Level,
1006. [in, out, unique, size\_is(cbBuf), disable\_consistency\_check] BYTE\*
1007. pPort,
1008. [in] DWORD cbBuf,
1009. [out] DWORD\* pcbNeeded,
1010. [out] DWORD\* pcReturned
1011. );
1012. DWORD
1013. RpcEnumMonitors(
1014. [in, string, unique] STRING\_HANDLE pName,
1015. [in] DWORD Level,
1016. [in, out, unique, size\_is(cbBuf), disable\_consistency\_check] BYTE\*
1017. pMonitor,
1018. [in] DWORD cbBuf,
1019. [out] DWORD\* pcbNeeded,
1020. [out] DWORD\* pcReturned
1021. );
1022. void
1023. Opnum37NotUsedOnWire();
1024. void
1025. Opnum38NotUsedOnWire();
1026. DWORD
1027. RpcDeletePort(
1028. [in, string, unique] STRING\_HANDLE pName,
1029. [in] ULONG\_PTR hWnd,
1030. [in, string] wchar\_t\* pPortName
1031. );
1032. DWORD
1033. RpcCreatePrinterIC(
1034. [in] PRINTER\_HANDLE hPrinter,
1035. [out] GDI\_HANDLE\* pHandle,
1036. [in] DEVMODE\_CONTAINER\* pDevModeContainer
1037. );
1038. DWORD
1039. RpcPlayGdiScriptOnPrinterIC(
1040. [in] GDI\_HANDLE hPrinterIC,
1041. [in, size\_is(cIn)] BYTE\* pIn,
1042. [in] DWORD cIn,
1043. [out, size\_is(cOut)] BYTE\* pOut,
1044. [in] DWORD cOut,
1045. [in] DWORD ul
1046. );
1047. DWORD
1048. RpcDeletePrinterIC(
1049. [in, out] GDI\_HANDLE\* phPrinterIC
1050. );
1051. void
1052. Opnum43NotUsedOnWire();
1053. void
1054. Opnum44NotUsedOnWire();
1055. void
1056. Opnum45NotUsedOnWire();
1057. DWORD
1058. RpcAddMonitor(
1059. [in, string, unique] STRING\_HANDLE Name,
1060. [in] MONITOR\_CONTAINER\* pMonitorContainer
1061. );
1062. DWORD
1063. RpcDeleteMonitor(
1064. [in, string, unique] STRING\_HANDLE Name,
1065. [in, string, unique] wchar\_t\* pEnvironment,
1066. [in, string] wchar\_t\* pMonitorName
1067. );
1068. DWORD
1069. RpcDeletePrintProcessor(
1070. [in, string, unique] STRING\_HANDLE Name,
1071. [in, string, unique] wchar\_t\* pEnvironment,
1072. [in, string] wchar\_t\* pPrintProcessorName
1073. );
1074. void
1075. Opnum49NotUsedOnWire();
1076. void
1077. Opnum50NotUsedOnWire();
1078. DWORD
1079. RpcEnumPrintProcessorDatatypes(
1080. [in, string, unique] STRING\_HANDLE pName,
1081. [in, string, unique] wchar\_t\* pPrintProcessorName,
1082. [in] DWORD Level,
1083. [in, out, unique, size\_is(cbBuf), disable\_consistency\_check] BYTE\*
1084. pDatatypes,
1085. [in] DWORD cbBuf,
1086. [out] DWORD\* pcbNeeded,
1087. [out] DWORD\* pcReturned
1088. );
1089. DWORD
1090. RpcResetPrinter(
1091. [in] PRINTER\_HANDLE hPrinter,
1092. [in, string, unique] wchar\_t\* pDatatype,
1093. [in] DEVMODE\_CONTAINER\* pDevModeContainer
1094. );
1095. DWORD
1096. RpcGetPrinterDriver2(
1097. [in] PRINTER\_HANDLE hPrinter,
1098. [in, string, unique] wchar\_t\* pEnvironment,
1099. [in] DWORD Level,
1100. [in, out, unique, size\_is(cbBuf), disable\_consistency\_check] BYTE\*
1101. pDriver,
1102. [in] DWORD cbBuf,
1103. [out] DWORD\* pcbNeeded,
1104. [in] DWORD dwClientMajorVersion,
1105. [in] DWORD dwClientMinorVersion,
1106. [out] DWORD\* pdwServerMaxVersion,
1107. [out] DWORD\* pdwServerMinVersion
1108. );
1109. void
1110. Opnum54NotUsedOnWire();
1111. void
1112. Opnum55NotUsedOnWire();
1113. DWORD
1114. RpcFindClosePrinterChangeNotification(
1115. [in] PRINTER\_HANDLE hPrinter
1116. );
1117. void
1118. Opnum57NotUsedOnWire();
1119. DWORD
1120. RpcReplyOpenPrinter(
1121. [in, string] STRING\_HANDLE pMachine,
1122. [out] PRINTER\_HANDLE\*phPrinterNotify,
1123. [in] DWORD dwPrinterRemote,
1124. [in] DWORD dwType,
1125. [in, range(0, 512)] DWORD cbBuffer,
1126. [in, unique, size\_is(cbBuffer), disable\_consistency\_check] BYTE\*
1127. pBuffer
1128. );
1129. DWORD
1130. RpcRouterReplyPrinter(
1131. [in] PRINTER\_HANDLE hNotify,
1132. [in] DWORD fdwFlags,
1133. [in, range(0, 512)] DWORD cbBuffer,
1134. [in, unique, size\_is(cbBuffer), disable\_consistency\_check] BYTE\*
1135. pBuffer
1136. );
1137. DWORD
1138. RpcReplyClosePrinter(
1139. [in, out] PRINTER\_HANDLE\*phNotify
1140. );
1141. DWORD
1142. RpcAddPortEx(
1143. [in, string, unique] STRING\_HANDLE pName,
1144. [in] PORT\_CONTAINER\* pPortContainer,
1145. [in] PORT\_VAR\_CONTAINER\* pPortVarContainer,
1146. [in, string] wchar\_t\* pMonitorName
1147. );
1148. DWORD
1149. RpcRemoteFindFirstPrinterChangeNotification(
1150. [in] PRINTER\_HANDLE hPrinter,
1151. [in] DWORD fdwFlags,
1152. [in] DWORD fdwOptions,
1153. [in, string, unique] wchar\_t\* pszLocalMachine,
1154. [in] DWORD dwPrinterLocal,
1155. [in, range(0, 512)] DWORD cbBuffer,
1156. [in, out, unique, size\_is(cbBuffer), disable\_consistency\_check]
1157. BYTE\* pBuffer
1158. );
1159. void
1160. Opnum63NotUsedOnWire();
1161. void
1162. Opnum64NotUsedOnWire();
1163. DWORD
1164. RpcRemoteFindFirstPrinterChangeNotificationEx(
1165. [in] PRINTER\_HANDLE hPrinter,
1166. [in] DWORD fdwFlags,
1167. [in] DWORD fdwOptions,
1168. [in, string, unique] wchar\_t\* pszLocalMachine,
1169. [in] DWORD dwPrinterLocal,
1170. [in, unique] RPC\_V2\_NOTIFY\_OPTIONS\* pOptions
1171. );
1172. DWORD
1173. RpcRouterReplyPrinterEx(
1174. [in] PRINTER\_HANDLE hNotify,
1175. [in] DWORD dwColor,
1176. [in] DWORD fdwFlags,
1177. [out] DWORD\* pdwResult,
1178. [in] DWORD dwReplyType,
1179. [in, switch\_is(dwReplyType)] RPC\_V2\_UREPLY\_PRINTER Reply
1180. );
1181. DWORD
1182. RpcRouterRefreshPrinterChangeNotification(
1183. [in] PRINTER\_HANDLE hPrinter,
1184. [in] DWORD dwColor,
1185. [in, unique] RPC\_V2\_NOTIFY\_OPTIONS\* pOptions,
1186. [out] RPC\_V2\_NOTIFY\_INFO\*\* ppInfo
1187. );
1188. void
1189. Opnum68NotUsedOnWire();
1190. DWORD
1191. RpcOpenPrinterEx(
1192. [in, string, unique] STRING\_HANDLE pPrinterName,
1193. [out] PRINTER\_HANDLE\* pHandle,
1194. [in, string, unique] wchar\_t\* pDatatype,
1195. [in] DEVMODE\_CONTAINER\* pDevModeContainer,
1196. [in] DWORD AccessRequired,
1197. [in] SPLCLIENT\_CONTAINER\* pClientInfo
1198. );
1199. DWORD
1200. RpcAddPrinterEx(
1201. [in, string, unique] STRING\_HANDLE pName,
1202. [in] PRINTER\_CONTAINER\* pPrinterContainer,
1203. [in] DEVMODE\_CONTAINER\* pDevModeContainer,
1204. [in] SECURITY\_CONTAINER\* pSecurityContainer,
1205. [in] SPLCLIENT\_CONTAINER\* pClientInfo,
1206. [out] PRINTER\_HANDLE\* pHandle
1207. );
1208. DWORD
1209. RpcSetPort(
1210. [in, string, unique] STRING\_HANDLE pName,
1211. [in, string, unique] wchar\_t\* pPortName,
1212. [in] PORT\_CONTAINER\* pPortContainer
1213. );
1214. DWORD
1215. RpcEnumPrinterData(
1216. [in] PRINTER\_HANDLE hPrinter,
1217. [in] DWORD dwIndex,
1218. [out, size\_is(cbValueName/sizeof(wchar\_t))] wchar\_t\* pValueName,
1219. [in] DWORD cbValueName,
1220. [out] DWORD\* pcbValueName,
1221. [out] DWORD\* pType,
1222. [out, size\_is(cbData)] BYTE\* pData,
1223. [in] DWORD cbData,
1224. [out] DWORD\* pcbData
1225. );
1226. DWORD
1227. RpcDeletePrinterData(
1228. [in] PRINTER\_HANDLE hPrinter,
1229. [in, string] wchar\_t\* pValueName
1230. );
1231. void
1232. Opnum74NotUsedOnWire();
1233. void
1234. Opnum75NotUsedOnWire();
1235. void
1236. Opnum76NotUsedOnWire();
1237. DWORD
1238. RpcSetPrinterDataEx(
1239. [in] PRINTER\_HANDLE hPrinter,
1240. [in, string] const wchar\_t\* pKeyName,
1241. [in, string] const wchar\_t\* pValueName,
1242. [in] DWORD Type,
1243. [in, size\_is(cbData)] BYTE\* pData,
1244. [in] DWORD cbData
1245. );
1246. DWORD
1247. RpcGetPrinterDataEx(
1248. [in] PRINTER\_HANDLE hPrinter,
1249. [in, string] const wchar\_t\* pKeyName,
1250. [in, string] const wchar\_t\* pValueName,
1251. [out] DWORD\* pType,
1252. [out, size\_is(nSize)] BYTE\* pData,
1253. [in] DWORD nSize,
1254. [out] DWORD\* pcbNeeded
1255. );
1256. DWORD
1257. RpcEnumPrinterDataEx(
1258. [in] PRINTER\_HANDLE hPrinter,
1259. [in, string] const wchar\_t\* pKeyName,
1260. [out, size\_is(cbEnumValues)] BYTE\* pEnumValues,
1261. [in] DWORD cbEnumValues,
1262. [out] DWORD\* pcbEnumValues,
1263. [out] DWORD\* pnEnumValues
1264. );
1265. DWORD
1266. RpcEnumPrinterKey(
1267. [in] PRINTER\_HANDLE hPrinter,
1268. [in, string] const wchar\_t\* pKeyName,
1269. [out, size\_is(cbSubkey/sizeof(wchar\_t))] wchar\_t\* pSubkey,
1270. [in] DWORD cbSubkey,
1271. [out] DWORD\* pcbSubkey
1272. );
1273. DWORD
1274. RpcDeletePrinterDataEx(
1275. [in] PRINTER\_HANDLE hPrinter,
1276. [in, string] const wchar\_t\* pKeyName,
1277. [in, string] const wchar\_t\* pValueName
1278. );
1279. DWORD
1280. RpcDeletePrinterKey(
1281. [in] PRINTER\_HANDLE hPrinter,
1282. [in, string] const wchar\_t\* pKeyName
1283. );
1284. void
1285. Opnum83NotUsedOnWire();
1286. DWORD
1287. RpcDeletePrinterDriverEx(
1288. [in, string, unique] STRING\_HANDLE pName,
1289. [in, string] wchar\_t\* pEnvironment,
1290. [in, string] wchar\_t\* pDriverName,
1291. [in] DWORD dwDeleteFlag,
1292. [in] DWORD dwVersionNum
1293. );
1294. DWORD
1295. RpcAddPerMachineConnection(
1296. [in, string, unique] STRING\_HANDLE pServer,
1297. [in, string] const wchar\_t\* pPrinterName,
1298. [in, string] const wchar\_t\* pPrintServer,
1299. [in, string] const wchar\_t\* pProvider
1300. );
1301. DWORD
1302. RpcDeletePerMachineConnection(
1303. [in, string, unique] STRING\_HANDLE pServer,
1304. [in, string] const wchar\_t\* pPrinterName
1305. );
1306. DWORD
1307. RpcEnumPerMachineConnections(
1308. [in, string, unique] STRING\_HANDLE pServer,
1309. [in, out, unique, size\_is(cbBuf), disable\_consistency\_check] BYTE\*
1310. pPrinterEnum,
1311. [in] DWORD cbBuf,
1312. [out] DWORD\* pcbNeeded,
1313. [out] DWORD\* pcReturned
1314. );
1315. DWORD
1316. RpcXcvData(
1317. [in] PRINTER\_HANDLE hXcv,
1318. [in, string] const wchar\_t\* pszDataName,
1319. [in, size\_is(cbInputData)] BYTE\* pInputData,
1320. [in] DWORD cbInputData,
1321. [out, size\_is(cbOutputData)] BYTE\* pOutputData,
1322. [in] DWORD cbOutputData,
1323. [out] DWORD\* pcbOutputNeeded,
1324. [in, out] DWORD\* pdwStatus
1325. );
1326. DWORD
1327. RpcAddPrinterDriverEx(
1328. [in, string, unique] STRING\_HANDLE pName,
1329. [in] DRIVER\_CONTAINER\* pDriverContainer,
1330. [in] DWORD dwFileCopyFlags
1331. );
1332. void
1333. Opnum90NotUsedOnWire();
1334. void
1335. Opnum91NotUsedOnWire();
1336. void
1337. Opnum92NotUsedOnWire();
1338. void
1339. Opnum93NotUsedOnWire();
1340. void
1341. Opnum94NotUsedOnWire();
1342. void
1343. Opnum95NotUsedOnWire();
1344. DWORD
1345. RpcFlushPrinter(
1346. [in] PRINTER\_HANDLE hPrinter,
1347. [in, size\_is(cbBuf)] BYTE\* pBuf,
1348. [in] DWORD cbBuf,
1349. [out] DWORD\* pcWritten,
1350. [in] DWORD cSleep
1351. );
1352. DWORD RpcSendRecvBidiData(
1353. [in] PRINTER\_HANDLE hPrinter,
1354. [in, string, unique] const wchar\_t\* pAction,
1355. [in] RPC\_BIDI\_REQUEST\_CONTAINER\* pReqData,
1356. [out] RPC\_BIDI\_RESPONSE\_CONTAINER\*\* ppRespData);
1357. void
1358. Opnum98NotUsedOnWire();
1359. void
1360. Opnum99NotUsedOnWire();
1361. void
1362. Opnum100NotUsedOnWire();
1363. void
1364. Opnum101NotUsedOnWire();
1365. HRESULT RpcGetCorePrinterDrivers(
1366. [in, string, unique] STRING\_HANDLE pszServer,
1367. [in, string] const wchar\_t \* pszEnvironment,
1368. [in] DWORD cchCoreDrivers,
1369. [in, size\_is(cchCoreDrivers)] const wchar\_t \* pszzCoreDriverDependencies,
1370. [in] DWORD cCorePrinterDrivers,
1371. [out, size\_is(cCorePrinterDrivers)] CORE\_PRINTER\_DRIVER \* pCorePrinterDrivers);
1372. void
1373. Opnum103NotUsedOnWire();
1374. HRESULT RpcGetPrinterDriverPackagePath(
1375. [in, string, unique] STRING\_HANDLE pszServer,
1376. [in, string] const wchar\_t \* pszEnvironment,
1377. [in, string, unique] const wchar\_t \* pszLanguage,
1378. [in, string] const wchar\_t \* pszPackageID,
1379. [in, out, unique, size\_is(cchDriverPackageCab)]
1380. wchar\_t \* pszDriverPackageCab,
1381. [in] DWORD cchDriverPackageCab,
1382. [out] LPDWORD pcchRequiredSize);
1383. void
1384. Opnum105NotUsedOnWire();
1385. void
1386. Opnum106NotUsedOnWire();
1387. void
1388. Opnum107NotUsedOnWire();
1389. void
1390. Opnum108NotUsedOnWire();
1391. void
1392. Opnum109NotUsedOnWire();
1393. DWORD RpcGetJobNamedPropertyValue(
1394. [in] PRINTER\_HANDLE hPrinter,
1395. [in] DWORD JobId,
1396. [in, string] const wchar\_t \*pszName,
1397. [out] RPC\_PrintPropertyValue \*pValue);
1398. DWORD RpcSetJobNamedProperty(
1399. [in] PRINTER\_HANDLE hPrinter,
1400. [in] DWORD JobId,
1401. [in] RPC\_PrintNamedProperty \*pProperty);
1402. DWORD RpcDeleteJobNamedProperty(
1403. [in] PRINTER\_HANDLE hPrinter,
1404. [in] DWORD JobId,
1405. [in, string] const wchar\_t \*pszName);
1406. DWORD RpcEnumJobNamedProperties(
1407. [in] PRINTER\_HANDLE hPrinter,
1408. [in] DWORD JobId,
1409. [out] DWORD \*pcProperties,
1410. [out, size\_is(,\*pcProperties)] RPC\_PrintNamedProperty \*\*ppProperties);
1411. void
1412. Opnum114NotUsedOnWire();}
1413. void
1414. Opnum115NotUsedOnWire();
1415. DWORD
1416. RpcLogJobInfoForBranchOffice(
1417. [in] PRINTER\_HANDLE hPrinter,
1418. [in, ref] RPC\_BranchOfficeJobDataContainer \*pBranchOfficeJobDataContainer);

# Appendix B: Product Behavior

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

Note: Some of the information in this section is subject to change because it applies to an unreleased, preliminary version of the Windows Server operating system, and thus may differ from the final version of the server software when released. All behavior notes that pertain to the unreleased, preliminary version of the Windows Server operating system contain specific references to Windows Server 2016 Technical Preview as an aid to the reader.

* Windows NT 3.1 operating system
* Windows NT 3.5 operating system
* Windows NT 3.51 operating system
* Windows NT 4.0 operating system
* Microsoft Windows 98 operating system
* Windows 2000 operating system
* Windows Millennium Edition 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 Technical Preview operating system

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms SHOULD or SHOULD NOT implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that the product does not follow the prescription.

[<1> Section 1.4](#Appendix_A_Target_1): In Windows NT 3.51 and Windows NT 4.0, print clients and servers do not interact with Active directory. Otherwise, print servers can publish printers to the Active Directory, and print clients can search the Active Directory for printers.

[<2> Section 1.7](#Appendix_A_Target_2): The dwBuildNumber value for [OSVERSIONINFO](#Section_5942648eb54f4e22a0e22d000e084b23) and [OSVERSIONINFOEX (section 2.2.3.10.2)](#Section_cd8e00e84c444e4f98d30db33d75d06d) for specific versions of Windows is shown in the table that follows.

| Version | dwBuildNumber value |
| --- | --- |
| Windows 8.1 and Windows Server 2012 R2 | >= 9431 |
| Windows 8 and Windows Server 2012 | >= 9200 |
| Windows 7 and Windows Server 2008 R2 | >= 7007 |
| Windows Vista operating system with Service Pack 1 (SP1) and Windows Server 2008 | >= 6001 |
| Windows Vista and Windows Server 2008 | >= 6000 |
| Windows XP operating system Service Pack 1 (SP1) | >= 2196 |
| Windows XP and Windows Server 2003 | >= 2196 |
| Windows 2000 | >= 1382 |
| Windows NT 4.0 | >= 1381 |

[<3> Section 2.1](#Appendix_A_Target_3): The Windows server impersonates the client when processing a method, and it registers [**security providers**](#gt_05fd3925-0672-4f24-9dd9-2b9d441eb333) as follows:

* Windows NT 3.1, Windows NT 3.5, Windows NT 3.51, Windows NT 4.0, and Windows 2000 servers do not register a security provider.
* Windows Server 2003 server registers the NT LAN Manager (NTLM) security provider.
* Otherwise, servers register the [**Simple and Protected GSS-API Negotiation Mechanism (SPNEGO)**](#gt_bc2f6b5e-e5c0-408b-8f55-0350c24b9838) security provider.

[<4> Section 2.2.1.2.7](#Appendix_A_Target_4): This feature is supported on the following Windows versions:

* Windows NT 3.5
* Windows NT 3.51
* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<5> Section 2.2.1.2.8](#Appendix_A_Target_5): This feature is supported on the following Windows versions:

* Windows NT 3.5
* Windows NT 3.51
* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<6> Section 2.2.1.2.10](#Appendix_A_Target_6): This feature is supported on the following Windows versions:

* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<7> Section 2.2.1.2.11](#Appendix_A_Target_7): This feature is supported on the following Windows versions:

* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<8> Section 2.2.1.2.12](#Appendix_A_Target_8): This feature is supported on the following Windows versions:

* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<9> Section 2.2.1.2.14](#Appendix_A_Target_9): This feature is supported on the following Windows versions:

* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<10> Section 2.2.1.2.15](#Appendix_A_Target_10): This feature is supported on the following Windows versions:

* Windows NT 3.5
* Windows NT 3.51
* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<11> Section 2.2.1.2.16](#Appendix_A_Target_11): This feature is supported on the following Windows versions:

* Windows NT 3.5
* Windows NT 3.51
* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<12> Section 2.2.1.2.17](#Appendix_A_Target_12): The [RPC\_BranchOfficeJobDataContainer (section 2.2.1.2.17)](#Section_8bed3226cdef4b90b762b35f1a3ad362) structure is supported on the following Windows versions:

* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<13> Section 2.2.1.3.1](#Appendix_A_Target_13): The Windows operating system uses the following values to indicate printer drivers on different OS versions:

| Value | Meaning |
| --- | --- |
| 0x00000000 | The printer driver for Windows 95 operating system, Windows 98, and Windows Millennium Edition. |
| 0x00000001 | The printer driver for Windows NT 3.51. |
| 0x00000002 | Kernel-mode printer driver for Windows NT 4.0. |
| 0x00000003 | User-mode printer driver for Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, and Windows Server 2008 R2. |
| 0x00000004 | User-mode printer driver for Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview. |

[<14> Section 2.2.1.3.1](#Appendix_A_Target_14): In Windows, a language monitor is specified for printers capable of bidirectional communication. The name is specific to a printer manufacturer. For example, the name of a language monitor could be "PJL monitor".

[<15> Section 2.2.1.3.1](#Appendix_A_Target_15): Windows print servers use this ordering for **pDependentFiles** members of returned [DRIVER\_INFO (section 2.2.1.5)](#Section_17943579a5144d0b8ded422265b23f45) and [DRIVER\_INFO and RPC\_DRIVER\_INFO Members (section 2.2.1.3.1)](#Section_4464eaf0f34f40d5b970736437a21913) structures on the following Windows versions:

* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<16> Section 2.2.1.3.1](#Appendix_A_Target_16): Windows operating systems use a combination of the OS major and minor numbers, the build number, and the revision. For example, the printer driver version number 0x000500020ECE0726 represents:

* OS Major Version: 0x0005
* OS Minor Version: 0x0002
* Build number: 0x0ECE (3790)
* Revision: 0x0726 (1830)

Windows clients use this value to check the versions of server printer driver, and when a version does not match, the user is prompted to update the driver.

[<17> Section 2.2.1.3.6](#Appendix_A_Target_17): In Windows, if non-NULL, the string length must be less than or equal to 1,041 characters.

[<18> Section 2.2.1.3.6](#Appendix_A_Target_18): In Windows, if non-NULL, the string length must be less than or equal to 256 characters.

[<19> Section 2.2.1.3.7](#Appendix_A_Target_19): The Windows implementation uses the following major version values.

| Value | Meaning |
| --- | --- |
| 0x00000004 | The operating system is Windows 95, Windows NT 4.0, Windows 98, or Windows Millennium Edition. |
| 0x00000005 | The operating system is Windows 2000, Windows XP, or Windows Server 2003. |
| 0x00000006 | The operating system is Windows Vista, Windows Server 2008, Windows Server 2008 R2, Windows 7, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, or Windows Server 2016 Technical Preview. |

[<20> Section 2.2.1.3.7](#Appendix_A_Target_20): The Windows implementation uses the following minor version values.

| Value | Meaning |
| --- | --- |
| 0x00000000 | The operating system is Windows 95, Windows NT 4.0, Windows 2000, Windows Vista, Windows Server 2008, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, or Windows Server 2016 Technical Preview. |
| 0x00000001 | The operating system is Windows XP, Windows 7, Windows Server 2008, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, or Windows Server 2016 Technical Preview. |
| 0x00000002 | The operating system is Windows XP Professional x64 Edition operating system or Windows Server 2003. |
| 0x0000000A | The operating system is Windows 98. |
| 0x0000005A | The operating system is Windows Millennium Edition. |

[<21> Section 2.2.1.3.7](#Appendix_A_Target_21): The Windows implementation uses the following processor architecture values:

| Name/Value | Meaning |
| --- | --- |
| PROCESSOR\_ARCHITECTURE\_INTEL  0x0000 | x86 architecture |
| PROCESSOR\_ARCHITECTURE\_IA64  0x0006 | Itanium architecture |
| PROCESSOR\_ARCHITECTURE\_AMD64  0x0009 | AMD64 architecture |
| PROCESSOR\_ARCHITECTURE\_ARM  0x0005 | ARM architecture |

[<22> Section 2.2.1.4](#Appendix_A_Target_22): In Windows the default job name is "No Document Name".

[<23> Section 2.2.1.5.3](#Appendix_A_Target_23): This feature is supported on the following Windows versions:

* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<24> Section 2.2.1.5.4](#Appendix_A_Target_24): This feature is supported on the following Windows version:

* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<25> Section 2.2.1.5.5](#Appendix_A_Target_25): This feature is supported on the following Windows version:

* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<26> Section 2.2.1.5.6](#Appendix_A_Target_26): This feature is supported on the following Windows versions:

* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<27> Section 2.2.1.5.6](#Appendix_A_Target_27): Windows uses INF files to define the printer driver configuration.

[<28> Section 2.2.1.5.6](#Appendix_A_Target_28): This flag was introduced with Windows Vista.

[<29> Section 2.2.1.5.6](#Appendix_A_Target_29): Windows print servers determine that a printer driver supports the Microsoft XML Paper Specification (XPS) format described in [[MSFT-XMLPAPER]](http://go.microsoft.com/fwlink/?LinkId=90214) if and only if the list of dependent files associated with that printer driver contains the [**file**](#gt_a04c146a-de3b-4e4b-829f-a9e772f3fe25) "PipelineConfig.xml".

This feature is supported on the following Windows versions:

* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<30> Section 2.2.1.5.6](#Appendix_A_Target_30): This flag is supported on Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<31> Section 2.2.1.5.6](#Appendix_A_Target_31): This feature is supported on the following Windows versions:

* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<32> Section 2.2.1.5.6](#Appendix_A_Target_32): This flag is supported on Windows 7 and Windows Server 2008 R2.

