



System i  
Programming  
Print APIs

*Version 6 Release 1*







System i  
Programming  
Print APIs

*Version 6 Release 1*

**Note**

Before using this information and the product it supports, read the information in "Notices," on page 439.

This edition applies to version 6, release 1, modification 0 of IBM i5/OS (product number 5761-SS1) and to all subsequent releases and modifications until otherwise indicated in new editions. This version does not run on all reduced instruction set computer (RISC) models nor does it run on CISC models.

© **Copyright International Business Machines Corporation 1998, 2008.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

---

# Contents

<b>Print APIs . . . . .</b>	<b>1</b>
APIs . . . . .	1
General Print APIs . . . . .	1
Add PDF Map Entry (QPQAPME) API . . . . .	3
Authorities and Locks . . . . .	4
Required Parameter Group . . . . .	4
APME0100 Format . . . . .	6
APME0100 Format Field Descriptions . . . . .	6
PMAP0100 Format . . . . .	8
PMAP0100 Field Descriptions. . . . .	9
Mail Information Entry . . . . .	10
Mail Information Entry Field Descriptions . . . . .	11
E-mail Address Entry . . . . .	17
E-mail Address Entry Field Description . . . . .	17
Stream File Entry . . . . .	17
Stream File Entry Field Description . . . . .	17
File Name Entry. . . . .	17
File Name Entry Field Description. . . . .	18
PDF Stream File Information Entry . . . . .	18
PDF Stream File Information Entry Field Descriptions . . . . .	18
PDF Spooled File Information Entry . . . . .	19
AFPDS Spooled File Information Entry . . . . .	19
PDF and AFPDS Spooled File Information Entry Field Descriptions . . . . .	19
Error Messages . . . . .	20
Build Open Time Commands (QSPBOPNC) API . . . . .	21
Authorities and Locks . . . . .	21
Required Parameter Group . . . . .	21
OPNC0100 Format . . . . .	22
Field Descriptions . . . . .	22
Error Messages . . . . .	22
Build Separator Pages (QSPBSEPP) API . . . . .	23
Authorities and Locks . . . . .	23
Required Parameter Group . . . . .	23
SEPP0100 Format . . . . .	25
Field Descriptions . . . . .	25
Error Messages . . . . .	26
Change Output Queue (QSPCHGOQ) API . . . . .	26
Authorities and Locks . . . . .	26
Required Parameter Group . . . . .	27
Format of the Variable Length Record . . . . .	27
Valid Keys. . . . .	28
Field Descriptions . . . . .	28
Error Messages . . . . .	30
Convert Image (QIMGCVTI, QimgCvtImg) API . . . . .	30
Authorities and Locks . . . . .	31
Required Parameter Group . . . . .	32
Control Structure Format . . . . .	33
Field Descriptions . . . . .	33
Input Image Structure Format . . . . .	36
Field Descriptions . . . . .	38
Output Image Structure Format . . . . .	41
Field Descriptions . . . . .	45
Feedback Structure Format . . . . .	51
Field Descriptions . . . . .	52

Multipage Output . . . . .	53
Color Reduction Field . . . . .	54
Values of Destination Image Configuration . . . . .	54
Error Messages . . . . .	57
Copy AFPDS Resource (QGSCPYSR) API . . . . .	60
Authorities and Locks . . . . .	60
Required Parameter Group . . . . .	60
Format of the User Space. . . . .	61
Input Parameter Section . . . . .	62
Field Descriptions . . . . .	63
Header Section . . . . .	63
Field Descriptions . . . . .	63
Resource Information Section . . . . .	64
Field Descriptions . . . . .	64
Resource Data Stream Section . . . . .	64
Error Messages . . . . .	64
Create PDF Map (QPQCRTPM) API . . . . .	65
Authorities and Locks . . . . .	65
Required Parameter Group . . . . .	65
Error Messages . . . . .	66
Delete PDF map object (QPQDLTPM) API . . . . .	66
Authorities and Locks . . . . .	67
Required Parameter Group . . . . .	67
Error Messages . . . . .	67
Extract Writer Status (QSPEXTWI) API . . . . .	68
Authorities and Locks . . . . .	68
Required Parameter Group . . . . .	68
EXTW0100 Format . . . . .	69
Field Descriptions . . . . .	69
Error Messages . . . . .	71
Host Print Transform (QWPZHPTR, QwpzHostPrintTransform) API . . . . .	71
Authorities and Locks . . . . .	72
Required Parameter Group . . . . .	72
Option Specific Input Information . . . . .	75
Field Descriptions . . . . .	76
Option Specific Output Information . . . . .	82
Field Descriptions . . . . .	82
Error Messages . . . . .	86
Open List of AFP Information (QGYOLAFP) API. . . . .	87
Authorities and Locks . . . . .	87
Required Parameter Group . . . . .	87
CFNI0100 Format . . . . .	90
FCSI0100 Format . . . . .	90
Field Descriptions . . . . .	90
PHCS0100 Format . . . . .	90
HPCS0100 Format . . . . .	91
PHCP0100 Format . . . . .	91
HPCP0100 Format . . . . .	92
PPCS0100 Format . . . . .	92
Field Descriptions . . . . .	92
Error Messages . . . . .	94
Open List of Output Queues (QSPOLOTQ) API . . . . .	95
Performance Impacts . . . . .	95
Authorities and Locks . . . . .	95
Required Parameter Group . . . . .	95

Filter Information . . . . .	96	FTBI0100 Format . . . . .	145
Format of Sort Information . . . . .	96	PHCS0100 Format . . . . .	146
Field Descriptions . . . . .	97	HPCS0100 Format . . . . .	146
Format of Receiver Variable . . . . .	98	PHCP0100 Format . . . . .	147
Field Descriptions . . . . .	98	HPCP0100 Format . . . . .	147
Error Messages . . . . .	99	PPCS0100 Format . . . . .	148
Open List of PDF Map Information (QPQOLPM)		Field Descriptions . . . . .	148
API . . . . .	99	Error Messages . . . . .	151
Authorities and Locks . . . . .	100	Retrieve Output Queue Information (QSPROUTQ)	
Required Parameter Group . . . . .	100	API . . . . .	152
PDFM0100 Format . . . . .	101	Authorities and Locks . . . . .	152
PDFM0100 Format Field Descriptions . . . . .	102	Required Parameter Group . . . . .	152
PDFM0200 Format . . . . .	103	OUTQ0100 Format . . . . .	153
PDFM0200 Format Field Descriptions . . . . .	104	OUTQ0200 Format . . . . .	154
Mail Information Entry . . . . .	106	Format of Spooled File Maximum Page Entry . . . . .	156
Mail Information Entry Field Descriptions . . . . .	107	Format of Writer Entry . . . . .	156
E-mail Address Entry . . . . .	112	Format of User-Defined Option Entry . . . . .	157
E-mail Address Entry Field Description . . . . .	112	Field Descriptions . . . . .	157
Stream File Entry . . . . .	112	Error Messages . . . . .	166
Stream File Entry Field Description . . . . .	113	Retrieve Printer Attributes (QGYRPRTA) API . . . . .	166
PDF Stream File Information Entry . . . . .	113	Authorities and Locks . . . . .	166
PDF Stream File Information Entry Field		Required Parameter Group . . . . .	166
Descriptions . . . . .	113	Format of Receiver Variable . . . . .	167
PDF Spooled File Information Entry . . . . .	114	RPTA0100 Format . . . . .	167
AFPDS Spooled File Information Entry . . . . .	114	Field Descriptions . . . . .	168
PDF and AFPDS Spooled File Information Entry		Error Messages . . . . .	175
Field Descriptions . . . . .	114	Retrieve Writer Information (QSPRWTRI) API . . . . .	176
File Name Entry . . . . .	115	Authorities and Locks . . . . .	176
File Name Entry Field Description . . . . .	115	Required Parameter Group . . . . .	176
FLTR0100 Format . . . . .	115	Optional Parameter . . . . .	177
FLTR0100 Format Field Descriptions . . . . .	116	WTRI0100 Format . . . . .	177
Error Messages . . . . .	118	Field Descriptions . . . . .	179
Open List of Printers (QGYRPRTL) API . . . . .	118	Error Messages . . . . .	184
Authorities and Locks . . . . .	119	Send Writer Message (QSPSNDWM) API . . . . .	184
Required Parameter Group . . . . .	119	Authorities and Locks . . . . .	185
Format of Receiver Variable . . . . .	120	Required Parameter Group . . . . .	185
Field Descriptions . . . . .	120	SNDM0100 Format . . . . .	187
Format of Filter Information . . . . .	123	Field Descriptions . . . . .	187
Field Descriptions . . . . .	123	Error Messages . . . . .	188
Error Messages . . . . .	123	Set Writer Status (QSPSETWI) API . . . . .	188
Publish Print Information (QSPPUBPI) API . . . . .	124	Authorities and Locks . . . . .	188
Authorities and Locks . . . . .	124	Required Parameter Group . . . . .	188
Required Parameter Group . . . . .	124	SETW0100 Format . . . . .	189
Printer Information Formats . . . . .	125	Field Descriptions . . . . .	190
Field Descriptions . . . . .	125	Error Messages . . . . .	192
Error Messages . . . . .	125	Transform AFP to ASCII (QWPZTAFP) API . . . . .	193
Remove PDF Map Object Entries (QPQRPME) API . . . . .	126	Authorities and Locks . . . . .	193
Authorities and Locks . . . . .	126	Required Parameter Group . . . . .	194
Required Parameter Group . . . . .	126	Output Controls Table Format . . . . .	194
Format for Entry Lengths and Entry Offsets . . . . .	128	Field Descriptions . . . . .	195
Remove Criteria Format . . . . .	128	Error Messages . . . . .	196
Field Descriptions . . . . .	128	Spooled File APIs . . . . .	196
Error Messages . . . . .	130	Close Spooled File (QSPCLOSP) API . . . . .	198
Retrieve AFP Information (QPQRAFPI) API . . . . .	130	Authorities and Locks . . . . .	198
Authorities and Locks . . . . .	131	Required Parameter Group . . . . .	198
Required Parameter Group . . . . .	131	Error Messages . . . . .	198
PSFI0100 Format . . . . .	133	Create Spooled File (QSPCRTSP) API . . . . .	198
Field Descriptions . . . . .	135	Authorities and Locks . . . . .	199
CFNI0100 Format . . . . .	143	Required Parameter Group . . . . .	199
FCSI0100 Format . . . . .	144	Considerations Using the QSPCRTSP API . . . . .	199
Field Descriptions . . . . .	144	System Values . . . . .	199

New Spooled File Ownership . . . . .	199	Optional Parameter Group 1 . . . . .	282
New Spooled File Placement . . . . .	200	Optional Parameter Group 2 . . . . .	282
Error Messages . . . . .	200	Optional Parameter Group 3 . . . . .	283
SPLA0200 Format Fields. . . . .	200	Optional Parameter Group 4 . . . . .	283
Field Descriptions . . . . .	216	Optional Parameter Group 5 . . . . .	285
Error Messages . . . . .	248	Format of the Generated List . . . . .	286
Get Spooled File Data (QSPGETSP) API . . . . .	249	Format SPLF0100 . . . . .	288
Authorities and Locks . . . . .	249	Format SPLF0200 . . . . .	288
Required Parameter Group . . . . .	250	Format SPLF0300 . . . . .	289
Format of the User Space . . . . .	251	Format SPLF0400 . . . . .	290
Generic Header Section . . . . .	252	Field Descriptions . . . . .	290
Field Descriptions . . . . .	253	Error Messages . . . . .	295
Buffer Information Section . . . . .	254	Move Spooled File (QSPMOVSP) API . . . . .	296
Field Descriptions . . . . .	254	Restrictions for Movement of Spooled Files . . . . .	297
General Information Section . . . . .	255	Authorities and Locks . . . . .	298
Field Descriptions . . . . .	256	Output Queue Authorities . . . . .	298
Page Data Section . . . . .	257	Spooled File Authorities . . . . .	298
Field Descriptions . . . . .	257	User Profile Highest Schedule Priority . . . . .	299
Print Data Section . . . . .	257	Required Parameter Group . . . . .	299
Field Descriptions . . . . .	259	Format of the Function Information . . . . .	300
Restrictions for Print Data Section . . . . .	260	MSPF0100 Format . . . . .	300
Error Messages . . . . .	260	MSPF0200 Format . . . . .	300
List Spooled File AFPDS Resources (QGSLRSC) API . . . . .	261	Field Descriptions . . . . .	301
How QGSLRSC Resolves to the AFPDS Resource Libraries . . . . .	261	How to Specify Spooled File Identifying Fields . . . . .	306
How QGSLRSC resolves to integrated file system imbedded object resources . . . . .	262	Error Messages . . . . .	307
How QGSLRSC Resolves to integrated file system font resources. . . . .	262	Open List of Spooled Files (QGYOLSPL) API. . . . .	308
Authorities and Locks . . . . .	262	Differences between QUSLSPL and QGYOLSPL . . . . .	309
Required Parameter Group . . . . .	263	Performance Impacts . . . . .	309
Optional Parameter Group 1 . . . . .	265	Authorities and Locks . . . . .	309
Optional Parameter Group 2 . . . . .	266	Required Parameter Group . . . . .	309
Format of the Generated List . . . . .	267	Optional Parameter Group . . . . .	311
Input Parameter Section . . . . .	267	Format of Receiver Variable . . . . .	311
Field Descriptions . . . . .	268	Format of Filter Information . . . . .	314
Header Section . . . . .	269	User Name Entry . . . . .	315
Field Descriptions . . . . .	269	Output Queue Name Entry . . . . .	316
Format LRSC0100 . . . . .	270	Spooled File Status Entry . . . . .	316
Field Descriptions . . . . .	270	Printer Device Name Entry . . . . .	316
Format LRSC0200 . . . . .	271	Format of Sort Information . . . . .	316
Field Descriptions . . . . .	271	Field Descriptions . . . . .	317
Format LRSC0300 . . . . .	271	Error Messages . . . . .	324
Field Descriptions . . . . .	272	Open Spooled File (QSPOPNSP) API . . . . .	325
Format LRSC0400 . . . . .	272	Authorities and Locks . . . . .	325
Field Descriptions . . . . .	273	Required Parameter Group . . . . .	325
Format LRSE0100 . . . . .	273	Optional Parameter Group . . . . .	327
Field Descriptions . . . . .	273	How to Select a Spooled File to Be Opened . . . . .	328
Format LRSE0200 . . . . .	274	Error Messages . . . . .	329
Field Descriptions . . . . .	275	Put Spooled File Data (QSPPUTSP) API . . . . .	330
Format LRSS0100 . . . . .	276	Authorities and Locks . . . . .	330
Field Descriptions . . . . .	276	Required Parameter Group . . . . .	330
Format LRSS0200 . . . . .	276	Considerations When Changing or Creating a User Space . . . . .	331
Field Descriptions . . . . .	277	Fields in the General Information Section . . . . .	331
Format LRSC0500 . . . . .	277	Fields in the Page Data Section . . . . .	332
Field Descriptions . . . . .	277	Error Messages . . . . .	332
Error Messages . . . . .	278	Retrieve Identity of Last Spooled File Created (QSPRILSP) API . . . . .	333
List Spooled Files (QUSLSPL) API . . . . .	279	Authorities and Locks . . . . .	333
Authorities and Locks . . . . .	280	Required Parameter Group . . . . .	333
Required Parameter Group . . . . .	280	Format of Receiver Variable . . . . .	334
		Field Descriptions . . . . .	334
		Error Messages . . . . .	335