[<33> Section 2.2.1.5.6](#Appendix_A_Target_33): This feature is supported on the following Windows versions:

* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<34> Section 2.2.1.5.6](#Appendix_A_Target_34): This flag is supported on Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<35> Section 2.2.1.5.6](#Appendix_A_Target_35): This feature is supported on the following Windows versions:

* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<36> Section 2.2.1.5.6](#Appendix_A_Target_36): This flag is supported on Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<37> Section 2.2.1.5.6](#Appendix_A_Target_37): This feature is supported on the following Windows versions:

* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<38> Section 2.2.1.5.6](#Appendix_A_Target_38): This flag is supported on Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<39> Section 2.2.1.5.6](#Appendix_A_Target_39): This feature is supported on the following Windows versions:

* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<40> Section 2.2.1.5.6](#Appendix_A_Target_40): This flag is supported on Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<41> Section 2.2.1.5.6](#Appendix_A_Target_41): This feature is supported on the following Windows versions:

* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<42> Section 2.2.1.5.6](#Appendix_A_Target_42): This flag is supported on Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<43> Section 2.2.1.5.6](#Appendix_A_Target_43): This feature is supported on the following Windows versions:

* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<44> Section 2.2.1.5.6](#Appendix_A_Target_44): This flag is supported on Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<45> Section 2.2.1.5.6](#Appendix_A_Target_45): This feature is supported on the following Windows versions:

* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<46> Section 2.2.1.5.6](#Appendix_A_Target_46): This flag is supported on Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<47> Section 2.2.1.5.6](#Appendix_A_Target_47): When the print queue is associated with a printer port corresponding to a USB printer, Windows print servers use the USB soft reset mechanism described in [[USBPRINT]](http://go.microsoft.com/fwlink/?LinkId=240056) section 4.2.3. This feature is supported on the following Windows versions:

* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<48> Section 2.2.1.5.6](#Appendix_A_Target_48): This flag is supported on Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<49> Section 2.2.1.5.6](#Appendix_A_Target_49): 3D printers are supported on the following Windows versions:

* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<50> Section 2.2.1.5.6](#Appendix_A_Target_50): This flag is supported on Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<51> Section 2.2.1.5.6](#Appendix_A_Target_51): The concepts of core printer driver and core driver dependencies are Windows–specific implementation details.

[<52> Section 2.2.1.5.6](#Appendix_A_Target_52): In Windows, this member applies to only package-aware printer driver.

[<53> Section 2.2.1.5.6](#Appendix_A_Target_53): Windows reads this value from the printer driver INF file.

[<54> Section 2.2.1.6.2](#Appendix_A_Target_54): This feature is supported on the following Windows versions:

* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<55> Section 2.2.1.6.2](#Appendix_A_Target_55): Printer drivers generate an implementation-specific unique identifier. Windows print servers generate a unique GUID.

[<56> Section 2.2.1.6.2](#Appendix_A_Target_56): Windows provides Multilingual User Interface (MUI) DLLs that contain localized string resources for inbox printer drivers. For third-party printer drivers, whether to localize strings is an implementation decision.

[<57> Section 2.2.1.7.3](#Appendix_A_Target_57): This feature is supported on the following Windows versions:

* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<58> Section 2.2.1.7.4](#Appendix_A_Target_58): This feature is supported on the following Windows versions:

* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<59> Section 2.2.1.9.2](#Appendix_A_Target_59): Windows specifies a descriptive name for the port monitor. For example: "Standard TCP/IP Port", "Fax Monitor Port", or "Local Port".

[<60> Section 2.2.1.9.3](#Appendix_A_Target_60): This feature is supported on the following Windows versions:

* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<61> Section 2.2.1.9.4](#Appendix_A_Target_61): This feature is supported on the following Windows versions:

* Windows NT 3.5
* Windows NT 3.51
* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<62> Section 2.2.1.10.1](#Appendix_A_Target_62): This feature is supported on the following Windows versions:

* Windows NT 3.5
* Windows NT 3.51
* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<63> Section 2.2.1.10.1](#Appendix_A_Target_63): Windows calculates the version by storing the build version in the high-order 16 bits, and the operation system release number in the low-order 16 bits. For example, 0x0A280005 corresponds to XP build 2600.

[<64> Section 2.2.1.10.1](#Appendix_A_Target_64): The Windows debugging build of the print server sets **fFreeBuild** to 0, and the release build of the print server sets **fFreeBuild** to 1.

[<65> Section 2.2.1.10.1](#Appendix_A_Target_65): Windows uses the following values:

| Value | Meaning |
| --- | --- |
| PROCESSOR\_INTEL\_386  0x00000182 | Intel 80386 compatible |
| PROCESSOR\_INTEL\_486  0x000001E6 | Intel 80486 compatible |
| PROCESSOR\_INTEL\_PENTIUM  0x0000024A | Intel Pentium compatible |
| PROCESSOR\_INTEL\_IA64  0x00000898 | Intel Itanium-based compatible |
| PROCESSOR\_AMD\_X8664  0x000022A0 | AMD x64 compatible |
| PROCESSOR\_ARM  0x00000000 | ARM compatible |

[<66> Section 2.2.1.10.1](#Appendix_A_Target_66): Windows returns a nonzero Win32 error code to indicate failure ([[MS-ERREF]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-ERREF%5d.pdf) section 2.2).

[<67> Section 2.2.1.10.1](#Appendix_A_Target_67): Windows uses the following values:

| Value | Meaning |
| --- | --- |
| PROCESSOR\_ARCHITECTURE\_INTEL  0x0000 | x86 architecture |
| PROCESSOR\_ARCHITECTURE\_IA64  0x0006 | Itanium architecture |
| PROCESSOR\_ARCHITECTURE\_AMD64  0x0009 | AMD64 architecture |
| PROCESSOR\_ARCHITECTURE\_ARM  0x0005 | ARM architecture |

[<68> Section 2.2.1.10.1](#Appendix_A_Target_68): Windows uses the value of 1 for **PROCESSOR\_ARCHITECTURE\_IA64** and **PROCESSOR\_ARCHITECTURE\_AMD64**.

For **PROCESSOR\_ARCHITECTURE\_INTEL** and **PROCESSOR\_ARCHITECTURE\_ARM**, Windows uses the value defined by the CPU vendor.

[<69> Section 2.2.1.10.3](#Appendix_A_Target_69): Windows servers select the first data type in the list of data types obtained from the print processor. This list is obtained by the same mechanism as in [RpcEnumPrintProcessorDatatypes (section 3.1.4.8.5)](#Section_c4a35c2040824ce6840102c52ee3158e).

[<70> Section 2.2.1.10.5](#Appendix_A_Target_70): This feature is supported on the following Windows versions:

* Windows NT 3.5
* Windows NT 3.51
* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<71> Section 2.2.1.10.6](#Appendix_A_Target_71): This feature is supported on the following Windows versions:

* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<72> Section 2.2.1.10.7](#Appendix_A_Target_72): This feature is supported on the following Windows versions:

* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<73> Section 2.2.1.10.8](#Appendix_A_Target_73): This feature is supported on the following Windows versions:

* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<74> Section 2.2.1.10.9](#Appendix_A_Target_74): This feature is supported on the following Windows version:

* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<75> Section 2.2.1.10.10](#Appendix_A_Target_75): This feature is supported on the following Windows version:

* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<76> Section 2.2.1.11.1](#Appendix_A_Target_76): This feature is supported on the following Windows versions:

* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<77> Section 2.2.1.11.3](#Appendix_A_Target_77): This feature is supported on the following Windows versions:

* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<78> Section 2.2.1.12.1](#Appendix_A_Target_78): This feature is supported on the following Windows versions:

* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<79> Section 2.2.1.12.1](#Appendix_A_Target_79): The bidirectional communications schema is a hierarchy of printer attributes, some of which are properties, with the rest being values or value entries. Bidirectional communications interfaces are implemented by printer-specific components. A detailed description of printer drivers and the bidirectional communications schema can be found in the Windows Device Driver Kit. See [[MSDN-MPD]](http://go.microsoft.com/fwlink/?LinkId=91840) and [[MSDN-BIDI]](http://go.microsoft.com/fwlink/?LinkId=122185) for further information.

[<80> Section 2.2.1.12.2](#Appendix_A_Target_80): This feature is supported on the following Windows versions:

* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<81> Section 2.2.1.12.2](#Appendix_A_Target_81): Windows returns a nonzero error code to indicate failure, as specified in [MS-ERREF].

[<82> Section 2.2.1.12.2](#Appendix_A_Target_82): The bidirectional communications schema is a hierarchy of printer attributes, some of which are properties, with the rest being values or value entries. Bidirectional communications interfaces are implemented by printer-specific components. A detailed description of printer drivers and the bidirectional communications schema can be found in the Windows Device Driver Kit.

[<83> Section 2.2.1.12.3](#Appendix_A_Target_83): This feature is supported on the following Windows versions:

* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<84> Section 2.2.1.13.1](#Appendix_A_Target_84): This feature is supported on the following Windows versions:

* Windows NT 3.5
* Windows NT 3.51
* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<85> Section 2.2.1.13.2](#Appendix_A_Target_85): This feature is supported on the following Windows versions:

* Windows NT 3.5
* Windows NT 3.51
* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<86> Section 2.2.1.13.2](#Appendix_A_Target_86): This feature is supported on the following Windows versions:

* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<87> Section 2.2.1.13.3](#Appendix_A_Target_87): This feature is supported on the following Windows versions:

* Windows NT 3.5
* Windows NT 3.51
* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<88> Section 2.2.1.13.4](#Appendix_A_Target_88): This feature is supported on the following Windows versions:

* Windows NT 3.5
* Windows NT 3.51
* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<89> Section 2.2.1.13.5](#Appendix_A_Target_89): This feature is supported on the following Windows versions:

* Windows NT 3.5
* Windows NT 3.51
* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<90> Section 2.2.1.13.6](#Appendix_A_Target_90): This feature is supported on the following Windows versions:

* Windows NT 3.5
* Windows NT 3.51
* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<91> Section 2.2.1.14.1](#Appendix_A_Target_91): The **RPC\_PrintPropertyValue** structure is supported by the following Windows versions:

* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<92> Section 2.2.1.14.2](#Appendix_A_Target_92): The **RPC\_PrintNamedProperty** structure is supported by the following Windows versions:

* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<93> Section 2.2.1.14.3](#Appendix_A_Target_93): The **RPC\_EPrintPropertyType** enumeration is supported by the following Windows versions:

* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<94> Section 2.2.1.14.4](#Appendix_A_Target_94): The **Job Named Property** with the name defined by the SPL\_FILE\_CONTENT\_TYPE\_PROP\_NAME constant is supported by the following Windows versions:

* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<95> Section 2.2.1.15.1](#Appendix_A_Target_95): The [EBranchOfficeJobEventType (section 2.2.1.15.1)](#Section_d070101180b3473d843cedf7e4c5bc86) enumeration is supported on the following Windows versions:

* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<96> Section 2.2.1.15.2](#Appendix_A_Target_96): The [RPC\_BranchOfficeJobData (section 2.2.1.15.2)](#Section_6e442be62c54497593c685895289ba0d) structure is supported on the following Windows versions:

* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<97> Section 2.2.1.15.3](#Appendix_A_Target_97): The [RPC\_BranchOfficeJobDataError (section 2.2.1.15.3)](#Section_d86fd29b95124c80a14b5f5fb25e5975) structure is supported on the following Windows versions:

* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<98> Section 2.2.1.15.4](#Appendix_A_Target_98): The [RPC\_BranchOfficeJobDataPipelineFailed (section 2.2.1.15.4)](#Section_31a2aecb2d7a4846a93be4fd708863e0) structure is supported on the following Windows versions:

* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<99> Section 2.2.1.15.5](#Appendix_A_Target_99): The [RPC\_BranchOfficeJobDataPrinted (section 2.2.1.15.5)](#Section_b4a220487837460eb1bd8d34f4270f29) structure is supported on the following Windows versions:

* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<100> Section 2.2.1.15.6](#Appendix_A_Target_100): The [RPC\_BranchOfficeJobDataRendered (section 2.2.1.15.6)](#Section_e90ee32f12f44310923871aca7dd9cb1) structure is supported on the following Windows versions:

* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<101> Section 2.2.1.15.7](#Appendix_A_Target_101): The [RPC\_BranchOfficeLogOfflineFileFull (section 2.2.1.15.7)](#Section_55fcc7c1d63e448888cd8ce5d9dd945c) structure is supported on the following Windows versions:

* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<102> Section 2.2.1.15.7](#Appendix_A_Target_102): The maximum allowed size is determined by the implementation. On Windows, the default value is 10 MB.

[<103> Section 2.2.2](#Appendix_A_Target_103): The 64-bit Edition of the following Windows versions do not correctly handle custom-marshaled **INFO** structures where the unused space is not between the end of the last **Fixed\_Portion** block and the beginning of the first **Variable\_Data** field.

* Windows XP
* Windows Server 2003
* Windows Vista
* Windows Server 2008
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<104> Section 2.2.2.1](#Appendix_A_Target_104): Versions of initialization data specifications correspond to versions of Windows operating systems as follows.

| Value | Meaning |
| --- | --- |
| 0x0320 | Windows NT 3.1, Windows NT 3.5, and Windows NT 3.51. |
| 0x0400 | Windows 95, Windows 98, and Windows Millennium Edition. |
| 0x0401 | Windows NT 4.0, Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview. |

[<105> Section 2.2.2.1](#Appendix_A_Target_105): Versions of printer drivers correspond to versions of Windows operating systems as follows.

| Value | Meaning |
| --- | --- |
| 0x0301 — 0x03FF | Windows NT 3.1, Windows NT 3.5, and Windows NT 3.51 user-mode printer drivers, and Windows NT 4.0 kernel-mode printer drivers. |
| 0x0500 — 0x05FF | Windows 2000, Windows XP, and Windows Server 2003 user-mode printer drivers. |
| 0x0600 — 0x06FF | Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview printer drivers. |

[<106> Section 2.2.2.1](#Appendix_A_Target_106): Supported on Windows NT 4.0, Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<107> Section 2.2.2.1](#Appendix_A_Target_107): Supported on Windows NT 4.0, Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<108> Section 2.2.2.1](#Appendix_A_Target_108): Supported on Windows NT 4.0, Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<109> Section 2.2.2.1](#Appendix_A_Target_109): Supported on Windows NT 4.0, Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<110> Section 2.2.2.1](#Appendix_A_Target_110): Supported on Windows NT 4.0, Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<111> Section 2.2.2.1](#Appendix_A_Target_111): Supported on Windows NT 4.0, Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<112> Section 2.2.2.1](#Appendix_A_Target_112): Supported on Windows NT 4.0, Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<113> Section 2.2.2.1](#Appendix_A_Target_113): Supported on Windows NT 4.0, Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<114> Section 2.2.2.1](#Appendix_A_Target_114): Supported on Windows NT 4.0, Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<115> Section 2.2.2.1](#Appendix_A_Target_115): Supported on Windows NT 4.0, Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<116> Section 2.2.2.1](#Appendix_A_Target_116): Supported on Windows NT 4.0, Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<117> Section 2.2.2.1](#Appendix_A_Target_117): Supported on Windows NT 4.0, Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<118> Section 2.2.2.1](#Appendix_A_Target_118): Supported on Windows NT 4.0, Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<119> Section 2.2.2.1](#Appendix_A_Target_119): Supported on Windows NT 4.0, Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<120> Section 2.2.2.1](#Appendix_A_Target_120): Supported on Windows NT 4.0, Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<121> Section 2.2.2.1](#Appendix_A_Target_121): Supported on Windows NT 4.0, Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<122> Section 2.2.2.1](#Appendix_A_Target_122): Supported on Windows NT 4.0, Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<123> Section 2.2.2.1](#Appendix_A_Target_123): Supported on Windows NT 4.0, Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<124> Section 2.2.2.1](#Appendix_A_Target_124): Supported on Windows NT 4.0, Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<125> Section 2.2.2.1](#Appendix_A_Target_125): Supported on Windows NT 4.0, Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<126> Section 2.2.2.1](#Appendix_A_Target_126): Supported on Windows NT 4.0, Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<127> Section 2.2.2.1](#Appendix_A_Target_127): Supported on Windows NT 4.0, Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<128> Section 2.2.2.1](#Appendix_A_Target_128): Supported on Windows NT 4.0, Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<129> Section 2.2.2.1](#Appendix_A_Target_129): Supported on Windows NT 4.0, Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<130> Section 2.2.2.1](#Appendix_A_Target_130): Supported on Windows NT 4.0, Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<131> Section 2.2.2.1](#Appendix_A_Target_131): Supported on Windows NT 4.0, Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<132> Section 2.2.2.1](#Appendix_A_Target_132): Supported on Windows NT 4.0, Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<133> Section 2.2.2.1](#Appendix_A_Target_133): Supported on Windows NT 4.0, Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<134> Section 2.2.2.1](#Appendix_A_Target_134): Supported on Windows NT 4.0, Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<135> Section 2.2.2.1](#Appendix_A_Target_135): Supported on Windows NT 4.0, Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<136> Section 2.2.2.1](#Appendix_A_Target_136): Supported on Windows NT 4.0, Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<137> Section 2.2.2.1](#Appendix_A_Target_137): Supported on Windows NT 4.0, Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<138> Section 2.2.2.1](#Appendix_A_Target_138): Supported on Windows NT 4.0, Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<139> Section 2.2.2.1](#Appendix_A_Target_139): Supported on Windows NT 4.0, Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<140> Section 2.2.2.1](#Appendix_A_Target_140): Supported on Windows NT 4.0, Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<141> Section 2.2.2.1](#Appendix_A_Target_141): Supported on Windows NT 4.0, Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<142> Section 2.2.2.1](#Appendix_A_Target_142): Supported on Windows NT 4.0, Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<143> Section 2.2.2.1](#Appendix_A_Target_143): Supported on Windows NT 4.0, Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<144> Section 2.2.2.1](#Appendix_A_Target_144): Supported on Windows NT 4.0, Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<145> Section 2.2.2.1](#Appendix_A_Target_145): Supported on Windows NT 4.0, Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<146> Section 2.2.2.1](#Appendix_A_Target_146): Supported on Windows NT 4.0, Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<147> Section 2.2.2.1](#Appendix_A_Target_147): Supported on Windows NT 4.0, Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<148> Section 2.2.2.1](#Appendix_A_Target_148): Supported on Windows NT 4.0, Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<149> Section 2.2.2.1](#Appendix_A_Target_149): Supported on Windows NT 4.0, Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<150> Section 2.2.2.1](#Appendix_A_Target_150): Supported on Windows NT 4.0, Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<151> Section 2.2.2.1](#Appendix_A_Target_151): Supported on Windows NT 4.0, Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<152> Section 2.2.2.1](#Appendix_A_Target_152): Supported on Windows NT 4.0, Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<153> Section 2.2.2.1](#Appendix_A_Target_153): Supported on Windows NT 4.0, Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<154> Section 2.2.2.1](#Appendix_A_Target_154): Supported on Windows NT 4.0, Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<155> Section 2.2.2.1](#Appendix_A_Target_155): Supported on Windows 95, Windows NT 4.0, Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<156> Section 2.2.2.1](#Appendix_A_Target_156): The value of this field is set by printer manufacturers, depending on the requirements of the printer driver.

[<157> Section 2.2.2.1.1](#Appendix_A_Target_157): **PSCRIPT** is the Windows PostScript core printer driver. It stores the private data that is required in the [\_DEVMODE](#Section_ccc2a501794e4d2bb312f69c75131c2e) **dmDriverExtraData** field with the following structures.

The **dmDriverExtraData** field contains one of the **PSDRVEXTRA** structures, immediately followed by zero or one [JTEXP (section 2.2.2.1.4)](#Section_2ca89612aa0a44f68fa49042b9c960a8) structures, followed by zero or more [OEM\_DMEXTRA (section 2.2.2.1.3)](#Section_8307eefb5edc47eab3effd8e6d58ba1b) structures.

These structures are not part of the protocol defined in this specification, and they are subject to change without notice. Implementations of compatible drivers must check the structure version and discard any data they do not handle.

**PSDRVEXTRA351**: Defined by the **PSCRIPT** driver released with Windows NT 3.51. This structure is used for **dmDriverExtraData** if the **dmDriverVersion** of the \_DEVMODE structure is 0x0350.

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| dwSignature | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwFlags | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| wchEPSFile | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (wchEPSFile cont'd for 12 rows) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| caSize | | | | | | | | | | | | | | | | caFlags | | | | | | | | | | | | | | | |
| caIlluminantIndex | | | | | | | | | | | | | | | | caRedGamma | | | | | | | | | | | | | | | |
| caGreenGamma | | | | | | | | | | | | | | | | caBlueGamma | | | | | | | | | | | | | | | |
| caReferenceBlack | | | | | | | | | | | | | | | | caReferenceWhite | | | | | | | | | | | | | | | |
| caContrast | | | | | | | | | | | | | | | | caBrightness | | | | | | | | | | | | | | | |
| caColorfulness | | | | | | | | | | | | | | | | caRedGreenTint | | | | | | | | | | | | | | | |

**PSDRVEXTRA400**: Defined by the **PSCRIPT** driver released with Windows NT 4.0. This structure is used for **dmDriverExtraData** if the **dmDriverVersion** of the \_DEVMODE structure is 0x0400.

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| dwSignature | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwFlags | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| wchEPSFile | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (wchEPSFile cont'd for 12 rows) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| caSize | | | | | | | | | | | | | | | | caFlags | | | | | | | | | | | | | | | |
| caIlluminantIndex | | | | | | | | | | | | | | | | caRedGamma | | | | | | | | | | | | | | | |
| caGreenGamma | | | | | | | | | | | | | | | | caBlueGamma | | | | | | | | | | | | | | | |
| caReferenceBlack | | | | | | | | | | | | | | | | caReferenceWhite | | | | | | | | | | | | | | | |
| caContrast | | | | | | | | | | | | | | | | caBrightness | | | | | | | | | | | | | | | |
| caColorfulness | | | | | | | | | | | | | | | | caRedGreenTint | | | | | | | | | | | | | | | |
| wChecksum | | | | | | | | | | | | | | | | wOptions | | | | | | | | | | | | | | | |
| aubOptions | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (aubOptions cont'd for 8 rows) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**wChecksum**: The value of this field is a checksum of the **aubOptions** array.

**wOptions**: The value of this field is the number of entries in the **aubOptions** array that are initialized.

**aubOptions**: This field is an array of 64 bytes in length and contains user interface selections. Unused fields should be initialized to zero. The meaning of the entries in this array differs for each supported printer model. Upon receipt, the checksum of this array is computed and compared to **wChecksum**. The array is used only if the checksums match.

**PSDRVEXTRA500**: Defined by the **PSCRIPT** driver released with Windows 2000 and Windows XP. This structure is used for **dmDriverExtraData** if the **dmDriverVersion** of the \_DEVMODE structure is 0x0501.

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| dwSignature | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwFlags | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| wchEPSFile | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (wchEPSFile cont'd for 12 rows) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| caSize | | | | | | | | | | | | | | | | caFlags | | | | | | | | | | | | | | | |
| caIlluminantIndex | | | | | | | | | | | | | | | | caRedGamma | | | | | | | | | | | | | | | |
| caGreenGamma | | | | | | | | | | | | | | | | caBlueGamma | | | | | | | | | | | | | | | |
| caReferenceBlack | | | | | | | | | | | | | | | | caReferenceWhite | | | | | | | | | | | | | | | |
| caContrast | | | | | | | | | | | | | | | | caBrightness | | | | | | | | | | | | | | | |
| caColorfulness | | | | | | | | | | | | | | | | caRedGreenTint | | | | | | | | | | | | | | | |
| wReserved1 | | | | | | | | | | | | | | | | wSize | | | | | | | | | | | | | | | |
| fxScrFreq | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| fxScrAngle | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| iDialect | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| iTTDLFmt | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| bReversePrint | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| iLayout | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| iPSLevel | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwReserved2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| wOEMExtra | | | | | | | | | | | | | | | | wVer | | | | | | | | | | | | | | | |
| dwX | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwY | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwWidthOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwHeightOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| wFeedDirection | | | | | | | | | | | | | | | | wCutSheet | | | | | | | | | | | | | | | |
| dwReserved3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwChecksum32 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwOptions | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| aOptions[0] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (aOptions array cont'd for 126 elements) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**wReserved1**: The value of this field should be set to zero when sent and must be ignored on receipt.

**wSize**: This field is the same as **wCoreFullSize** in **PSDRVEXTRA**.

**PSDRVEXTRA**: Defined by the **PSCRIPT** driver released with Windows Vista, Windows Server 2008, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview. This structure is used for **dmDriverExtraData** if the **dmDriverVersion** of the \_DEVMODE structure is 0x0600.

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| dwSignature | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwFlags | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| wchEPSFile | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (wchEPSFile cont'd for 12 rows) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| caSize | | | | | | | | | | | | | | | | caFlags | | | | | | | | | | | | | | | |
| caIlluminantIndex | | | | | | | | | | | | | | | | caRedGamma | | | | | | | | | | | | | | | |
| caGreenGamma | | | | | | | | | | | | | | | | caBlueGamma | | | | | | | | | | | | | | | |
| caReferenceBlack | | | | | | | | | | | | | | | | caReferenceWhite | | | | | | | | | | | | | | | |
| caContrast | | | | | | | | | | | | | | | | caBrightness | | | | | | | | | | | | | | | |
| caColorfulness | | | | | | | | | | | | | | | | caRedGreenTint | | | | | | | | | | | | | | | |
| wCoreJTExpSize | | | | | | | | | | | | | | | | wCoreFullSize | | | | | | | | | | | | | | | |
| fxScrFreq | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| fxScrAngle | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| iDialect | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| iTTDLFmt | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| bReversePrint | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| iLayout | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| iPSLevel | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwReserved2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| wOEMExtra | | | | | | | | | | | | | | | | wVer | | | | | | | | | | | | | | | |
| dwX | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwY | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwWidthOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwHeightOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| wFeedDirection | | | | | | | | | | | | | | | | wCutSheet | | | | | | | | | | | | | | | |
| dwReserved3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwChecksum32 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwOptions | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| aOptions[0] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (aOptions array cont'd for 126 elements) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwNupDirection | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwNupBorderFlags | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwBookletFlags | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwPadding | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**wCoreJTExpSize**: The value of this field specifies the size of the JTEXP structure if one is present, following directly after this structure.

**wCoreFullSize**: The value of this field specifies the size of the **PSDRVEXTRA** structure plus the value of **wCoreJTExpSize**.

**dwNupDirection**: This field is used only if N-Up printing is selected, and the value of this field must be one of the following.

| Value | Meaning |
| --- | --- |
| 0x00000001 | Print N-Up pages left-to-right, top-to-bottom. |
| 0x00000002 | Print N-Up pages top-to-bottom, left-to-right. |
| 0x00000004 | Print N-Up pages right-to-left, top-to-bottom. |
| 0x00000008 | Print N-Up pages top-to-bottom, right-to-left. |

**dwNupBorderFlags**: This field is used only if N-Up printing is selected, and the value of this field must be one of the following.

| Value | Meaning |
| --- | --- |
| 0x00000000 | Print borders around N-Up pages. |
| 0x00000001 | Do not print borders around N-Up pages. |

**dwBookletFlags**: This field is used only if booklet printing is selected, and the value of this field must be one of the following.

| Value | Meaning |
| --- | --- |
| 0x00000000 | Print booklet so that pages flip to the left (western style). |
| 0x00000001 | Print booklet so that pages flip to the right. |

**dwPadding**: The value of this field should be set to zero when sent and must be ignored on receipt.

Description of common members for **PSDRVEXTRA** structures:

**dwSignature**: The value of this field must be 0x56495250.

**dwFlags**: The value of this field must be the result of the bitwise OR of zero or more of the following values.

| Value | Meaning |
| --- | --- |
| 0x00000002 | Send PostScript driver error handler code. |
| 0x00000004 | Print mirror image. |
| 0x00000010 | Print negative image of page. |
| 0x00000040 | Compress bitmaps. |
| 0x00000200 | If \_DEVMODE field **dmOrientation** is set to 2 (LANDSCAPE), rotate page by an additional 180 degrees. |
| 0x00002000 | If driver is initialized by GDI, inform GDI that metafile spooling is requested. |

**wchEPSFile**: This field is not used.

**caSize**: The value of this field must be 24.

**caFlags**: This field is not used.

**caIlluminantIndex**: This field is not used.

**caRedGamma**: This field is not used.

**caGreenGamma**: This field is not used.

**caBlueGamma**: This field is not used.

**caReferenceBlack**: This field is not used.

**caReferenceWhite**: This field is not used.

**caContrast**: This field is not used.

**caBrightness**: This field is not used.

**caColorfulness**: This field is not used.

**caRedGreenTint**: This field is not used.

**fxScrFreq**: This field is not used.

**fxScrAngle**: This field is not used.

**iDialect**: The value of this field must be one of the following.

| Value | Meaning |
| --- | --- |
| 0x00000000 | Optimize generated PostScript for speed. |
| 0x00000001 | Optimize generated PostScript for portability. |
| 0x00000002 | Optimize generated PostScript for EPS use. |
| 0x00000003 | Optimize generated PostScript for archival. |

**iTTDLFmt**: The value of this field must be one of the following.

| Value | Meaning |
| --- | --- |
| 0x00000000 | Download fonts in default format, pick the best suited format for font. |
| 0x00000001 | Download fonts as Type 1 outlines. |
| 0x00000002 | Download fonts as Type 3 bitmaps. |
| 0x00000003 | Download fonts as Type 42 fonts. |
| 0x00000004 | Same as 0x00000000. |
| 0x00000005 | Same as 0x00000000. |

**bReversePrint**: If the value of this field is nonzero, print pages in reverse order; otherwise, print pages in normal order.

**iLayout**: The value of this field must be one of the following.

| Value | Meaning |
| --- | --- |
| 0x00000000 | N-Up printing disabled. |
| 0x00000001 | Print 2-Up. |
| 0x00000002 | Print 4-Up. |
| 0x00000003 | Print 6-Up. |
| 0x00000004 | Print 9-Up. |
| 0x00000005 | Print 16-Up. |
| 0x00000006 | Print as a booklet. |

**iPSLevel**: The value of this field must be one of the following.

| Value | Meaning |
| --- | --- |
| 0x00000001 | Use only PostScript level 1 features. |
| 0x00000002 | Use only PostScript level 1 and level 2 features. |
| 0x00000003 | Use all PostScript features available for level 1, level 2, and level 3. |

**dwReserved2**: The value of this field should be set to zero when sent and MUST be ignored on receipt.

**wOEMExtra**: The value of this field must be the total size of the private data that is used by the vendor-supplied driver [**plug-in**](#gt_c731fa43-8a11-4f66-91f8-8519571ce1bf). This data immediately follows this structure.

**wVer**: The value of this field must be 0x0010.

**dwX**: The value of this field specifies the width, in 1/1000th millimeter units, of the custom paper size. Used only if the **dmPaperSize** field of \_DEVMODE is set to 0x7FFF.

**dwY**: The value of this field specifies the height, in 1/1000th millimeter units, of the custom paper size. Used only if the **dmPaperSize** field of \_DEVMODE is set to 0x7FFF.

**dwWidthOffset**: The value of this field specifies the left unprintable margin, in 1/1000th of a millimeter, of the custom paper size. Used only if the **dmPaperSize** field of \_DEVMODE is set to 0x7FFF.

**dwHeightOffset**: The value of this field specifies the top unprintable margin, in 1/1000th of a millimeter, of the custom paper size. Used only if the **dmPaperSize** field of \_DEVMODE is set to 0x7FFF.

**wFeedDirection**: The value of this field must be one of the following.

| Value | Meaning |
| --- | --- |
| 0x0000 | The paper is physically fed into the print mechanism with its long edge first. |
| 0x0001 | The paper is physically fed into the print mechanism with its short edge first. |
| 0x0002 | The paper is physically fed into the print mechanism with its long edge first, but upside down. |
| 0x0003 | The paper is physically fed into the print mechanism with its long edge first, but upside down. |

**wCutSheet**: The value of this field is zero for roll-fed custom paper sizes and nonzero for cut sheet custom paper. This field is used only if the **dmPaperSize** field of \_DEVMODE is set to 0x7FFF.

**dwReserved3**: The value of this field should be set to zero when sent and MUST be ignored on receipt.

**dwChecksum32**: The value of this field is the checksum of the names of the vendor-defined features and feature options that are supported by the printer model, as provided by the printer driver. The checksum is calculated using the 32-bit [**cyclic redundancy check (CRC)**](#gt_9cb45a36-92bb-4c14-b2fd-2ad7e2979bfd) function defined in section 3.2.9 of [[IEEE802.3-2008]](http://go.microsoft.com/fwlink/?LinkId=161001). For each feature, the checksum is calculated on the null-terminated [**ASCII**](#gt_79fa85ca-ac61-467c-b819-e97dc1a7a599) string representations of the feature name and each of the feature options in order. The checksum is accumulated in this manner over all of the features supported by the printer model.

**dwOptions**: The value of this field specifies the number of entries in the **aOptions** array.

**aOptions**: This field is an array that is 512 bytes long and contains the options selected by the user for each vendor-defined feature. Unused fields should be initialized to zero. The meaning of the entries differs for each supported printer model.

[<158> Section 2.2.2.1.2](#Appendix_A_Target_158): **UNIDRV** is the generic Windows core printer driver for all printers that do not use PostScript. It stores its private data in the \_DEVMODE **dmDriverExtraData** area using the following structures.

The **dmDriverExtraData** field contains one of the **UNIDRVEXTRA** structures, immediately followed by zero or one JTEXP (section 2.2.2.1.4) structure, followed by zero or more OEM\_DMEXTRA (section 2.2.2.1.3) structures.

These structures are not part of the protocol defined in this specification, and they are subject to change without notice. Implementations of compatible drivers must check the structure version and discard any data they do not handle.

**UNIDRVEXTRA3\_4**: defined by the **UNIDRV** driver released with Windows NT 3.5 and Windows NT 4.0. This structure is used for **dmDriverExtraData** if the **dmDriverVersion** field of the \_DEVMODE structure is 0x0301.

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| wReserved[0] | | | | | | | | | | | | | | | | ... | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | (repeats for total of 56 reserved words) | | | | | | | | | | | | | | | |

**wReserved**: This field is no longer used by **UNIDRV**.

**UNIDRVEXTRA500**: defined by the **UNIDRV** driver released with Windows 2000 and Windows XP. This structure is used for **dmDriverExtraData** if the **dmDriverVersion** field of the DEVMODE structure is 0x0500.

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| dwSignature | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| wVer | | | | | | | | | | | | | | | | sPadding | | | | | | | | | | | | | | | |
| wSize | | | | | | | | | | | | | | | | wOEMExtra | | | | | | | | | | | | | | | |
| dwChecksum32 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwFlags | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| bReversePrint | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| iLayout | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| iQuality | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| wReserved | | | | | | | | | | | | | | | | ... | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | ... | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | ... | | | | | | | | | | | | | | | |
| dwOptions | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| aOptions[0] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (repeats for a total of 128 aOptions array elements) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| aOptions[127] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**sPadding**: The value of this field SHOULD be set to zero when sent and MUST be ignored on receipt.

**wSize**: This field is the same as **wCoreFullSize** in **UNIDRVEXTRA**.

**UNIDRVEXTRA**: defined by the **UNIDRV** driver released with Windows Vista, Windows Server 2008, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview. This structure is used for **dmDriverExtraData** if the **dmDriverVersion** field of the DEVMODE structure is 0x0600.

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| dwSignature | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| wVer | | | | | | | | | | | | | | | | wCoreJTExpSize | | | | | | | | | | | | | | | |
| wCoreFullSize | | | | | | | | | | | | | | | | wOEMExtra | | | | | | | | | | | | | | | |
| dwChecksum32 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwFlags | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| bReversePrint | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| iLayout | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| iQuality | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| wReserved | | | | | | | | | | | | | | | | ... | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | ... | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | ... | | | | | | | | | | | | | | | |
| dwOptions | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| aOptions[0] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| (repeats for a total of 128 aOptions array elements) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| aOptions[127] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwNupDirection | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwNupBorderFlags | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwBookletFlags | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**wCoreJTExpSize**: The value of this field specifies the size of the JTEXP structure if one is present, following directly after this structure.

**wCoreFullSize**: The value of this field specifies the size of the **UNIDRVEXTRA** structure plus the value of **wCoreJTExpSize**.

**dwNupDirection**: This field is used only if N-Up printing is selected, and it must be one of the following values:

| Value | Meaning |
| --- | --- |
| 0x00000001 | Print N-Up pages left-to-right, top-to-bottom. |
| 0x00000002 | Print N-Up pages top-to-bottom, left-to-right. |
| 0x00000004 | Print N-Up pages right-to-left, top-to-bottom. |
| 0x00000008 | Print N-Up pages top-to-bottom, right-to-left. |

**dwNupBorderFlags**: This field is used only if N-Up Printing is selected, and it must be one of the following values:

| Value | Meaning |
| --- | --- |
| 0x00000000 | Print borders around N-Up pages. |
| 0x00000001 | Do not print borders around N-Up pages. |

**dwBookletFlags**: This field is used only if booklet printing is selected, and it must be one of the following values:

| Value | Meaning |
| --- | --- |
| 0x00000000 | Print booklet so that pages flip to the left (western style). |
| 0x00000001 | Print booklet so that pages flip to the right. |

Description of common members for **UNIDRVEXTRA** structures:

**dwSignature**: The value of this field must be 0x44494E55.

**dwFlags**: The value of this field must be the result of the bitwise OR of zero or more of the following values:

| Value | Meaning |
| --- | --- |
| 0x00000002 | Print text as graphics (do not use fonts). |
| 0x00000010 | Do not use EMFSPOOL spooling. |
| 0x00000080 | Use Custom Quality [**halftoning**](#gt_3934fcec-759d-45ac-8539-8cf2adeda698). |

**bReversePrint**: If the value of this field is nonzero, print pages in reverse order; otherwise, print pages in normal order.

**iLayout**: The value of this field must be one of the following values

| Value | Meaning |
| --- | --- |
| 0x00000000 | N-Up printing disabled. |
| 0x00000001 | Print 2-Up. |
| 0x00000002 | Print 4-Up. |
| 0x00000003 | Print 6-Up. |
| 0x00000004 | Print 9-Up. |
| 0x00000005 | Print 16-Up. |
| 0x00000006 | Print as a booklet. |

**iQuality**: The value of this field must be one of the following values:

| Value | Meaning |
| --- | --- |
| 0x00000000 | Best Quality |
| 0x00000001 | Medium Quality |
| 0x00000002 | Draft Quality |

**wReserved**: The value of this field SHOULD be set to zero when sent and MUST be ignored on receipt.

**wOEMExtra**: The value of this field specifies the total size of private data that is used by the vendor-supplied driver plug-in. This data immediately follows this structure.

**wVer**: The value of this field must be 0x0022.

**dwChecksum32**: The value of this field is the checksum of the names of the vendor-defined features and feature options that are supported by the printer model, as provided by the printer driver. The checksum is calculated using the 32-bit CRC function defined in section 3.2.9 of [IEEE802.3-2008]. For each feature, the checksum is calculated on the null-terminated ASCII string representations of the feature name and each of the feature options in order. The checksum is accumulated in this manner over all of the features supported by the printer model.

**dwOptions**: The value of this field is the number of entries in the **aOptions** array.

**aOptions**: This field is an array that is 512 bytes in length and contains the options selected by the user for each vendor-defined feature. Unused fields should be initialized to zero. The meaning of the entries differs for each supported printer model.

[<159> Section 2.2.2.1.3](#Appendix_A_Target_159): This is the **OEM\_DMEXTRA** structure, which contains the \_DEVMODE **dmDriverExtraData** defined by vendor-supplied driver plug-in modules.

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| dwSize | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwSignature | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwVersion | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Vendor-defined Data (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**dwSize**: Must be the total size of the vendor-defined data.

**dwSignature**: Must be a vendor-defined unique number.

**dwVersion**: Must be the version of the vendor-supplied plug-in.

**Vendor-defined Data (variable)**: A variable-length field that holds vendor-defined data.

[<160> Section 2.2.2.1.4](#Appendix_A_Target_160): This is the **JTEXP** structure, which contains the \_DEVMODE **dmDriverExtraData** defined by Windows to hold feature selection information.

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 20 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 30 | 1 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| dwSize | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwSignature | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| dwVersion | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| wJTHdrSize | | | | | | | | | | | | | | | | wCoreMFOSize | | | | | | | | | | | | | | | |
| ModelName (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| FeatureOptionPairs (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**dwSize**: Must be the total size of the **JTEXP** structure. This size must be large enough to accommodate any possible combination of options for the vendor-specified features stored in **FeatureOptionPairs**.

**dwSignature**: Must be 0x534D544A.

**dwVersion**: Must be zero.

**wJTHdrSize**: Must be set to 16.

**wCoreMFOSize**: Must be the combined size of **ModelName** and **FeatureOptionPairs**. This must be exactly 16 bytes less than **dwSize**.

**ModelName (variable)**: Must be a zero-terminated UTF-16LE encoded string specifying the name of the printer model.

**FeatureOptionPairs (variable)**: Must be a concatenation of an even number of zero-terminated ASCII strings, terminated by an additional zero character. Each pair of two consecutive strings specifies a vendor-defined feature and the currently selected option for that feature. Each printer driver provides its own list of features and possible option values for each feature.

[<161> Section 2.2.2.4.3](#Appendix_A_Target_161): This feature is supported on the following Windows versions:

* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<162> Section 2.2.2.4.4](#Appendix_A_Target_162): This feature is supported on the following Windows version:

* Windows 2000

[<163> Section 2.2.2.4.5](#Appendix_A_Target_163): This feature is supported on the following Windows versions:

* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<164> Section 2.2.2.4.6](#Appendix_A_Target_164): This feature is supported on the following Windows versions:

* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<165> Section 2.2.2.4.7](#Appendix_A_Target_165): This feature is supported on the following Windows versions:

* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<166> Section 2.2.2.4.7](#Appendix_A_Target_166): Windows uses INF files for installation configuration data. For more information, see [[MSDN-UINF]](http://go.microsoft.com/fwlink/?LinkId=90152) for more details.

[<167> Section 2.2.2.4.8](#Appendix_A_Target_167): This feature is supported on the following Windows versions:

* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<168> Section 2.2.2.4.10](#Appendix_A_Target_168): Windows print clients use this field to detect changes to printer driver files and to decide whether to update their local copies.

[<169> Section 2.2.2.5.2](#Appendix_A_Target_169): This feature is supported on the following Windows versions:

* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<170> Section 2.2.2.6.3](#Appendix_A_Target_170): This feature is supported on the following Windows versions:

* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<171> Section 2.2.2.6.4](#Appendix_A_Target_171): This feature is supported on the following Windows versions:

* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<172> Section 2.2.2.9.1](#Appendix_A_Target_172): This feature is supported on the following Windows versions:

* Windows NT 3.5
* Windows NT 3.51
* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<173> Section 2.2.2.9.5](#Appendix_A_Target_173): This feature is supported on the following Windows versions:

* Windows NT 3.5
* Windows NT 3.51
* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<174> Section 2.2.2.9.6](#Appendix_A_Target_174): This feature is supported on the following Windows versions:

* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<175> Section 2.2.2.9.7](#Appendix_A_Target_175): This feature is supported on the following Windows versions:

* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<176> Section 2.2.2.9.8](#Appendix_A_Target_176): This feature is supported on the following Windows version:

* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<177> Section 2.2.2.9.9](#Appendix_A_Target_177): This feature is supported on the following Windows version:

* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<178> Section 2.2.2.11](#Appendix_A_Target_178): This feature is supported on the following Windows version:

* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<179> Section 2.2.2.12](#Appendix_A_Target_179): The Windows print server obtains this checksum by calling the **GdiQueryFonts** API method.

[<180> Section 2.2.2.13](#Appendix_A_Target_180): This feature is supported on the following Windows versions:

* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<181> Section 2.2.2.13](#Appendix_A_Target_181): In Windows implementations, the driver version is matched to the version portion of the INF file **DriverVer** member. For information about INF file syntax, see [MSDN-UINF].

[<182> Section 2.2.2.14.4](#Appendix_A_Target_182): This feature is supported on the following Windows versions:

* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<183> Section 2.2.2.14.5](#Appendix_A_Target_183): This feature is supported on the following Windows versions:

* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<184> Section 2.2.2.15.3](#Appendix_A_Target_184): This feature is supported on the following Windows versions:

* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<185> Section 2.2.3.1](#Appendix_A_Target_185): This feature is supported on the following Windows version:

* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<186> Section 2.2.3.1](#Appendix_A_Target_186): This feature is supported on the following Windows versions:

* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<187> Section 2.2.3.1](#Appendix_A_Target_187): This feature is supported on the following Windows versions:

* Windows NT 3.5
* Windows NT 3.51
* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<188> Section 2.2.3.1](#Appendix_A_Target_188): This feature is supported on the following Windows versions:

* Windows NT 3.5
* Windows NT 3.51
* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<189> Section 2.2.3.1](#Appendix_A_Target_189): This feature is supported on the following Windows versions:

* Windows NT 3.5
* Windows NT 3.51
* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<190> Section 2.2.3.2](#Appendix_A_Target_190): These change notification flags are supported on the following Windows versions:

* Windows NT 3.51
* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<191> Section 2.2.3.3](#Appendix_A_Target_191): These job notification values are supported on the following Windows versions:

* Windows NT 3.51
* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<192> Section 2.2.3.4](#Appendix_A_Target_192): This feature is supported on the following Windows versions:

* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<193> Section 2.2.3.6.1](#Appendix_A_Target_193): This feature is supported on the following Windows versions:

* Windows NT 3.5
* Windows NT 3.51
* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<194> Section 2.2.3.6.1](#Appendix_A_Target_194): PRINTER\_CHANGE\_ALL (0x7777FFFF) is supported by the following Windows versions:

* Windows NT 3.51
* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows Vista
* Windows Server 2008
* Windows 7
* Windows Server 2008 R2

[<195> Section 2.2.3.6.1](#Appendix_A_Target_195): PRINTER\_CHANGE\_ALL\_2 (0x7F77FFFF) is supported by the following Windows versions:

* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<196> Section 2.2.3.6.2](#Appendix_A_Target_196): This feature is supported on the following Windows versions:

* Windows NT 3.5
* Windows NT 3.51
* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<197> Section 2.2.3.6.2](#Appendix_A_Target_197): This feature is supported on the following Windows versions:

* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<198> Section 2.2.3.7](#Appendix_A_Target_198): Windows displays this printer in its list of network-available printers.

[<199> Section 2.2.3.7](#Appendix_A_Target_199): Windows displays this printer in its list of network-available printers.

[<200> Section 2.2.3.7](#Appendix_A_Target_200): Windows displays this printer in its list of network-available printers.

[<201> Section 2.2.3.7](#Appendix_A_Target_201): Windows displays this printer in its list of network-available printers.

[<202> Section 2.2.3.7](#Appendix_A_Target_202): This feature is supported on the following Windows versions:

* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<203> Section 2.2.3.8](#Appendix_A_Target_203): 3D printers are not supported on the following Windows versions:

* Windows NT 3.1
* Windows NT 3.5
* Windows NT 3.51
* Windows NT 4.0
* Windows 98
* Windows 2000
* Windows Millennium Edition
* Windows XP
* Windows Server 2003
* Windows Vista
* Windows Server 2008
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012

[<204> Section 2.2.3.8](#Appendix_A_Target_204): This printer notification value is not supported on the following Windows versions:

* Windows NT 3.1
* Windows NT 3.5
* Windows NT 3.51
* Windows NT 4.0
* Windows 98
* Windows 2000
* Windows Millennium Edition
* Windows XP
* Windows Server 2003
* Windows Vista
* Windows Server 2008
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012

[<205> Section 2.2.3.8](#Appendix_A_Target_205): This printer notification value is not supported on the following Windows versions:

* Windows NT 3.1
* Windows NT 3.5
* Windows NT 3.51
* Windows NT 4.0
* Windows 98
* Windows 2000
* Windows Millennium Edition
* Windows XP
* Windows Server 2003
* Windows Vista
* Windows Server 2008
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012

[<206> Section 2.2.3.8](#Appendix_A_Target_206): Unless noted otherwise, these printer notification values are not supported on the following Windows versions:

* Windows NT 3.1
* Windows 98
* Windows Millennium Edition

[<207> Section 2.2.3.8](#Appendix_A_Target_207): This feature is not supported on the following Windows versions:

* Windows NT 3.1
* Windows NT 3.5
* Windows NT 3.51
* Windows NT 4.0
* Windows 98
* Windows 2000
* Windows Millennium Edition
* Windows XP
* Windows Server 2003
* Windows Vista
* Windows Server 2008
* Windows 7
* Windows Server 2008 R2

[<208> Section 2.2.3.8](#Appendix_A_Target_208): This feature is not supported on the following Windows versions:

* Windows NT 3.1
* Windows NT 3.5
* Windows NT 3.51
* Windows NT 4.0
* Windows 98
* Windows Millennium Edition

[<209> Section 2.2.3.10](#Appendix_A_Target_209): This feature is supported on the following Windows versions:

* Windows NT 3.51
* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003

[<210> Section 2.2.3.10](#Appendix_A_Target_210): This feature is supported on the following Windows versions:

* Windows 2000
* Windows Server 2003

[<211> Section 2.2.3.10](#Appendix_A_Target_211): Windows does not use this key name remotely.

[<212> Section 2.2.3.10](#Appendix_A_Target_212): Windows does not use this key name remotely.

[<213> Section 2.2.3.10](#Appendix_A_Target_213): Windows does not use this key name remotely.

[<214> Section 2.2.3.10](#Appendix_A_Target_214): Windows does not use this key name remotely.

[<215> Section 2.2.3.10](#Appendix_A_Target_215): This feature is supported on the following Windows versions:

* Windows NT 3.51
* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003

[<216> Section 2.2.3.10](#Appendix_A_Target_216): Windows does not use this key name remotely.

[<217> Section 2.2.3.10](#Appendix_A_Target_217): Windows does not use this key name remotely.

[<218> Section 2.2.3.10](#Appendix_A_Target_218): Windows does not use this key name remotely.

[<219> Section 2.2.3.10](#Appendix_A_Target_219): This feature is supported on the following Windows versions:

* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<220> Section 2.2.3.10](#Appendix_A_Target_220): This feature is supported on the following Windows versions:

* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<221> Section 2.2.3.10](#Appendix_A_Target_221): This feature is supported on the following Windows versions:

* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<222> Section 2.2.3.10](#Appendix_A_Target_222): This feature is supported on the following Windows versions:

* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<223> Section 2.2.3.10](#Appendix_A_Target_223): This feature is supported on the following Windows versions:

* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<224> Section 2.2.3.10](#Appendix_A_Target_224): This feature is supported on the following Windows versions:

* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<225> Section 2.2.3.10](#Appendix_A_Target_225): Windows by default does not configure this value, allowing drivers with a driver version (see *cVersion* in section 2.2.1.3.1) of 0x00000004 to run printer UI applications if available. An administrator can change this behavior by creating the "HKLM\Software\Policies\Microsoft\Windows NT\Printers\V4DriverDisallowPrinterUIApp" REG\_DWORD value in the registry and initializing this value with 0x00000001. This feature is supported on the following Windows versions:

* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<226> Section 2.2.3.10.1](#Appendix_A_Target_226): The dwBuildNumber value for OSVERSIONINFO and OSVERSIONINFOEX for specific versions of Windows is shown in the table that follows.

| Version | dwBuildNumber value |
| --- | --- |
| Windows 8.1 and Windows Server 2012 R2 | >= 9431 |
| Windows 8 and Windows Server 2012 | >= 9200 |
| Windows 7 and Windows Server 2008 R2 | >= 7007 |
| Windows Vista SP1 and Windows Server 2008 | >= 6001 |
| Windows Vista and Windows Server 2008 | >= 6000 |
| Windows XP SP1 | >= 2196 |
| Windows XP and Windows Server 2003 | >= 2196 |
| Windows 2000 | >= 1382 |
| Windows NT 4.0 | >= 1381 |

[<227> Section 2.2.3.10.3](#Appendix_A_Target_227): This value indicates Windows 2000 Professional operating system, Windows XP, Windows Vista, Windows 7, Windows 8, Windows 8.1, Windows 10, or Windows Server 2016 Technical Preview.

[<228> Section 2.2.3.10.3](#Appendix_A_Target_228): This value indicates Windows 2000 Server operating system, Windows Server 2003, Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, Windows 10, or Windows Server 2016 Technical Preview.

[<229> Section 2.2.3.10.3](#Appendix_A_Target_229): This value indicates Windows 2000 Server, Windows Server 2003, Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, Windows 10, or Windows Server 2016 Technical Preview.

[<230> Section 2.2.3.10.5](#Appendix_A_Target_230): Windows uses the following values, which can be combined to specify multiple product suites.

| Value | Meaning |
| --- | --- |
| VER\_SUITE\_SMALLBUSINESS  0x00000001 | Microsoft Small Business Server was once installed on the system, but it might have been upgraded to another version of Windows. |
| VER\_SUITE\_ENTERPRISE  0x00000002 | Windows NT Server 4.0 operating system, Enterprise Edition, Windows 2000 Advanced Server operating system, Windows Server 2003 Enterprise Edition operating system, or Windows Server 2008 Enterprise operating system is installed. |
| VER\_SUITE\_BACKOFFICE  0x00000004 | Microsoft BackOffice components are installed. |
| VER\_SUITE\_TERMINAL  0x00000010 | Terminal Services is installed. If **VER\_SUITE\_TERMINAL** is set but **VER\_SUITE\_SINGLEUSERTS** is not set, the system is running in [**application server mode**](#gt_b95e0f34-13ff-496c-8cad-bf2fe26a5b59). |
| VER\_SUITE\_SMALLBUSINESS\_RESTRICTED  0x00000020 | Microsoft Small Business Server is installed with the restrictive client license in force. |
| VER\_SUITE\_EMBEDDEDNT  0x00000040 | Windows XP Embedded is installed. |
| VER\_SUITE\_DATACENTER  0x00000080 | Windows 2000 Datacenter Server operating system, Windows Server 2003 Datacenter Edition operating system, or Windows Server 2008 Datacenter operating system is installed. |
| VER\_SUITE\_SINGLEUSERTS  0x00000100 | Remote Desktop is supported, but only one interactive session. This value is set unless the system is running in application server mode. |
| VER\_SUITE\_PERSONAL  0x00000200 | Windows XP Home Edition operating system, Windows Vista Home Basic, or Windows Vista Home Premium is installed. |
| VER\_SUITE\_BLADE  0x00000400 | Windows Server 2003 Web Edition operating system is installed. |
| VER\_SUITE\_STORAGE\_SERVER  0x00002000 | Windows Storage Server 2003 R2 or Windows Storage Server 2003 is installed. |
| VER\_SUITE\_COMPUTE\_SERVER  0x00004000 | Windows Server 2003 operating system Compute Cluster Edition is installed. |
| VER\_SUITE\_WH\_SERVER  0x00008000 | Windows Home Server is installed. |

[<231> Section 2.2.3.11](#Appendix_A_Target_231): This feature is supported on the following Windows versions:

* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<232> Section 2.2.3.11](#Appendix_A_Target_232): This feature is supported on the following Windows versions:

* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<233> Section 2.2.3.11](#Appendix_A_Target_233): This feature is supported on the following Windows versions:

* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<234> Section 2.2.3.11](#Appendix_A_Target_234): This feature is supported on the following Windows versions:

* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<235> Section 2.2.3.11](#Appendix_A_Target_235): Windows reports as the last driver identifier in the list the Microsoft Universal Sharing driver ({A9838643-5862-4F72-ACAF-F4CECE098759}). This value is supported on the following Windows versions:

* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<236> Section 2.2.3.11](#Appendix_A_Target_236): This value is supported on the following Windows versions:

* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<237> Section 2.2.3.11](#Appendix_A_Target_237): This value is supported on the following Windows versions:

* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<238> Section 2.2.3.11](#Appendix_A_Target_238): This value is supported on the following Windows versions:

* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<239> Section 2.2.3.11](#Appendix_A_Target_239): This feature is supported on the following Windows versions:

* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<240> Section 2.2.3.11](#Appendix_A_Target_240): This feature is supported on the following Windows versions:

* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<241> Section 2.2.3.11](#Appendix_A_Target_241): The default value on Windows implementations is 10 MB.