Retrieve Spool Information (QSPSPLI) API . . . . .	335	Option Specific Input Information to Print	
Authorities and Locks . . . . .	335	Driver . . . . .	403
Required Parameter Group . . . . .	335	Field Descriptions . . . . .	404
Format of Receiver Variable . . . . .	336	Option Specific Output Information from Print	
Field Descriptions . . . . .	337	Driver . . . . .	407
Error Messages . . . . .	337	Field Descriptions . . . . .	407
Retrieve Spooled File Attributes (QUSRSPLA) API . . . . .	337	PSF/400 Mapping Exit Program . . . . .	409
Authorities and Locks . . . . .	338	Authorities and Locks . . . . .	410
Required Parameter Group . . . . .	338	Required Parameter Group . . . . .	410
Optional Parameter Group 1 . . . . .	339	Mapping Exit Program Input Information . . . . .	410
Optional Parameter Group 2 . . . . .	340	Field Descriptions . . . . .	411
How to Select a Spooled File to Retrieve Its		Mapping Exit Program Output Information . . . . .	412
Attributes . . . . .	341	Field Descriptions . . . . .	412
SPLA0100 Format . . . . .	342	Extension Area Format . . . . .	414
SPLA0200 Format . . . . .	346	Field Descriptions . . . . .	415
User-Defined Option Entry . . . . .	352	Integrated File System Authorizations . . . . .	419
User Resource Library List Entry . . . . .	352	Stream File Format . . . . .	419
Edge-Stitch Staple Offset Entry . . . . .	352	Field Descriptions . . . . .	419
Saddle-Stitch Staple Offset Entry . . . . .	353	Individual Stream File Information Format . . . . .	419
Library List Entry . . . . .	353	Field Descriptions . . . . .	420
Internet Print Protocol Spooled File Attributes . . . . .	353	Spooled File PDF Distribution . . . . .	420
Save and Restore Spooled File Attributes . . . . .	353	Field Descriptions . . . . .	420
Field Descriptions . . . . .	354	Spooled File AFPDS Distribution . . . . .	421
Error Messages . . . . .	384	Field Descriptions . . . . .	421
PrintManager APIs . . . . .	385	PDF Encryption Structure . . . . .	422
Exit Programs . . . . .	385	Field Descriptions . . . . .	422
Exit Program for a Customized Separator Page . . . . .	385	Writer Transform Exit Program . . . . .	424
Authorities and Locks . . . . .	385	Authorities and Locks . . . . .	425
Required Parameter Group . . . . .	385	Required Parameter Group . . . . .	425
Separator Data Format . . . . .	386	Option Specific Input Information to Transform	
Field Descriptions . . . . .	387	Exit Program . . . . .	427
Separator Information Format . . . . .	390	Field Descriptions . . . . .	428
Field Descriptions . . . . .	390	Option Specific Output Information From	
Image Print Transform Exit Program . . . . .	391	Transform Exit . . . . .	431
Authorities and Locks . . . . .	393	Field Descriptions . . . . .	432
Required Parameters . . . . .	393	UIM Spooled File User-defined List Action Exit	
Transform Values to Exit Program . . . . .	393	Program . . . . .	436
XFRM0100 Format . . . . .	393	Authorities and Locks . . . . .	436
Field Descriptions . . . . .	394	Required Parameters . . . . .	436
Process Values to Exit Program . . . . .	398	LASP0100 Format . . . . .	437
Field Descriptions . . . . .	398	Field Descriptions . . . . .	437
IPDS to PDF Transform Interface Exit Program . . . . .	399	Concepts . . . . .	438
Authorities and Locks . . . . .	400	Spool and Print Tools in Library QUSRTOOL . . . . .	438
Parameters . . . . .	400		
PDF Transform Input Information . . . . .	400		
Field Descriptions . . . . .	401		
Print Driver Exit Program . . . . .	402	<b>Appendix. Notices . . . . .</b>	<b>439</b>
Authorities and Locks . . . . .	402	Programming interface information . . . . .	440
Required Parameter Group . . . . .	402	Trademarks . . . . .	441
		Terms and conditions . . . . .	442

---

## Print APIs

See the Printing topic collections for information about i5/OS<sup>®</sup> printing capabilities and about planning for and configuring printing environments.

The print APIs consist of:

- “General Print APIs”
- “Spooled File APIs” on page 196
- “PrintManager APIs” on page 385

Also see “Spool and Print Tools in Library QUSRTOOL” on page 438.

APIs by category

---

## APIs

These are the APIs for this category.

---

### General Print APIs

General print APIs can obtain information about or perform printing activities on the system. They can:

- Retrieve output queue information such as status and number of entries on the queue.
- Retrieve information about specific printer writers.
- Transform data streams from one type to another.

The general print APIs are:

- “Add PDF Map Entry (QPQAPME) API” on page 3 (QPQAPME) allows you to add an entry to a PDF map object.
- “Build Open Time Commands (QSPBOPNC) API” on page 21 (QSPBOPNC) uses the attributes of the spooled file to build commands that must be sent ahead of the data to prepare the printer to print the spooled file. These commands are referred to as open time commands. The open time commands contain most of the file level commands needed to format the printed output.
- “Build Separator Pages (QSPBSEPP) API” on page 23 (QSPBSEPP) builds the system separator page to be printed for the spooled file. A user exit program is called to build the separator page or the system default is used. The separator page is returned in the receiver variable parameter. This data can then be sent to the printer.
- “Change Output Queue (QSPCHGOQ) API” on page 26 (QSPCHGOQ) changes the attributes of the output queue specified.
- “Convert Image (QIMGCVTI, QimgCvtImg) API” on page 30 (QIMGCVTI, QimgCvtImg) converts an image or print data stream from one format to another, and optionally prints the result.
- “Copy AFPDS Resource (QGSCPYRS) API” on page 60 (QGSCPYRS) puts an AFPDS data stream equivalent of the specified AFPDS resource into the specified user space.
- “Create PDF Map (QPQCRTPM) API” on page 65 (QPQCRTPM) allows you to create a PDF map.
- “Delete PDF map object (QPQDLTPM) API” on page 66 (QPQDLTPM) allows you to delete a PDF map object.

- “Extract Writer Status (QSPEXTWI) API” on page 68 (QSPEXTWI) is used by a driver program to access information about the writer, the spooled file the writer is processing, and to determine if the information for the spooled file has been changed. A call to this API should be made by a driver program after each buffer is sent to the printer.
- “Host Print Transform (QWPZHPTR, QwpzHostPrintTransform) API” on page 71 (QWPZHPTR, QwpzHostPrintTransform) provides an interface to the Host Print Transform function. The Host Print Transform function transforms SCS or AFP data to ASCII data.
- “Open List of AFP Information (QGYOLAFP) API” on page 87 (QGYOLAFP) lists entries from a font mapping table or list font character set or code page information from a coded font and places the list in a receiver variable.
- “Open List of PDF Map Information (QPQOLPM) API” on page 99 (QPQOLPM) allows you to obtain a list of entries within a PDF map object.
- “Open List of Printers (QGYRPRTL) API” on page 118 (QGYRPRTL) generates a list of printers on the system. Information about each printer is included with the list.
- “Publish Print Information (QSPPUBPI) API” on page 124 (QSPPUBPI) collects print object information and queues it for publishing in an LDAP directory.
- “Remove PDF Map Object Entries (QPQRPME) API” on page 126 (QPQRPME) allows you to remove PDF map entries.
- “Retrieve AFP Information (QPQRAFPI) API” on page 130 (QPQRAFPI) returns specific information about a PSF configuration object, a font table, or a coded font.
- “Retrieve Output Queue Information (QSPROUTQ) API” on page 152 (QSPROUTQ) retrieves information associated with the specified output queue. Information returned includes the parameters used to create the queue, the current status of the queue, and the number of entries on the queue.
- “Retrieve Printer Attributes (QGYRPRTA) API” on page 166 (QGYRPRTA) retrieves attributes about a specific printer device.
- “Retrieve Writer Information (QSPRWTRI) API” on page 176 (QSPRWTRI) retrieves printer writer information associated with the specified printer only when a printer writer is started to the printer. The information retrieved is similar to what can be seen when running the Work with Writer (WRKWTR) command for a particular printer writer.
- “Send Writer Message (QSPSNDWM) API” on page 184 (QSPSNDWM) is used by a driver exit program to send either an informational or inquiry message to the writer’s message queue.
- “Set Writer Status (QSPSETWI) API” on page 188 (QSPSETWI) is used by a driver exit program to update information about a spooled file that a writer is processing. This information is used on certain spooled file displays.
- “Transform AFP to ASCII (QWPZTAFP) API” on page 193 (QWPZTAFP) transforms an Advanced Function Presentation data stream (AFPDS) into an ASCII data stream. This ASCII data stream can be formatted for IBM, Hewlett-Packard, or PostScript-capable printers.

The general print exit programs are:

- “Exit Program for a Customized Separator Page” on page 385 allows a user to customize the separator page preceding a spooled file. For example, an exit program can be used to create a banner-style separator page.
- “Image Print Transform Exit Program” on page 391 can be used to transform user ASCII data streams.
- “IPDS to PDF Transform Interface Exit Program” on page 399 submits an IPDS to PDF transform when a PDF transform is requested to the QIBM\_QPQ\_Transform exit point and the PDFGEN parameter on the PSF configuration object has a value other than \*NONE.
- “Print Driver Exit Program” on page 402 defines how a user-defined print driver exit program must be written in order to be used with the i5/OS printer writer program.
- “PSF/400 Mapping Exit Program” on page 409 allows a customer to specify a mail tag instead of a mail address in the USRDFNDDTA parameter of the printer file or the DDS STRPAGGRP parameter.

- Spool File Creation describes the layout of the message sent when using the QIBM\_NOTIFY\_CRTSPLF environment variable.
- Spool File Ready describes the layout of the message sent when the DTAQ parameter of CRTOUTQ or CHGOUTQ is used to specify a data queue.
- “Writer Transform Exit Program” on page 424 describes the interface between a user-defined transform exit program (transform exit) and an i5/OS printer writer (writer). A transform exit program can be enabled by an operator when a writer is started to an output queue.

Top | “Print APIs,” on page 1 | APIs by category

---

## Add PDF Map Entry (QPQAPME) API

Required Parameter Group:

1	Returned library name	Output	Char(10)
2	Qualified PDF map object name	Input	Char(20)
3	PDF map entry spooled file attributes	Input	Char(*)
4	Length of PDF map entry spooled file attributes	Input	Binary(4)
5	Format of PDF map entry spooled file attributes	Input	Char(8)
6	PDF mapping action	Input	Char(*)
7	Length of PDF mapping action	Input	Binary(4)
8	Format of PDF mapping action	Input	Char(8)
9	Add type	Input	Binary(4)
10	Error code	I/O	Char(*)

Default Public Authority: \*Use  
Threadsafe: Yes

The Add PDF Map Entry (QPQAPME) API inserts an entry into the specified PDF Map object. You can retrieve the current entries by using the Open List of PDF Map Information (QPQOLPM) API.

This API allows you to set up PDF mapping entries for spooled files you want processed after they have been converted to PDF.

There are two types of entries. Segmented and non-segmented. Segmented entries are for spooled files that are divided into multiple PDF files. The division is based upon the use of the STRPAGGRP DDS keyword and other products such as the AFP™ Toolbox or the Infoprint® Server. For all segmented entries of a spooled file, the input spooled file criteria (format APME0100), all the selection parameters should be the same except the mail tag information. Non-segmented entries are for spooled files that will be converted to one PDF file.

The entry consists of two parts:

- Selection for the input spooled file. You can specify which spooled file attributes should be used to match a spooled file being converted to PDF. This information is used by products, such as Infoprint Server to determine what additional processing needs to be done with the PDF file.  
The selection for the input spooled file includes such spool attributes as output queue name and library, spooled file name, job name, user name, user data, form type, mail tag, and also a sequence number. To allow for the input selection criteria of multiple entries in the PDF map which can match a particular spooled file, the entries are sorted in the order of the sequence number specified. Multiple entries can be a match for a spooled file because generic and \*ALL are supported as part of input selection criteria, for example output queue, file, job name, user name, user data, and form type.
- PDF map actions for the spooled file. For each spooled file or segmented entry of a spooled file, you may define one or more map actions and their associated parameters. The supported actions are e-mail, stream file, PDF respool, and AFPDS respool.

For e-mail, you can enter the following additional information:

- Tolist, Copy Control, and Blind Copy Control
- Subject of email
- Body of e-mail
- Attachments (in addition to PDF spool file)
- Reply to address
- Mail sender

For stream file, you can enter the following additional information:

- Full path name where stream file will be stored in Integrated File System
- Authority for PDF stream file

For re-spooling PDF file, you can enter the following additional information:

- Output queue name
- File name
- User data
- User defined data
- Form type

For re-spooling AFPDS file, you can enter the following additional information:

- Output queue name
- File name
- User data
- User defined data
- Form type

When you request to add an entry to a PDF Map object, the request may not be successful in the following situations:

- The entry you requested to add already exists in the PDF Map. For non-segmented entries, the sequence number must be unique. For segmented entries (which all have the same sequence number), the mail tag must be unique.
- The user profile storage limit is exceeded when an entry is added.

## Authorities and Locks

*Library authority*

\*EXECUTE

*PDF Map object authority*

\*CHANGE

*PDF Map object lock*

\*EXCL

## Required Parameter Group

**Returned library name**

OUTPUT; CHAR(10)

The name of the library that contains the PDF map object to which the entry was added if it was added successfully. This parameter is not set if the entry was not successfully added. This information helps you identify the specific library used when \*LIBL or \*CURLIB is specified in the qualified PDF map object name parameter.

**Qualified Map object name**

INPUT; CHAR(20)

The Map object for which you would like to add information. The qualified object name has two parts. The first 10 characters contain the object name.

The second 10 characters contain the name of the library in which the object is located.

The following special values are allowed:

- \**CURLIB*            The job's current library is used to locate the object. If no library is specified as the current library for the job, the QGPL library is used.
- \**LIBL*             The library list is used to locate the object.

The actual library that is used is returned in the Returned library name parameter.

**PDF map entry spooled file attributes**

INPUT; CHAR(\*)

The PDF map entry spooled file attributes for the entry to be added to the PDF map object. The parameter uniquely identifies the entry in the PDF map object.

Specify which spooled file attributes should be used to match a spooled file being converted to PDF.

**Length of PDF map entry spooled file attributes**

INPUT; BINARY(4)

The length of the PDF map entry spooled file attributes parameter. This value must be greater than or equal to 4.

**Format of PDF map entry spooled file attributes**

INPUT; CHAR(8)

The format of the PDF map for the entry to be added to the PDF map object.

The valid format names are:

- "APME0100 Format"* on page 6    Contains the input selection criteria for the entry. This includes a sequence number for the entry and spooled file information, such as output queue, spooled file name, job name, user name, user data, form type, and mail tag.

**PDF mapping action**

INPUT; CHAR(\*)

The PDF mapping action for the entry to be added to the PDF map object.

See "PMAP0100 Format" on page 8 for details on the data structure.

**Length of PDF mapping action**

INPUT; BINARY(4)

The length of the PDF mapping action parameter. This value must be greater than or equal to 8.

**Format of PDF mapping action**

INPUT; CHAR(8)

The format of the PDF mapping action to be added to the PDF map object.

The valid format names are:

- "PMAP0100 Format"* on page 8    Contains the mapping action for the entry. This can include mail information, PDF stream file information, PDF and AFPDS spooled file information, and a spooled file segmentation indicator.

## Add type

INPUT; BINARY(4)

The type of add to be performed.

Valid values are:

### 1 Insert with replacement

This value requests to replace the PDF map portion of the entry if the PDF map entry spooled file attributes is already in the PDF map object. If the entry does not exist, it will be added into the PDF map object.

### 2 Add without replacement

This value requests to add the entry only if the PDF map entry spooled file attributes portion of the entry is not already in PDF map object. If the entry does not exist, it will be added into the PDF map object. If the entry is already in the PDF map object, an error is returned.

## Error code

I/O; CHAR(\*)

The structure in which to return error information. For the format of the structure, see Error code parameter.

## APME0100 Format

For more details about the fields in the following table, see "APME0100 Format Field Descriptions."