[<242> Section 2.2.3.12](#Appendix_A_Target_242): This feature is supported on the following Windows versions:

* Windows NT 3.5
* Windows NT 3.51
* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<243> Section 2.2.3.12](#Appendix_A_Target_243): This feature is supported on the following Windows versions:

* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<244> Section 2.2.3.12](#Appendix_A_Target_244): This feature is supported on the following Windows versions:

* Windows NT 3.5
* Windows NT 3.51
* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<245> Section 2.2.3.12](#Appendix_A_Target_245): This feature is supported on the following Windows versions:

* Windows NT 3.5
* Windows NT 3.51
* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<246> Section 2.2.3.12](#Appendix_A_Target_246): This feature is supported on the following Windows versions:

* Windows NT 3.5
* Windows NT 3.51
* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<247> Section 2.2.3.12](#Appendix_A_Target_247): This feature is supported on the following Windows versions:

* Windows NT 3.5
* Windows NT 3.51
* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<248> Section 2.2.3.12](#Appendix_A_Target_248): This feature is supported on the following Windows versions:

* Windows NT 3.5
* Windows NT 3.51
* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<249> Section 2.2.3.12](#Appendix_A_Target_249): This feature is supported on the following Windows versions:

* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<250> Section 2.2.3.12](#Appendix_A_Target_250): This feature is supported on the following Windows versions:

* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<251> Section 2.2.3.12](#Appendix_A_Target_251): This feature is supported on the following Windows versions:

* Windows NT 3.5
* Windows NT 3.51
* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<252> Section 2.2.3.12](#Appendix_A_Target_252): This feature is supported on the following Windows versions:

* Windows NT 3.5
* Windows NT 3.51
* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<253> Section 2.2.3.12](#Appendix_A_Target_253): This feature is supported on the following Windows versions:

* Windows NT 3.5
* Windows NT 3.51
* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<254> Section 2.2.3.12](#Appendix_A_Target_254): This feature is supported on the following Windows versions:

* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<255> Section 2.2.3.12](#Appendix_A_Target_255): This feature is supported on the following Windows versions:

* Windows NT 3.5
* Windows NT 3.51
* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<256> Section 2.2.4.2](#Appendix_A_Target_256): Windows supports the spool file data type formats EMFSPOOL, RAW format, XML Paper Specification (XPS), and custom data types defined by custom print processors. For more information on these formats, see [[MS-EMFSPOOL]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-EMFSPOOL%5d.pdf), [[MSDN-SPOOL]](http://go.microsoft.com/fwlink/?LinkId=90134), [[MSDN-XMLP]](http://go.microsoft.com/fwlink/?LinkId=90172), and [MSFT-XMLPAPER].

Windows standard print processors support the following case-insensitive data type name strings:

| Name | Description |
| --- | --- |
| "RAW" | RAW |
| "RAW [FF appended]" | RAW |
| "RAW [FF auto]" | RAW |
| "NT EMF 1.003" | EMFSPOOL |
| "NT EMF 1.006" | EMFSPOOL |
| "NT EMF 1.007" | EMFSPOOL |
| "NT EMF 1.008" | EMFSPOOL |
| "TEXT" | Plain text |
| "XPS\_PASS" | XPS passthrough (supported on Windows Vista, Windows Server 2008, Windows Vista SP1, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview). |
| "XPS2GDI" | XPS data to be converted into a new Graphics Device Interface (GDI) print job to send to the device (supported on Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview). |

[<257> Section 2.2.4.3](#Appendix_A_Target_257): Windows restricts driver name strings to 260 characters, including the terminating null character.

[<258> Section 2.2.4.4](#Appendix_A_Target_258): The environment name strings "Windows 4.0" and "Windows NT x86" are supported on the following Windows versions:

* Windows NT 3.5
* Windows NT 3.51
* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

The environment name string "Windows IA64" is supported on the following Windows versions:

* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

The environment name string "Windows x64" is supported on the following Windows versions:

* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

The environment name string "Windows ARM" is supported on the following Windows versions, and only on print servers with the "Windows ARM" operating system environment:

* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<259> Section 2.2.4.5](#Appendix_A_Target_259): Windows restricts form name strings to 260 characters, including the terminating null character.

[<260> Section 2.2.4.7](#Appendix_A_Target_260): Windows restricts key name strings to 260 characters, including the terminating null character.

[<261> Section 2.2.4.8](#Appendix_A_Target_261): Windows restricts monitor name strings to 260 characters, including the terminating null character.

[<262> Section 2.2.4.9](#Appendix_A_Target_262): IPv6 names are supported only on Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<263> Section 2.2.4.9](#Appendix_A_Target_263): Windows uses the following pattern for local files:

1. NAME = <any TEXT except "\">
2. DIRECTORY = "\" 1#NAME
3. FILENAME = "\" 1#NAME
4. PATH = [alpha ":"] #DIRECTORY FILENAME

[<264> Section 2.2.4.9](#Appendix_A_Target_264): Windows restricts path name strings to 519 characters, including the terminating null character.

[<265> Section 2.2.4.10](#Appendix_A_Target_265): Windows uses the following patterns for port names:

1. PARALLEL\_PORT = "LPT" DIGIT ":"
2. SERIAL\_PORT = "COM" DIGIT ":"
3. FILE\_PORT = "FILE:"
4. USB\_PORT = "USB" 1#DIGIT ":"
5. UNC\_PORT = SERVER\_NAME DIRECTORY FILENAME
6. LOCAL\_FILE\_PORT = PATH
7. PORT\_NAME = (PARALLEL\_PORT | SERIAL\_PORT | FILE\_PORT | USB\_PORT |
8. UNC\_PORT | LOCAL\_FILE\_PORT)

where:

* **SERVER\_NAME** is specified in section [2.2.4.16](#Section_9b3f813570224b72accbaefcc360c83b).
* **DIRECTORY** is specified in section [2.2.4.9](#Section_1d75f622d07c45ccbf1ea6191e408516).
* **PARALLEL\_PORT** is used for devices attached through a parallel port.
* **SERIAL\_PORT** is used for devices attached through a serial port.
* **FILE\_PORT** is used to send data to a file.
* **USB\_PORT** is used for devices attached through a [**universal serial bus (USB)**](#gt_2922ead8-cc31-4cea-a2aa-885ac2c9455d) port.
* **UNC\_PORT** is used for network printers attached directly through an IP address or a network address.

Windows supports the pooling of ports. When printing to a printer associated with a pool of ports, the first available port is picked by the print server. Port pooling allows representation of multiple identical physical printers as a single logical printer. Pooled port names are represented as a comma-separated list of port names, for example, "LPT1:,LPT2:".

Clients connecting to a Windows print server need to be prepared to handle pooled ports correctly; for example, they cannot rely on individual port names enumerated by the [RpcEnumPorts](#Section_4f018733e78e4088ac48f6e19ee00863) method to match the string pointed to by the **pPortName** member of a [PRINTER\_INFO (section 2.2.1.10)](#Section_125fb29b3c5141fca333b1e0e8794a91) structure.

[<266> Section 2.2.4.11](#Appendix_A_Target_266): Windows restricts print processor name strings to 260 characters, including the terminating NULL character.

[<267> Section 2.2.4.12](#Appendix_A_Target_267): Windows restricts print provider name strings to 260 characters, including the terminating null character.

[<268> Section 2.2.4.14](#Appendix_A_Target_268): Windows uses the following postfix string (PRINTER\_NAME\_POSTFIX) values: "LocalOnly", "LocalsplOnly", and "DrvConvert". "LocalOnly" means that the client asks the server to use only local printer settings for [RpcGetPrinterData](#Section_9c04a0f2adc443938df47c36a14183f5) and [RpcSetPrinterData](#Section_e0d3e41e097c459da7fbacaf1343ea27), and it specifies that the client is not interested in printing to this local printer but only in accessing the printer's local settings. "LocalsplOnly" and "DrvConvert" are treated the same way and mean that the client asks the server to open only the local printer with the respective name (PRINTER\_NAME\_PREFIX) if such local printer exists. These postfix strings are used on Windows during the upgrade of a printer driver and in other cases where it is preferred that a printer is accessed locally.

[<269> Section 2.2.4.14](#Appendix_A_Target_269): Windows restricts printer name strings to 539 characters (259 + 260 + 20), including all backslashes, other separators, and the terminating null character.

[<270> Section 2.2.4.16](#Appendix_A_Target_270): IPv6 names are supported only on Windows Vista, Windows Server 2008, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<271> Section 2.2.4.16](#Appendix_A_Target_271): Windows restricts server name strings to 259 characters, including the two leading backslash characters and trailing backslash character.

[<272> Section 2.2.4.18](#Appendix_A_Target_272): Windows restricts value name strings to 260 characters, including the terminating null character.

[<273> Section 2.3.1](#Appendix_A_Target_273): In Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview, print servers can publish printers to the Active Directory. In Windows NT 4.0 and Windows NT 3.51, print servers do not interact with the directory.

[<274> Section 2.3.3.1](#Appendix_A_Target_274): Windows print servers attempt to set the following attributes based on internal print spooler state:

* uNCName
* serverName
* shortServerName
* versionNumber
* printerName
* description
* driverName
* location
* portName
* printStartTime
* printEndTime
* printKeepPrintedJobs
* printSeparatorFile
* printShareName
* printSpooling
* priority
* url
* flags
* printStatus
* printAttributes
* driverVersion

Windows print servers attempt to set the following attributes based on the results of querying the printer driver configuration module:

* printBinNames
* printCollate
* printColor
* printDuplexSupported
* printStaplingSupported
* printMaxXExtent
* printMaxYExtent
* printMinXExtent
* printMinYExtent
* printMediaSupported
* printMediaReady
* printNumberUp
* printMemory
* printOrientationsSupported
* printMaxResolutionSupported
* printLanguage
* printRate
* printRateUnits
* printPagesPerMinute

All other attributes in the print schema can be set indirectly by print clients using [RpcSetPrinterDataEx (section 3.1.4.2.18)](#Section_58520b8b44ce46fc90d30362457098e0), as specified in section [2.3.3.1](#Section_3f4f0a52e417495aad460466a9d0ef9d).

[<275> Section 2.3.3.1](#Appendix_A_Target_275): Windows print servers will publish a value under one of the "DsSpooler", "DsDriver", and "DsUser" keys to the directory, if its name corresponds to an attribute in the schema for the print queue object class.

[<276> Section 2.3.3.2](#Appendix_A_Target_276): Windows print servers persist a string representing the GUID for the published object in the directory. This string corresponds to the **pszObjectGUID** member of the [PRINTER\_INFO\_7](#Section_91a6cb808df743d08a5d8f1d23f96de6) structure. Windows print servers then call the **IDL\_DRSCrackNames** method of the DRSR protocol, documented in [[MS-DRSR]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-DRSR%5d.pdf) section 4.1.4, on the domain controller to convert the GUID string into the fully qualified DN of the object.

[<277> Section 2.3.3.2](#Appendix_A_Target_277): When a Windows print server performs an LDAP update operation for a print queue, it always updates all of the LDAP attributes corresponding to the printer data values under one printer data key ("DsSpooler", "DsDriver", or "DsUser") where the server has stored attributes as described in section 2.3.3.1.

[<278> Section 2.3.3.3](#Appendix_A_Target_278): In Windows 2000, Windows XP, and Windows Server 2003, print clients search the GC for print queues in all naming contexts. In Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview, print clients only search the current naming context by default, but users can still choose to search other naming contexts.

[<279> Section 2.3.3.3](#Appendix_A_Target_279): In Windows 2000, Windows XP, and Windows Server 2003, print client negotiate encryption for LDAP requests. In Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview, print clients do not use encryption except when users choose to customize the printer search by specifying other NCs or more detailed filters.

[<280> Section 2.3.3.4](#Appendix_A_Target_280): By default, Windows servers never perform this periodic operation. An administrator can change this behavior by writing a time period, in minutes, to the "HKLM\Software\Policies\Microsoft\Windows NT\Printers\VerifyPublishedState" value (REG\_DWORD type) in the registry.

[<281> Section 2.3.3.4](#Appendix_A_Target_281): On Windows servers that are configured as [**writable**](#gt_5e5f9997-e6b4-4a73-aa22-984814b83a18) domain controllers (DC), the periodic search procedure is, by default, executed every 8 hours. This default period of 8 hours can be changed by writing the new time period, in minutes, to the "HKLM\Software\Policies\Microsoft\Windows NT\Printers\PruningInterval" value (REG\_DWORD type) in the registry. The Windows servers that are not configured as writable domain controllers do not execute this periodic search.

[<282> Section 3.1.1](#Appendix_A_Target_282): Windows implements port pooling. A printer object can manage references to multiple port objects. A physical print device is connected to each of the port objects, but the physical print devices are substantially the same. Windows transparently distributes incoming jobs to the multiple port objects to balance workload.

[<283> Section 3.1.1](#Appendix_A_Target_283): Windows creates the print job SECURITY\_DESCRIPTOR by inheriting the SECURITY\_DESCRIPTOR of the printer (which by default grants JOB\_ALL\_ACCESS to members of the Administrators group and to the creator/owner client) and by adding an access allowed entry that grants JOB\_READ access to the submitting client.

[<284> Section 3.1.1](#Appendix_A_Target_284): **Job Named Properties** are supported on the following Windows versions:

* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<285> Section 3.1.1](#Appendix_A_Target_285): In Windows implementations, the printer driver version is a DWORD set to one of the following values:

| Value | Meaning |
| --- | --- |
| 0x00000001 | Windows NT 3.51 user-mode printer drivers |
| 0x00000002 | Windows NT 4.0 user-mode printer drivers. |
| 0x00000003 | Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, and Windows Server 2008 R2 user-mode printer drivers. |
| 0x00000004 | Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview user-mode printer drivers. |

[<286> Section 3.1.1](#Appendix_A_Target_286): Windows print servers initialize the list to contain the TCPMON port monitor, which provides support for [**TCP**](#gt_b08d36f6-b5c6-4ce4-8d2d-6f2ab75ea4cb)/IP-connected printers.

[<287> Section 3.1.1](#Appendix_A_Target_287): Print providers are Windows–specific and not required by this protocol.

[<288> Section 3.1.1](#Appendix_A_Target_288): In addition to the **list of print providers** persisted in the registry, the **list of print providers** on Windows print servers also includes a print provider named "Windows NT Local Print Provider".

[<289> Section 3.1.1](#Appendix_A_Target_289): The only versions of Windows servers that maintain such a list are Windows XP and Windows Server 2003. Administrators can configure this list. In all other versions of Windows, this list does not exist.

[<290> Section 3.1.1](#Appendix_A_Target_290): Windows print servers that maintain a **list of warned printer drivers** persist the list in an INF file called printupg.inf.

[<291> Section 3.1.1](#Appendix_A_Target_291): The **Job Named Properties** are supported on the following Windows versions:

* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<292> Section 3.1.1](#Appendix_A_Target_292): Branch office print mode is supported on the following Windows versions:

* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<293> Section 3.1.1](#Appendix_A_Target_293): **Persistence of Abstract Data Model in the Registry**

Print system management tools such as **printmig**, **brm**, and other third-party applications read and write registry values that are used by the Windows print server to persist its abstract data model. Windows print servers also persist objects from the abstract data model and other settings in the registry, as described in the following table.

Note that the value of the registry key entry shown in the table below is wrapped to the following line at the backslash character for easier reading. The actual string value is one contiguous string.

For example: A value in the table such as

1. HKLM\
2. SYSTEM\
3. CurrentControlSet\
4. Control\
5. Print

represents a registry key path string that would be entered in a string variable as:

1. "HKLM\SYSTEM\CurrentControlSet\Control\Print"

| Registry key | Value | Type | Description |
| --- | --- | --- | --- |
| HKLM\  SYSTEM\  CurrentControlSet\  Control\  Print\ |  |  |  |
|  | MaxRPCSize | REG\_DWORD | Maximum buffer size accepted by the server in RPC calls. The default used, if this value is not present, is 50MB. |
| HKLM\  SYSTEM\  CurrentControlSet\  Control\  Print\  Environments\  [Env Name]\  Drivers |  |  | The **list of printer drivers** installed on the print server for the environment named [Env Name]. |
| HKLM\  SYSTEM\  CurrentControlSet\  Control\  Print\  Environments\  [Env Name]\  Drivers\  [Version]\  [Driver Name] |  |  | Information about the printer driver named [Driver Name] having the version identified by [Version] for the environment named [Env Name]. |
|  | Configuration File | REG\_SZ | Persisted DRIVER\_INFO member **pConfigFile**. |
|  | Data File | REG\_SZ | Persisted DRIVER\_INFO member **pDataFile**. |
|  | Driver | REG\_SZ | Persisted DRIVER\_INFO member **pDriverPath**. |
|  | Help File | REG\_SZ | Persisted DRIVER\_INFO member **pHelpFile**. |
|  | Monitor | REG\_SZ | Persisted DRIVER\_INFO member **pMonitorName**. |
|  | Datatype | REG\_SZ | Persisted DRIVER\_INFO member **pDefaultDataType**. |
|  | Dependent Files | REG\_MULTI\_SZ | Persisted DRIVER\_INFO member **pDependentFiles**. |
|  | Previous Names | REG\_MULTI\_SZ | Persisted DRIVER\_INFO member **pszzPreviousNames**. |
|  | Version | REG\_DWORD | Persisted DRIVER\_INFO member **cVersion**. |
|  | Manufacturer | REG\_SZ | Persisted DRIVER\_INFO member **pMfgName**. |
|  | OEM URL | REG\_SZ | Persisted DRIVER\_INFO member **pOEMUrl**. |
|  | HardwareID | REG\_SZ | Persisted DRIVER\_INFO member **pHardwareID**. |
|  | Provider | REG\_SZ | Persisted DRIVER\_INFO member **pProvider**. |
|  | DriverDate | REG\_BINARY | Persisted DRIVER\_INFO member **ftDriverDate**. |
|  | DriverVersion | REG\_BINARY | Persisted DRIVER\_INFO member **dwlDriverVersion**. |
| HKLM\  SYSTEM\  CurrentControlSet\  Control\  Print\  Environments\  [Env Name]\  Print Processors |  |  | The **list of print processors** installed on the print server for the environment named [Env Name]. |
| HKLM\  SYSTEM\  CurrentControlSet\  Control\  Print\  Environments\  [Env Name]\  Print Processors\  [Processor Name] |  |  | Information about the print processor named [Processor Name] for the environment named [Env Name]. |
|  | Driver | REG\_SZ | Persisted **pPathName** parameter used when print processor was added by the [RpcAddPrintProcessor (section 3.1.4.8.1)](#Section_d7e6fdb25256483592c127aa6f76758a) method. |
| HKLM\  SYSTEM\  CurrentControlSet\  Control\  Print\  Forms |  |  | The **list of form objects** installed on the print server. |
| HKLM\  SYSTEM\  CurrentControlSet\  Control\  Print\  Monitors |  |  | The **list of language monitors** and port monitors installed on the print server. |
| HKLM\  SYSTEM\  CurrentControlSet\  Control\  Print\  Monitors\  [Monitor Name] |  |  | Information about the port monitor named [Monitor Name]. |
|  | Driver | REG\_SZ | Persisted [MONITOR\_INFO\_2 (section 2.2.1.8.2)](#Section_f05b335b203c454f87da69ad96930bd7) member **pDLLName**. |
| HKLM\  SYSTEM\  CurrentControlSet\  Control\  Print\  Monitors\  LPR Port\  Ports |  |  | Information about LPR ports. |
| HKLM\  SYSTEM\  CurrentControlSet\  Control\  Print\  Monitors\  LPR Port\  Ports\  [Port name] |  |  | Information about the LPR port named [Port Name]. |
|  | Server Name | REG\_SZ | Host address of server exposing LPR port. |
|  | Printer Name | REG\_SZ | Queue name on LPR server. |
| HKLM\  SYSTEM\  CurrentControlSet\  Control\  Print\  Monitors\  Standard TCP/IP Port\  Ports |  |  |  |
| HKLM\  SOFTWARE\  DigitalEquipmentCorporation\  Network Printing Software | All subkeys and values |  | Opaque data for DEC Networking monitors. |
| HKLM\  SOFTWARE\  Hewlett-Packard\  HP JetAdmin | All subkeys and values |  | Opaque data for HP JetAdmin monitors. |
| HKLM\  SOFTWARE\  Lexmark | All subkeys and values |  | Opaque data for Lexmark Networking monitors. |
| HKLM\  SOFTWARE\  Microsoft\  Windows NT operating system\  CurrentVersion\  Ports |  |  | List of all ports installed with the print server. |
|  | [Port Name] | REG\_SZ | Implementation-specific initialization parameters for the port named [Port Name]. If there are no initialization parameters, the value specifies an empty string. |
| HKLM\  SOFTWARE\  Microsoft\  Windows NT\  CurrentVersion\  Print\  Connections |  |  | The **list of per-machine connections** on the print server. |
| HKLM\  SOFTWARE\  Microsoft\  Windows NT\  CurrentVersion\  Print\  Printers |  |  | The **persisted list of printers** installed on the print server. |
|  | DefaultSpoolDirectory | REG\_SZ | The directory used by the print server for storing temporary print job data. |
| HKLM\  SOFTWARE\  Microsoft\  Windows NT\  CurrentVersion\  Print\  Printers\  [Printer Name] |  |  | Information about the printer named [Printer Name]. |
|  | Share Name | REG\_SZ | Persisted [PRINTER\_INFO](#Section_c07c2bfde2c94d369259354a229d645d) member **pShareName**. |
|  | Print Processor | REG\_SZ | Persisted PRINTER\_INFO member **pPrintProcessor**. |
|  | Datatype | REG\_SZ | Persisted PRINTER\_INFO member **pDatatype**. |
|  | Parameters | REG\_SZ | Persisted PRINTER\_INFO member **pParameters**. |
|  | ObjectGUID |  | Persisted PRINTER\_INFO member **pszObjectGUID**. |
|  | Printer Driver | REG\_SZ | Persisted PRINTER\_INFO member **pDriverName**. |
|  | Default \_DEVMODE | REG\_SZ | Persisted [PRINTER\_INFO\_8](#Section_abe75b29b0924398a5af9d639f430cde) member **pDevMode**. |
|  | Priority | REG\_DWORD | Persisted PRINTER\_INFO member **Priority**. |
|  | Default Priority | REG\_DWORD | Persisted PRINTER\_INFO member **DefaultPriority**. |
|  | StartTime | REG\_DWORD | Persisted PRINTER\_INFO member **StartTime**. |
|  | UntilTime | REG\_DWORD | Persisted PRINTER\_INFO member **UntilTime**. |
|  | Separator File | REG\_SZ | Persisted PRINTER\_INFO member **pSepFile**. |
|  | Location | REG\_SZ | Persisted PRINTER\_INFO member **pLocation**. |
|  | Attributes | REG\_DWORD | Persisted PRINTER\_INFO member **Attributes**. |
|  | Security | REG\_BINARY | Persisted PRINTER\_INFO member **pSecurityDescriptor**. |
|  | Port | REG\_SZ | Persisted PRINTER\_INFO member **pPortName**. |
| HKLM\  SOFTWARE\  Microsoft\  Windows NT\  CurrentVersion\  Print\  Providers |  |  | The **list of print providers** installed on the print server. |

Definitions for placeholders in the preceding table:

* [Env Name] is an environment name as specified in [2.2.4.4](#Section_ad7758938a194786ab4634731f52d1af).
* [Version] is "Version-0", "Version-1", "Version-2", "Version-3", or "Version 4". The numbers 0–4 (0x00000000 – 0x00000004) are described in section 2.2.1.3.1, member cVersion.
* [Driver Name] is a driver name as specified in section [2.2.4.3](#Section_a4c7686fa88c48f996ebbf4dfc3fe519).
* [Processor Name] is a print processor name as specified in section [2.2.4.11](#Section_807e0bdd21a54a729a6a1a7fb1b08a8e).
* [Monitor Name] is a monitor name as specified in section [2.2.4.8](#Section_808e2e6c7f914bdd92da56fbca513404).
* [Port Name] is a port name as specified in section [2.2.4.10](#Section_03c5feb74fbb4569b0649eb6ad833f50).
* [Printer Name] is a printer name as specified in section [2.2.4.14](#Section_24fcd124035c4988a8583a7d8d6f7b43).

[<294> Section 3.1.1](#Appendix_A_Target_294): **Locations of Print System Components in the Server File System**

Print system management tools such as **printmig**, **brm**, and other, third-party applications remotely access files that are loaded or executed by the Windows print server. Windows print server loads or executes files as described in the following table.

| File description | Location on print server file system |
| --- | --- |
| Printer driver files | %systemroot%\system32\spool\drivers\[env-dir]\[version-dir] |
| Print processor files | %systemroot%\system32\spool\prtprocs\[env-dir] |
| Separator files | %systemroot%\system32\spool\sepfiles |

Definitions for placeholders in the preceding table:

[env-dir] specifies one of the following strings, depending on the [Env Name] of the driver:

* "W32X86" for the environment string "Windows NT x86"
* "IA64" for the environment string "Windows IA64"
* "WIN40" for the environment string "Windows 4.0"
* "W32ALPHA" for the environment string "Windows NT Alpha\_AXP"
* "X64" for the environment string "Windows X64"
* "ARM" for the environment string "Windows ARM"

[version-dir] is one of the strings "0", "1", "2", "3", or "4", matching the DRIVER\_INFO member **cVersion**.

The %systemroot% environment variable is shared by the print server using the share name "admin$".

[<295> Section 3.1.3](#Appendix_A_Target_295): In Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview, print servers can publish printers to the Active Directory, and print clients can search the Active Directory for printers. In Windows NT 3.51 and Windows NT 4.0, print clients and servers do not interact with the directory.

[<296> Section 3.1.4](#Appendix_A_Target_296): The job named property management methods are supported on the following Windows versions:

* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<297> Section 3.1.4](#Appendix_A_Target_297): Branch office print remote logging methods are supported on the following Windows versions:

* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<298> Section 3.1.4](#Appendix_A_Target_298):

| Opnum | Description |
| --- | --- |
| 37 | Only used locally by Windows, never remotely. |
| 38 | Only used locally by Windows, never remotely. |
| 43 | Only used locally by Windows, never remotely. |
| 44 | Only used locally by Windows, never remotely. |
| 45 | Only used locally by Windows, never remotely. |
| 49 | Only used locally by Windows, never remotely. |
| 50 | Only used locally by Windows, never remotely. |
| 54 | Only used locally by Windows, never remotely. |
| 55 | Only used locally by Windows, never remotely. |
| 57 | Only used locally by Windows, never remotely. |
| 63 | Only used locally by Windows, never remotely. |
| 64 | Only used locally by Windows, never remotely. |
| 68 | Only used locally by Windows, never remotely. |
| 74 | Only used locally by Windows, never remotely. |
| 75 | Only used locally by Windows, never remotely. |
| 76 | Only used locally by Windows, never remotely. |
| 83 | Only used locally by Windows, never remotely. |
| 90 | Only used locally by Windows, never remotely. |
| 91 | Only used locally by Windows, never remotely. |
| 92 | Only used locally by Windows, never remotely. |
| 93 | Only used locally by Windows, never remotely. |
| 94 | Only used locally by Windows, never remotely. |
| 95 | Only used locally by Windows, never remotely. |
| 98 | Only used locally by Windows, never remotely. |
| 99 | Only used locally by Windows, never remotely. |
| 100 | Only used locally by Windows, never remotely. |
| 101 | Only used locally by Windows, never remotely. |
| 102 | Only used locally by Windows, never remotely. |
| 103 | Only used locally by Windows, never remotely. |
| 104 | Only used locally by Windows, never remotely. |
| 105 | Only used locally by Windows, never remotely. |
| 106 | Only used locally by Windows, never remotely. |
| 107 | Only used locally by Windows, never remotely. |
| 108 | Only used locally by Windows, never remotely. |
| 109 | Only used locally by Windows, never remotely. |
| 114 | Only used locally by Windows, never remotely. |
| 115 | Only used locally by Windows, never remotely. |

[<299> Section 3.1.4.1.4](#Appendix_A_Target_299): IPv6 names are supported only on Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

[<300> Section 3.1.4.1.4](#Appendix_A_Target_300): All Windows client implementations derive the RPC binding directly from the [STRING\_HANDLE](#Section_59883eb85ce64ad4bdd2599434cfdcf2) binding parameter, and all Windows server implementations perform this validation step.

[<301> Section 3.1.4.1.5](#Appendix_A_Target_301): All Windows client implementations derive the RPC binding directly from the STRING\_HANDLE binding parameter, and all Windows server implementations perform this validation step.

[<302> Section 3.1.4.1.8.1](#Appendix_A_Target_302): In Windows, when the *pDevModeContainer* parameter is not declared with the "unique" IDL attribute and is set to a value of NULL, the underlying RPC protocol [[MS-RPCE]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-RPCE%5d.pdf) implementation stops the invalid call and throws an exception before the call reaches the server.

[<303> Section 3.1.4.1.8.2](#Appendix_A_Target_303): In Windows, when the *pDocInfoContainer* parameter is not declared with the "unique" IDL attribute and is set to a value of NULL, the underlying RPC protocol [MS-RPCE] implementation stops the invalid call and throws an exception before the call reaches the server.

[<304> Section 3.1.4.1.8.7](#Appendix_A_Target_304): If the value of the **pSecurity** member in the [SECURITY\_CONTAINER](#Section_6cf2b4ff1b2b453c87f985243823cea8) structure is NULL, a default security descriptor is used. Security descriptors are specified in [[MS-DTYP]](file:///E:\Target\Windows\Published\Books\MS-RPRN\%5bMS-DTYP%5d.pdf).

Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview:

The owner in the security descriptor is the local system. The discretionary access control list (DACL) contains [**access control entries (ACE)**](#gt_b581857f-39aa-4979-876b-daba67a40f15), which grant the following permissions:

Full control to the administrator's group and to the user who added the printer.

Print permissions to "everyone".

Permissions to control (cancel, pause, resume) the job to the user who submits the job.

Windows NT, Windows 2000, Windows XP, and Windows Server 2003:

The owner in the security descriptor is the user who added the printer. The DACL contains ACEs that grant the following permissions:

Full control to administrators and power users groups.

Print permissions to "everyone".

Permissions to control (cancel, pause, resume) the job to the user who submits the job.

[<305> Section 3.1.4.1.8.8](#Appendix_A_Target_305): Windows does not use the following members: **pUserName**, **dwBuildNum**, **dwMajorVersion**, **dwMinorVersion**, and **wProcessorArchitecture**. **pMachineName** is used only if the server cannot determine the client machine name using remote procedure call (RPC) functions. The **pMachineName** member can be NULL.

[<306> Section 3.1.4.1.8.9](#Appendix_A_Target_306): This validation step can be performed on Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview.

An administrator can configure a print server to not perform this validation step by writing a nonzero value to the "HKLM\System\CurrentControlSet\Control\Print\ARMTestMode" value (REG\_DWORD type) in the registry.

[<307> Section 3.1.4.1.10](#Appendix_A_Target_307): This verification is done by the following Windows versions:

* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<308> Section 3.1.4.2.1](#Appendix_A_Target_308): Windows Vista, Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview servers can return 0x00004001 for error conditions or if no printers are found matching the requested flags.

[<309> Section 3.1.4.2.1](#Appendix_A_Target_309): Windows Vista, Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview servers can return 0x00004001 for error conditions or if no printers are found matching the requested flags.

[<310> Section 3.1.4.2.1](#Appendix_A_Target_310): The Windows server checks that the client user has the [SERVER\_ACCESS\_ENUMERATE](#Section_945c4df5496949c5b1ce59c1b3f35024) permission.

[<311> Section 3.1.4.2.1](#Appendix_A_Target_311): Windows returns the printers in alphabetical order.

[<312> Section 3.1.4.2.1](#Appendix_A_Target_312): Windows Vista, Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview servers can return 0x00004001 for error conditions or if no printers are found matching the requested flags.

[<313> Section 3.1.4.2.1](#Appendix_A_Target_313): Windows uses a policy-defined period (default: 60 minutes) from the first call to [RpcAddPrinter](#Section_bd0bb8d526fd470581f5614d32b9f38f) with *Level* set to 0x00000001 to determine whether ERROR\_CAN\_NOT\_COMPLETE is returned.

[<314> Section 3.1.4.2.1](#Appendix_A_Target_314): In Windows, the enumeration is restricted as follows:

* Unshared printers are enumerated only if the user has the SERVER\_ACCESS\_ADMINISTER permission.
* Only those printers whose security descriptor grants PRINTER\_ACCESS\_USE to the caller are enumerated.

[<315> Section 3.1.4.2.3](#Appendix_A_Target_315): The Windows server checks that the client user has implementation-specific permission, such as SERVER\_ACCESS\_ADMINISTER, to install a printer.

[<316> Section 3.1.4.2.3](#Appendix_A_Target_316): Windows stores the time when the printer is added to the list. When a printer is added, Windows removes any printer on the list that was added more than 70 minutes ago. Windows stores a maximum of 256 printers in the list. If the limit is reached, no new printers are added, and ERROR\_OUTOFMEMORY, as specified in [MS-ERREF], is returned.

[<317> Section 3.1.4.2.4](#Appendix_A_Target_317): In Windows, the server verifies that the printer object has been opened with an access value including the generic **DELETE** permission, for example, PRINTER\_ALL\_ACCESS.

[<318> Section 3.1.4.2.4](#Appendix_A_Target_318): The Windows server verifies that the client user has the PRINTER\_ACCESS\_ADMINISTER permission.

[<319> Section 3.1.4.2.4](#Appendix_A_Target_319): This feature is supported on the following Windows versions:

* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<320> Section 3.1.4.2.5](#Appendix_A_Target_320): The Windows server verifies for a printer object that the object has been opened with PRINTER\_ACCESS\_ADMINISTER permission, and for a server object that the object has been opened with SERVER\_ACCESS\_ADMINISTER permission.

[<321> Section 3.1.4.2.5](#Appendix_A_Target_321): In Windows 2000, Windows Millennium Edition, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview, the Windows server can return the ERROR\_IO\_PENDING error code from the [RpcSetPrinter](#Section_382997b5b02a4b6a8c6849a64b8c45f8) call when the **Level** member of the **PRINTER\_CONTAINER** pointed to by *pPrinterContainer* is 0x7 (indicating a directory service operation) and the server implementation uses an asynchronous mechanism to perform the operation.

[<322> Section 3.1.4.2.5](#Appendix_A_Target_322): This feature is supported on the following Windows versions:

* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<323> Section 3.1.4.2.6](#Appendix_A_Target_323): In the following Windows versions, this parameter is a handle to a printer or print server object if the value of the *Level* parameter is 0x00000003.

* Windows Vista
* Windows Server 2008
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<324> Section 3.1.4.2.6](#Appendix_A_Target_324): The Windows server verifies for a printer object that the object has been opened with PRINTER\_ACCESS\_USE permission, and for a server object that the object has been opened with SERVER\_ACCESS\_ENUMERATE permission.

[<325> Section 3.1.4.2.7](#Appendix_A_Target_325): In Windows, the detection of the string "ChangeID" is case-insensitive.

[<326> Section 3.1.4.2.7](#Appendix_A_Target_326): In Windows, the data identified by *pValueName* for printer objects is stored in the registry under the key "PrinterDriverData"; therefore, [RpcGetPrinterDataEx](#Section_74a9f79910bf4ef6a2a40dcd9c43fb5c) is used with *pKeyName* pointing to the string "PrinterDriverData" to access the identical set of values.

[<327> Section 3.1.4.2.8](#Appendix_A_Target_327): In Windows, the detection of the string "ChangeID" is case-insensitive.

[<328> Section 3.1.4.2.8](#Appendix_A_Target_328): In Windows, if *hPrinter* is a server object handle, the server checks that the client user has SERVER\_ACCESS\_ADMINISTER permission.

If *hPrinter* is a printer object handle, the server checks that the client user has PRINTER\_ACCESS\_ADMINISTER permission. In Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview, the server checks that the client user has either PRINTER\_ACCESS\_ADMINISTER or PRINTER\_ACCESS\_MANAGE LIMITED permission.

[<329> Section 3.1.4.2.8](#Appendix_A_Target_329): In Windows the data identified by *pValueName* for printer objects is stored in the registry under the key "PrinterDriverData"; therefore, RpcSetPrinterDataEx is used with *pKeyName* pointing to the string "PrinterDriverData" to access the identical set of values.

For server objects, there is no aliasing of value and key because the server ignores the *pKeyName* parameter of the RpcSetPrinterDataEx method when called with a handle to a server object.

[<330> Section 3.1.4.2.11](#Appendix_A_Target_330): In Windows, the underlying RPC protocol [MS-RPCE] implementation stops the invalid call and throws an exception before the call reaches the server.

[<331> Section 3.1.4.2.11](#Appendix_A_Target_331): In Windows, the underlying RPC protocol [MS-RPCE] implementation stops the invalid call and throws an exception before the call reaches the server.

[<332> Section 3.1.4.2.11](#Appendix_A_Target_332): When creating print jobs in EMFSPOOL format [MS-EMFSPOOL], Windows print clients use the [UNIVERSAL\_FONT\_ID](#Section_97cd8dd84c4b4a7aa160fae63f3e59bf) structures returned by [RpcPlayGdiScriptOnPrinterIC](#Section_eff167adf4004c6eb0e2c48b62ac0826) to determine if the server has copies of specific fonts, and if so, the client does not embed those fonts in the EMFSPOOL data sent to the print server.

[<333> Section 3.1.4.2.14](#Appendix_A_Target_333): This feature is supported on the following Windows versions:

* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<334> Section 3.1.4.2.15](#Appendix_A_Target_334): This feature is supported on the following Windows versions:

* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<335> Section 3.1.4.2.15](#Appendix_A_Target_335): In Windows, the server checks that the client user has implementation-specific permissions to install a printer, typically SERVER\_ACCESS\_ADMINISTER.

[<336> Section 3.1.4.2.15](#Appendix_A_Target_336): Windows stores the time when the printer is added to the list. When a printer is added, Windows removes any printer on the list that was added more than a policy-defined period (default 70 minutes) ago. Windows stores a maximum of 256 printers in the list. If the limit is reached, no new printers are added, and ERROR\_OUTOFMEMORY is returned, as specified in [MS-ERREF].

[<337> Section 3.1.4.2.15](#Appendix_A_Target_337): Windows fails this call if, at the time of this call, the server does not have installed all the following: the printer driver, the port, and the print processor.

[<338> Section 3.1.4.2.16](#Appendix_A_Target_338): This feature is supported on the following Windows versions:

* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<339> Section 3.1.4.2.16](#Appendix_A_Target_339): In Windows, for printer objects, the data identified by *pValueName* is stored in the registry under the key named "PrinterDriverData"; therefore, [RpcEnumPrinterDataEx](#Section_9574b3e74a294bfaba67c35dab674162) is used with *pKeyName* pointing to the string "PrinterDriverData" to access the identical set of values.

[<340> Section 3.1.4.2.17](#Appendix_A_Target_340): This feature is supported on the following Windows versions:

* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<341> Section 3.1.4.2.17](#Appendix_A_Target_341): In Windows, the detection of the string "ChangeID" is case-insensitive.

[<342> Section 3.1.4.2.17](#Appendix_A_Target_342): In Windows, the server checks that the client user has PRINTER\_ACCESS\_ADMINISTER permission.

[<343> Section 3.1.4.2.17](#Appendix_A_Target_343): In Windows, the data identified by *pValueName* for printer objects is stored in the registry under the key named "PrinterDriverData"; therefore, [RpcDeletePrinterDataEx](#Section_19c2e0e5b7ec4d068b211488f2275505) is used with *pKeyName* pointing to the string "PrinterDriverData" to access the identical set of values.

[<344> Section 3.1.4.2.18](#Appendix_A_Target_344): This feature is supported on the following Windows version:

* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<345> Section 3.1.4.2.18](#Appendix_A_Target_345): In Windows, the detection of the string "ChangeID" is case-insensitive.

[<346> Section 3.1.4.2.18](#Appendix_A_Target_346): In Windows, if *hPrinter* is a server object handle, the server checks that the client user has SERVER\_ACCESS\_ADMINISTER permission. If *hPrinter* is a printer object handle, the server checks that the client user has PRINTER\_ACCESS\_ADMINISTER permission.

[<347> Section 3.1.4.2.19](#Appendix_A_Target_347): This feature is supported on the following Windows version:

* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<348> Section 3.1.4.2.19](#Appendix_A_Target_348): This verification is done by the following Windows versions:

* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<349> Section 3.1.4.2.19](#Appendix_A_Target_349): This verification is done by the following Windows versions:

* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<350> Section 3.1.4.2.20](#Appendix_A_Target_350): This feature is supported on the following Windows version:

* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<351> Section 3.1.4.2.21](#Appendix_A_Target_351): This feature is supported on the following Windows version:

* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<352> Section 3.1.4.2.22](#Appendix_A_Target_352): This feature is supported on the following Windows version:

* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<353> Section 3.1.4.2.22](#Appendix_A_Target_353): In Windows, the detection of the string "ChangeID" is case-insensitive.

[<354> Section 3.1.4.2.22](#Appendix_A_Target_354): In Windows, the server checks that the client user has the PRINTER\_ACCESS\_ADMINISTER permission.

[<355> Section 3.1.4.2.23](#Appendix_A_Target_355): This feature is supported on the following Windows version:

* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<356> Section 3.1.4.2.23](#Appendix_A_Target_356): In Windows, the server checks that the client user has the PRINTER\_ACCESS\_ADMINISTER permission.

[<357> Section 3.1.4.2.24](#Appendix_A_Target_357): This feature is supported on the following Windows version:

* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<358> Section 3.1.4.2.24](#Appendix_A_Target_358): In Windows, the name of the print provider file is used and is stored in the Windows registry. If *pProvider* is NULL, the Windows operating system uses Win32spl.dll as the name of the executable object.

Print providers are a Windows implementation detail and are not required by this protocol. Windows print clients do not use a non-NULL *pProvider* parameter remotely, but third-party software can do so. As there is no protocol method to enumerate print providers remotely, a client would need specific knowledge about the internal implementation of the server to specify a meaningful print provider name.

[<359> Section 3.1.4.2.24](#Appendix_A_Target_359): In Windows, the server checks that the client user has the SERVER\_ACCESS\_ADMINISTER permission.

[<360> Section 3.1.4.2.25](#Appendix_A_Target_360): This feature is supported on the following Windows version:

* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<361> Section 3.1.4.2.25](#Appendix_A_Target_361): The Windows server checks that the client user has SERVER\_ACCESS\_ADMINISTER permission.

[<362> Section 3.1.4.2.26](#Appendix_A_Target_362): This feature is supported on the following Windows version:

* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<363> Section 3.1.4.2.27](#Appendix_A_Target_363): This feature is supported on the following Windows versions:

* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<364> Section 3.1.4.2.27](#Appendix_A_Target_364): Port monitors shipping with Windows support the following actions only for [TCPMON](#Section_4f6172c708314ca6a37472c841f27c7c) and [WSDMON](#Section_e1760a13634e4a7fa7dd9d02e4048f9b), and only if connected to a printer capable of bidirectional communication: BIDI\_ACTION\_ENUM\_SCHEMA, BIDI\_ACTION\_GET, and BIDI\_ACTION\_GET\_ALL. All other Port Monitors shipping with Windows do not support [RpcSendRecvBidiData](#Section_5aec83472ebf464eb6d1540aa60b5c71).

[<365> Section 3.1.4.2.27](#Appendix_A_Target_365): Port monitors shipping with Windows 10 and Windows Server 2016 Technical Preview support the BIDI\_ACTION\_GET\_WITH\_ARGUMENT action for WSDMON if connected to a printer capable of bidirectional communication.

[<366> Section 3.1.4.2.27](#Appendix_A_Target_366): The Windows system port monitors do not support any additional action strings.

[<367> Section 3.1.4.2.27](#Appendix_A_Target_367): TCPMON and WSDMON check that the port or printer object has been opened with an access value that includes the PRINTER\_ACCESS\_USE bit.

[<368> Section 3.1.4.3.1](#Appendix_A_Target_368): In Windows, the command codes JOB\_CONTROL\_CANCEL (0x00000003) and JOB\_CONTROL\_DELETE (0x00000005) are equivalent. The redundancy is caused by historical reasons, but both must be supported.

[<369> Section 3.1.4.3.1](#Appendix_A_Target_369): In Windows, the command codes JOB\_CONTROL\_CANCEL (0x00000003) and JOB\_CONTROL\_DELETE (0x00000005) are equivalent. The redundancy is caused by historical reasons, but both must be supported.

[<370> Section 3.1.4.3.1](#Appendix_A_Target_370): For JOB\_CONTROL\_CANCEL and JOB\_CONTROL\_DELETE requests, the server verifies that the client has generic DELETE permission. For all other requests, the server verifies that the client has JOB\_ACCESS\_ADMINISTER permission for the job. For a job position change request, the server additionally verifies that the handle is a printer or server object and has been opened with PRINTER\_ACCESS\_ADMINISTER requested.

[<371> Section 3.1.4.4.1](#Appendix_A_Target_371): The Windows server checks that the client user has SERVER\_ACCESS\_ADMINISTER permission.

[<372> Section 3.1.4.4.1](#Appendix_A_Target_372): These validation steps are performed on the following Windows versions:

* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<373> Section 3.1.4.4.1](#Appendix_A_Target_373): These validation steps are performed on the following Windows versions:

* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<374> Section 3.1.4.4.1](#Appendix_A_Target_374): When a print client adds a printer driver to a Windows print server using [RpcAddPrinterDriver](#Section_f23a75191c7740699acea6d8eae47c22), the print server sets the Boolean values representing each of the printer driver's attributes to a Boolean value of FALSE, indicating that the printer driver does not have any of these attributes.

[<375> Section 3.1.4.4.2](#Appendix_A_Target_375): In Windows, if the string contains "All" or "AllCluster", all printer drivers for all environments must be enumerated.

[<376> Section 3.1.4.4.2](#Appendix_A_Target_376): The Windows server checks that the client user has SERVER\_ACCESS\_ENUMERATE permission.

[<377> Section 3.1.4.4.2](#Appendix_A_Target_377): This validation step is performed on the following Windows versions:

* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<378> Section 3.1.4.4.2](#Appendix_A_Target_378): This validation step is performed on the following Windows versions:

* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<379> Section 3.1.4.4.3](#Appendix_A_Target_379): Windows servers complete the request only if the specified environment is "Windows 4.0" and **pszzPreviousNames** contains the name of a printer driver that is installed on the server for the "Windows 4.0" environment; otherwise, the server returns ERROR\_UNKNOWN\_PRINTER\_DRIVER. For other environments, the clients can use [RpcGetPrinterDriver2](#Section_a73e77d07b2448cfa6d38b4061749495) when the server supports this method.

[<380> Section 3.1.4.4.4](#Appendix_A_Target_380): The Windows server checks that the client user has SERVER\_ACCESS\_ENUMERATE permission.

[<381> Section 3.1.4.4.5](#Appendix_A_Target_381): In Windows, the server checks that the client user has the SERVER\_ACCESS\_ADMINISTER permission.

[<382> Section 3.1.4.4.6](#Appendix_A_Target_382): This feature is supported on the following Windows versions:

* Windows NT 3.5
* Windows NT 3.51
* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<383> Section 3.1.4.4.6](#Appendix_A_Target_383): The following table shows the unsigned 32-bit major operating system version number that is used by Windows clients and servers.

| Major version | Meaning |
| --- | --- |
| 0x00000004 | The operating system is Windows 95, Windows NT 4.0, Windows 98, or Windows Millennium Edition. |
| 0x00000005 | The operating system is Windows 2000, Windows XP, Windows Server 2003, or Windows Server 2003 R2 operating system. |
| 0x00000006 | The operating system is Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, or Windows Server 2016 Technical Preview. |

[<384> Section 3.1.4.4.6](#Appendix_A_Target_384): The following table shows the unsigned 32-bit minor operating system version number that is used by Windows Server 2008 clients and servers.

| Minor version | Meaning |
| --- | --- |
| 0x00000000 | The operating system is Windows 95, Windows NT 4.0, Windows 2000, Windows Vista, or Windows Server 2008. |
| 0x00000001 | The operating system is Windows XP, Windows 7, or Windows Server 2008 R2. |
| 0x00000002 | The operating system is Windows XP Professional x64 Edition, Windows Server 2003, Windows Server 2003 R2, Windows 8, or Windows Server 2012. |
| 0x00000003 | The operating system is Windows 8.1, Windows Server 2012 R2, Windows 10, or Windows Server 2016 Technical Preview. |
| 0x0000000A | The operating system is Windows 98. |
| 0x0000005A | The operating system is Windows Millennium Edition. |

[<385> Section 3.1.4.4.6](#Appendix_A_Target_385): *pdwServerMaxVersion* and *pdwServerMinVersion* are ignored by the Windows print server, and therefore no values are returned. However, the caller is expected to submit valid pointer values for *pdwServerMaxVersion* and *pdwServerMinVersion* in order to ensure that the call can be correctly received by the server.

**Note**  If both *dwClientMajorVersion* and *dwClientMinorVersion* are set to 0xFFFFFFFF, the print server will return printer driver information for the printer driver version matching the operating system version on which the print server is running.

[<386> Section 3.1.4.4.6](#Appendix_A_Target_386): This validation step is performed on the following Windows versions:

* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<387> Section 3.1.4.4.7](#Appendix_A_Target_387): This feature is supported on the following Windows version:

* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<388> Section 3.1.4.4.7](#Appendix_A_Target_388): Windows sets this parameter to one of the following values.

| Value | Meaning |
| --- | --- |
| 0x00000001 | Windows NT 3.51 user-mode printer drivers |
| 0x00000002 | Windows NT 4.0 user-mode printer drivers |
| 0x00000003 | Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, and Windows Server 2008 R2 user-mode printer drivers. |
| 0x00000004 | Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10, and Windows Server 2016 Technical Preview user-mode printer drivers. |

[<389> Section 3.1.4.4.7](#Appendix_A_Target_389): In Windows, the server checks that the client user has SERVER\_ACCESS\_ADMINISTER (section 2.2.3.1) permission.

[<390> Section 3.1.4.4.8](#Appendix_A_Target_390): This feature is supported on the following Windows version:

* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<391> Section 3.1.4.4.8](#Appendix_A_Target_391): Only in Windows XP and Windows Server 2003, as described in [3.1.1](#Section_1bd61448db4b4284a4b18346732402d4), the Windows server maintains a warned list of printer drivers. Drivers in this list can only be added to the server if the **APD\_INSTALL\_WARNED\_DRIVER** bit is set. If a client attempts to add a printer driver that is in this list and the bit is not set, the server returns the ERROR\_PRINTER\_DRIVER\_WARNED error code. If a client attempts to add a printer driver that is in this list and the bit is set, the server attempts to add the printer driver as described in [3.1.4.4.8](#Section_b96cc49759e54510ab045484993b259b). On all other versions of Windows, this list does not exist, and the flag is ignored.

On Windows XP and Windows Server 2003, the Windows client detects the ERROR\_PRINTER\_DRIVER\_WARNED error code returned from the RpcAddPrinterDriverEx method and asks the user whether to continue adding the printer driver. If the user continues, the client sets the **APD\_INSTALL\_WARNED\_DRIVER** bit in the **dwFileCopyFlags** field and calls the method again. On Windows Vista, Windows Server 2008, Windows 7, and Windows Server 2008 R2, the Windows client detects the ERROR\_PRINTER\_DRIVER\_WARNED error code and informs the user that the driver could not be installed. Clients with these versions of the operating system do not use the **APD\_INSTALL\_WARNED\_DRIVER** flag.

[<392> Section 3.1.4.4.8](#Appendix_A_Target_392): In Windows, if this bit is set, ERROR\_PRINTER\_DRIVER\_BLOCKED is returned. if this bit is not set, ERROR\_UNKNOWN\_PRINTER\_DRIVER is returned.

[<393> Section 3.1.4.4.8](#Appendix_A_Target_393): In Windows, the server checks that the client user has the SERVER\_ACCESS\_ADMINISTER permission.

[<394> Section 3.1.4.4.8](#Appendix_A_Target_394): These validation steps are performed on the following Windows versions:

* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<395> Section 3.1.4.4.8](#Appendix_A_Target_395): These validation steps are performed on the following Windows versions:

* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<396> Section 3.1.4.4.8](#Appendix_A_Target_396): When a print client adds a printer driver to a

[<397> Section 3.1.4.4.9](#Appendix_A_Target_397): This method is used remotely only if the client is at least Windows 7 operating system and the server is at least Windows Server 2008 R2 operating system.

[<398> Section 3.1.4.4.9](#Appendix_A_Target_398): All Windows versions: The IDs are the GUIDString representations of 128-bit GUIDs.

[<399> Section 3.1.4.4.10](#Appendix_A_Target_399): This method is used remotely only if the client is at least Windows 7 operating system and the server is at least Windows Server 2008 R2 operating system.

[<400> Section 3.1.4.4.10](#Appendix_A_Target_400): All Windows versions: The *pszLanguage* string is specified using the identifiers specified for the Locale Name in [[MSDN-MUI]](http://go.microsoft.com/fwlink/?LinkId=90048).

[<401> Section 3.1.4.4.10](#Appendix_A_Target_401): All Windows versions: *pszDriverPackageCab* points to a string containing the path name of a cabinet file for the driver package. For more information, see [[MSDN-CAB]](http://go.microsoft.com/fwlink/?LinkId=226293).

[<402> Section 3.1.4.4.10](#Appendix_A_Target_402): All Windows versions: If the parameter is zero, Windows fills in the variable pointed to by *pcchRequiredSize* with the valid size.

[<403> Section 3.1.4.5.1](#Appendix_A_Target_403): In Windows, the server checks that the client user has SERVER\_ACCESS\_ADMINISTER permission.

[<404> Section 3.1.4.5.2](#Appendix_A_Target_404): In Windows, the server checks that the client user has SERVER\_ACCESS\_ADMINISTER permission.

[<405> Section 3.1.4.5.4](#Appendix_A_Target_405): The Windows server checks that the client user has SERVER\_ACCESS\_ADMINISTER permission.

[<406> Section 3.1.4.6.1](#Appendix_A_Target_406): The Windows server checks that the client user has SERVER\_ACCESS\_ENUMERATE permission.

[<407> Section 3.1.4.6.2](#Appendix_A_Target_407): The RpcDeletePort method is only called over the wire by Windows NT 4.0 operating system clients.

[<408> Section 3.1.4.6.2](#Appendix_A_Target_408): Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, and Windows Server 2012 R2 check that the client user has SERVER\_ACCESS\_ENUMERATE permission. All other Windows versions check that the client user has SERVER\_ACCESS\_ADMINISTER permission.

[<409> Section 3.1.4.6.3](#Appendix_A_Target_409): This feature is supported on the following Windows versions:

* Windows NT 3.5
* Windows NT 3.51
* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<410> Section 3.1.4.6.3](#Appendix_A_Target_410): In Windows, the server checks that the client user has the SERVER\_ACCESS\_ADMINISTER permission.

[<411> Section 3.1.4.6.4](#Appendix_A_Target_411): This feature is supported on the following Windows versions:

* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<412> Section 3.1.4.6.4](#Appendix_A_Target_412): In Windows, the server checks that the client user has SERVER\_ACCESS\_ADMINISTER permission.

[<413> Section 3.1.4.6.5](#Appendix_A_Target_413): In Windows, the actions and data retrieval supported by [RpcXcvData](#Section_bf9d2351aac0481e9d3705eb5cc961da) are performed by executing the XcvDataPort method on the port monitor module referenced by the input handle. For more information about monitor module methods, see section [3.1.4.11](#Section_07ce205123c64d508f55c25d7ac09cb3). This feature is supported on the following Windows versions:

* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<414> Section 3.1.4.6.5](#Appendix_A_Target_414): For Windows–specific behavior, see section 3.1.4.11.