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Sequence number of entry
4	4	CHAR(10)	Output queue name
14	E	CHAR(10)	Output queue library name
24	18	CHAR(10)	Spooled file name
34	18	CHAR(10)	Job name
44	2C	CHAR(10)	User name
54	36	CHAR(10)	User data
64	40	CHAR(10)	Form type
74	4A	CHAR(250)	Mail tag
324	144	CHAR(2)	Reserved - initialized to null

## APME0100 Format Field Descriptions

**Form type.** The type of forms that should be loaded on the printer before this spooled file is printed. A least one form type or special value must be specified. This field can be used in conjunction with the output queue name, spooled file name, user, user data, and mail tag to select a spooled file to be processed.

The following special values can be used for the form type:

*\*ALL* Specifies that all form types will be considered a match regardless of the value for form type.

A default of *\*ALL* is assumed if the form type is not specified.

**Job name.** The name of the job for the spooled file. At least one job name or special value must be specified. This field can be used in conjunction with the output queue name, spooled file name, form type, and mail tag to select a spooled file to be processed.

The following special value is used for the job name:

<i>*ALL</i>	Specifies that all job names will be considered a match for input selection regardless of the job name.
<i>*Generic*-job</i>	Specify the generic name of a job. A generic name is a character string followed by an asterisk(*). If you specify a generic name, all user names that begin with the generic name are considered a match in the input selection

A default of *\*ALL* is assumed if the job name is not specified.

**Mail tag.** Specifies the mail tag for the spooled file. For non-segmented spooled files, the mail tag is specified on the CRTPRTE, CHGPRTE, or OVRPRTE commands using the USRDFNDDTA parameter. For segmented spooled files, the mail tag is specified on the STRPAGGRP DDS keyword. This field can be used in conjunction with the output queue name, spooled file name, user, user data, and form type to select a spooled file to be processed.

Here is information on specifying a mail tag:

- Embedded blanks will be used as part of the mail tag.
- Do not use single quotes around the text string, as they will be interpreted as part of the text string.
- Case is preserved.

The following special values can be used for the mail tag:

<i>*ALL</i>	Specifies that all mail tags will be considered a match in the input selection regardless of the value for mail tag.
-------------	--

A default of *\*ALL* is assumed if the mail tag is not specified.

**Output queue library name.** The name of the library where the output queue resides.

**Output queue name.** The name of the output queue in which the spooled file is located. At least one output queue or special value must be specified. This field can be used in conjunction with the spooled file name, user, form type, and mail tag to select a spooled file to be processed.

The following special value is used for the output queue name:

<i>*ALL</i>	Specifies that all output queues will be considered a match in the input selection regardless of the output queue name. When this value is specified, the output queue library name must be blanks.
<i>*Generic*-output-queue</i>	Specify the generic name of an output queue. A generic name is a character string of one or more characters followed by an asterisk(*). If you specify a generic value, all output queues with the names that begin with the generic name are considered a match in the input selection

**Reserved - initialized to null.** A reserved field.

**Sequence number of entry.** The user-defined sequence number for an entry in a PDF Map object. To allow for the input selection criteria of multiple entries in the PDF map that can match a particular spooled file, the entries are sorted in the order of the sequence numbers. Multiple entries can be a match for a spooled file because generic and *\*ALL* are supported as part of input selection criteria, for example output queue, file, user name, user data, and form type.

**Spoiled file name.** The name of the spoiled file. At least one spoiled file name or special value must be specified. This field can be used in conjunction with the output queue name, user, form type, and mail tag to to select a spoiled file to be processed.

The following special value is used for the output queue name:

- \*ALL* Specifies that all spoiled file names will be considered a match in the input selection regardless of the spoiled file name.
- \*Generic\*-spoiled-file* Specify the generic name of a spoiled file. A generic name is a character string followed by an askerisk(\*). If you specify a generic name, all spoiled files with the names that begin with the generic name are considered a match in the input selection

A default of *\*ALL* is assumed if the spoiled file name is not specified.

**User data.** The value of the user data. You must specify user data or a special value. This field can be used in conjunction with the output queue name, spoiled file name, user, form type, and mail tag to select to select a spoiled file to be processed.

The following special value is used for the user data:

- \*ALL* Specifies that all user data will be considered a match in the input selection regardless of the user data.

A default of *\*ALL* is assumed if the user data is not specified.

**User name.** The name of the user that created the spoiled file. At least one user name or special value must be specified. This field can be used in conjunction with the output queue name, spoiled file name, form type, and mail tag to to select a spoiled file to be processed.

The following special value is used for the output queue name:

- \*ALL* Specifies that all user names will be considered a match for input selection regardless of the user name.
- \*Generic\*-user-name* Specify the generic name of a user. A generic name is a character string followed by an askerisk(\*). If you specify a generic name, all user names that begin with the generic name are considered a match in the input selection

A default of *\*ALL* is assumed if the user name is not specified.

## PMAP0100 Format

For more details about the fields in the following table, see “PMAP0100 Field Descriptions” on page 9.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Offset to mail information entry
4	4	BINARY(4)	Length of to mail information entry
8	8	BINARY(4)	Offset to PDF stream file information
12	C	BINARY(4)	Length of to PDF stream file information
16	10	BINARY(4)	Offset to PDF spoiled file information
20	14	BINARY(4)	Length of PDF spoiled file information
24	18	BINARY(4)	Offset to AFPDS spoiled file information

Offset		Type	Field
Dec	Hex		
28	1C	BINARY(4)	Length of AFPDS spooled file information
32	20	CHAR(1)	Segmented entry
33	21	CHAR(50)	Text description
83	53	CHAR(49)	Reserved - initialized to null

## PMAP0100 Field Descriptions

**Length of AFPDS spooled file information.** The size, in bytes, of the AFPDS Spooled file information entry. For information on the format of a Spooled file information entry, see AFPDS Spooled File Information Entry.

**Length of mail information entry.** The size, in bytes, of the mail information entry. For information on the format of a mail information entry, see Mail Information Entry.

**Length of PDF spooled file information.** The size, in bytes, of the PDF spooled file information entry. For information on the format of a spooled file information entry, see PDF Spooled File Information Entry.

**Length of PDF stream file information.** The size, in bytes, of the PDF stream file information entry. This identifies the path where PSF/400 has written the PDF stream file. For information on the format of a stream file entry, see PDF Stream File Information Entry.

**Offset to AFPDS spooled file information.** The offset, in bytes, from the beginning of the PMAP0100 format structure, to an AFPDS Spooled File Information entry. If this value is 0, an AFPDS Spooled File Information entry is not specified.

**Offset to mail information entry.** The offset, in bytes, from the beginning of the PMAP0100 format, to a Mail information entry.

**Offset to PDF spooled file information.** The offset, in bytes, from the beginning of the PMAP0100 format, to a Spooled File Information entry.

**Offset to PDF stream file information.** The offset, in bytes, from the beginning of the PMAP0100 format, to a PDF Stream File Information entry.

**Reserved - initialized to null.** A reserved field.

**Segmented entry.** Indicates whether an entry is a segment. Segmented entries are for AFPDS spooled file that is divided into multiple PDF files. The division would be based upon the use of the STRPAGGRP DDS keyword, the Toolbox APIs, or sucs products as as Infoprint Server.

The following values are used:

- 0 The entry is not a segment.
- 1 The entry is a segment.

If the segmented entry indicator is not specified, then 0 is assumed (entry is not segmented).

**Text description.** A brief description of the PDF Map entry.

## Mail Information Entry

Information in this table should be filled in when e-mail is being requested for a PDF map. If e-mail is not being requested, the length of mail information and offset to mail information fields above should be set to 0. For more details about the fields in the following table, see “Mail Information Entry Field Descriptions” on page 11.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Offset to message text
4	4	BINARY(4)	Length of message text
8	8	BINARY(4)	Offset to subject
12	C	BINARY(4)	Length of subject
16	10	BINARY(4)	CCSID of message text and subject
20	14	BINARY(4)	Offset to ReplyTo e-mail address
24	18	BINARY(4)	Length of ReplyTo e-mail address
28	1C	BINARY(4)	Offset to TO e-mail address entries
32	20	BINARY(4)	Number of TO e-mail address entries
36	24	BINARY(4)	Length of TO e-mail address entry
40	28	BINARY(4)	Offset to CC e-mail address entries
44	2C	BINARY(4)	Number of CC e-mail address entries
48	30	BINARY(4)	Length of CC e-mail address entry
52	34	BINARY(4)	Offset to BCC e-mail address entries
56	38	BINARY(4)	Number of BCC e-mail address entries
60	3C	BINARY(4)	Length of BCC e-mail address entry
64	40	BINARY(4)	Offset to Body of e-mail entries
68	44	BINARY(4)	Number of Body of e-mail entries
72	48	BINARY(4)	Length of Body of e-mail entry
76	4C	BINARY(4)	Offset to attachment e-mail entries
80	50	BINARY(4)	Number of attachment e-mail entries
84	54	BINARY(4)	Length of attachment e-mail entry
88	58	BINARY(4)	Offset to TO e-mail stream file
92	5C	BINARY(4)	Length of TO e-mail stream file
96	60	BINARY(4)	Offset to CC e-mail stream file
100	64	BINARY(4)	Length of CC e-mail stream file
104	68	BINARY(4)	Offset to BCC e-mail stream file
108	6C	BINARY(4)	Length of BCC e-mail stream file
112	70	BINARY(4)	Offset to PDF file for e-mail
116	74	BINARY(4)	Length of PDF file for e-mail
120	78	CHAR(10)	Mail Sender
130	82	CHAR(10)	User password
140	8C	CHAR(10)	Owner password
150	96	CHAR(10)	Encryption level
160	A0	CHAR(10)	Print PDF file

Offset		Type	Field
Dec	Hex		
170	AA	CHAR(10)	Copy PDF file
180	B4	CHAR(10)	Change PDF file
190	BE	CHAR(10)	Change Comments in PDF file
200	C8	CHAR(10)	Content Access to PDF file
210	D2	CHAR(1)	Use stream file for TO e-mail address entries
211	D3	CHAR(1)	Use stream file for CC e-mail address entries
212	D6	CHAR(1)	Use stream file for BCC e-mail address entries
213	D5	CHAR(43)	Reserved - initialized to null

## Mail Information Entry Field Descriptions

**CCSID of message text and subject.** The CCSID associated with the message text and subject, if specified.

0 Use the job's default CCSID.  
*ccsid-value* Specifies the CCSID. Values are in the range 1 to 65533.

This field is used only when an SMTP mail server is specified in your PSF configuration object and must be set to zero otherwise.

This CCSID will be used to convert the data to Unicode for inclusion within the e-mail. The receiver of the e-mail may have to have the receiving e-mail program configured for the appropriate character set to be able to view the e-mail correctly.

**Change Comments in PDF file.** This field is applicable only for an Encryption Level of \*40RC4. Specifies whether users are allowed to add or change comments (annotations) or form fields in the PDF file.

The following special values are used:

\*YES The user is allowed to change the PDF file.  
\*NO The user is not allowed to change the PDF file.

**Change PDF file.** This field is applicable for Encryption levels \*40RC4 and \*128RC4. The definition is different for the type of Encryption level.

For an Encryption level of \*40RC4, Change PDF file specifies whether users are allowed to change the document, or to prevent them from creating form fields, as well as making any other changes.

When the Encryption level is \*40RC4, the following special values are used:

\*YES The user is allowed to change the PDF file.  
\*NO The user is not allowed to change the PDF file.

For an Encryption level of \*128RCR, Change PDF file specifies the types of changes which are allowed for the PDF file.

When the Encryption level is \*128RCR, the following special values are used:

- \*NONE The user is not allowed to change the PDF file.
- \*ASSEMBLY The user is allowed document assembly to let user's insert delete, rotate pages, and create bookmarks and thumbnails.

**Content Access to PDF file.** Specifies whether content accessibility for the visually impaired is enabled for the PDF file.

This field is applicable only for an Encryption Level of \*128RCR.

When the Encryption level is \*128RCR, the following special values are used:

- \*YES content accessibility for the visually impaired is enabled in the PDF file..
- \*NO Content accessibility for the visually impaired is disabled in the PDF file. .

**Copy PDF file.** Specifies whether users are allowed to copy or extract text and graphics. Also specifies whether the accessibility interface is disabled.

This field is applicable for Encryption levels \*40RC4 and \*128RCR.

The following special values are used for Copy PDF file:

- \*YES The user is allowed to copy and extract text and graphics in the PDF file. The accessibility interface is enabled.
- \*NO The user is not allowed to copy and extract text and graphics in the PDF file. The accessibility interface is disabled.

**Encryption level.** The encryption level for the PDF file.

The following special values are used for the encryption level:

- \*NONE The file is not encrypted.
- \*40RCR The PDF file is created with the 40-bit RC4 encryption level. This level of encryption has a lower level of security that is compatible with Adobe Acrobat Reader 3.X, 4.X, and 5.
- \*128RCR The PDF file is created with the 128-bit RC4 encryption level. This level of encryption is has a higher level of security but is only compatible with Adobe Acrobat 5.

**Length of attachment e-mail entry.** The size, in bytes, of each entry used for naming an e-mail attachment. Use one entry for each stream file. The entries must be concatenated, as in an array. For information on the format of a stream file entry, see Stream File Entry.

This field is used only when an SMTP mail server is specified in your PSF configuration object.

**Length of BCC e-mail address entry.** The size, in bytes, of each E-mail Address entry used for the BCC of an e-mail. For information on the format of an e-mail addresses entry, see E-mail Address Entry.

Use stream file for BCC e-mail address entries should be set to 0.

This field is used only when an SMTP mail server is specified in your PSF configuration object.

**Length of BCC e-mail stream file.** The size, in bytes, of the Stream File entry which lists the BCCs used for an e-mail. For information on the format of an e-mail addresses entry, see Stream File Entry.

Use stream file for BCC e-mail address entries should be set to 1.

This field is used only when an SMTP mail server is specified in your PSF configuration object.

**Length of Body of e-mail entry.** The size, in bytes, of each entry used for naming a stream file placed into the body of the e-mail. Use one entry for each stream file. The entries must be concatenated, as in an array. For information on the format of a stream file entry, see Stream File Entry.

This field is used only when an SMTP mail server is specified in your PSF configuration object.

**Length of CC e-mail address entry.** The size, in bytes, of each E-mail Address entry used for the CC of an e-mail. For information on the format of an e-mail addresses entry, see E-mail Address Entry.

Use stream file for CC e-mail address entries should be set to 0.

This field is used only when an SMTP mail server is specified in your PSF configuration object.

**Length of CC e-mail stream file.** The size, in bytes, of the Stream File entry containing the stream file which lists the CCs used for an e-mail. For information on the format of a stream file entry, see Stream File Entry.

Use stream file for CC e-mail address entries should be set to 1.

This field is used only when an SMTP mail server is specified in your PSF configuration object.

**Length of message text.** The size, in bytes, of the message text that gets placed into the body of the e-mail.

The following special values can be used for the message text:

*\*NONE* No message text will be included in the body of the e-mail.

If the length of the message text is 0, then the default message text is obtained from message PQT4133 in message file QPQMSGF.

**Length of PDF file for e-mail** The size, in bytes, of a File Name Entry. This identifies the name to be used for the PDF file produced from a spooled file, that is attached to an e-mail. For information on the format of a file name entry, see File Name Entry.

**Length of ReplyTo to e-mail address.** The size, in bytes, of an E-mail Address entry used for the ReplyTo of an e-mail. For information on the format of an e-mail addresses entry, see E-mail Address Entry.

This field is used only when an SMTP mail server is specified in your PSF configuration object.

The following special values can be used for the ReplyTo e-mail address:

*\*MAILSENDER* Replies will be sent to the FROM e-mail address.

**Length of subject.** The size, in bytes, of the text that is used for the subject of the e-mail.

The following special values can be used for the subject:

*\*NONE* No subject will be used in the body of the e-mail.

If the length of the subject is 0, then the default subject is obtained from message PQT4133 in message file QPQMSGF.

**Length of TO e-mail address entry.** The size, in bytes, of each E-mail Address entry used for the TO of an e-mail. For information on the format of an e-mail addresses entry, see E-mail Address Entry.

Use stream file for TO e-mail address entries should be set to 0.

The following special values can be used for the TO e-mail address entry:

*\*SPLF*        The e-mail address to be used is specified as an attribute of the spooled file (USRDFNDTA('MAILTAG(e-mail-address)'). You can only specify one e-mail address on the USRDFNDTA parameter. The e-mail address can also be specified on the STRPAGGRP DDS keyword (group name).

**Length of TO e-mail stream file.** The size, in bytes, of the Stream File entry containing the stream file which lists the TOs used for an e-mail. If the stream file name is surrounded by quotes, use all characters within the quotes as the stream file name. For information on the format of an e-mail addresses entry, see Stream File Entry.

Use stream file for TO e-mail address entries should be set to X'F1'.

**Mail Sender.** The user profile from whom the e-mail is being sent.

The following special values can be used for the mail sender:

*\*PSFCFG*        The mail sender specified on the PSF configuration object will be used for the mail sender.

**Number of attachment e-mail entries.** The number of stream files to be attached to the e-mail. This value does not include the spooled file converted to PDF.

This field is used only when an SMTP mail server is specified in your PSF configuration object.

**Number of BCC e-mail address entries.** The number of BCC e-mail addresses to be used for sending an e-mail.

Use stream file for BCC e-mail address entries should be set to 0.

This field is used only when an SMTP mail server is specified in your PSF configuration object.

**Number of Body of e-mail entries.** The number of stream files to be placed into the body of the e-mail.

This field is used only when an SMTP mail server is specified in your PSF configuration object.

**Number of CC e-mail address entries.** The number of CC e-mail addresses to be used for sending an e-mail.

Use stream file for CC e-mail address entries should be set to 0.

This field is used only when an SMTP mail server is specified in your PSF configuration object.

**Number of TO e-mail address entries.** The number of TO e-mail addresses to be used for sending an e-mail.

Use stream file for TO e-mail address entries should be set to 0.

This field is used only when an SMTP mail server is specified in your PSF configuration object.

**Offset to attachment e-mail entries.** The offset, in bytes, from the beginning of the PMAP0100 format structure, to a list of attachment entries.

If this value is 0, a list of attachment entries is not specified.

**Offset to BCC e-mail address entries.** The offset, in bytes, from the beginning of the PMAP0100 format structure, to a list of BCC e-mail address entries. This area contains the BCC e-mail addresses to which the PDF file is sent.

If Use stream file for BCC e-mail address entries is set to X'F1', then this value will be 0.

**Offset to BCC e-mail stream file.** The offset, in bytes, from the beginning of the PMAP0100 format structure, to a file name which contains a list of BCC e-mail address entries.

If Use stream file for BCC e-mail address entries is set to 0, then this value will be 0.

**Offset to Body of e-mail entries.** The offset, in bytes, from the beginning of the PMAP0100 format structure, to a list of Stream file entries.

If this value is 0, a list of Stream file entries is not specified.

**Offset to CC e-mail address entries.** The offset, in bytes, from the beginning of the PMAP0100 format structure, to a list of CC e-mail address entries. This area contains the CC e-mail addresses to which the PDF file is sent.

If Use stream file for CC e-mail address entries is set to 1, then this value will be 0.

**Offset to CC e-mail stream file.** The offset, in bytes, from the beginning of the PMAP0100 format structure, to a file name which contains a list of CC e-mail address entries.

If Use stream file for CC e-mail address entries is set to 0, then this value will be 0.

**Offset to message text.** The offset, in bytes, from the beginning of the PMAP0100 format, to the message text which gets placed in to the body of the e-mail.

If this value is 0, message text is not specified.

**Offset to subject.** The offset, in bytes, to the subject of the e-mail. If this value is 0, a subject is not specified.

**Offset to PDF file for e-mail** The offset, in bytes, from the beginning of the PMAP0100 format, to the PDF File For E-mail Entry.

**Offset to ReplyTo e-mail address.** The offset, in bytes, from the beginning of the PMAP0100 format, to the E-mail address Entry for a ReplyTo e-mail address. If this value is 0, a ReplyTo e-mail address is not specified.

**Offset to TO e-mail address entries.** The offset, in bytes, from the beginning of the PMAP0100 format, to a list of TO e-mail address entries. This area contains the TO e-mail addresses to which the PDF file is sent.

**Offset to TO e-mail stream file.** The offset, in bytes, from the beginning of the PMAP0100 format, to a file name which contains a list of TO e-mail address entries.

If Use stream file for TO e-mail address entries is set to 0, then this value will be 0.

**Owner password.** The owner password associated with the PDF file to be created.

This field is applicable for Encryption levels \*40RC4 and \*128RC4.

The following can be used for the owner password:

\*NONE            No owner password is specified for the PDF file.  
*owner-password*    A owner password is specified for the PDF file. The password is encrypted.

**Print PDF file.** This field is applicable for Encryption levels \*40RC4 and \*128RCR. The definition is different for the type of Encryption level.

For an Encryption level of \*40RC4, Print PDF file specifies whether users are allowed to print the PDF file.

When the Encryption level is \*40RC4, the following special values are used:

\*YES            The user is allowed to print the PDF file.  
\*NO             The user is not allowed to print the PDF file.

For an Encryption level of \*128RC4, Print PDF file specifies print options for the PDF file.

When the Encryption level is \*128RC4, the following special values are used:

\*YES            The user is not allowed to print the PDF file.  
\*NO             The user is not allowed to print the PDF file.  
\*IMAGE         The user is only allowed to print at a low resolution (image). This prevents the user from recreating the PDF file with different security settings. Printing may be slower because each page will be printed as a bit-mapped image.

**Use stream file for BCC e-mail address entries.** Indicates if a stream file containing BCC e-mail addresses is being used.

The following values are used:

X'F1'            A stream file for BCC e-mail address entries is being used.  
0                A stream file for BCC e-mail address entries is not being used.

**Use stream file for CC e-mail address entries.** Indicates if a stream file containing CC e-mail addresses is being used.

The following values are used:

0                A stream file for CC e-mail address entries is not being used.  
1                A stream file for CC e-mail address entries is being used.

**Use stream file for TO e-mail address entries.** Indicates if a stream file containing TO e-mail addresses is being used.

The following values are used:

0                A stream file for TO e-mail address entries is not being used.  
1                A stream file for TO e-mail address entries is being used.

**User password.** The user password associated with the PDF file to be created.

This field is applicable for Encryption levels \*40RC4 and \*128RC4.

The following can be used for the user password:

\*NONE No user password is specified for the PDF file.  
user-password An user password is specified for the PDF file. The password is encrypted.

## E-mail Address Entry

This table is used for mapping the TO, CC, BCC and ReplyTo e-mail addresses. For more details about the fields that can be used with the following table, see “E-mail Address Entry Field Description.”

Offset		Type	Field
Dec	Hex		
0	0	CHAR(80)	E-mail address

## E-mail Address Entry Field Description

**E-mail address.** An e-mail address.

For the TO e-mail address, the following special value can be specified:

\*SPLF Specifies that the e-mail address is specified as an attribute of the spooled file on the USRDFNDDTA parameter in the format USRDFNDDTA('MAILTAG(e-mail address)'). You can only specify one e-mail address on the USRDFNDDTA parameter. The e-mail address can also be specified in the spooled file with the STRPAGGRP DDS keyword (group name).

## Stream File Entry

This table is used for mapping the stream files that can be used for the Body, Attachment, TO, CC, and BCC stream files. Use one entry for each stream file. If you are specifying a stream file for the Body, the file extension (for example, txt or htm) must be specified in lower case. Otherwise, the file will be sent as an attachment. If necessary, the field is padded with blanks (X'40'). For more details about the fields that can be used with the following table, see “Stream File Entry Field Description.”

Offset		Type	Field
Dec	Hex		
0	0	CHAR(255)	Stream file name

## Stream File Entry Field Description

**Stream file name.** A path-qualified stream file name.

## File Name Entry

This table is used for specifying a file name that is not path-qualified. Use this entry for specifying a PDF file for e-mail entry. This will be the name of the attachment for the PDF file (original spooled file) that is included in the e-mail. For more details about the field that can be used with the following table, see “File Name Entry Field Description” on page 18.

Offset		Type	Field
Dec	Hex		
0	0	CHAR(80)	File name

## File Name Entry Field Description

**File name.** A file name which does not contain path information.

## PDF Stream File Information Entry

For more details about the fields in the following table, see “PDF Stream File Information Entry Field Descriptions.”

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Offset to PDF file name
4	4	BINARY(4)	Length of PDF file name
8	8	CHAR(10)	Public Authority
18	11	CHAR(2)	Reserved

## PDF Stream File Information Entry Field Descriptions

**Length of PDF file name.** The size, in bytes, of the Stream File entry which contains the PDF file name. For information on the format of a Stream File entry, see Stream File Entry.

**Offset to PDF file name.** The offset, in bytes, from the beginning of the PMAP0100 format, to the Stream File entry for a PDF file.

**Public Authority.** The public authority for the PDF stream file. The default is \*EXCLUDE.

The following special values may be specified:

<i>*EXCLUDE</i>	Users are excluded from accessing the object.
<i>*NONE</i>	Users do not have any of the data authorities to the objects.
<i>*RWX</i>	Users are given *RWX authority to perform all operations on the object except those limited to the owner or controlled by object existence, object management, object alter, and object reference authority. Users can change the object and perform basic functions on the object. *RWX authority provides object operational authority and all the data authorities.
<i>*RX</i>	The users are given *RX authority to perform basic operations on object, such as run a program or display the contents of a file. The user is prevented from changing the object. *RX authority provides object operational authority and read and execute authorities.
<i>*RW</i>	The users are given *RW authority to view the contents of an object and change the contents of an object. *RW authority provides object operational authority and data read, add, update, and delete authorities.
<i>*WX</i>	Users are given *WX authority to change the contents of an object and run a program or search a library or directory. *WX authority provides object operational authority and data add, update, delete, and execute authorities.
<i>*R</i>	Users are given *R authority to view the contents of an object. *R authority provides object operational authority and data read authority.
<i>*W</i>	Users are given *W authority to change the contents of an object. *W authority provides object operational authority and data add, update, and delete authorities.

\*X users are given \*X authority to run a program or search a library or directory. \*X authority provides object operational authority and data execute authority.

## PDF Spooled File Information Entry

These fields are used for mapping the PDF spooled file information. If necessary, text fields are padded with blanks (X'40'). For more details about the fields in the following table, see "PDF and AFPDS Spooled File Information Entry Field Descriptions."

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	Output queue name for output spooled file
10	A	CHAR(10)	Output queue library name for output spooled file
20	14	CHAR(10)	Spooled file name for output spooled file
30	1E	CHAR(10)	User data for output spooled file
40	28	CHAR(10)	Form type for output spooled file
50	32	CHAR(255)	User defined data for output spooled file
305	13B	CHAR(3)	Reserved

## AFPDS Spooled File Information Entry

These fields are used for mapping the AFPDS spooled file information. If necessary, text fields are padded with blanks (X'40'). For more details about the fields in the following table, see "PDF and AFPDS Spooled File Information Entry Field Descriptions."

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	Output queue name for output spooled file
10	A	CHAR(10)	Output queue library name for output spooled file
20	14	CHAR(10)	Spooled file name for output spooled file
30	1E	CHAR(10)	User data for output spooled file
40	28	CHAR(10)	Form type for output spooled file
50	32	CHAR(255)	User defined data for output spooled file
305	13B	CHAR(3)	Reserved

## PDF and AFPDS Spooled File Information Entry Field Descriptions

**Form type for output spooled file.** The type of forms that should be loaded on the printer before this spooled file is printed. This is the value which will be used when a file is spooled.

The following special values can be used for the form type for output spooled file:

\**SPLF* The form type of the original spooled file will be used.

**Output queue library name for output spooled file.** The name of the library where the output queue for the file which will be spooled resides.

**Output queue name for output spooled file.** The name of the output queue to which the file to be spooled will be written.

The following special values can be used for the form type for output queue name for output spooled file:

*\*PSFCFG*            The output queue specified in the PSF configuration object will be used when writing the PDF file to spool.

**Spooled file name for output spooled file.** The name which will be used for the file that gets spooled.

The following special values can be used for the spooled file name for output spooled file:

*\*SPLF*            The name of the original spooled file will be used.

**User data for output spooled file.** The user-specified data which will describe the file to be spooled.

The following special values can be used for the user data for output spooled file:

*\*SPLF*            The user data of the original spooled file will be used.

**User defined data for output spooled file.** The user-defined data that will be used for the file to be spooled.

The following special values can be used for the user defined data for output spooled file:

*\*SPLF*            The user defined data of the original spooled file will be used.

## Error Messages

Message ID	Error Message Text
CPD0172 D	Parameters passed on CALL do not match those required.
CPF5F00 E	Add type &1 is not valid.
CPF5F01 E	Length of PDF map entry spooled file attributes is not valid.
CPF5F02 E	Length of PDF mapping action is not valid.
CPF5F03 E	Value of length, offset, or number in parameter list is not valid.
CPF5F04 E	Entry is already in the PDF Map object.
CPF5F05 E	No e-mail address specified with mail information.
CPF5F06 E	No PDF mapping action was specified for entry.
CPF5F07 E	Invalid value &2 specified in PDF map action.
CPF8100 E	All CPF81xx messages could be returned. xx is from 01 to FF.
CPF9801 E	Object &2 in library &3 not found.
CPF9802 E	Not authorized to object &2 in &3.
CPF9803 E	Cannot allocate object &2 in library &3.
CPF9808 E	Cannot allocate one or more libraries on library list.
CPF9810 E	Library &1 not found.
CPF9820 E	Not authorized to use library &1.
CPF9838 E	User profile storage limit exceeded.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.
CPF9999 E	Function check. &1 unmonitored by &2 at statement &5, instruction &3.

API introduced: V5R3

---

## Build Open Time Commands (QSPBOPNC) API

Required Parameter Group:

1	Receiver variable	Output	Char(*)
2	Length of receiver variable	Input	Binary(4)
3	Format name	Input	Char(8)
4	Printer device name	Input	Char(10)
5	Printer data stream type	Input	Char(10)
6	Spoiled file attributes	Input	Char(*)
7	Error code	I/O	Char(*)

Default Public Authority: \*USE

Threadsafe: No

The Build Open Time Commands (QSPBOPNC) API uses the attributes of the spoiled file to build commands that must be sent ahead of the data to prepare the printer to print the spoiled file. These commands are referred to as *open time* commands. The open time commands contain most of the file level commands needed to format the printed output.

### Authorities and Locks

The user must have \*USE authority to the printer device description.

### Required Parameter Group

#### Receiver variable

OUTPUT; CHAR(\*)

The receiver variable receives the information requested. When the size of the area is smaller than the format requested, the API returns only the data the area can hold. If the size of the area is greater than the format, the area past the format is undefined.

#### Length of receiver variable

INPUT; BINARY(4)

The length of the receiver variable. The amount of data is truncated if the receiver variable is too small. Anything less than 8 bytes is not valid.

#### Format name

INPUT; CHAR(8)

The format of the information returned in the receiver variable.

The valid format name is:

*"OPNC0100 Format"* on page 22 The open time commands for the spoiled file.

#### Printer device name

INPUT; CHAR(10)

The name of the printer device description of the writer that was started.

The following special value is allowed:

\*NONE The printer device is unknown.

### Print data stream type

INPUT; CHAR(10)

The type of data stream recognized by the printer writer. This is used to determine the type of data to generate for open time command.

The valid special value is:

\*SCS Systems Network Architecture (SNA) character string. A full set of SCS commands (similar to the set sent to a 3812 printer) is generated.

### Spooled file attributes

INPUT; CHAR(\*)

The attributes of the spooled file to be printed. These attributes are obtained by using the Retrieve Spooled File Attributes (QUSRSPPL) API, specifying format SPLA0200.

### Error code

I/O; CHAR(\*)

The structure in which to return error information. For the format of the structure, see Error code parameter.

## OPNC0100 Format

The following table shows the information returned for the OPNC0100 format. For more details about the fields in the following table see, "Field Descriptions."