[<415> Section 3.1.4.6.5](#Appendix_A_Target_415): These validation steps are performed on the following Windows versions:

* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<416> Section 3.1.4.7.1](#Appendix_A_Target_416): In Windows, the server checks that the client user has SERVER\_ACCESS\_ENUMERATE permission.

[<417> Section 3.1.4.7.1](#Appendix_A_Target_417): On Windows servers, the monitors supporting these methods are LOCALMON, LPRMON, TCPMON, [**USBMON**](#gt_3feb8a26-c933-4bab-b725-41760216ef4a), and WSDMON.

[<418> Section 3.1.4.7.2](#Appendix_A_Target_418): In Windows, the server checks that the client user has SERVER\_ACCESS\_ADMINISTER permission.

[<419> Section 3.1.4.7.3](#Appendix_A_Target_419): In Windows, the server checks that the client user has SERVER\_ACCESS\_ADMINISTER permission.

[<420> Section 3.1.4.8.1](#Appendix_A_Target_420): This validation step is performed on the following Windows versions:

* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<421> Section 3.1.4.8.1](#Appendix_A_Target_421): In Windows, the server checks that the client user has SERVER\_ACCESS\_ADMINISTER permission.

[<422> Section 3.1.4.8.2](#Appendix_A_Target_422): The Windows server checks that the client user has SERVER\_ACCESS\_ENUMERATE permission.

[<423> Section 3.1.4.8.3](#Appendix_A_Target_423): In Windows, the server checks that the client user has SERVER\_ACCESS\_ENUMERATE permission.

[<424> Section 3.1.4.8.4](#Appendix_A_Target_424): The Windows server checks that the client user has SERVER\_ACCESS\_ADMINISTER permission.

[<425> Section 3.1.4.8.5](#Appendix_A_Target_425): In Windows, the server checks that the client user has SERVER\_ACCESS\_ENUMERATE permission.

[<426> Section 3.1.4.9.6](#Appendix_A_Target_426): For a job object, the Windows server verifies that the calling client has been granted an access value that includes the **JOB\_READ** permission (section 2.2.3.1). For a port object, the Windows server verifies that the calling client has, for the currently printing job on that port, an access value that includes the **JOB\_READ** permission.

[<427> Section 3.1.4.9.6](#Appendix_A_Target_427): The Local Port monitor in Windows supports reading from COM ports.

[<428> Section 3.1.4.9.8](#Appendix_A_Target_428): This feature is supported on the following Windows version:

* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<429> Section 3.1.4.10.1](#Appendix_A_Target_429): Windows waits for up to 600 seconds

[<430> Section 3.1.4.10.2](#Appendix_A_Target_430): This feature is supported on the following Windows versions:

* Windows NT 3.5
* Windows NT 3.51
* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<431> Section 3.1.4.10.3](#Appendix_A_Target_431): This method is called only by Windows NT 3.5 clients.

[<432> Section 3.1.4.10.3](#Appendix_A_Target_432): This feature is supported on the following Windows versions:

* Windows NT 3.5
* Windows NT 3.51
* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<433> Section 3.1.4.10.3](#Appendix_A_Target_433): Windows clients generate an arbitrary unique value for each known printer.

[<434> Section 3.1.4.10.4](#Appendix_A_Target_434): This method is called only by Windows NT 3.5 clients.

[<435> Section 3.1.4.10.4](#Appendix_A_Target_435): This feature is supported on the following Windows versions:

* Windows NT 3.51
* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<436> Section 3.1.4.10.4](#Appendix_A_Target_436): Windows clients generate an arbitrary unique value for each known printer.

[<437> Section 3.1.4.10.5](#Appendix_A_Target_437): This feature is supported on the following Windows versions:

* Windows NT 3.5
* Windows NT 3.51
* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<438> Section 3.1.4.10.5](#Appendix_A_Target_438): Windows clients pass in monotonically increasing values for the *dwColor* parameter in calls to this method. Windows servers use these values in calls to [RpcRouterReplyPrinterEx (section 3.2.4.1.4)](#Section_1b44cb8295b5485c9ec90360c3a3ecb0). Clients determine the most recent notification data returned by the server by comparing the values of this parameter. Clients discard the notification data received in a call to RpcRouterReplyPrinterEx if the *dwColor* value is different from the value that was received most recently from [RpcRouterRefreshPrinterChangeNotification](#Section_cb5d22ed0352430383d5b5afd32fb461).

Windows uses this mechanism because of the possible ordering reversal of calls caused by network latency.

[<439> Section 3.1.4.11](#Appendix_A_Target_439): The Windows port monitors support these functions in the following cases:

* In Windows NT 3.5 and Windows NT 4.0, [LOCALMON](#Section_85d1c3126bba4b01bea52ec153334c1b) supports AddPortEx, AddPort, ConfigurePort, and DeletePort.
* In the following Windows versions, LOCALMON supports only AddPortEx, and TCPMON supports AddPortEx, AddPort, ConfigurePort, and DeletePort.
  + Windows 2000
  + Windows XP
  + Windows Server 2003
  + Windows Vista
  + Windows Server 2008
  + Windows 7
  + Windows Server 2008 R2
  + Windows 8
  + Windows Server 2012
  + Windows 8.1
  + Windows Server 2012 R2
  + Windows 10
  + Windows Server 2016 Technical Preview

[<440> Section 3.1.4.11](#Appendix_A_Target_440): The Windows operating system provides port monitor modules in the form of executable dynamic-link libraries (DLLs). Examples are:

* **localmon.dll**, which manages local serial ("COM") and parallel ("LPT") ports on a Windows machine (section 3.1.4.11.1).
* **lprmon.dll**, which allows a Windows print server to send print jobs to machines that support [**UNIX**](#gt_6084f060-d2f4-4ebb-8679-2d2ffa43b3ed) print-server functions (section [3.1.4.11.2](#Section_faf227a5b8a84a0eadcfa206059a91a9)).
* **tcpmon.dll**, which manages standard TCP/IP ports on a Windows machine (section 3.1.4.11.3).
* **usbmon.dll**, which manages local USB ports on a Windows machine.
* **wsdmon.dll**, which manages Web Services for Devices (WSD) ports (section 3.1.4.11.4), except on Windows NT 3.1, Windows NT 3.5, Windows NT 3.51, Windows NT 4.0, Windows 98, Windows 2000, Windows Millennium Edition, Windows XP, and Windows Server 2003.

All the functions that LOCALMON exports have been incorporated into **localspl.dll** in Windows, except for the following versions.

* Windows NT 3.1
* Windows NT 3.5
* Windows NT 3.51
* Windows NT 4.0
* Windows 98

Also in Windows, except for the preceding versions, port monitors are divided into two DLLs, a port monitor server DLL and a port monitor user interface DLL.

[<441> Section 3.1.4.11.1](#Appendix_A_Target_441): For the following versions of Windows, LOCALMON is packaged inside another executable component called **localspl.dll**; the presentation is captured in a separate monitor module. The user interface module corresponding to LOCALMON is called localui.

* Windows 2000
* Windows XP
* Windows Server 2003
* Windows Vista
* Windows Server 2008
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<442> Section 3.1.4.11.2](#Appendix_A_Target_442): The LPRMON monitor module is supported in the following versions of Windows. The user interface module corresponding to LPRMON is called **lprmonui**.

* Windows 2000
* Windows XP
* Windows Server 2003
* Windows Vista
* Windows Server 2008
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<443> Section 3.1.4.11.3](#Appendix_A_Target_443): The TCPMON monitor module is supported for the following versions of Windows. The user interface module corresponding to TCPMON is called **tcpmonui**.

* Windows 2000
* Windows XP
* Windows Server 2003
* Windows Vista
* Windows Server 2008
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<444> Section 3.1.4.11.3](#Appendix_A_Target_444): This feature is supported on the following Windows versions:

* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<445> Section 3.1.4.11.3](#Appendix_A_Target_445): This feature is supported on the following Windows versions:

* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<446> Section 3.1.4.11.3](#Appendix_A_Target_446): This feature is supported on the following Windows versions:

* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<447> Section 3.1.4.11.3](#Appendix_A_Target_447): This feature is supported on the following Windows versions:

* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<448> Section 3.1.4.11.4](#Appendix_A_Target_448): This monitor module is supported on the following Windows versions:

* Windows Vista
* Windows Server 2008
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<449> Section 3.1.4.11.4](#Appendix_A_Target_449): This feature is supported on the following Windows versions:

* Windows Vista SP1
* Windows Server 2008
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<450> Section 3.1.4.11.4](#Appendix_A_Target_450): This feature is supported on the following Windows versions:

* Windows Vista SP1
* Windows Server 2008
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<451> Section 3.1.4.11.4](#Appendix_A_Target_451): This feature is supported on the following Windows versions:

* Windows Vista SP1
* Windows Server 2008
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<452> Section 3.1.4.11.4](#Appendix_A_Target_452): This feature is supported on the following Windows versions:

* Windows Vista SP1
* Windows Server 2008
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<453> Section 3.1.4.11.4](#Appendix_A_Target_453): This feature is supported on the following Windows versions:

* Windows Vista SP1
* Windows Server 2008
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<454> Section 3.1.4.11.4](#Appendix_A_Target_454): This feature is supported on the following Windows versions:

* Windows Vista SP1
* Windows Server 2008
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<455> Section 3.1.4.11.4](#Appendix_A_Target_455): This feature is supported on the following Windows versions:

* Windows Vista SP1
* Windows Server 2008
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<456> Section 3.1.4.11.4](#Appendix_A_Target_456): This feature is supported on the following Windows versions:

* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<457> Section 3.1.4.11.4](#Appendix_A_Target_457): This command is only supported on stand-alone servers. The required value for the *pInputData* parameter is a non-NULL pointer to a string specifying the **GlobalID** of the WSD endpoint. The required value for the *pOutputData* parameter is a non-NULL pointer to a buffer that is required to receive a string identifying the new port name if a WSD Printer Service is found; otherwise, ERROR\_PRINTER\_NOT\_FOUND is returned, as specified in [MS-ERREF].

[<458> Section 3.1.4.11.4](#Appendix_A_Target_458): This feature is supported on the following Windows versions:

* Windows 10
* Windows Server 2016 Technical Preview

[<459> Section 3.1.4.12.1](#Appendix_A_Target_459): The **RpcGetJobNamedPropertyValue** method is supported by the following Windows versions:

* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<460> Section 3.1.4.12.2](#Appendix_A_Target_460): The **RpcSetJobNamedProperty** method is supported by the following Windows versions:

* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<461> Section 3.1.4.12.3](#Appendix_A_Target_461): The **RpcDeleteJobNamedProperty** method is supported by the following Windows versions:

* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<462> Section 3.1.4.12.4](#Appendix_A_Target_462): The **RpcEnumJobNamedProperties** method is supported by the following Windows versions:

* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<463> Section 3.1.4.13.1](#Appendix_A_Target_463): The [RpcLogJobInfoForBranchOffice (section 3.1.4.13.1)](#Section_65a54d4df30b41438299d61f086ac7fe) method is supported on the following Windows versions:

* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<464> Section 3.2.1](#Appendix_A_Target_464): Branch office print remote logging is supported on the following Windows versions:

* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<465> Section 3.2.1](#Appendix_A_Target_465): In Windows, this data element is stored in the print spooler.

[<466> Section 3.2.4](#Appendix_A_Target_466): The Windows implementation ignores errors and passes them back to the invoker.

[<467> Section 3.2.4.1.1](#Appendix_A_Target_467): This feature is supported on the following Windows versions:

* Windows NT 3.5
* Windows NT 3.51
* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<468> Section 3.2.4.1.1](#Appendix_A_Target_468): The Windows server calls [RpcReplyOpenPrinter](#Section_7fcd3036d45a4ec7b081f2b860e66676) during the processing of a call to [RpcRemoteFindFirstPrinterChangeNotificationEx](#Section_eb66b2211c1f4249b8bcc5befec2314d). The return value indicates the success of that processing.

[<469> Section 3.2.4.1.2](#Appendix_A_Target_469): This feature is supported on the following Windows versions:

* Windows NT 3.5
* Windows NT 3.51
* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<470> Section 3.2.4.1.3](#Appendix_A_Target_470): This feature is supported on the following Windows versions:

* Windows NT 3.5
* Windows NT 3.51
* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<471> Section 3.2.4.1.4](#Appendix_A_Target_471): This feature is supported on the following Windows versions:

* Windows NT 3.51
* Windows NT 4.0
* Windows 2000
* Windows XP
* Windows Server 2003
* Windows XP SP1
* Windows Vista
* Windows Server 2008
* Windows Vista SP1
* Windows 7
* Windows Server 2008 R2
* Windows 8
* Windows Server 2012
* Windows 8.1
* Windows Server 2012 R2
* Windows 10
* Windows Server 2016 Technical Preview

[<472> Section 3.2.4.2.3](#Appendix_A_Target_472): The Windows system will select a number of jobs to be 0xFFFFFFFF to obtain the full list in a single operation.

[<473> Section 3.2.4.2.5](#Appendix_A_Target_473): Windows clients only enumerate their shared printers if no directory services are available in the domain.

[<474> Section 3.2.4.2.5](#Appendix_A_Target_474): Windows clients enumerate their shared printers up to a default of three print servers and a default of two workstations and one additional server per 32 print servers found. The values can be changed via policy settings.

[<475> Section 3.2.4.2.6](#Appendix_A_Target_475): A Windows client uses [RpcOpenPrinterEx](#Section_0e81ce1872b146c38584a205393b04ff) to open a handle for the server object. Then the Windows client uses RpcGetPrinterData with the value names "Architecture", "MajorVersion", and "OSVersion" to obtain information about the server's environment. This information is used to select an appropriate printer driver. The client subsequently closes the server object handle using [RpcClosePrinter](#Section_8482c18677734fb58034e9bae7a91e52). The Windows-based client also uses RpcEnumPorts to populate a list with ports available on the server from which the end-user can select the port for the new printer. The Windows-based client uses [RpcEnumMonitors](#Section_6c5836fed9ea41a0ad7e01152f39e8a1) to determine a port monitor for the new printer. The Windows-based client uses [RpcGetPrinterDriverDirectory](#Section_9df11cf440984852ad72d1f75a82bffe) to determine the destination directory in case the new printer requires that a printer driver be installed.

[<476> Section 5.2](#Appendix_A_Target_476): Windows print server follows a security model in which the print server, print queues, and print job are securable resources. Each of these resources has a security descriptor that is associated with it. The [**SD**](#gt_33e80269-dca4-4c3b-90df-b0f957571213) contains the security information that is associated with a resource on the print server. The print server checks the client access to resources by comparing the security information that is associated with the caller against the SD of the resource. SDs are specified by the SECURITY\_DESCRIPTOR structure ([MS-DTYP] section 2.4.6).

Each RPC client has associated with it an access token, which contains the [**security identifier (SID)**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d) of the user making the RPC call. The SD identifies the owner of the printing resource and contains a discretionary access control list (DACL). The DACL contains access control entries (ACEs), which specify the SID of a user or group of users, and whether access rights are to be allowed, denied, or audited. For resources on a print server, the ACEs specify operations including: print, manage printers, and manage documents in a print queue.

The SD that is associated with the print server or print queue controls the creation of the context handle ([PRINTER\_HANDLE](#Section_9c5d49b8692047b0946687ba910e6071) section 2.2.1.1.4) and the outcome of operations that use the handle. These operations range from printing and job management to listening for notifications.

The SDs for a Windows-based print server are used to control the creation and deletion of print queues on the server; the installation of print system components, including printer drivers, print processors, and port monitors; and the enumeration of resources on the server. A server SD is not write-accessible to callers. In addition to being used to control callers' access to resources, the server SD is also used as the "parent" in the creation of the print queue's SD.

**Note:** The SD of the Windows-based print server is different from the SD that is applied on the named pipe of the spool. The SD of the spool's named pipe controls the RPC client's access to make RPC calls to the print server. The SD of the Windows-based print server is used to control the caller's permissions to perform various operations on the print server.

The SD of the print queue controls the setting of print queue properties, such as the port and printer driver that are used for printing; the rendering and device settings; or the sharing or security parameters. The printer SD allows auditing operations such as printing; managing printers and documents; reading and changing permissions; and taking ownership.

Each print job has an associated SD, which is created by using the SD of the print queue as parent. The user who submitted the document for printing is the owner of the print job and has permission to manage the print job during its lifetime.

When a PRINTER\_HANDLE is opened for a specific printing resource, the caller must specify the access that is needed for subsequent operations, including printer or server administration; printing on a printer or print server; and reading, writing, or administrating a print job. If the caller has the specified permissions, the print handle is created and can be used for subsequent calls.

In addition to handle-based operations, the SD is used for access checks when enumerations, printer driver package installation, or other non-handle-based operations are performed.

# Change Tracking

This section identifies changes that were made to this document since the last release. Changes are classified as New, Major, Minor, Editorial, or No change.

The revision class **New** means that a new document is being released.

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

* A document revision that incorporates changes to interoperability requirements or functionality.
* The removal of a document from the documentation set.

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

The revision class **Editorial** means that the formatting in the technical content was changed. Editorial changes apply to grammatical, formatting, and style issues.

The revision class **No change** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the technical content of the document is identical to the last released version.

Major and minor changes can be described further using the following change types:

* New content added.
* Content updated.
* Content removed.
* New product behavior note added.
* Product behavior note updated.
* Product behavior note removed.
* New protocol syntax added.
* Protocol syntax updated.
* Protocol syntax removed.
* New content added due to protocol revision.
* Content updated due to protocol revision.
* Content removed due to protocol revision.
* New protocol syntax added due to protocol revision.
* Protocol syntax updated due to protocol revision.
* Protocol syntax removed due to protocol revision.
* Obsolete document removed.

Editorial changes are always classified with the change type **Editorially updated**.

Some important terms used in the change type descriptions are defined as follows:

* **Protocol syntax** refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.
* **Protocol revision** refers to changes made to a protocol that affect the bits that are sent over the wire.

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

| Section | Tracking number (if applicable) and description | Major change (Y or N) | Change type |
| --- | --- | --- | --- |
| [2.2.3.8](#Section_41d5c622ec7749ff95e369b325ce4e77) Printer Notification Values | 72315 : Added printer notification categories for 2D and 3D printers. | Y | New content added. |
| 2.2.3.8 Printer Notification Values | 72315 : Added product behavior note for 3D printer support. | Y | New product behavior note added. |
| [2.2.3.11](#Section_4a81aa1c469e49418c4db8c4093b7fc2) Printer Data Values | Updated content for Windows 10 and Windows Server 2016 Technical Preview. | Y | Content update. |
| [3.1.4.2.27](#Section_5aec83472ebf464eb6d1540aa60b5c71) RpcSendRecvBidiData (Opnum 97) | Updated content for Windows 10 and Windows Server 2016 Technical Preview. | Y | Content update. |
| [3.1.4.7.1](#Section_6c5836fed9ea41a0ad7e01152f39e8a1) RpcEnumMonitors (Opnum 36) | Added USB monitor to Windows server print monitor modules. | Y | Product behavior note updated. |
| [3.1.4.10.3](#Section_b8b414d9f1cd4191bb6b87d09ab2fd83) RpcRemoteFindFirstPrinterChangeNotification (Opnum 62) | 72315 : Added details of values for the fdwOptions parameter. | Y | New content added. |
| [3.1.4.10.4](#Section_eb66b2211c1f4249b8bcc5befec2314d) RpcRemoteFindFirstPrinterChangeNotificationEx (Opnum 65) | 72315 : Added details of values for the fdwOptions parameter. | Y | New content added. |
| [3.1.4.11](#Section_07ce205123c64d508f55c25d7ac09cb3) Monitor Module Methods | Added USB monitor module DLL to the list of print monitor module DLLs supported on Windows. | Y | Product behavior note updated. |
| [3.1.4.11.4](#Section_e1760a13634e4a7fa7dd9d02e4048f9b) WSDMON | Added content for Windows 10. | Y | Content update. |
| [7](#Section_e81cbc09ab054a32ae4a8ec57b436c43) Appendix B: Product Behavior | Added Windows 10 to applicability list. | Y | Content update. |
| 7 Appendix B: Product Behavior | Updated product behavior notes for Windows 10 and Windows Server 2016 Technical Preview. | Y | Product behavior note updated. |

# Index

**\_**

[\_DEVMODE packet](#section_ccc2a501794e4d2bb312f69c75131c2e) 77

[\_DRIVER\_FILE\_INFO packet](#section_37ac9d7f9ee54e1a98e81186007ca52e) 108

[\_DRIVER\_INFO](#section_62b7b83c16144e53bccbb609aa2a93e4) 92

[\_DRIVER\_INFO\_1 packet](#section_59eab35482fd4299884e9a1226073a4a) 92

[\_DRIVER\_INFO\_101 packet](#section_821d675e39444c77b4ec3b519883cfa6) 106

[\_DRIVER\_INFO\_2 packet](#section_2825d22ec5a547cda2163e903fd6e030) 92

[\_DRIVER\_INFO\_3 packet](#section_b9a54cc1e79f45c8a1261acd1f0f8663) 94

[\_DRIVER\_INFO\_4 packet](#section_2a53e91004c34aba905a26741dcd59dd) 95

[\_DRIVER\_INFO\_5 packet](#section_e20b5b5573494f50a9f2bcc02d958cfc) 97

[\_DRIVER\_INFO\_6 packet](#section_9a048522474644c2ac92d881ac1f36a8) 99

[\_DRIVER\_INFO\_7 packet](#section_6cf1b6abf8a1429ba2cd2c3e0ae80c28) 101

[\_DRIVER\_INFO\_8 packet](#section_f9b749d608fc48208d7bfe0ee33e42c5) 103

[\_FORM\_INFO](#section_af01dfcd84ba455ba03c3ba656afa764) 110

[\_FORM\_INFO\_1 packet](#section_95e918defbdb4d0aad4cd7005ab33305) 110

[\_FORM\_INFO\_2 packet](#section_b15260d2493c49ac9062720162e1aee1) 111

[\_JOB\_INFO](#section_aed5b928e9d24c65b4136644531ab8d5) 112

[\_JOB\_INFO\_1 packet](#section_34f34c553d024991b5d3e24159a90936) 112

[\_JOB\_INFO\_2 packet](#section_3e09be6b388941cd97522ee81c758e87) 114

[\_JOB\_INFO\_3 packet](#section_4c779cf650274c04934f808bc0815087) 116

[\_JOB\_INFO\_4 packet](#section_fdda558466164904b98884726b5915ee) 117

[\_MONITOR\_INFO](#section_083e6e86c87a49a0a67975827d993511) 120

[\_MONITOR\_INFO\_1 packet](#section_0496f2b4c42c4363bb032a5df819f7ba) 120

[\_MONITOR\_INFO\_2 packet](#section_a69e678518be483a90f97a283df84fbf) 120

[\_PORT\_INFO](#section_f4735f9ace70472f92b37bbed62609ab) 121

[\_PORT\_INFO\_1 packet](#section_086847b9a37041499ceda41ecb436d3b) 121

[\_PORT\_INFO\_2 packet](#section_d09458b84d2b48f2b7500d07ea24f47a) 122

[\_PRINTER\_INFO](#section_fc749a2738ec4063b0c1dc477c120ffd) 123

[\_PRINTER\_INFO\_1 packet](#section_39aa2f28ffe04f918bddc0fa5dee40c5) 125

[\_PRINTER\_INFO\_2 packet](#section_9e3273757ac9419fa61061b88bbbd918) 126

[\_PRINTER\_INFO\_3 packet](#section_0cc3c48367144682aee454f588cab949) 128

[\_PRINTER\_INFO\_4 packet](#section_b739452348b74e16bc9d3c87d8a5f0c0) 129

[\_PRINTER\_INFO\_5 packet](#section_68a546b0d7484a3da5d737f713e037d9) 130

[\_PRINTER\_INFO\_6 packet](#section_f784565a2b814cc0a1e31c77212f7e28) 131

[\_PRINTER\_INFO\_7 packet](#section_01d350511b3945f8ad3964391d8fb445) 131

[\_PRINTER\_INFO\_8 packet](#section_55f96f18a1344778b1fa556a801997b7) 132

[\_PRINTER\_INFO\_STRESS packet](#section_b026cf5c680f464a8e08675228ddd42c) 123

**A**

Abstract data model

[client](#section_fa5a51d585cc40dc951a1e288b62d820) 302

[server](#section_1bd61448db4b4284a4b18346732402d4) 180

[Adding a printer driver to a server example](#section_bc4d18308e1f428a9073d655916494cf) 313

[Adding a printer to a server example](#section_a33ebcedd8d544dba7287a77d0e40231) 311

[Adding printer driver to server example](#section_bc4d18308e1f428a9073d655916494cf) 313

[Adding printer to server example](#section_a33ebcedd8d544dba7287a77d0e40231) 311

[Applicability](#section_256291f073704f60b6c176847cb97fa2) 28

**B**

[BIDI\_TYPE enumeration](#section_4ebaf54761f64b2eb32872c1a5dc685d) 168

[Bidirectional communication data](#section_38fb51d87f9e4b16ac406f48244214ed) 64

[Branch Office Print Remote Logging Methods method](#section_fd04e0de264746e0a06aa3ccb138dbb6) 301

**C**

[Capability negotiation](#section_0c08f94330bd4653983342447e5093d5) 28

[Change tracking](#section_32b7aecb8a7a48bdae1d35cac60bff8d) 482

Client

[abstract data model](#section_fa5a51d585cc40dc951a1e288b62d820) 302

[Client Interaction with the Print Server method](#section_594b211ba073475a9bcd74a79f065024) 307

[Client-Side Notification Processing Methods method](#section_ee34aeeec9464991be9fc6d0049fb023) 303

[initialization](#section_b40da97efa2544cf8c951cd93b0d8286) 303

[interaction with print server](#section_594b211ba073475a9bcd74a79f065024) 307

[local events](#section_1f7170188df248c5b818c40112941eb5) 310

[message processing](#section_0f30f3700d7f43df922737f9155cf821) 303

[sequencing rules](#section_0f30f3700d7f43df922737f9155cf821) 303

[timer events](#section_d640a376b59c40239761e49a0db24431) 310

[timers](#section_4b32ff7006804aeca52049a6ec47b36f) 303

[Client Interaction with the Print Server method](#section_594b211ba073475a9bcd74a79f065024) 307

[Client-Side Notification Processing Methods method](#section_ee34aeeec9464991be9fc6d0049fb023) 303

[Client-side notification-processing methods](#section_ee34aeeec9464991be9fc6d0049fb023) 303

[Common data types](#section_9b03c5103a0e48219aa5fc79dbf8a188) 30

[Common data types - overview](#section_9b03c5103a0e48219aa5fc79dbf8a188) 30

[Common IDL data types](#section_8c5b1f140514438da966e327c1dad594) 32

[Commonly Used Parameters method](#section_7eb63f7a637a4b7582e091853f6f6013) 194

[CONFIG\_INFO\_DATA\_1](#section_9764f82547a1440c973a32328d204296) 136

[CONFIG\_INFO\_DATA\_1 packet](#section_9764f82547a1440c973a32328d204296) 136

[Constants](#section_a27734227f094844858afd59c0121b2a) 144

[Containers](#section_8ca20b730d864e57a009f797ef1b2de8) 35

[CORE\_PRINTER\_DRIVER packet](#section_7cdc8cf152854420bfa53233ed20125b) 135

[Custom marshaled data types](#section_99fdf2cf79bb43bf997c88b33538a5e1) 75

[Custom\_Marshaled\_Data\_Types packet](#section_99fdf2cf79bb43bf997c88b33538a5e1) 75