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Bytes returned
4	4	BINARY(4)	Bytes available
8	8	BINARY(4)	Offset to open time commands
12	C	BINARY(4)	Size of open time commands
		CHAR(*)	Open time commands

## Field Descriptions

**Bytes available.** The number of bytes of data available to be returned. All available data is returned if enough space is provided.

**Bytes returned.** The number of bytes of data returned.

**Offset to open time commands.** The byte offset to the open time commands from the beginning of the format.

**Open time commands.** The open time commands, which contain most of the file level commands to format the output.

**Size of open time commands.** The size of the open time commands, in bytes.

## Error Messages

Message ID	Error Message Text
CPF24B4 E	Severe error while addressing parameter list.
CPF2634 E	Not authorized to object &1.

Message ID	Error Message Text
CPF2702 E	Device description &1 not found.
CPF3CF1 E	Error code parameter not valid.
CPF3C19 E	Error occurred with receiver variable specified.
CPF3C21 E	Format name &1 is not valid.
CPF3C24 E	Length of the receiver variable is not valid.
CPF33E0 E	Incomplete set of attributes provided.
CPF334C E	Value &1 not valid for print data stream type.
CPF8105 E	Device description &4 damaged.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V3R7

Top | "Print APIs," on page 1 | APIs by category

---

## Build Separator Pages (QSPBSEPP) API

Required Parameter Group:

1	Receiver variable	Output	Char(*)
2	Length of receiver variable	Input	Binary(4)
3	Format name	Input	Char(8)
4	Printer device name	Input	Char(10)
5	Printer data stream type	Input	Char(10)
6	Type of separator page	Input	Char(10)
7	Qualified name of separator page program	Input	Char(20)
8	Writer handle	Input	Char(16)
9	Spoiled file handle	Input	Char(10)
10	Drawer for separator page	Input	Binary(4)
11	Error code	I/O	Char(*)

Default Public Authority: \*USE

Threadsafe: No

The Build Separator Pages (QSPBSEPP) API builds the system separator page to be printed for the spooled file. A user exit program is called to build the separator page or the system default is used. The separator page is returned in the receiver variable parameter. This data can then be sent to the printer. This API is only valid when called from a program that was called by a printer writer program. This API is not valid when called from a program that was called by a remote writer program.

If Build Separator Pages (QSPBSEPP) API encounters a problem when running a separator exit program, the default system separator will be returned. No error will be signalled or returned in the error code parameter.

### Authorities and Locks

If a qualified separator page program is specified or implied the user must have \*EXECUTE authority to the program and the library the program is stored in.

The user must have \*USE authority to the printer device description.

### Required Parameter Group

Receiver variable

OUTPUT; CHAR(\*)

The receiver variable that receives the information requested. You can specify the size of the area to be smaller than the format requested as long as you specify the length parameter correctly. As a result, the API returns only the data that the area can hold.

**Length of receiver variable**

INPUT; BINARY(4)

The length of the receiver variable. The amount of data is truncated if the receiver variable is too small. Anything less than 8 bytes is not valid.

**Format name**

INPUT; CHAR(8)

The format of the information returned in the receiver variable.

The valid format name is:

*"SEPP0100 Format"* on page 25      The formatted separator page data.

**Printer device name**

INPUT; CHAR(10)

The name of the printer device description for the writer that was started.

**Print data stream type**

INPUT; CHAR(10)

The type of data stream recognized by the printer writer. This is used to determine the type of data to generate for the separator page.

The valid special value is:

\*SCS      Systems Network Architecture (SNA) character string

**Type of separator**

INPUT; CHAR(10)

The type of separator page to be built.

Valid values are:

\*FILE      A file separator page is returned in the separator page data. This separator is printed before each copy of a spooled file.

\*JOB      A job separator page is returned in the separator page data. This separator is printed between spooled files created by different jobs.

**Qualified name of separator page program**

INPUT; CHAR(20)

The name of the separator page exit program to use to build the separator page. The first 10 characters contain the separator page program name and the second 10 characters contain the name of the library in which the separator page program is located.

The following special values are supported for the program name:

\*SYSDFT      The system default separator page is generated.

\*DEV D      The separator program specified in the device description for the printer is used.

The following special values are supported for the library name:

\*LIBL      The library list of the writer job is used to locate the message file.

\**CURLIB*            The current library for the writer job is used to locate the message file.

### Writer handle

INPUT;CHAR(16)

The handle to the writer job. This handle is provided to the driver program on a writer call to the driver exit program at initialization time (the initialize option).

### Spooled file handle

INPUT;CHAR(10)

The spooled file handle to the current file that is being processed by the writer. This handle is provided to the driver program on a writer call to the driver exit program (using the process file option).

### Drawer for separator page

INPUT; BINARY(4)

The following values are allowed:

- 0            The separator drawer specified in the device description for the printer device name is used.
- 1 - 255     The drawer from which the separator page is printed.

### Error code

I/O; CHAR(\*)

The structure in which to return error information. For the format of the structure, see Error code parameter.

## SEPP0100 Format

The following table shows the information returned for the SEPP0100 format. For more details about the fields in the following table, see "Field Descriptions."

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Bytes returned
4	4	BINARY(4)	Bytes available
8	8	BINARY(4)	Offset to separator page data
12	C	BINARY(4)	Size of separator page data
16	10	CHAR(*)	Separator page data

## Field Descriptions

**Bytes available.** The number of bytes of data available to be returned. All available data is returned if enough space is provided.

**Bytes returned.** The number of bytes of data returned.

**Offset to separator page data.** The byte offset to the separator page data (from the beginning of the format).

**Separator page data.** The separator page data, which includes printer commands and the data to be printed.

**Size of separator page data.** The size of the separator page data, in bytes.

## Error Messages

Message ID	Error Message Text
CPF2207 E	Not authorized to use object &1 in library &3 type *&2.
CPF24B4 E	Severe error while addressing parameter list.
CPF2702 E	Device description &1 not found.
CPF3CF1 E	Error code parameter not valid.
CPF3C19 E	Error occurred with receiver variable specified.
CPF3C21 E	Format name &1 is not valid.
CPF3C24 E	Length of the receiver variable is not valid.
CPF3C40 E	Spooled file &4 not found.
CPF3C90 E	Literal value cannot be changed.
CPF33CC E	No writer found for specified handle &1.
CPF33CD E	No file found for specified handle &1.
CPF3309 E	No files named &1 are active.
CPF334C E	Value &1 not valid for print data stream type.
CPF334D E	Value &1 not valid for type of separator.
CPF334E E	Value &1 not valid for separator drawer.
CPF3344 E	File &1 number &2 no longer in the system.
CPF9810 E	Library &1 not found.
CPF9811 E	Program &1 in library &2 not found.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V3R7

[Top](#) | ["Print APIs," on page 1](#) | [APIs by category](#)

---

## Change Output Queue (QSPCHGOQ) API

Required Parameter Group:

1	Qualified output queue name	Input	Char(20)
2	Changes	Input	Char(*)
3	Error code	I/O	Char(*)

Default Public Authority: \*USE  
Threadsafe: No

The Change Output Queue (QSPCHGOQ) API changes the following attributes of the specified output queue:

- User-defined options
- Qualified user-defined object name and type
- Qualified user driver program name
- Qualified user data transform program name
- User-defined data

## Authorities and Locks

*Output Queue Library Authority*  
\*EXECUTE

*Output Queue Authority*

The caller needs one of the following authorities:

- \*OBJMGT, \*READ, \*ADD, and \*DLT authority to the output queue if the output queue authority to check is data authority (AUTCHK(\*DTAAUT)).

- Owner of the output queue if the output queue authority to check is owner (AUTCHK(\*OWNER)).
- Job control special authority (\*JOBCTL) if the output queue is operator controlled (OPRCTL(\*YES)).
- Spool control special authority (\*SPLCTL).

*Output Queue Lock*

\*EXCLRD

## Required Parameter Group

### Qualified output queue name

INPUT;CHAR(20)

The name of the output queue for which information is changed. The first 10 characters contain the output queue name and the second 10 characters contain the name of the library in which the output queue resides.

The following special values are supported for the library name:

- \*LIBL                    The library list used to locate the output queue.  
 \*CURLIB                The current library for the job is used to locate the output queue.

### Changes

INPUT; CHAR(\*)

The changes to be made to the specified output queue. The information must be in the following format:

*Number of variable length records*            BINARY(4)

Total number of all of the variable length records.

*Variable length records*                    Each variable length record contains a keyword plus its associated new value. Refer to "Format of the Variable Length Record" for the structure of this field.

### Error code

I/O; CHAR(\*)

The structure in which to return error information. For the format of the structure, see Error code parameter.

## Format of the Variable Length Record

The following table shows the field structure of the variable length record. For more detail about the fields in the following table, see "Field Descriptions" on page 28.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Length of variable length record
4	8	BINARY(4)	Key identifier
8	8	BINARY(4)	Length of new value
12	C	CHAR(*)	New value

## Valid Keys

Key	Type	Field
201	CHAR(*)	User-defined options
202	CHAR(30)	Qualified user-defined object name and type
203	CHAR(20)	Qualified user driver program name
204	CHAR(20)	Qualified user data transform program name
205	CHAR(*)	User-defined data

## Field Descriptions

**Key identifier.** The key value associated with the attribute to be changed. Only specific attributes of the output queue can change. Refer to “Valid Keys” for a list of valid keys.

**Length of new value.** The length of the new value to be assigned to the attribute.

**Length of variable length record.** The length of the record.

**New value.** The value to which a specific attribute is to be set.

**Qualified user data transform program name.** The qualified transform program name has two parts (each part is CHAR(10)):

*User data transform program name* The name of the user-specified data transform program that is used by the driver program.

The following values are supported:

\*NONE No data transform program is specified.

*Data transform program name*

The name of the data transform program.

*Data transform program library name* The name of the library that contains the data transform program. The following special values are supported for the library name:

\*LIBL The library list used to locate the data transform program.

\*CURLIB

The current library for the job is used to locate the data transform program.

**Qualified user-defined object name and type.** The qualified user-defined object name and type has three parts (each part is CHAR(10)):

*User-defined object name* The name of the user-defined object that is used by user applications or user-specified programs that process spooled files.

The following values are supported:

\*NONE No user-defined object is specified.

*User-defined object name*

The name of the object defined by the user.

*User-defined object library name* The name of the library that contains the user-defined object.  
The following special values are supported for the library name:

*\*LIBL* The library list used to locate the user-defined object.

*\*CURLIB*

The current library for the job is used to locate the user-defined object.

*User object type* The type of the user-defined object.  
The following values are supported:

*User object type*

The user object type can be:

Data area (\*DTAARA)

Data queue (\*DTAQ)

File (\*FILE)

PSF configuration object (\*PSFCFG)

User index (\*USRIDX)

User queue (\*USRQ)

User space (\*USRSPC)

**Note:** The value must be blanks when the name of the user-defined object is \*NONE

**Qualified user driver program.** The qualified user driver program name has two parts (each part is CHAR(10)):

*User driver program name* The name of the user-specified driver program that is used to process the spooled files on the output queue.

*\*NONE* No user driver program is specified.

*User driver program name*

The name of the user-specified driver program.

The following values are supported:

*User driver program library name* The name of the library that contains the user driver program.

The following special values are supported for the library name:

*\*LIBL* The library list used to locate the user driver program.

*\*CURLIB*

The current library for the job is used to locate the user driver program.

**User-defined data.** Data defined by the user to be used by user applications or user-specified programs that process spooled files.

The following table shows the layout of the user-defined extended data field:

Type	Field
BINARY(4)	Length of user-defined data. The values supported range from 0 through 5000.
CHAR(*)	User-defined data

**User-defined options.** The following table shows the layout of the user-defined options field.

**Note:** To change the user-defined options to \*NONE, set the number of user-defined options to 0.

Type	Field
BINARY(4)	Number of user-defined options The number of user-defined options for the output queue. Specify a number from 0 to 4.
CHAR(*)	User-defined options array An array of options defined by the user to be used by user applications or user-specified programs that process spooled files. Each option is defined as CHAR(10).

## Error Messages

Message ID	Error Message Text
CPF2207 E	Not authorized to use object &1 in library &3 type *&2.
CPF24B4 E	Severe error while addressing parameter list.
CPF3CF1 E	Error code parameter not valid.
CPF3C4D E	Length &1 for key &2 not valid.
CPF3C81 E	Value for key &1 not valid.
CPF3C82 E	Key &1 not valid for API &2.
CPF3319 E	Cannot change parameters specified. A writer is active to output queue &1.
CPF3330 E	Necessary resource not available.
CPF34B1 E	Output queue &1 in library &2 not found.
CPF34D9 E	User defined object type &1 not valid.
CPF34DA E	Output queue &1 in &2 not changed due to errors.
CPF8122 E	&8 damage on library &4.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V3R7

Top | "Print APIs," on page 1 | APIs by category

---

## Convert Image (QIMGCVTI, QimgCvtlmg) API

Required Parameter Group:

1	Control structure	Input	Char(*)
2	Input image structure	Input	Char(*)
3	Main storage input image	Input	Char(*)
4	Output image structure	Input	Char(*)
5	Main storage output image	Output	Char(*)
6	Feedback structure	Output	Char(*)
7	Error code	I/O	Char(*)

Service Program Name: QIMGCIMG

Default Public Authority: \*USE

Threadsafe: No

The Convert Image (OPM, QIMGCVTI; ILE, QimgCvtlmg) API converts an image or print data stream from one format to another, and optionally prints the result. The input and output data streams can reside in a stream file, a spooled file, or main storage. The supported input data stream formats are Graphics Interchange Format (GIF), Tag Image File Format (TIFF), OS/2<sup>®</sup> and Windows<sup>®</sup> Bitmap (BMP), and PostScript Level. The supported output data stream formats are Advanced Function Presentation<sup>™</sup>, Hewlett-Packard Printer Control Language (PCL), and PostScript Level 1.

In addition to the input data stream formats listed above, the following input data stream formats may be supported through the use of a valid registered exit program: Hewlett-Packard Printer Control Language (PCL) and Portable Document Format (PDF). In addition to the output data stream formats listed above, the following output data stream format may be supported through the use of a valid registered exit program: PostScript (including Level 2 and above). By specifying UNKNOWN for the input data stream or the output data stream, the user may also, through the use of a valid registered exit program, transform data streams that are not listed here.

For more information, see the Printing topic collections.

## Authorities and Locks

*Image Configuration Authority*

\*USE

*Image Configuration Lock*

\*SHRRD

*Input Stream File Authority*

\*R

*Input Stream File Open Sharing Mode*

Share with readers

*All Directory in Input Path Authority*

\*X

*Input Spooled File Authority*

See "Input Spooled File Authorities" on page 32.

*Input Spooled File Lock*

\*LSRD

*Font Stream File Authority*

\*R

*Font Stream File Open Sharing Mode*

Share with readers

*All Directory in Font Path Authority*

\*X

*Output Stream File Authority*

\*W

*Output Stream File Open Sharing Mode*

Share with neither readers nor writers

*All Directory in Output Path Authority*

\*X

*Lowest Directory in Output Path Authority*

\*XW

*Printer Device Authority*

\*USE

*Output Queue Authority*

\*USE

*Printer File Authority*

\*USE

*Printer File Lock*

\*SHRNUP

## Input Spooled File Authorities

The requester is authorized to the input spooled file if one or more of the following conditions are met:

- The requester is the owner of the spooled file.
- The requester has \*READ authority to the queue on which the spooled file resides, and the queue is specified as DSPDTA(\*YES).
- The requester has \*SPLCTL authority.
- The requester has \*JOBCTL authority, and the queue on which the spooled file resides is specified as OPRCTL(\*YES).
- The requester has ownership of the queue on which the spooled file resides, and the queue is specified as AUTCHK(\*OWNER).
- The requester has read, add, and delete authorities to the queue on which the spooled file resides, and the queue is specified as AUTCHK(\*DTAAUT).

## Required Parameter Group

### Control structure

INPUT; CHAR(\*)

Information that controls the conversion process. This information does not pertain specifically to either the input or output image or print data streams. See “Control Structure Format” on page 33 for information about the format.

### Input image structure

INPUT;CHAR(\*)

A description of the input image or print data stream. See “Input Image Structure Format” on page 36 for information about the format.

### Main storage input image

INPUT;CHAR(\*)

When the format field of the input image structure is IMGIO300, this parameter contains the input image or print data stream.

When this parameter does not contain the input data stream, it must contain 4 bytes of hexadecimal zeros.

### Output image structure

INPUT;CHAR(\*)

A description of the output image or print data stream and where it is to be written. See “Output Image Structure Format” on page 41 for information about the format.

### Main storage output image

OUTPUT; CHAR(\*)

When the format field of the output image structure is IMGO0300, this parameter is the destination of the converted image or print data stream. **Notes:**

1. The space provided field of the output image structure specifies the available size of this storage area.
2. The output data stream length field of the feedback structure specifies the amount of storage used by the converted data stream.
3. If the space provided field is insufficient to contain the converted data stream, an error condition is returned, and no data stream is returned in this parameter.

### Feedback structure

OUTPUT;CHAR(\*)

Additional output information from the API, such as a spooled file identifier, size of main storage output, or a handle for future API calls for multiple page output. See “Feedback Structure Format” on page 51 for information about the format.

**Error code**

I/O; CHAR(\*)

The structure in which to return error information. For the format of the structure, see Error code parameter.

**Control Structure Format**

The table below shows the organization of the control structure parameter. For detailed descriptions of each field, see “Field Descriptions.”

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Length
4	4	CHAR(8)	Format
12	C	BINARY(4)	Operation
16	10	CHAR(32)	Handle for multipage output
48	30	BINARY(4)	Space provided for feedback structure
52	34	CHAR(8)	Feedback structure format
60	3C	BINARY(4)	Reverse
64	40	BINARY(4)	Color reduction
68	44	BINARY(4)	Resize
72	48	BINARY(4)	Stretch
76	4C	BINARY(4)	Horizontal justify
80	50	BINARY(4)	Vertical justify
84	54	BINARY(4)	Keep color
88	58	BINARY(4)	Keep quality
92	5C	BINARY(4)	Cancel on error
96	60	BINARY(4)	Severity level

**Field Descriptions**

**Cancel on error.** Whether the conversion should be ended when a non catastrophic error condition is detected. When yes is specified, the severity level field may be used to control which errors end the conversion.

Possible values are:

- 0 No
- 1 Yes

**Color reduction.** Whether or not to reduce the output data stream’s pixel information content. If color reduction is set to same, the pixel information content of the output data stream will be either that of the input data stream or the destination device, whichever is less. Setting color reduction to gray scale will

reduce the output pixel information content to gray scale if it would have been color and will otherwise have no effect. Setting color reduction to black and white will force the output to black and white. See “Color Reduction Field” on page 54 for more details.

Possible values are:

- 0 Same
- 1 Gray scale
- 2 Black and white

**Feedback structure format.** The format of the feedback structure. See “Feedback Structure Format” on page 51 for a description of the feedback structure. This field must contain the value IMGF0100.

**Format.** The format of the control structure. This field must contain the value IMGC0100.

**Handle for multipage output.** This parameter is not passed to any transform program registered under exit point QIBM\_QIMG\_TRANSFORMS. The handle that was returned in the feedback structure of a prior call to this API that specified first for the operation field. See “Feedback Structure Format” on page 51 and “Multipage Output” on page 53 for more information.

The handle is required when append, last, or end is specified for the operation field. For all other values of operation, this field must be set to hexadecimal zeros.

**Horizontal justify.** The horizontal position of an image on the page if the image is smaller than the page.

Possible values are:

- 0 Center
- 1 Left
- 2 Right

**Note:** This field has no effect for PostScript input data streams. It also may not work as expected if the input image is larger than the page.

**Keep color.** Whether all coloring in the source data stream must be kept. If the conversion would result in loss of the number of colors or number of gray scale shades, it is ended with an error condition and no converted data stream is produced.

Possible values are:

- 0 No
- 1 Yes

**Note:** This field has no effect for PostScript input data streams.

**Keep quality.** This parameter is not passed to any transform program registered under exit point QIBM\_QIMG\_TRANSFORMS. Whether image or print resolution must be kept. If the conversion would result in loss of resolution, it is ended with an error condition and no converted data stream is produced.

Possible values are:

- 0 No
- 1 Yes

**Note:** This field has no effect for PostScript input data streams or for any other data streams which do not contain resolution information.

**Length.** The length of the control structure. This length must be 52 when the end option is specified for the operation field. In all other cases, this length must be 100, or an error is returned.

**Operation.** Whether or not a conversion is done, and if so, when the conversion is to take place.

Possible values are:

- 0 Calculate. An immediate or delayed conversion is selected, based on the destination of the converted data stream. If the destination is a spooled file, delayed is selected; otherwise, immediate is selected.
- 1 Immediate. The input data stream is converted before control is returned to the API caller.
- 2 Delayed. The input data stream is placed in a spooled file and converted (if necessary) when the spooled file is printed. This value is only valid for output format IMGO0200.
- 3 No conversion. The input data stream is copied to the output destination as is.
- 4 First. A multipage conversion is initiated with the first input data stream.
- 5 Append. A multipage conversion is continued with another input data stream.
- 6 Last. A multipage conversion is ended with the last input data stream.
- 7 End. A multipage conversion is ended with no input data stream.

The calculate option selects the immediate or delayed option, based on the destination of the converted data stream.

The delayed option is only valid when output is spooled. See format IMGO0200 in “Output Image Structure Format” on page 41 for output specifications.

When no conversion is specified for operation and output is spooled to a PCL or PostScript printer, the input data stream is passed through to the printer. The caller must ensure that the input image or print data stream is compatible with the destination printer. Otherwise, results are unpredictable.

When first, append, or last is specified, the input is converted before control is returned to the API caller.

Refer to “Multipage Output” on page 53 for more details about the first, append, last, and end options.

Multipage conversion is not supported when transforming to an exit program.

**Resize.** This parameter is not passed to any transform program registered under exit point QIBM\_QIMG\_TRANSFORMS. Controls resizing of the output print or image data stream relative to the page size. Resize does not change the aspect ratio of the image unless the stretch field is set to 1 (yes).

Possible values are:

- 0 Scale down if needed to fit page
- 1 Keep size
- 2 Keep pixels
- 3 Scale up or down to fit page
- 4 Scale up to fit page.

**Note:** Resize works by changing the resolution in the output data stream and relies on the printer’s ability to support that resolution. Consequently, resize will not work on all printers. Resize has no effect for PostScript input data streams.

**Reverse.** If yes, black becomes white and white becomes black. This parameter has no effect for color output. It also has no effect for color images output as gray on a Printer Control Language (PCL) color printer.

Possible values are:

0	No
1	Yes

**Note:** This can also be done by changing the photometric interpretation of pixels for the input or output data stream, either using image configurations or the photometric interpretation fields that describe the input and output data streams.

**Severity level.** The severity level of error required to end the conversion or print request when yes is specified for the cancel on error field. Valid values are 0 through 99.

**Space provided for feedback structure.** The length of the storage that is provided for the “Feedback Structure Format” on page 51. The length must be either 0, 108, or 130.

**Stretch.** This parameter is not passed to any transform program registered under exit point QIBM\_QIMG\_TRANSFORMS. Whether, when resizing the image, to stretch the image to fit the output size exactly, or to preserve the image aspect ratio. If no, the image aspect ratio will not be changed. If yes, the horizontal and vertical dimensions of the image will be scaled independently and the aspect ratio may change if the image is resized.

Possible values are:

0	No
1	Yes

**Note:** This field has no effect for PCL or Advanced Function Presentation data stream (AFPDS) output or for PostScript input data streams. It only takes effect if the image is resized; therefore this field has no effect when the resize field contains 1 or 2 and possibly for other resize values.

**Vertical justify.** The vertical position of an image on the page if the image is smaller than the page.

Possible values are:

0	Center
1	Top
2	Bottom

**Note:** This field has no effect for PostScript input data streams. It also may not work as expected if the input image is larger than the page.

## Input Image Structure Format

The input image structure describes the input image or print data stream.

The different formats specify the location of the input image or print data stream to be converted: file, spooled file, or main storage. The formats contain many common fields that specify attributes of the input image or print data stream:

- Data stream format
- Photometric interpretation

- Resolution units
- Horizontal resolution
- Vertical resolution

### IMGI0100 Format

Use this format for input from a stream file.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Length
4	4	CHAR(8)	Format
12	C	CHAR(20)	Reserved
32	20	BINARY(4)	Data stream format
36	24	BINARY(4)	Photometric interpretation
40	28	BINARY(4)	Resolution units
44	2C	BINARY(4)	Horizontal resolution
48	30	BINARY(4)	Vertical resolution
52	34	BINARY(4)	Input object type
56	38	BINARY(4)	Offset to input object name
60	3C	BINARY(4)	Length of input object name
64	40	BINARY(4)	Reserved
68	44	CHAR(*)	Input object name

### IMGI0200 Format

Use this format for input from a spooled file.

This format contains fields used in combination to uniquely identify the input spooled file:

- Qualified job name
- Internal job identifier
- Internal spooled file identifier
- Spooled file name
- Spooled file number
- Job system name
- Spooled file create date
- Spooled file create time

See “Open Spooled File (QSPOPNP) API” on page 325 for information about the valid combinations of these fields.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Length
4	4	CHAR(8)	Format
12	C	CHAR(20)	Reserved
32	20	BINARY(4)	Data stream format
36	24	BINARY(4)	Photometric interpretation

Offset		Type	Field
Dec	Hex		
40	28	BINARY(4)	Resolution units
44	2C	BINARY(4)	Horizontal resolution
48	30	BINARY(4)	Vertical resolution
52	34	CHAR(26)	Qualified job name
78	4E	CHAR(16)	Internal job identifier
94	5E	CHAR(16)	Internal spooled file identifier
110	6E	CHAR(10)	Spooled file name
120	78	BINARY(4)	Spooled file number
124	7C	CHAR(8)	Job system name
132	84	CHAR(7)	Spooled file create date
139	8B	CHAR(1)	Reserved
140	8C	CHAR(6)	Spooled file create time

## IMGI0300 Format

Use this format for input from main storage. See the Main storage input image (page 32).

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Length
4	4	CHAR(8)	Format
12	C	CHAR(20)	Reserved
32	20	BINARY(4)	Data stream format
36	24	BINARY(4)	Photometric interpretation
40	28	BINARY(4)	Resolution units
44	2C	BINARY(4)	Horizontal resolution
48	30	BINARY(4)	Vertical resolution
52	34	BINARY(4)	Length of input data stream

## Field Descriptions

**Data stream format.** Format of the input print or image data stream. Possible values are:

- 0 Format is determined from the input data stream.
- 1 TIFF (Tag Image File Format) (Only the first image of a multiple-page TIFF image is transformed.)
- 2 BMP (Bitmap)
- 6 GIF (Graphics Interchange Format) (Only the first image of an animated GIF image is transformed.)
- 22 PCL (Printer Control Language) (Available only if a transform exit program that supports PCL input has been registered to the QIBM\_QIMG\_TRANSFORMS exit point)
- 23 PostScript (PostScript level 2 and above available only if a transform exit program that supports PostScript Level 2 or above has been registered to the QIBM\_QIMG\_TRANSFORMS exit point)
- 25 PDF (Portable Document Format) (Available only if a transform exit program that supports PDF input has been registered to the QIBM\_QIMG\_TRANSFORMS exit point)
- 99 Unknown (Input data stream format unknown)

**Format.** The format of the input image structure, determined by the location of the input data stream.

The following formats are defined:

<i>IMGIO100</i>	Input is from a stream file.
<i>IMGIO200</i>	Input is from a spooled file.
<i>IMGIO300</i>	Input is from main storage.

**Horizontal resolution.** This parameter is not passed to any transform program registered under exit point QIBM\_QIMG\_TRANSFORMS.

The number of horizontal pixels per resolution unit, or the following special value:

0 Resolution determined from the input data stream. If resolution cannot be determined from the data stream, a resolution of 300 dots per inch is used.

**Input object type.** For format IMGIO100, the type of the input object name.

0 Stream file

**Input object name.** For format IMGIO100, the input object name. For stream file object type, this is the full name of the input stream file, including path.

**Internal job identifier.** For format IMGIO200, the internal job identifier for the job that owns the spooled file. This field is used only when the job name component of the qualified job name field is \*INT, and the other two components contain all blanks. For more information, see the List Job (QUSLJOB) API in the Work Management part.

**Internal spooled file identifier.** For structure format IMGIO200, the internal spooled file identifier. This field is used only when the spooled file name is \*INT. For more information, see "List Spooled Files (QUSLSPL) API" on page 279.

**Job system name.** The name of the system where the job that created the spooled file ran or blank when the spooled file name is \*INT.

The possible values are:

<i>*ONLY</i>	There is one job with the specified job name, user name, job number, spooled file name, spooled file number, spooled file creation date, and spooled file creation time.
<i>*CURRENT</i>	The job on the current system with the specified job name, user name, job number, spooled file name, spooled file number, spooled file create date, and spooled file create time is used.
<i>*ANY</i>	The job system name is not considered when selecting a spooled file. Use this value when you want the Spooled file create date and Spooled file create time fields to take precedence over the job system name when selecting a spooled file.
<i>system-name</i>	The name of the system where the job that created the spooled file ran.

When the Length field is less than 146 bytes, the API assumes blanks when the spooled file name is \*INT. When spooled file name is not \*INT, the API assumes \*ONLY.

**Length.** The length of this structure, which varies depending on the format specified.

When the convert field in the control structure specifies the end option, the length must be 4.

For format IMGIO100, the length must be 68 plus the length of the input object name.

For format IMGIO200, the length must be 124 or 146.

For format IMGIO300, the length must be 56.

**Length of input data stream.** For format IMGIO300, the length of the input data stream in main storage.

**Length of input object name.** For format IMGIO100, the length of the input object name.

**Offset to input object name.** For format IMGIO100, the offset in bytes from the start of the Input Image Structure to the input object name. This value must be 68.

**Photometric interpretation.** This parameter is not passed to any transform program registered under exit point QIBM\_QIMG\_TRANSFORMS. The interpretation of the pixel data contained in the input image or print data stream.

Possible values are:

- 0 Photometric interpretation is determined from the input data stream.
- 1 Monochrome where white is zero
- 2 Monochrome where black is zero

**Qualified job name.** For format IMGIO200, the qualified job name for the job that owns the spooled file.

The qualified job name has three parts:

<i>job name</i>	CHAR(10). A specific job name, or one of the following special values: <ul style="list-style-type: none"><li>* The job in which this program is running. The rest of the job name parameter must be blank.</li><li>*INT The internal job identifier that is used to locate the spooled file. The user name and job number must be set to blank.</li></ul>
<i>user name</i>	CHAR(10). A specific user profile name, or blanks when the job name is * or *INT.
<i>job number</i>	CHAR(6). A specific job number, or blanks when the job name is * or *INT.

**Reserved.** Space reserved for future use. This field must be set to hexadecimal zero(s).

**Resolution units.** This parameter is not passed to any transform program registered under exit point QIBM\_QIMG\_TRANSFORMS. The units of measure for the horizontal and vertical resolution fields.

Possible values are:

- 0 Resolution units are determined from the input data stream.
- 1 No resolution is given
- 2 Inch
- 3 Centimeter

**Spooled file create date.** ➤ The date, based on local job time, that the spooled file was created on the system or blank when the spooled file name is \*INT. ⚡ This field is considered after the job name, user name, job number, spooled file name, spooled file number, and job system name field requirements have been met.

The date must be in the CYYMMDD format or one of the following special values:

<i>*ONLY</i>	There is only one spooled file with the specified job name, user name, job number, spooled file name, spooled file number, and job system name.
<i>*LAST</i>	The spooled file with the latest date and time which also has the specified job name, user name, job number, spooled file name, spooled file number, and job system name is used.
<i>date</i>	The date the spooled file was created on the system in the format CYYMMDD. » See field Date file opened (created) in API QUSRSPLA under field « descriptions for more information on the date format.