**D**

Data model - abstract

[client](#section_fa5a51d585cc40dc951a1e288b62d820) 302

[server](#section_1bd61448db4b4284a4b18346732402d4) 180

Data types

[common](#section_9b03c5103a0e48219aa5fc79dbf8a188) 30

[common - overview](#section_9b03c5103a0e48219aa5fc79dbf8a188) 30

[custom marshaled](#section_99fdf2cf79bb43bf997c88b33538a5e1) 75

[IDL](#section_df6af0abee9745aaae87186f83ab2265) 32

[DATATYPES\_INFO\_1 packet](#section_7224d5eac0e1474b8e24034f0032e09c) 91

[DELETE](#section_945c4df5496949c5b1ce59c1b3f35024) 144

[DELETE\_PORT\_DATA\_1](#section_98b804cc08694f1f8f91dacf63769d7e) 136

[DELETE\_PORT\_DATA\_1 packet](#section_98b804cc08694f1f8f91dacf63769d7e) 136

[DEVMODE structure](#section_0136fad4f02c44dd857f820477b19c1d) 32

[DEVMODE\_CONTAINER structure](#section_c2b14fe764794460a7ba633a845dd61a) 35

Directory service

[interaction details - overview](#section_9e24210cdc8e474d8ffd7b9dbe168440) 175

[interaction summary](#section_1104f58a038247e3946336d5225ed856) 173

[schema elements](#section_8b0f1957cb264966b55f371858ea4406) 174

[Discovery methods](#section_5ea67df3c4e745efb42540b8b066d276) 205

[DOC\_INFO\_1 structure](#section_20f7b510e6a04f6c97a05630d1e70879) 47

[DOC\_INFO\_CONTAINER structure](#section_c4f39f63301745bf8b8a81ab9458f052) 35

[Document printing methods](#section_93e6c5a794d6487ebe20d2c91e5277ad) 278

[Document Printing Methods method](#section_93e6c5a794d6487ebe20d2c91e5277ad) 278

[DRIVER\_CONTAINER structure](#section_3a3f9cf78ec44921b1f686cf8d139bc2) 35

[DRIVER\_INFO](#section_17943579a5144d0b8ded422265b23f45) 47

[DRIVER\_INFO members](#section_4464eaf0f34f40d5b970736437a21913) 43

[DRIVER\_INFO\_1 structure](#section_df308bbf8ff74b0f930bcb9b90080b90) 47

[DRIVER\_INFO\_2 structure](#section_39bbfc3087684cd49930434857e2c2a2) 48

**E**

[EBranchOfficeJobEventType enumeration](#section_d070101180b3473d843cedf7e4c5bc86) 70

[Enumerating and managing printers example](#section_e35fa2d28ca14369be526e606759bd0e) 314

[Enumerating jobs and modifying job settings example](#section_7307d04985784769b2c50e684ab0f8e3) 317

[Enumerating print jobs example](#section_7307d04985784769b2c50e684ab0f8e3) 317

[Enumerating printers example](#section_e35fa2d28ca14369be526e606759bd0e) 314

[EVENTLOG\_AUDIT\_FAILURE](#section_34284809abff4ecea5ad71f47160e5e4) 162

[EVENTLOG\_AUDIT\_SUCCESS](#section_34284809abff4ecea5ad71f47160e5e4) 162

[EVENTLOG\_ERROR\_TYPE](#section_34284809abff4ecea5ad71f47160e5e4) 162

[EVENTLOG\_INFORMATION\_TYPE](#section_34284809abff4ecea5ad71f47160e5e4) 162

[EVENTLOG\_WARNING\_TYPE](#section_34284809abff4ecea5ad71f47160e5e4) 162

Events

[local - client](#section_1f7170188df248c5b818c40112941eb5) 310

[local - server](#section_06246c515560435ab514b180917584db) 302

[timer - client](#section_d640a376b59c40239761e49a0db24431) 310

[timer - server](#section_615e516d7d8e4a50a0a62607a4401500) 302

Examples

[adding a printer driver to a server](#section_bc4d18308e1f428a9073d655916494cf) 313

[adding a printer to a server](#section_a33ebcedd8d544dba7287a77d0e40231) 311

[adding printer driver to server](#section_bc4d18308e1f428a9073d655916494cf) 313

[adding printer to server](#section_a33ebcedd8d544dba7287a77d0e40231) 311

[enumerating and managing printers](#section_e35fa2d28ca14369be526e606759bd0e) 314

[enumerating jobs and modifying job settings](#section_7307d04985784769b2c50e684ab0f8e3) 317

[enumerating print jobs and modifying job settings](#section_7307d04985784769b2c50e684ab0f8e3) 317

[receiving notifications on printing events](#section_20bc2e9cce884fd399612a4f06ebe41f) 319

**F**

[Form management methods](#section_0b9ceec4c17747aea4bea5d3167fc205) 260

[Form Management Methods method](#section_0b9ceec4c17747aea4bea5d3167fc205) 260

[FORM\_CONTAINER structure](#section_49508bc38fcb49f5b92d0724666d76a4) 36

[FORM\_INFO](#section_2fef1fa4518f4e958572c6491442cf2a) 51

[FORM\_INFO members](#section_84fe09faea4b4cc58ae49c8582fd1ffe) 44

[FORM\_INFO\_1 structure](#section_7821930259dd4d3f82d73a2e95466c28) 51

[Full IDL](#section_e8f9dad8d11441cc9a52fc927e908cf4) 323

**G**

[Generic Driver Extra Data](#section_b054db1d026a4485b3cef36d86135bf5) 90

[GENERIC\_ALL](#section_945c4df5496949c5b1ce59c1b3f35024) 144

[GENERIC\_EXECUTE](#section_945c4df5496949c5b1ce59c1b3f35024) 144

[GENERIC\_READ](#section_945c4df5496949c5b1ce59c1b3f35024) 144

[GENERIC\_WRITE](#section_945c4df5496949c5b1ce59c1b3f35024) 144

[Glossary](#section_831cd729be7c451eb729bd8d84ce4d24) 11

**I**

[IDL](#section_e8f9dad8d11441cc9a52fc927e908cf4) 323

[IDL - data types](#section_df6af0abee9745aaae87186f83ab2265) 32

[Implementer - security considerations](#section_fc2fbb2b10be440090efeb748090ade4) 322

[Index of security parameters](#section_1586a2e8c4cb4ed0a1325fecf217fea5) 322

[Informative references](#section_88d64aedfcbb49578ecef3eaac0a27b0) 22

Initialization

[client](#section_b40da97efa2544cf8c951cd93b0d8286) 303

[server](#section_50cafaeac64b4022b27b9c28d7b799ab) 184

[Introduction](#section_d3e4e6f380514340a5d611c871fcc40b) 11

**J**

[Job management methods](#section_bd8e1517a5b14850933866575d2b8cf4) 239

[Job Management Methods method](#section_bd8e1517a5b14850933866575d2b8cf4) 239

[Job named property management methods](#section_95aa7a44c15841a0b2ad2a08ccd0881f) 297

[Job Named Property Management Methods method](#section_95aa7a44c15841a0b2ad2a08ccd0881f) 297

[JOB\_ACCESS\_ADMINISTER](#section_945c4df5496949c5b1ce59c1b3f35024) 144

[JOB\_ACCESS\_READ](#section_945c4df5496949c5b1ce59c1b3f35024) 144

[JOB\_ALL\_ACCESS](#section_945c4df5496949c5b1ce59c1b3f35024) 144

[JOB\_CONTAINER structure](#section_0cd5f47b54214b199ed05a29d63f3c2a) 36

[JOB\_EXECUTE](#section_945c4df5496949c5b1ce59c1b3f35024) 144

[JOB\_INFO](#section_87ea3e7722fb4914b803ef3b2619195a) 52

[JOB\_INFO members](#section_94d1857ec44449b18fb483eac6696060) 45

[JOB\_INFO\_1 structure](#section_794c1ba9ba0a4498ba9c2635563460dd) 52

[JOB\_INFO\_2 structure](#section_5a48c6c9fc5f42ffa5a7f111fee58d1a) 53

[JOB\_INFO\_3 structure](#section_1385aef9d3234e8d9d285f2f011bd39c) 53

[JOB\_INFO\_4 structure](#section_99086f6e3db14034ac6ccd4a5a77f22c) 54

[JOB\_NOTIFY\_FIELD\_BYTES\_PRINTED](#section_76acb7fbb63f4cad88dd6920096a2f13) 147

[JOB\_NOTIFY\_FIELD\_DATATYPE](#section_76acb7fbb63f4cad88dd6920096a2f13) 147

[JOB\_NOTIFY\_FIELD\_DEVMODE](#section_76acb7fbb63f4cad88dd6920096a2f13) 147

[JOB\_NOTIFY\_FIELD\_DOCUMENT](#section_76acb7fbb63f4cad88dd6920096a2f13) 147

[JOB\_NOTIFY\_FIELD\_DRIVER\_NAME](#section_76acb7fbb63f4cad88dd6920096a2f13) 147

[JOB\_NOTIFY\_FIELD\_MACHINE\_NAME](#section_76acb7fbb63f4cad88dd6920096a2f13) 147

[JOB\_NOTIFY\_FIELD\_NOTIFY\_NAME](#section_76acb7fbb63f4cad88dd6920096a2f13) 147

[JOB\_NOTIFY\_FIELD\_PAGES\_PRINTED](#section_76acb7fbb63f4cad88dd6920096a2f13) 147

[JOB\_NOTIFY\_FIELD\_PARAMETERS](#section_76acb7fbb63f4cad88dd6920096a2f13) 147

[JOB\_NOTIFY\_FIELD\_PORT\_NAME](#section_76acb7fbb63f4cad88dd6920096a2f13) 147

[JOB\_NOTIFY\_FIELD\_POSITION](#section_76acb7fbb63f4cad88dd6920096a2f13) 147

[JOB\_NOTIFY\_FIELD\_PRINT\_PROCESSOR](#section_76acb7fbb63f4cad88dd6920096a2f13) 147

[JOB\_NOTIFY\_FIELD\_PRINTER\_NAME](#section_76acb7fbb63f4cad88dd6920096a2f13) 147

[JOB\_NOTIFY\_FIELD\_PRIORITY](#section_76acb7fbb63f4cad88dd6920096a2f13) 147

[JOB\_NOTIFY\_FIELD\_SECURITY\_DESCRIPTOR](#section_76acb7fbb63f4cad88dd6920096a2f13) 147

[JOB\_NOTIFY\_FIELD\_START\_TIME](#section_76acb7fbb63f4cad88dd6920096a2f13) 147

[JOB\_NOTIFY\_FIELD\_STATUS](#section_76acb7fbb63f4cad88dd6920096a2f13) 147

[JOB\_NOTIFY\_FIELD\_STATUS\_STRING](#section_76acb7fbb63f4cad88dd6920096a2f13) 147

[JOB\_NOTIFY\_FIELD\_SUBMITTED](#section_76acb7fbb63f4cad88dd6920096a2f13) 147

[JOB\_NOTIFY\_FIELD\_TIME](#section_76acb7fbb63f4cad88dd6920096a2f13) 147

[JOB\_NOTIFY\_FIELD\_TOTAL\_BYTES](#section_76acb7fbb63f4cad88dd6920096a2f13) 147

[JOB\_NOTIFY\_FIELD\_TOTAL\_PAGES](#section_76acb7fbb63f4cad88dd6920096a2f13) 147

[JOB\_NOTIFY\_FIELD\_UNTIL\_TIME](#section_76acb7fbb63f4cad88dd6920096a2f13) 147

[JOB\_NOTIFY\_FIELD\_USER\_NAME](#section_76acb7fbb63f4cad88dd6920096a2f13) 147

[JOB\_READ](#section_945c4df5496949c5b1ce59c1b3f35024) 144

[JOB\_STATUS\_BLOCKED\_DEVQ](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[JOB\_STATUS\_COMPLETE](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[JOB\_STATUS\_DELETED](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[JOB\_STATUS\_DELETING](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[JOB\_STATUS\_ERROR](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[JOB\_STATUS\_OFFLINE](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[JOB\_STATUS\_PAPEROUT](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[JOB\_STATUS\_PAUSED](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[JOB\_STATUS\_PRINTED](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[JOB\_STATUS\_PRINTING](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[JOB\_STATUS\_RESTART](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[JOB\_STATUS\_SPOOLING](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[JOB\_STATUS\_USER\_INTERVENTION](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[JOB\_WRITE](#section_945c4df5496949c5b1ce59c1b3f35024) 144

**L**

Local events

[client](#section_1f7170188df248c5b818c40112941eb5) 310

[server](#section_06246c515560435ab514b180917584db) 302

**M**

[Managing printers example](#section_e35fa2d28ca14369be526e606759bd0e) 314

Members

in INFO structures

[custom-marshaled](#section_35bed5da52e24bf9aeefed33f64cd375) 91

[overview](#section_964b517f67304b4a99edee9648e7f1f7) 42

[rules](#section_6c6af00855314ccab0113da5215505ca) 169

[vendor-extensible](#section_39112098aa6d440ab661a5061dfd100d) 29

Message processing

[client](#section_0f30f3700d7f43df922737f9155cf821) 303

[server](#section_700f007d72b74b44951756569704bb8d) 184

Messages

[common data types](#section_9b03c5103a0e48219aa5fc79dbf8a188) 30

[transport](#section_fdf1138af6b94b00ada61fbb6097d683) 30

Methods

[Branch Office Print Remote Logging Methods](#section_fd04e0de264746e0a06aa3ccb138dbb6) 301

[Client Interaction with the Print Server](#section_594b211ba073475a9bcd74a79f065024) 307

[Client-Side Notification Processing Methods](#section_ee34aeeec9464991be9fc6d0049fb023) 303

[Commonly Used Parameters](#section_7eb63f7a637a4b7582e091853f6f6013) 194

[Document Printing Methods](#section_93e6c5a794d6487ebe20d2c91e5277ad) 278

[Form Management Methods](#section_0b9ceec4c17747aea4bea5d3167fc205) 260

[Job Management Methods](#section_bd8e1517a5b14850933866575d2b8cf4) 239

[Job Named Property Management Methods](#section_95aa7a44c15841a0b2ad2a08ccd0881f) 297

[Monitor Module Methods](#section_07ce205123c64d508f55c25d7ac09cb3) 292

[Notification Methods](#section_72b88737108f4e398fa9ee3af45da015) 286

[Port Management Methods](#section_1cf0e23868da43bdbaba44b35b115cc0) 264

[Port Monitor Management Methods](#section_2c505550d9b24123bfa9ae204f4d88d6) 270

[Print Processor Management Methods](#section_701f22dc32934ef28aa64d03ad4ef868) 273

[Printer Driver Management Methods](#section_84e21d0dd5da4db2ab19d7ece51e42aa) 245

[Printer Management and Discovery Methods](#section_5ea67df3c4e745efb42540b8b066d276) 205

[Modifying job settings example](#section_7307d04985784769b2c50e684ab0f8e3) 317

[Monitor module methods](#section_07ce205123c64d508f55c25d7ac09cb3) 292

[Monitor Module Methods method](#section_07ce205123c64d508f55c25d7ac09cb3) 292

[MONITOR\_CONTAINER structure](#section_b23382165e3e4379a722af2230d06df2) 37

[MONITOR\_INFO](#section_822a6b4d37814a579eefd572dcf4d86f) 54

[MONITOR\_INFO members](#section_70e8e10238754ce8a292aaabc88e1458) 46

[MONITOR\_INFO\_1 structure](#section_7ee6a75f988941a68b1490d4e378b956) 54

[MONITOR\_INFO\_2 structure](#section_f05b335b203c454f87da69ad96930bd7) 54

**N**

[Normative references](#section_1249140046a74cee9ce70c8eb7ceac28) 21

[Notification methods](#section_72b88737108f4e398fa9ee3af45da015) 286

[Notification Methods method](#section_72b88737108f4e398fa9ee3af45da015) 286

**O**

[OEM Driver Extra Data](#section_8307eefb5edc47eab3effd8e6d58ba1b) 90

[OS\_TYPE enumeration](#section_682158a5b093400b8dd9b31f3fdd04d5) 161

[OSVERSIONINFO packet](#section_5942648eb54f4e22a0e22d000e084b23) 160

[OSVERSIONINFOEX packet](#section_cd8e00e84c444e4f98d30db33d75d06d) 161

[Overview (synopsis)](#section_a7cceedb3afc45f082e1af2b2cbf20e4) 23

**P**

[Parameter index - security](#section_1586a2e8c4cb4ed0a1325fecf217fea5) 322

[Parameters](#section_7eb63f7a637a4b7582e091853f6f6013) 194

[Parameters - security index](#section_1586a2e8c4cb4ed0a1325fecf217fea5) 322

[Port management methods](#section_1cf0e23868da43bdbaba44b35b115cc0) 264

[Port Management Methods method](#section_1cf0e23868da43bdbaba44b35b115cc0) 264

[Port monitor management methods](#section_2c505550d9b24123bfa9ae204f4d88d6) 270

[Port Monitor Management Methods method](#section_2c505550d9b24123bfa9ae204f4d88d6) 270

[PORT\_CONTAINER structure](#section_e05adf6ca04e4e398b34e7c118ce2bc8) 38

[PORT\_DATA\_1](#section_bd1e99bb8a2747c3b276e38fda1f9d34) 137

[PORT\_DATA\_1 packet](#section_bd1e99bb8a2747c3b276e38fda1f9d34) 137

[PORT\_DATA\_2](#section_2d256a4b819c4b848e8fb1edb1dcf0c2) 139

[PORT\_DATA\_2 packet](#section_2d256a4b819c4b848e8fb1edb1dcf0c2) 139

[PORT\_DATA\_LIST\_1](#section_345c0cf7ae98467a8df5849ce3eb7e69) 141

[PORT\_DATA\_LIST\_1 packet](#section_345c0cf7ae98467a8df5849ce3eb7e69) 141

[PORT\_INFO](#section_05c1a8d20bcb4292bc65c8cc2e7f90cf) 55

[PORT\_INFO members](#section_2607e270ebe64d84a2f712b056fae964) 46

[PORT\_INFO\_1 structure](#section_9bab3fb5b36948f28f083082ca13e739) 55

[PORT\_INFO\_2 structure](#section_7a923364dffc40b3bca5bf2eed610f31) 55

[PORT\_INFO\_3 structure](#section_411e442325e44f6184b745336e40094e) 56

[PORT\_INFO\_FF structure](#section_6366af02dc304987a1a3f0ec4176ff94) 57

[PORT\_VAR\_CONTAINER structure](#section_0cf200e2c00a4c538cb7c150c44cbdbd) 38

[PostScript Driver Extra Data](#section_2809187d8eef4fd28029ab43d78bd01f) 90

[Preconditions](#section_254b87395be24995b192a070fc33966c) 28

[Prerequisites](#section_254b87395be24995b192a070fc33966c) 28

[Print processor management methods](#section_701f22dc32934ef28aa64d03ad4ef868) 273

[Print Processor Management Methods method](#section_701f22dc32934ef28aa64d03ad4ef868) 273

[Print Ticket Driver Extra Data](#section_2ca89612aa0a44f68fa49042b9c960a8) 90

[Printer driver management methods](#section_84e21d0dd5da4db2ab19d7ece51e42aa) 245

[Printer Driver Management Methods method](#section_84e21d0dd5da4db2ab19d7ece51e42aa) 245

[Printer Management and Discovery Methods method](#section_5ea67df3c4e745efb42540b8b066d276) 205

[Printer management methods](#section_5ea67df3c4e745efb42540b8b066d276) 205

[Printer notification data](#section_c7e020b1c21948d29130a4e96ac03c3f) 65

[PRINTER\_ACCESS\_ADMINISTER](#section_945c4df5496949c5b1ce59c1b3f35024) 144

[PRINTER\_ACCESS\_MANAGE\_LIMITED](#section_945c4df5496949c5b1ce59c1b3f35024) 144

[PRINTER\_ACCESS\_USE](#section_945c4df5496949c5b1ce59c1b3f35024) 144

[PRINTER\_ALL\_ACCESS](#section_945c4df5496949c5b1ce59c1b3f35024) 144

[PRINTER\_ATTRIBUTE\_DEFAULT](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[PRINTER\_ATTRIBUTE\_DIRECT](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[PRINTER\_ATTRIBUTE\_DO\_COMPLETE\_FIRST](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[PRINTER\_ATTRIBUTE\_ENABLE\_BIDI](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[PRINTER\_ATTRIBUTE\_ENABLE\_DEVQ](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[PRINTER\_ATTRIBUTE\_FAX](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[PRINTER\_ATTRIBUTE\_KEEPPRINTEDJOBS](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[PRINTER\_ATTRIBUTE\_LOCAL](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[PRINTER\_ATTRIBUTE\_NETWORK](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[PRINTER\_ATTRIBUTE\_PUBLISHED](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[PRINTER\_ATTRIBUTE\_QUEUED](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[PRINTER\_ATTRIBUTE\_RAW\_ONLY](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[PRINTER\_ATTRIBUTE\_SHARED](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[PRINTER\_ATTRIBUTE\_TS](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[PRINTER\_ATTRIBUTE\_WORK\_OFFLINE](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[PRINTER\_CHANGE\_ADD\_FORM](#section_badfe1eba5214686b2accac29dff311a) 151

[PRINTER\_CHANGE\_ADD\_JOB](#section_1d63f9efa4cd41e99152004fd2a8855a) 150

[PRINTER\_CHANGE\_ADD\_PORT](#section_badfe1eba5214686b2accac29dff311a) 151

[PRINTER\_CHANGE\_ADD\_PRINT\_PROCESSOR](#section_badfe1eba5214686b2accac29dff311a) 151

[PRINTER\_CHANGE\_ADD\_PRINTER](#section_badfe1eba5214686b2accac29dff311a) 151

[PRINTER\_CHANGE\_ADD\_PRINTER\_DRIVER](#section_badfe1eba5214686b2accac29dff311a) 151

[PRINTER\_CHANGE\_ALL](#section_1d63f9efa4cd41e99152004fd2a8855a) 150

[PRINTER\_CHANGE\_ALL\_2](#section_1d63f9efa4cd41e99152004fd2a8855a) 150

[PRINTER\_CHANGE\_CONFIGURE\_PORT](#section_badfe1eba5214686b2accac29dff311a) 151

[PRINTER\_CHANGE\_DELETE\_FORM](#section_badfe1eba5214686b2accac29dff311a) 151

[PRINTER\_CHANGE\_DELETE\_JOB](#section_1d63f9efa4cd41e99152004fd2a8855a) 150

[PRINTER\_CHANGE\_DELETE\_PORT](#section_badfe1eba5214686b2accac29dff311a) 151

[PRINTER\_CHANGE\_DELETE\_PRINT\_PROCESSOR](#section_badfe1eba5214686b2accac29dff311a) 151

[PRINTER\_CHANGE\_DELETE\_PRINTER](#section_1d63f9efa4cd41e99152004fd2a8855a) 150

[PRINTER\_CHANGE\_DELETE\_PRINTER\_DRIVER](#section_badfe1eba5214686b2accac29dff311a) 151

[PRINTER\_CHANGE\_FAILED\_CONNECTION\_PRINTER](#section_badfe1eba5214686b2accac29dff311a) 151

[PRINTER\_CHANGE\_FORM](#section_badfe1eba5214686b2accac29dff311a) 151

[PRINTER\_CHANGE\_JOB](#section_1d63f9efa4cd41e99152004fd2a8855a) 150

[PRINTER\_CHANGE\_PORT](#section_badfe1eba5214686b2accac29dff311a) 151

[PRINTER\_CHANGE\_PRINT\_PROCESSOR](#section_badfe1eba5214686b2accac29dff311a) 151

[PRINTER\_CHANGE\_PRINTER](#section_1d63f9efa4cd41e99152004fd2a8855a) 150

[PRINTER\_CHANGE\_PRINTER\_DRIVER](#section_badfe1eba5214686b2accac29dff311a) 151

[PRINTER\_CHANGE\_SERVER](#section_badfe1eba5214686b2accac29dff311a) 151

[PRINTER\_CHANGE\_SET\_FORM](#section_badfe1eba5214686b2accac29dff311a) 151

[PRINTER\_CHANGE\_SET\_JOB](#section_1d63f9efa4cd41e99152004fd2a8855a) 150

[PRINTER\_CHANGE\_SET\_PRINTER](#section_1d63f9efa4cd41e99152004fd2a8855a) 150

[PRINTER\_CHANGE\_SET\_PRINTER\_DRIVER](#section_1d63f9efa4cd41e99152004fd2a8855a) 150

[PRINTER\_CHANGE\_TIMEOUT](#section_1d63f9efa4cd41e99152004fd2a8855a) 150

[PRINTER\_CHANGE\_WRITE\_JOB](#section_1d63f9efa4cd41e99152004fd2a8855a) 150

[PRINTER\_CONTAINER structure](#section_ad7d4d4e0feb45e093edb3b2b9a313d1) 39

[PRINTER\_ENUM\_CONNECTIONS](#section_2a1fe8a4e8be4cf38b378d19f9a2edcd) 152

[PRINTER\_ENUM\_CONTAINER](#section_2a1fe8a4e8be4cf38b378d19f9a2edcd) 152

[PRINTER\_ENUM\_EXPAND](#section_2a1fe8a4e8be4cf38b378d19f9a2edcd) 152

[PRINTER\_ENUM\_HIDE](#section_2a1fe8a4e8be4cf38b378d19f9a2edcd) 152

[PRINTER\_ENUM\_ICON1](#section_2a1fe8a4e8be4cf38b378d19f9a2edcd) 152

[PRINTER\_ENUM\_ICON2](#section_2a1fe8a4e8be4cf38b378d19f9a2edcd) 152

[PRINTER\_ENUM\_ICON3](#section_2a1fe8a4e8be4cf38b378d19f9a2edcd) 152

[PRINTER\_ENUM\_ICON8](#section_2a1fe8a4e8be4cf38b378d19f9a2edcd) 152

[PRINTER\_ENUM\_LOCAL](#section_2a1fe8a4e8be4cf38b378d19f9a2edcd) 152

[PRINTER\_ENUM\_NAME](#section_2a1fe8a4e8be4cf38b378d19f9a2edcd) 152

[PRINTER\_ENUM\_NETWORK](#section_2a1fe8a4e8be4cf38b378d19f9a2edcd) 152

[PRINTER\_ENUM\_REMOTE](#section_2a1fe8a4e8be4cf38b378d19f9a2edcd) 152

[PRINTER\_ENUM\_SHARED](#section_2a1fe8a4e8be4cf38b378d19f9a2edcd) 152

[PRINTER\_ENUM\_VALUES packet](#section_1daff8b90e2546d586a2a965ce67f74e) 133

[PRINTER\_EXECUTE](#section_945c4df5496949c5b1ce59c1b3f35024) 144

[PRINTER\_INFO](#section_125fb29b3c5141fca333b1e0e8794a91) 57

[PRINTER\_INFO members](#section_c07c2bfde2c94d369259354a229d645d) 46

[PRINTER\_INFO\_1 structure](#section_0d1a296b7027441c83938aa5c3764b27) 59

[PRINTER\_INFO\_2 structure](#section_d40734e1ac9b4b2a862883a1bc7bbeed) 59

[PRINTER\_INFO\_3 structure](#section_00af009b231e4ae6bfd7cb8476c32c19) 60

[PRINTER\_INFO\_4 structure](#section_ac391864cd9945099ff3ea0366dd9b2b) 60

[PRINTER\_INFO\_5 structure](#section_d7e5feaa16be4dbdb6d922f62f69afbb) 61

[PRINTER\_INFO\_6 structure](#section_436ef38839cd499295ac02ddd2a51193) 61

[PRINTER\_INFO\_7 structure](#section_91a6cb808df743d08a5d8f1d23f96de6) 61

[PRINTER\_INFO\_8 structure](#section_abe75b29b0924398a5af9d639f430cde) 62

[PRINTER\_INFO\_9 structure](#section_6a5df26640c94661966f27caa209b1fd) 62

[PRINTER\_INFO\_STRESS structure](#section_9e94b8f4a84b4c96b9ae4b4acb3469b8) 57

[PRINTER\_NOTIFY\_FIELD\_ATTRIBUTES](#section_41d5c622ec7749ff95e369b325ce4e77) 153

[PRINTER\_NOTIFY\_FIELD\_AVERAGE\_PPM](#section_41d5c622ec7749ff95e369b325ce4e77) 153

[PRINTER\_NOTIFY\_FIELD\_BRANCH\_OFFICE\_PRINTING](#section_41d5c622ec7749ff95e369b325ce4e77) 153