When the Length field is less than 146 bytes, the API assumes blanks when the spooled file name is \*INT. When spooled file name is not \*INT, the API assumes \*ONLY.

**Spooled file create time.** » The time, in local job time, that the spooled file was created on the system or blank when the spooled file name is \*INT. « This field must be set to blanks when special values \*LAST or \*ONLY are used for field Spooled file create date. This field must have a value set if a date is specified for field Spooled file create date. This field is considered after the job name, user name, job number, spooled file name, spooled file number, job system name, and spooled file create date field requirements have been met. The time must be in the HHMMSS format or one of the following special values:

<i>*ONLY</i>	There is only one spooled file with the specified job name, user name, job number, spooled file name, spooled file number, job system name, and spooled file create date.
<i>*LAST</i>	>The spooled file with the latest time which also has the specified job name, user name, job number, spooled file name, spooled file number, job system name, and spooled file create date is used.
<i>time</i>	The time the spooled file was created on the system in the format HHMMSS. » See field Time file opened (created) in API QUSRSPLA under field « descriptions for more information on the time format.
<i>blanks</i>	This field must be set to blanks if the spooled file create date field is set to *LAST or *ONLY.

When the Length field is less than 146 bytes, the API assumes blanks for this field.

**Spooled file name.** For format IMGIO200, the spooled file name. The special value \*INT means that the internal spooled file identifier is used.

**Spooled file number.** For format IMGIO200, the spooled file number. When the spooled file name is given (that is, not \*INT), this uniquely identifies the spooled file if the job has multiple spooled files with the same name.

The following special values are allowed:

- 0 Use the only spooled value with the specified name.
- 1 Use the highest-numbered spooled file with the specified spooled file name.
- 2 The spooled file number is not used to determine which spooled file to process. Use this value when you want the Job system name field or the Spooled file create date and Spooled file create time fields to take precedence over the spooled file number when selecting a spooled file.

**Vertical resolution.** This parameter is not passed to any transform program registered under exit point QIBM\_QIMG\_TRANSFORMS.

The number of vertical pixels per resolution unit, or the following special value:

- 0 Resolution determined from the input data stream.

## Output Image Structure Format

The output image structure describes the output image or print data stream.

The different formats specify the destination of the converted image or print data stream: file, spooled file, or main storage. The formats contain many common fields that specify attributes of the output image or print data stream:

- Data stream format
- Photometric interpretation
- Resolution units
- Horizontal and vertical resolutions
- Size units
- Horizontal and vertical sizes
- Compression type and quality
- Paper size
- User paper size units
- Horizontal and vertical user paper sizes
- Paper orientation
- Left, right, top, and bottom unprintable borders

These attribute values override any corresponding values present in the destination image configuration object, if specified.

For multipage considerations regarding some fields in this structure, see “Multipage Output” on page 53.

### IMGO0100 Format

Use this format to store output in a stream file.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Length
4	4	CHAR(8)	Format
12	C	CHAR(10)	Destination image configuration
22	16	CHAR(10)	Reserved
32	20	BINARY(4)	Data stream format
36	24	BINARY(4)	Photometric interpretation
40	28	BINARY(4)	Resolution units
44	2C	BINARY(4)	Horizontal resolution
48	30	BINARY(4)	Vertical resolution
52	34	BINARY(4)	Size units
56	38	PACKED(15,5)	Horizontal size
64	40	PACKED(15,5)	Vertical size
72	48	BINARY(4)	Compression type
76	4C	BINARY(4)	Compression quality
80	50	BINARY(4)	Bits per sample
84	54	BINARY(4)	Paper size
88	58	BINARY(4)	User paper size units
92	5C	PACKED(15,5)	Horizontal user paper size
100	64	PACKED(15,5)	Vertical user paper size
108	6C	BINARY(4)	Paper orientation

Offset		Type	Field
Dec	Hex		
112	70	BINARY(4)	Left unprintable border
116	74	BINARY(4)	Right unprintable border
120	78	BINARY(4)	Top unprintable border
124	7C	BINARY(4)	Bottom unprintable border
128	80	BINARY(4)	Output object type
132	84	BINARY(4)	Offset to output object name
136	88	BINARY(4)	Length of output object name
140	8C	BINARY(4)	Reserved
144	90	CHAR(*)	Output object name

## IMGO0200 Format

Use this format to store output in a spooled file.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Length
4	4	CHAR(8)	Format
12	C	CHAR(10)	Destination image configuration
22	16	CHAR(10)	Reserved
32	20	BINARY(4)	Data stream format
36	24	BINARY(4)	Photometric interpretation
40	28	BINARY(4)	Resolution units
44	2C	BINARY(4)	Horizontal resolution
48	30	BINARY(4)	Vertical resolution
52	34	BINARY(4)	Size units
56	38	PACKED(15,5)	Horizontal size
64	40	PACKED(15,5)	Vertical size
72	48	BINARY(4)	Compression type
76	4C	BINARY(4)	Compression quality
80	50	BINARY(4)	Bits per sample
84	54	BINARY(4)	Paper size
88	58	BINARY(4)	User paper size units
92	5C	PACKED(15,5)	Horizontal user paper size
100	64	PACKED(15,5)	Vertical user paper size
108	6C	BINARY(4)	Paper orientation
112	70	BINARY(4)	Left unprintable border
116	74	BINARY(4)	Right unprintable border
120	78	BINARY(4)	Top unprintable border
124	7C	BINARY(4)	Bottom unprintable border
128	80	CHAR(10)	Output device

Offset		Type	Field
Dec	Hex		
138	8A	CHAR(10)	Output queue
148	94	CHAR(10)	Output queue library name
158	9E	CHAR(10)	Printer file
168	A8	CHAR(10)	Printer file library name
178	B2	CHAR(10)	User data
188	BC	BINARY(4)	Copies
192	C0	BINARY(4)	Save
196	C4	BINARY(4)	Request spooled feedback

### IMGO0300 Format

Use this format to store output in main storage. See Main storage output image (page 32).

**Note:** This format is not allowed for multipage output. See “Multipage Output” on page 53 for details.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Length
4	4	CHAR(8)	Format
12	C	CHAR(10)	Destination image configuration
22	16	CHAR(10)	Reserved
32	20	BINARY(4)	Data stream format
36	24	BINARY(4)	Photometric interpretation
40	28	BINARY(4)	Resolution units
44	2C	BINARY(4)	Horizontal resolution
48	30	BINARY(4)	Vertical resolution
52	34	BINARY(4)	Size units
56	38	PACKED(15,5)	Horizontal size
64	40	PACKED(15,5)	Vertical size
72	48	BINARY(4)	Compression type
76	4C	BINARY(4)	Compression quality
80	50	BINARY(4)	Bits per sample
84	54	BINARY(4)	Paper size
88	58	BINARY(4)	User paper size units
92	5C	PACKED(15,5)	Horizontal user paper size
100	64	PACKED(15,5)	Vertical user paper size
108	6C	BINARY(4)	Paper orientation
112	70	BINARY(4)	Left unprintable border
116	74	BINARY(4)	Right unprintable border
120	78	BINARY(4)	Top unprintable border
124	7C	BINARY(4)	Bottom unprintable border
128	80	BINARY(4)	Space provided

## Field Descriptions

**Bits per sample.** This parameter is not passed to any transform program registered under exit point QIBM\_QIMG\_TRANSFORMS. The number of bits per image sample.

The following values are allowed:

- 0 The value in the destination image configuration object is used.
- 1 One bit per sample
- 8 Eight bits per sample

**Bottom unprintable border.** The number of pixels of unprintable area on the bottom border, or the following special value:

- 1 The value in the destination image configuration object is used.

**Compression quality.** This field is not used and must be set to 0.

**Compression type.** The algorithm used to compress the output image or print data stream.

Possible values are:

- 0 The value in the destination image configuration object is used.
- 1 No compression
- 4 CCITT Group 4 Fax
- 7 PackBits
- 8 IBM<sup>®</sup> MMR (modified-modified read)

**Copies.** For format IMG00200, the number of copies to produce when the spooled file is printed. The valid values for the number of copies are 1 through 255, or the special value:

- 0 Use the copies value from the printer file.

**Data stream format.** The format of the output data stream.

- 0 The value in the destination image configuration object is used.
- 20 AFPDS (Advanced Function Presentation data stream)
- 22 PCL (Printer Control Language)
- 23 PostScript Level 1
- 99 Unknown (Output data stream format unknown)

**Destination image configuration.** The value for the image configuration object that contains fields describing the output device and data stream. The string must be filled with blanks. See “Values of Destination Image Configuration” on page 54 for a complete list of available image configuration objects.

Field values from this image configuration will be substituted for the values of other fields in this structure, whenever the special value \*IMGCFG is used for a field.

The following special values are also allowed:

- \*DEV The image configuration for the output device is used. This special value is allowed only for format IMGO0200.
- \*NONE No image configuration is specified. Other fields cannot specify that the value from the destination image configuration object be used.  
**Note:** \*DEV is the recommended value for this field. The destination image configuration and any fields in the output image structure will override the values in the image configuration for the output printer device. Later, if the spooled file is changed to another output queue or another printer, these fields will still be in effect and may cause errors. If this is undesirable, configure the printer with another image configuration.

**Format.** The format identifier of the output image structure, determined by the type of output.

The following formats are defined:

- IMGO0100 Output is to a stream file.
- IMGO0200 Output is to a spooled file.
- IMGO0300 Output is to main storage.

**Horizontal resolution.** The number of horizontal pixels per resolution unit, or the following special value:

- 0 The value in the destination image configuration object is used.

**Horizontal size.** This parameter is not passed to any transform program registered under exit point QIBM\_QIMG\_TRANSFORMS. The size of the image in the horizontal direction, in size units. This is used to determine the size of the output image. If the stretch field in the control structure parameter is no, the aspect ratio is maintained such that the image fits within the specified size. That is, if the vertical direction determines the scale factor, the actual horizontal size may be smaller than specified. If the size units field is -2 (paper size), this field must be zero.

The following special value is allowed:

- 0 The value in the destination image configuration object is used.  
**Note:** This field has no effect for PostScript input data streams.  
**Note:** Output image size is determined by manipulating horizontal and vertical resolutions, not by changing the number of output pixels. Because resolution often depends on printer device capabilities, some sizing may not be possible.

**Horizontal user paper size.** The size of the paper in the horizontal direction, in user paper size units. This field must be set to zero unless the value of the paper size field is -1 (user paper size).

**Left unprintable border.** The number of pixels of unprintable area on the left border, or the following special value:

- 1 The value in the destination image configuration object is used.

**Length.** The length of the structure, which varies depending on the format specified.

When the operation field in the control structure format specifies the end option, this field must be set to 4.

For format IMGO0100, the length must be 144 plus the length of the output object name.

For format IMGO0200, the length must be 200.

For format IMGO0300, the length must be 132.

**Length of output object name.** For format IMGO0100, the length of the output object name.

**Offset to output object name.** For format IMGO0100, the offset in bytes from the start of the output image structure to the input object name. This value must be 144.

**Output device.** For format IMGO0200, the name of the output device. This field is used to get the default device characteristics if a destination image configuration is not given. If the output queue is configured with a remote location, any device attributes will be retrieved on the output queue.

The following special values are allowed:

<i>*NONE</i>	No output device is specified. Either an output queue with a remote location is specified or the destination image configuration is specified explicitly.
<i>*JOB</i>	The output device is taken from the current job.
<i>*USRPRF</i>	The output device is taken from the user profile of the current job.
<i>*SYSVAL</i>	The default output device from the QPRTDEV system value is used.
<i>*WRKSTN</i>	The output device is taken from the work station device description of the current job. If the current job is not interactive, this value is ignored and processing continues as if <i>*SYSVAL</i> were specified.

**Output object name.** For format IMGO0100, the output object name. For stream file object type, this is the full name of the output stream file, including path.

**Output object type.** For format IMGO0100, the type of the output object name.

The possible value follows:

0 Stream file

**Output queue.** For format IMGO0200, the name of the output queue. The first 10 characters contain the output queue and the second 10 characters contain the name of the library in which the output queue resides. For AFPDS output, only valid device output queues are allowed.

The following special values are allowed:

<i>*DEV</i>	The device output queue in QUSRSYS is used, after defaults for the device are resolved. Once the device is determined, its output queue will be used.
<i>*JOB</i>	The output queue is retrieved from the current job.
<i>*PRTF</i>	The output queue is retrieved from the printer file.
<i>*USRPRF</i>	The output queue is retrieved from the user profile of the current job.
<i>*WRKSTN</i>	The output queue is retrieved from the work station device description of the current job. If the current job is not interactive, this value is ignored and processing continues as if <i>*DEV</i> were specified.

**Output queue library name.** The name of the library that contains the output queue.

The following special values are supported for the library name:

<i>*LIBL</i>	The library list is used to locate the output queue.
<i>*CURLIB</i>	The current library for the job is used to locate the output queue.

**Paper orientation.** How the output is oriented on paper.

Possible values follow:

- 1 The value in the destination image configuration object is used.
- 0 Best fit. Portrait will be used unless the input image is wider than it is tall. This value is the same as portrait for PostScript input data streams.
- 1 Portrait. For image input files, the top of each output page is placed on the narrowest dimension of the paper. For PostScript input data streams, the orientation specified in the input data stream is used.
- 2 Landscape. For image input files, the top of each output page is placed on the widest dimension of the paper. For PostScript input data streams, the output is rotated 90 degrees from the orientation specified in the input data stream.
- 3 Portrait, rotated 180 degrees. For PostScript input data streams, the output is rotated 180 degrees from the orientation specified in the input data stream.
- 4 Landscape, rotated 180 degrees. For PostScript input data streams, the output is rotated 270 degrees from the orientation specified in the input data stream.

**Paper size.** The paper size to be used for the output image or print data stream. See the table following the values for dimensions of each size value.

Possible values are:

- 1 The value is specified in the user paper size field.
- 0 The value in the destination image configuration object is used.
- 1 Executive
- 2 Letter
- 3 Legal
- 6 Ledger
- 13 A5
- 26 A4
- 27 A3
- 45 B5
- 46 B4
- 50 CONT80
- 51 CONT132
- 80 Monarch envelope
- 81 COMM10 envelope
- 89 COMM9 envelope
- 90 DL envelope
- 91 C5 envelope
- 100 B5 envelope

Name	Millimeters	Inches
Paper:		
Letter	215.9 x 279.4	8.5 x 11
Legal	215.9 x 355.6	8.5 x 14
Ledger		
Executive	184.2 x 266.7	7.25 x 10.5
A3		
A4	210 x 297	8.27 x 11.7

Name	Millimeters	Inches
A5	148 x 210	5.83 x 8.27
B4		
B5	182 x 257	7.17 x 10.1
CONT80	203.2 x 279.4	8 x 11
CONT132	335.3 x 279.4	13.2 x 11
Envelope:		
Monarch	98.4 x 190.5	3.875 x 7.5
Commercial 9	98.4 x 225.4	3.875 x 8.875
Commercial 10	104.8 x 241.3	4.125 x 9.5
DL	110 x 220	4.33 x 8.66
B5	176 x 250	6.93 x 9.84
C5	162 x 229	6.38 x 9.02

**Photometric interpretation.** This parameter is not passed to any transform program registered under exit point QIBM\_QIMG\_TRANSFORMS. The interpretation of the pixel data that is contained in the output image or print data stream.

Possible values are:

- 0 The value in the destination image configuration object is used.
- 1 Monochrome where white is zero
- 2 Monochrome where black is zero
- 3 RGB (red, green, blue) color

**Printer file.** For format IMGO0200, the name of a printer file.

An IBM supplied printer file, such as QSYSPRT in library QSYS, may be used if desired.