[PRINTER\_NOTIFY\_FIELD\_BYTES\_PRINTED](#section_41d5c622ec7749ff95e369b325ce4e77) 153

[PRINTER\_NOTIFY\_FIELD\_CJOBS](#section_41d5c622ec7749ff95e369b325ce4e77) 153

[PRINTER\_NOTIFY\_FIELD\_COMMENT](#section_41d5c622ec7749ff95e369b325ce4e77) 153

[PRINTER\_NOTIFY\_FIELD\_DATATYPE](#section_41d5c622ec7749ff95e369b325ce4e77) 153

[PRINTER\_NOTIFY\_FIELD\_DEFAULT\_PRIORITY](#section_41d5c622ec7749ff95e369b325ce4e77) 153

[PRINTER\_NOTIFY\_FIELD\_DEVMODE](#section_41d5c622ec7749ff95e369b325ce4e77) 153

[PRINTER\_NOTIFY\_FIELD\_DRIVER\_NAME](#section_41d5c622ec7749ff95e369b325ce4e77) 153

[PRINTER\_NOTIFY\_FIELD\_LOCATION](#section_41d5c622ec7749ff95e369b325ce4e77) 153

[PRINTER\_NOTIFY\_FIELD\_OBJECT\_GUID](#section_41d5c622ec7749ff95e369b325ce4e77) 153

[PRINTER\_NOTIFY\_FIELD\_PAGES\_PRINTED](#section_41d5c622ec7749ff95e369b325ce4e77) 153

[PRINTER\_NOTIFY\_FIELD\_PARAMETERS](#section_41d5c622ec7749ff95e369b325ce4e77) 153

[PRINTER\_NOTIFY\_FIELD\_PORT\_NAME](#section_41d5c622ec7749ff95e369b325ce4e77) 153

[PRINTER\_NOTIFY\_FIELD\_PRINT\_PROCESSOR](#section_41d5c622ec7749ff95e369b325ce4e77) 153

[PRINTER\_NOTIFY\_FIELD\_PRINTER\_NAME](#section_41d5c622ec7749ff95e369b325ce4e77) 153

[PRINTER\_NOTIFY\_FIELD\_PRIORITY](#section_41d5c622ec7749ff95e369b325ce4e77) 153

[PRINTER\_NOTIFY\_FIELD\_SECURITY\_DESCRIPTOR](#section_41d5c622ec7749ff95e369b325ce4e77) 153

[PRINTER\_NOTIFY\_FIELD\_SEPFILE](#section_41d5c622ec7749ff95e369b325ce4e77) 153

[PRINTER\_NOTIFY\_FIELD\_SERVER\_NAME](#section_41d5c622ec7749ff95e369b325ce4e77) 153

[PRINTER\_NOTIFY\_FIELD\_SHARE\_NAME](#section_41d5c622ec7749ff95e369b325ce4e77) 153

[PRINTER\_NOTIFY\_FIELD\_START\_TIME](#section_41d5c622ec7749ff95e369b325ce4e77) 153

[PRINTER\_NOTIFY\_FIELD\_STATUS](#section_41d5c622ec7749ff95e369b325ce4e77) 153

[PRINTER\_NOTIFY\_FIELD\_TOTAL\_BYTES](#section_41d5c622ec7749ff95e369b325ce4e77) 153

[PRINTER\_NOTIFY\_FIELD\_TOTAL\_PAGES](#section_41d5c622ec7749ff95e369b325ce4e77) 153

[PRINTER\_NOTIFY\_FIELD\_UNTIL\_TIME](#section_41d5c622ec7749ff95e369b325ce4e77) 153

[PRINTER\_NOTIFY\_INFO\_COLORMISMATCH](#section_033938dbcdc74541819510b7b217d96a) 146

[PRINTER\_NOTIFY\_INFO\_DISCARDED](#section_033938dbcdc74541819510b7b217d96a) 146

[PRINTER\_NOTIFY\_INFO\_DISCARDNOTED](#section_033938dbcdc74541819510b7b217d96a) 146

[PRINTER\_NOTIFY\_OPTIONS\_REFRESH](#section_033938dbcdc74541819510b7b217d96a) 146

[PRINTER\_READ](#section_945c4df5496949c5b1ce59c1b3f35024) 144

[PRINTER\_STATUS\_BUSY](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[PRINTER\_STATUS\_DOOR\_OPEN](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[PRINTER\_STATUS\_ERROR](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[PRINTER\_STATUS\_INITIALIZING](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[PRINTER\_STATUS\_IO\_ACTIVE](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[PRINTER\_STATUS\_MANUAL\_FEED](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[PRINTER\_STATUS\_NO\_TONER](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[PRINTER\_STATUS\_NOT\_AVAILABLE](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[PRINTER\_STATUS\_OFFLINE](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[PRINTER\_STATUS\_OUT\_OF\_MEMORY](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[PRINTER\_STATUS\_OUTPUT\_BIN\_FULL](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[PRINTER\_STATUS\_PAGE\_PUNT](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[PRINTER\_STATUS\_PAPER\_JAM](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[PRINTER\_STATUS\_PAPER\_OUT](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[PRINTER\_STATUS\_PAPER\_PROBLEM](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[PRINTER\_STATUS\_PAUSED](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[PRINTER\_STATUS\_PENDING\_DELETION](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[PRINTER\_STATUS\_POWER\_SAVE](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[PRINTER\_STATUS\_PRINTING](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[PRINTER\_STATUS\_PROCESSING](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[PRINTER\_STATUS\_SERVER\_OFFLINE](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[PRINTER\_STATUS\_SERVER\_UNKNOWN](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[PRINTER\_STATUS\_TONER\_LOW](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[PRINTER\_STATUS\_USER\_INTERVENTION](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[PRINTER\_STATUS\_WAITING](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[PRINTER\_STATUS\_WARMING\_UP](#section_1625e9d929e448f4b83d3bd0fdaea787) 165

[PRINTER\_WRITE](#section_945c4df5496949c5b1ce59c1b3f35024) 144

[PRINTPROCESSOR\_INFO\_1 packet](#section_cb458dcc1ae34b00864f6b2c369759e4) 133

[Product behavior](#section_e81cbc09ab054a32ae4a8ec57b436c43) 347

**R**

[READ\_CONTROL](#section_945c4df5496949c5b1ce59c1b3f35024) 144

[Receiving notifications on printing events example](#section_20bc2e9cce884fd399612a4f06ebe41f) 319

[RECTL structure](#section_d81607da4aa744b0ae4407d0c2227b34) 34

[References](#section_7f4274c6b4fc495fade513e8502a5f69) 21

[informative](#section_88d64aedfcbb49578ecef3eaac0a27b0) 22

[normative](#section_1249140046a74cee9ce70c8eb7ceac28) 21

[REG\_BINARY](#section_25cce7007fcf4bb6a2f30f6d08430a55) 156

[REG\_DWORD](#section_25cce7007fcf4bb6a2f30f6d08430a55) 156

[REG\_DWORD\_BIG\_ENDIAN](#section_25cce7007fcf4bb6a2f30f6d08430a55) 156

[REG\_DWORD\_LITTLE\_ENDIAN](#section_25cce7007fcf4bb6a2f30f6d08430a55) 156

[REG\_EXPAND\_SZ](#section_25cce7007fcf4bb6a2f30f6d08430a55) 156

[REG\_LINK](#section_25cce7007fcf4bb6a2f30f6d08430a55) 156

[REG\_MULTI\_SZ](#section_25cce7007fcf4bb6a2f30f6d08430a55) 156

[REG\_NONE](#section_25cce7007fcf4bb6a2f30f6d08430a55) 156

[REG\_QWORD](#section_25cce7007fcf4bb6a2f30f6d08430a55) 156

[REG\_QWORD\_LITTLE\_ENDIAN](#section_25cce7007fcf4bb6a2f30f6d08430a55) 156

[REG\_RESOURCE\_LIST](#section_25cce7007fcf4bb6a2f30f6d08430a55) 156

[REG\_SZ](#section_25cce7007fcf4bb6a2f30f6d08430a55) 156

[Relationship to other protocols](#section_dadadc2d0dcd47e0b78f13d32875a126) 27

[RPC\_BIDI\_DATA structure](#section_d6e679dc5a244936ac6858251dbd1555) 64

[RPC\_BIDI\_REQUEST\_CONTAINER structure](#section_4ea856c44c374485868ae7ac8e331e5e) 40

[RPC\_BIDI\_REQUEST\_DATA structure](#section_c3b78db75fec440584ff412e82d491f1) 64

[RPC\_BIDI\_RESPONSE\_CONTAINER structure](#section_bdfdec23d5034f86a8486c3386d609d6) 40

[RPC\_BIDI\_RESPONSE\_DATA structure](#section_f42d17bd05d643a4872e6e6513ee0922) 64

[RPC\_BINARY\_CONTAINER structure](#section_bc9f774d66fc45209563091553c3cabd) 41

[RPC\_BranchOfficeJobData structure](#section_6e442be62c54497593c685895289ba0d) 71

[RPC\_BranchOfficeJobDataContainer structure](#section_8bed3226cdef4b90b762b35f1a3ad362) 42

[RPC\_BranchOfficeJobDataError structure](#section_d86fd29b95124c80a14b5f5fb25e5975) 72

[RPC\_BranchOfficeJobDataPipelineFailed structure](#section_31a2aecb2d7a4846a93be4fd708863e0) 73

[RPC\_BranchOfficeJobDataPrinted structure](#section_b4a220487837460eb1bd8d34f4270f29) 73

[RPC\_BranchOfficeJobDataRendered structure](#section_e90ee32f12f44310923871aca7dd9cb1) 74

[RPC\_BranchOfficeLogOfflineFileFull structure](#section_55fcc7c1d63e448888cd8ce5d9dd945c) 74

[RPC\_DRIVER\_INFO members](#section_4464eaf0f34f40d5b970736437a21913) 43

[RPC\_DRIVER\_INFO\_3 structure](#section_b4d3b5f3c6314fc393484c983590a0d8) 48

[RPC\_DRIVER\_INFO\_4 structure](#section_459d24e40b754fc7b246fe40a1981eed) 48

[RPC\_DRIVER\_INFO\_6 structure](#section_ae12474015ef482c824e56e5436ed44b) 49

[RPC\_DRIVER\_INFO\_8 structure](#section_937d5fc867f145f8ab878b56a8fca558) 49

[RPC\_EPrintPropertyType enumeration](#section_0360160c35514c5bbcee9164122d34e2) 70

[RPC\_FORM\_INFO members](#section_84fe09faea4b4cc58ae49c8582fd1ffe) 44

[RPC\_FORM\_INFO\_2 structure](#section_dca49b4f3eb146569e972f0b0c569bf6) 51

[RPC\_PrintNamedProperty structure](#section_31bdf5289b8947f9823ba45bbc877716) 69

[RPC\_PrintPropertyValue structure](#section_2ed8404520264696b2e263dbffdc6d21) 69

[RPC\_V2\_NOTIFY\_INFO structure](#section_79db335febe74a6f8533efdda32cc5d8) 66

[RPC\_V2\_NOTIFY\_INFO\_DATA structure](#section_440ed0c046a4420193cdcc1e5d1fd022) 67

[RPC\_V2\_NOTIFY\_OPTIONS structure](#section_4599c052e81441749bd500473329a9f8) 65

[RPC\_V2\_NOTIFY\_OPTIONS\_TYPE structure](#section_3cce2507707d4a8e8ce8388463a24952) 66

[RpcAbortPrinter method](#section_924cd448b8b74dbaaa56d9d3244cd386) 282

[RpcAddForm method](#section_d6eb54304a274e5ab6988140a12fa889) 260

[RpcAddJob method](#section_7ba3267737294b5f96b5c8a9f99d34a5) 244

[RpcAddMonitor method](#section_dee4feea274940bdab7cc1c0ca1fbcf8) 272

[RpcAddPerMachineConnection method](#section_3648c131b1554cc69a2d79b95b422a5b) 234

[RpcAddPortEx method](#section_1303a48b47724226a8b8c495d255304e) 267

[RpcAddPrinter method](#section_bd0bb8d526fd470581f5614d32b9f38f) 210

[RpcAddPrinterDriver method](#section_f23a75191c7740699acea6d8eae47c22) 246

[RpcAddPrinterDriverEx method](#section_b96cc49759e54510ab045484993b259b) 255

[RpcAddPrinterEx method](#section_3af45717308f4b4f8428430d12856110) 224

[RpcAddPrintProcessor method](#section_d7e6fdb25256483592c127aa6f76758a) 274

[RpcClosePrinter method](#section_8482c18677734fb58034e9bae7a91e52) 218

[RpcCreatePrinterIC method](#section_b0686ba23a6b4a6c943fb26600dbd88c) 219

[RpcDeleteForm method](#section_1b25a56720084ca1ac93feaeaec7ec54) 261

[RpcDeleteJobNamedProperty method](#section_02b59e33f1534870af67199267dc0b64) 299

[RpcDeleteMonitor method](#section_2483519c230c4e8bb52d29677d941ad8) 272

[RpcDeletePerMachineConnection method](#section_8861fe40b2764c778ef7cc6a28063e0e) 236

[RpcDeletePort method](#section_cb13f514f95a4d00936bf934641b6a4b) 266

[RpcDeletePrinter method](#section_e46c727493db443e8787ad49e5a08466) 212

[RpcDeletePrinterData method](#section_a67521b7c92d4ebf900836fb321196ff) 227

[RpcDeletePrinterDataEx method](#section_19c2e0e5b7ec4d068b211488f2275505) 233

[RpcDeletePrinterDriver method](#section_e03b14adabe84158ab0ce0033d820116) 251

[RpcDeletePrinterDriverEx method](#section_21d247cd66e24373a187b858c907e13f) 253

[RpcDeletePrinterIC method](#section_d9792de2d19c4c9bb7a4034523b7784b) 221

[RpcDeletePrinterKey method](#section_cde0dcc363264509ad3fe170977931ba) 234

[RpcDeletePrintProcessor method](#section_e08179d3d3c3421bab77a0ba50a4254e) 276

[RpcEndDocPrinter method](#section_209fcf3205b444aba3a4779cfae181ca) 284

[RpcEndPagePrinter method](#section_c4a17128584d402e9bac6a050e9da6d0) 282

[RpcEnumForms method](#section_0b9dde1473d14f1a9f748e82255a9612) 263

[RpcEnumJobNamedProperties method](#section_6b1559f993494d90b8cf24292ab736ad) 300

[RpcEnumJobs method](#section_0c77f1ff08274f4da06d5a3d802ffc0c) 243

[RpcEnumMonitors method](#section_6c5836fed9ea41a0ad7e01152f39e8a1) 271

[RpcEnumPerMachineConnections method](#section_3b1e7385d1ca48dfb372461c524f78d9) 236

[RpcEnumPorts method](#section_4f018733e78e4088ac48f6e19ee00863) 265

[RpcEnumPrinterData method](#section_51bfa051e7e44fc4b79eea9d4b08cb84) 226

[RpcEnumPrinterDataEx method](#section_9574b3e74a294bfaba67c35dab674162) 231

[RpcEnumPrinterDrivers method](#section_857d00ac36824a0d86ca3d3c372e5e4a) 247

[RpcEnumPrinterKey method](#section_128ff57da4a44a23b2ee36f71ca5c0d5) 232

[RpcEnumPrinters method](#section_0d83c60dec7546718a3b99517651ff1b) 207

[RpcEnumPrintProcessorDatatypes method](#section_c4a35c2040824ce6840102c52ee3158e) 277

[RpcEnumPrintProcessors method](#section_8c3c2db185d84d8894c11e09cda435c7) 275

[RpcFindClosePrinterChangeNotification method](#section_f50c8a861b2d4124b2205b840aacd2c7) 287

[RpcFlushPrinter method](#section_be7ec83d52eb42e9bc5d7244e5b44b94) 285

[RpcGetCorePrinterDrivers method](#section_3d6282322aa944f484e01168d3208f27) 257

[RpcGetForm method](#section_be369d1d1ede49b181c209c69411c25d) 261

[RpcGetJob method](#section_f7090127287b4aa7ba8b2b6b7247dfc6) 242

[RpcGetJobNamedPropertyValue method](#section_64d3b0c08eef4a2b859960904de7c823) 297

[RpcGetPrinter method](#section_ccbf288a0768416dbede0ef1987fc286) 215

[RpcGetPrinterData method](#section_9c04a0f2adc443938df47c36a14183f5) 216

[RpcGetPrinterDataEx method](#section_74a9f79910bf4ef6a2a40dcd9c43fb5c) 229

[RpcGetPrinterDriver method](#section_00d91852785d42e4a774b1d96361b0ce) 249

[RpcGetPrinterDriver2 method](#section_a73e77d07b2448cfa6d38b4061749495) 251

[RpcGetPrinterDriverDirectory method](#section_9df11cf440984852ad72d1f75a82bffe) 250

[RpcGetPrinterDriverPackagePath method](#section_e453b11350c64a129ca6e327b5accca6) 258

[RpcGetPrintProcessorDirectory method](#section_17eaee3be5754f82855ce60657d5b2c8) 275

[RpcLogJobInfoForBranchOffice method](#section_65a54d4df30b41438299d61f086ac7fe) 301

[RpcOpenPrinter method](#section_989357e2446e4872bb381dce21e1313f) 208

[RpcOpenPrinterEx method](#section_0e81ce1872b146c38584a205393b04ff) 222

[RpcPlayGdiScriptOnPrinterIC method](#section_eff167adf4004c6eb0e2c48b62ac0826) 220

[RpcReadPrinter method](#section_58fc87857f004f00bc784c3b438acbb7) 283

[RpcRemoteFindFirstPrinterChangeNotification method](#section_b8b414d9f1cd4191bb6b87d09ab2fd83) 288

[RpcRemoteFindFirstPrinterChangeNotificationEx method](#section_eb66b2211c1f4249b8bcc5befec2314d) 289

[RpcReplyClosePrinter method](#section_d6c73ecbc02b4150a01bad73a642990a) 305

[RpcReplyOpenPrinter method](#section_7fcd3036d45a4ec7b081f2b860e66676) 304

[RpcResetPrinter method](#section_6d979ef765e848af94c76bbb9d2be06a) 222

[RpcRouterRefreshPrinterChangeNotification method](#section_cb5d22ed0352430383d5b5afd32fb461) 291

[RpcRouterReplyPrinter method](#section_38bf5d2ce4a744b2b16f91ff8226f836) 305

[RpcRouterReplyPrinterEx method](#section_1b44cb8295b5485c9ec90360c3a3ecb0) 306

[RpcScheduleJob method](#section_4d6454a7f7d44d308ba18644cb6b2186) 245

[RpcSendRecvBidiData method](#section_5aec83472ebf464eb6d1540aa60b5c71) 237

[RpcSetForm method](#section_c17cbf6601a24ac0b57b1d2d8d1f73de) 262

[RpcSetJob method](#section_f978bd2ce99a43ffba0802e892948dac) 239

[RpcSetJobNamedProperty method](#section_a08c060ad4e24a88a4874f6aaecfcf09) 298

[RpcSetPort method](#section_b0a0984e6c724dc8b8c9db98276c94cd) 268

[RpcSetPrinter method](#section_382997b5b02a4b6a8c6849a64b8c45f8) 212

[RpcSetPrinterData method](#section_e0d3e41e097c459da7fbacaf1343ea27) 217

[RpcSetPrinterDataEx method](#section_58520b8b44ce46fc90d30362457098e0) 228

[RpcStartDocPrinter method](#section_90f95ff03ae04ddea83d35f457c710a6) 279

[RpcStartPagePrinter method](#section_76261f544c634fc2a2dabaee584e076b) 280

[RpcWaitForPrinterChange method](#section_e40cb68f679e4b84a33e86a94a141d83) 286

[RpcWritePrinter method](#section_6dd263aac8b547748d211daa0d0d9920) 281

[RpcXcvData method](#section_bf9d2351aac0481e9d3705eb5cc961da) 268

**S**

Security

[implementer considerations](#section_fc2fbb2b10be440090efeb748090ade4) 322

[parameter index](#section_1586a2e8c4cb4ed0a1325fecf217fea5) 322

[SECURITY\_CONTAINER structure](#section_6cf2b4ff1b2b453c87f985243823cea8) 41

Sequencing rules

[client](#section_0f30f3700d7f43df922737f9155cf821) 303

[server](#section_700f007d72b74b44951756569704bb8d) 184

Server

[abstract data model](#section_1bd61448db4b4284a4b18346732402d4) 180

[Branch Office Print Remote Logging Methods method](#section_fd04e0de264746e0a06aa3ccb138dbb6) 301

[Commonly Used Parameters method](#section_7eb63f7a637a4b7582e091853f6f6013) 194

[Document Printing Methods method](#section_93e6c5a794d6487ebe20d2c91e5277ad) 278

[Form Management Methods method](#section_0b9ceec4c17747aea4bea5d3167fc205) 260

[initialization](#section_50cafaeac64b4022b27b9c28d7b799ab) 184

[Job Management Methods method](#section_bd8e1517a5b14850933866575d2b8cf4) 239

[Job Named Property Management Methods method](#section_95aa7a44c15841a0b2ad2a08ccd0881f) 297

[local events](#section_06246c515560435ab514b180917584db) 302

[message processing](#section_700f007d72b74b44951756569704bb8d) 184

[Monitor Module Methods method](#section_07ce205123c64d508f55c25d7ac09cb3) 292

[Notification Methods method](#section_72b88737108f4e398fa9ee3af45da015) 286

[Port Management Methods method](#section_1cf0e23868da43bdbaba44b35b115cc0) 264

[Port Monitor Management Methods method](#section_2c505550d9b24123bfa9ae204f4d88d6) 270

[Print Processor Management Methods method](#section_701f22dc32934ef28aa64d03ad4ef868) 273

[Printer Driver Management Methods method](#section_84e21d0dd5da4db2ab19d7ece51e42aa) 245

[Printer Management and Discovery Methods method](#section_5ea67df3c4e745efb42540b8b066d276) 205

[sequencing rules](#section_700f007d72b74b44951756569704bb8d) 184

[timer events](#section_615e516d7d8e4a50a0a62607a4401500) 302

[timers](#section_c3e345350fb34dad966d297b48703bb8) 184

[SERVER\_ACCESS\_ADMINISTER](#section_945c4df5496949c5b1ce59c1b3f35024) 144

[SERVER\_ACCESS\_ENUMERATE](#section_945c4df5496949c5b1ce59c1b3f35024) 144

[SERVER\_ALL\_ACCESS](#section_945c4df5496949c5b1ce59c1b3f35024) 144

[SERVER\_EXECUTE](#section_945c4df5496949c5b1ce59c1b3f35024) 144

[SERVER\_NOTIFY\_FIELD\_PRINT\_DRIVER\_ISOLATION\_GROUP](#section_61d9014b16e44e5e9f12723392c6ab66) 149

[SERVER\_READ](#section_945c4df5496949c5b1ce59c1b3f35024) 144

[SERVER\_WRITE](#section_945c4df5496949c5b1ce59c1b3f35024) 144

[SIZE structure](#section_c76b61d3166543f59584a40d42bf05c5) 34

[SPECIFIC\_RIGHTS\_ALL](#section_945c4df5496949c5b1ce59c1b3f35024) 144

[SPLCLIENT\_CONTAINER structure](#section_cca15c7257d342fc91e749442b623c11) 41

[SPLCLIENT\_INFO](#section_d23fee1a3f624cbb8a06d01180fe2ff9) 63

[SPLCLIENT\_INFO members](#section_fae766fc31d343cc902f8e6cc4c75f31) 47

[SPLCLIENT\_INFO\_1 structure](#section_dd99a589aecd4c259111ded01b10f347) 63

[SPLCLIENT\_INFO\_2 structure](#section_4459a08bec5f46a482534893a49dd4f8) 63

[SPLCLIENT\_INFO\_3 structure](#section_f0e50c0a19a14297bdcf15e2ada9525f) 63

[STANDARD\_RIGHTS\_ALL](#section_945c4df5496949c5b1ce59c1b3f35024) 144

[STANDARD\_RIGHTS\_EXECUTE](#section_945c4df5496949c5b1ce59c1b3f35024) 144

[STANDARD\_RIGHTS\_READ](#section_945c4df5496949c5b1ce59c1b3f35024) 144

[STANDARD\_RIGHTS\_REQUIRED](#section_945c4df5496949c5b1ce59c1b3f35024) 144

[STANDARD\_RIGHTS\_WRITE](#section_945c4df5496949c5b1ce59c1b3f35024) 144

[Standards assignments](#section_848b8334134a4d02aea403b673d6c515) 29

[STRING\_CONTAINER structure](#section_5f6aaa8dad83427892fa2afaaf8b75fc) 42

Structures ([section 2.2.2.14](#section_c6430e00b96a4e58b0f467d17c4adeb5) 136, [section 2.2.2.15](#section_6e2abe47b5c24c43a2328fca3897287f) 141)

[SYNCHRONIZE](#section_945c4df5496949c5b1ce59c1b3f35024) 144

[SYSTEMTIME\_CONTAINER structure](#section_1438066b4b56488898213e2432dc0f07) 42

**T**

[TABLE\_DEVMODE](#section_6e1ac55984a34b3ea89707159974661e) 150

[TABLE\_DWORD](#section_6e1ac55984a34b3ea89707159974661e) 150

[TABLE\_SECURITYDESCRIPTOR](#section_6e1ac55984a34b3ea89707159974661e) 150

[TABLE\_STRING](#section_6e1ac55984a34b3ea89707159974661e) 150

[TABLE\_TIME](#section_6e1ac55984a34b3ea89707159974661e) 150

[Tcpmon](#section_4f6172c708314ca6a37472c841f27c7c) 294

[THREAD\_PRIORITY\_ABOVE\_NORMAL](#section_f0f64e02a00a4b72832f4cc9a035d8de) 162

[THREAD\_PRIORITY\_BELOW\_NORMAL](#section_f0f64e02a00a4b72832f4cc9a035d8de) 162

[THREAD\_PRIORITY\_HIGHEST](#section_f0f64e02a00a4b72832f4cc9a035d8de) 162

[THREAD\_PRIORITY\_LOWEST](#section_f0f64e02a00a4b72832f4cc9a035d8de) 162

[THREAD\_PRIORITY\_NORMAL](#section_f0f64e02a00a4b72832f4cc9a035d8de) 162

Timer events

[client](#section_d640a376b59c40239761e49a0db24431) 310

[server](#section_615e516d7d8e4a50a0a62607a4401500) 302

Timers

[client](#section_4b32ff7006804aeca52049a6ec47b36f) 303

[server](#section_c3e345350fb34dad966d297b48703bb8) 184

[Tracking changes](#section_32b7aecb8a7a48bdae1d35cac60bff8d) 482

[Transport](#section_fdf1138af6b94b00ada61fbb6097d683) 30

**U**

[UNIVERSAL\_FONT\_ID packet](#section_97cd8dd84c4b4a7aa160fae63f3e59bf) 135

**V**

[Vendor-extensible members](#section_39112098aa6d440ab661a5061dfd100d) 29

[Versioning](#section_0c08f94330bd4653983342447e5093d5) 28

**W**

[WRITE\_DAC](#section_945c4df5496949c5b1ce59c1b3f35024) 144

[WRITE\_OWNER](#section_945c4df5496949c5b1ce59c1b3f35024) 144

[WSD\_BACKUP\_PORT\_DATA](#section_45f18ad3e1a44c44a7800dc2f0422914) 142

[WSD\_BACKUP\_PORT\_DATA packet](#section_45f18ad3e1a44c44a7800dc2f0422914) 142

[WSD\_BACKUP\_PORT\_DATA\_EX packet](#section_226099c983cb426097e1f796444b4502) 142

[WSD\_DRIVER\_DATA](#section_3a887ca485f94dd3b83f6794a0b55353) 141

[WSD\_DRIVER\_DATA packet](#section_3a887ca485f94dd3b83f6794a0b55353) 141

[Wsdmon](#section_e1760a13634e4a7fa7dd9d02e4048f9b) 295