**Printer file library name.** The name of the library that contains the printer file.

The following special values are supported for the library name:

- \*LIBL The library list is used to locate the output queue.
- \*CURLIB The current library for the job is used to locate the output queue.

**Request spooled feedback.** For format IMGO0200, whether spooled file identification feedback is requested. See "Feedback Structure Format" on page 51 for details about the information returned. The information is returned only when it is requested using this field.

Possible values are:

- 0 No
- 1 Yes

**Reserved.** Space reserved for future use. This field must be set to hexadecimal zero(s).

**Resolution units.** This parameter is not passed to any transform program registered under exit point QIBM\_QIMG\_TRANSFORMS. The units of measure of the horizontal and vertical resolution fields.

Possible values follow:

- 0 The value in the destination image configuration object is used.
- 2 Inch
- 3 Centimeter

**Right unprintable border.** The number of pixels of unprintable area on the right border, or the following special value:

- 1 The value in the destination image configuration object is used.

**Save.** For format IMGO0200, whether the spooled file will be saved on the output queue after printing.

Possible values are:

- 0 The value from the printer file is used.
- 1 No
- 2 Yes

**Size units.** This parameter is not passed to any transform program registered under exit point QIBM\_QIMG\_TRANSFORMS. The units of measure of the horizontal and vertical size fields.

Possible values are:

- 2 The value from the paper size fields is used and the horizontal size and vertical size fields must be zero.
- 0 The value in the destination image configuration object is used.
- 1 Pixel
- 2 Inch
- 3 Centimeter

**Space provided.** For format IMGO0300, the number of bytes of main storage provided for the output data stream in the Main storage output image (page 32). If the converted data stream will not fit within the space provided, an error condition is returned.

**Top unprintable border.** The number of pixels of unprintable area on the top border, or the following special value:

- 1 The value in the destination image configuration object is used.

**User data.** For format IMGO0200, up to 10 characters of information to be displayed on the WRKSPLF screen.

The following special value is allowed:

- \**INOBJ* The characters will be determined from the input object.
  - If the input is from a stream file, the first 10 characters of the filename (after the last slash in the path, if any) are used.
  - If the input is from a spooled file, the spooled file name is used.
  - If the input is from main storage, the value is blank.

**User paper size units.** This parameter is not passed to any transform program registered under exit point QIBM\_QIMG\_TRANSFORMS. The units of measure of the horizontal and vertical user paper size fields. This field must be zero unless the value of the paper size field is -1 (user paper size).

Possible values are:

- 0 The value in the destination image configuration object is used.
- 2 Inch
- 4 Millimeter

**Vertical resolution.** The number of vertical pixels per resolution unit, or the following special value:

- 0 The value in the destination image configuration object is used.

**Vertical size.** This parameter is not passed to any transform program registered under exit point QIBM\_QIMG\_TRANSFORMS. The size of the image in the vertical direction, in size units. This is used to determine the size of the output image. If the stretch field in the control structure parameter is no, the aspect ratio is maintained such that the image fits within the specified size. That is, if the horizontal direction determines the scale factor, the actual vertical size may be smaller than specified. If the size units field is -2 (paper size), this field must be zero.

The following special value is allowed:

- 0 The value in the destination image configuration object is used.  
**Note:** This field has no effect for PostScript input data streams.  
**Note:** Output image size is determined by manipulating horizontal and vertical resolutions, not by changing the number of output pixels. Because resolution often depends on printer device capabilities, some sizing may not be possible.

**Vertical user paper size.** The size of the paper in the vertical direction, in user paper size units. This field must be set to zero unless the value of paper size is -1 (user paper size).

## Feedback Structure Format

The feedback structure returns additional output information from the API, such as a spooled file identifier, size of main storage output, or a handle for future API calls for multipage output.

Offset		Type	Field
Dec	Hex		
0	0	CHAR(32)	Handle for multipage output
32	20	CHAR(10)	Spooled file name
42	2A	CHAR(10)	Job name
52	34	CHAR(10)	Job user name
62	3E	CHAR(6)	Job number
68	44	BINARY(4)	Spooled file number
72	48	CHAR(16)	Internal job identifier
88	58	CHAR(16)	Internal spooled file identifier
104	68	BINARY(4)	Output data stream length
108	6C	CHAR(8)	Job system name
116	74	CHAR(7)	Spooled file create date

Offset		Type	Field
Dec	Hex		
123	7B	CHAR(1)	Reserved
124	7C	CHAR(6)	Spooled file create time

This structure contains output data that is returned for any of the following conditions:

- The operation field in the control structure specifies first.  
The handle for multipage output is returned. This returned handle is required as input in the control structure on subsequent API calls where append, last, or end is specified for the operation field.
- The request spooled feedback field in the output image structure specifies yes.  
Spooled file information is returned, if requested, when a spooled file is created by the conversion.
- The output image structure format parameter uses format IMGO0300.  
The length of the output data stream is returned. This is the length of the data stream stored in the main storage output image parameter.

## Field Descriptions

**Handle for multipage output.** When first is specified for the operation field in the control structure format, this field is returned. For all other values of the operation field, this field is set to hexadecimal zeros.

**Internal job identifier.** When spooled feedback is requested, the internal identifier of the job that is creating the spooled file (the job in which this API is called).

**Internal spooled file identifier.** When spooled feedback is requested, the internal identifier of the created spooled file.

**Job name.** When spooled feedback is requested, the name of the job that owns the spooled file (the job in which this API is called).

**Job number.** When spooled feedback is requested, the number of the job that owns the spooled file (the job in which this API is called).

**Job system name.** The name of the system where the job that created the spooled file ran (the job in which this API is called).

**Job user name.** When spooled feedback is requested, the user name of the job that generated the spooled file (the job in which this API is called).

**Output data stream length.** When the format field in the output image structure is IMGO0300, this field contains the length of the output data stream in main storage. This is the length of the data stream that is stored in the main storage output image.

**Spooled file create date.** When spooled feedback is requested, » the date, based on local job time, that the spooled file was created on the system in the format CYYMMDD. See field Date file opened (created) in API QUSRSPPLA under field descriptions for more information on the date format. «

**Spooled file create time.** When spooled feedback is requested, » the time, in local job time, that the spooled file was created on the system in the format HHMMSS. See field Time file opened (created) in API QUSRSPPLA under field descriptions for more information on the time format. «

**Spooled file name.** When spooled feedback is requested, the name of the created spooled file.

**Spooled file number.** When spooled feedback is requested, the number of the created spooled file.

## Multipage Output

For the operation field in the control structure, the first, append, last, and end options are related. These are used to iteratively call the API with multiple input data streams, while producing a single output data stream. The output data stream format, specified on the first call, must be AFPDS (Advanced Function Printing data stream), PostScript, or PCL (Printer Control Language). Calls that use these options must all be done in the same job, and must follow a prescribed sequence:

1. A single first call
2. Any number of append calls
3. A single last or end call

After a first call, only append, last, or end calls are allowed until the sequence is finished by a last or end call. For example, an immediate call following a first call is not allowed.

The end option is like the last option in that it ends the sequence of calls. However, no input data stream processing is done when end is specified; therefore, no new pages are added to the output data stream.

Parameter rules are defined for multipage output, which are summarized in the Parameters for Multipage Output (page 53) table below. In particular:

1. When the first option is specified, a handle is returned in the feedback structure. This handle must be specified in the control structure on all subsequent API calls that specify the append, last, and end options.
2. When the append or last options are specified, the following field values in the output image structure are ignored. These fields are as follows:
  - Length
  - Format
  - Destination image configuration
  - Data stream format
  - Paper size
  - User paper size units
  - Horizontal user paper size
  - Vertical user paper size
  - Output object type (format IMGO0100)
  - Length of output object name (format IMGO0100)
  - Output object name (format IMGO0100)
  - Output device (format IMGO0200)
  - Output queue (format IMGO0200)
  - Printer file (format IMGO0200)
  - User data (format IMGO0200)
  - Copies (format IMGO0200)
  - Save (format IMGO0200)
  - Request spooled feedback (format IMGO0200)If one of these field values changes, the value from the first call is used.
3. When the end option is specified, the length field in both the input image structure and the output image structure must be set to 4, and the length field in the control structure must be set to 52. Again, this is because no other information is accepted in these parameters.

## Parameters for Multipage Output

Parameter No.	First	Append	Last	End
1	REQ	REQ	REQ	LEN=52
2	REQ	REQ	REQ	LEN=4
3	REQ	REQ	REQ	N/A
4	REQ	REQ	REQ	LEN=4
5	N/A	N/A	N/A	N/A
6	REQ	LEN=0	LEN=0	LEN=0
7	REQ	REQ	REQ	REQ

**Legend:**

*REQ*                Required  
*OPT*                Optional  
*LEN=n*              Length of structure must be n  
*N/A*                 Not applicable (ignored)

**Color Reduction Field**

The color reduction field of the control structure determines the characteristics of the output data stream according to the following table. In this table, the input image type is the type of the actual input image file. The output image type is the type of the data that is printed or stored in a file. The destination image type is the type of output file that is described in the output image structure, determined by the two fields, photometric interpretation and bits per sample. A — (dash) means that the value does not affect the result for that combination.

Input Image Type	Destination			Color Reduction	Output Image Type
	Image Type	Photometric Interpretation	Bits per Sample		
Black and White	-	-	-	-	Black and White
-	Black and White	1 or 2	1	-	Black and White
-	-	-	-	Black and White	Black and White
Color	Color	3	8	Same	Color
Color	Color	3	8	Gray Scale	Gray Scale
Color	Gray Scale	1 or 2	8	Same	Gray Scale
Color	Gray Scale	1 or 2	8	Gray Scale	Gray Scale
Gray Scale	Color	3	8	Same	Gray Scale
Gray Scale	Color	3	8	Gray Scale	Gray Scale
Gray Scale	Gray Scale	1 or 2	8	Same	Gray Scale
Gray Scale	Gray Scale	1 or 2	8	Gray Scale	Gray Scale

**Values of Destination Image Configuration**

The following special values are allowed for the destination image configuration field of parameter 4. Each special value is described in terms of the data streams supported, the maximum resolution in dots per inch (dpi), and whether the printer has color or supports compression.

### Printers supporting PCL data streams

*IMGA01	PCL 300-dpi printer
*IMGA02	PCL 600-dpi printer
*IMGA03	PCL 1200-dpi printer
*IMGA04	PCL 300-dpi color printer
*IMGA05	PCL 600-dpi color printer
*IMGA06	PCL 1200-dpi color printer
*IMGA07	PCL 75-dpi printer (No compression)
*IMGA08	PCL 600-dpi color printer with larger no-print border
*IMGA09	PCL 300-dpi printer (No compression)

### Printers supporting PostScript data streams

*IMGB01	PostScript 300-dpi printer
*IMGB02	PostScript 600-dpi printer
*IMGB03	PostScript 1200-dpi printer
*IMGB04	PostScript 300-dpi color printer
*IMGB05	PostScript 600-dpi color printer
*IMGB06	PostScript 1200-dpi color printer
*IMGB07	PostScript 600x300-dpi color printer
*IMGB08	PostScript 1200x300-dpi color printer
*IMGB09	PostScript 360-dpi color printer
*IMGB10	PostScript 720-dpi color printer
*IMGB11	PostScript 1440x720-dpi color printer
*IMGB12	PostScript 400-dpi printer
*IMGB13	PostScript 800-dpi color printer
*IMGB14	PostScript 600-dpi color printer with larger no-print border
*IMGB15	PostScript 300-dpi color printer with larger no-print border

### Printers supporting IPDS™ data streams

*IMGC01	IPDS 240-dpi printer
*IMGC02	IPDS 300-dpi printer
*IMGC03	IPDS 600-dpi printer
*IMGC04	IPDS 1200-dpi printer
*IMGC05	IPDS 240-dpi printer with no-print border
*IMGC06	IPDS 300-dpi printer with no-print border
*IMGC07	IPDS 600-dpi printer with no-print border
*IMGC08	IPDS 1200-dpi printer with no-print border
*IMGC09	IPDS 240-dpi printer (IM/1 image)
*IMGC10	IPDS 240-dpi printer (IM/1 image) with no-print border
*IMGC11	IPDS 240-dpi printer (CCITT G4)

### Printers supporting PCL and PostScript data streams

*IMGD01	PCL/PostScript 300-dpi printer
*IMGD02	PCL/PostScript 600-dpi printer

*IMGD03	PCL/PostScript 1200-dpi printer
*IMGD04	PCL/PostScript 300-dpi color printer
*IMGD05	PCL/PostScript 600-dpi color printer
*IMGD06	PCL/PostScript 1200-dpi color printer
*IMGD07	PCL 300-dpi/PostScript 600-dpi printer
*IMGD08	PCL 300-dpi/PostScript 1200-dpi printer
*IMGD09	PCL 600-dpi/PostScript 300-dpi printer
*IMGD10	PCL 600-dpi/PostScript 1200-dpi printer
*IMGD11	PCL/PostScript 600-dpi color printer with larger no-print border

The recommended image configuration objects are listed below for some common printers.

*IMGD01	Compaq Pagemarc 20
*IMGA01	Epson EPCL-4 Printer
*IMGA02	Epson EPCL-5 Printer
*IMGB10	Epson Stylus Photo with PostScript
*IMGB11	Epson Stylus Color 600, 800 with PostScript
*IMGA04	HP Color LaserJet 5
*IMGD04	HP Color LaserJet 5M
*IMGA04	HP DeskJet 560C, 820C, 1200C
*IMGA01	HP DeskJet 500, 600, 1200
*IMGA04	HP DeskJet 1600C, 1600CN
*IMGD04	HP DeskJet 1600CM
*IMGA09	HP LaserJet II, IID, IIP
*IMGB01	HP LaserJet II, IID, IIP with PostScript
*IMGA01	HP LaserJet III, IIID, IIISi, 4L
*IMGD01	HP LaserJet III, IIID, IIISi, 4L with PostScript
*IMGA02	HP LaserJet 4, 4P, 4V, 4Si, 4 Plus
*IMGD02	HP LaserJet 4M, 4MP, 4MV, 4Si MX, 4M Plus
*IMGA02	HP LaserJet 5, 5P, 5Si
*IMGD02	HP LaserJet 5M, 5MP, 5Si MX
*IMGA02	HP LaserJet 6, 6P, 6L
*IMGD02	HP LaserJet 6M, 6MP
*IMGD02	IBM 3112, 3116 Page Printer with IPDS feature
*IMGA02	IBM 3112, 3116 Page Printer (ASCII/LAN)
*IMGD02	IBM 3112, 3116 Page Printer with PostScript
*IMGC01	IBM 3130, 3160-1 AF Printer (240-pel mode)
*IMGC02	IBM 3130 AF Printer (300-pel mode)
*IMGC09	IBM 3825, 3827, 3828 AF Printer
*IMGC01	IBM 3825, 3827, 3828 AF Printer (with AFIG)
*IMGC01	IBM 3829 AF Printer
*IMGC10	IBM 3835-001 AF Printer
*IMGC05	IBM 3835-001 AF Printer (with AFIG)
*IMGC05	IBM 3835-002, 3900 AF Printer
*IMGA01	IBM 3912, 3916 Page Printer (ASCII/LAN)
*IMGC06	IBM 3912, 3916 Page Printer with IPDS feature (twinax)
*IMGA01	IBM 3930-03 Page Printer
*IMGD01	IBM 3930-03 Page Printer with PostScript
*IMGC02	IBM 3935 AF Printer
*IMGA09	IBM 4019 LaserPrinters (HP mode)
*IMGB01	IBM 4019 LaserPrinters with PostScript
*IMGC06	IBM 4028 LaserPrinters
*IMGA01	IBM 4029 LaserPrinters

*IMGB02	IBM 4029 LaserPrinters with PostScript
*IMGA01	IBM 4039 LaserPrinters
*IMGD07	IBM 4039 LaserPrinters with PostScript
*IMGA02	IBM 4049 LaserPrinters
*IMGD02	IBM 4049 LaserPrinters with PostScript
*IMGB09	IBM 4079 Color Jetprinter PS
*IMGB05	IBM 4303 Network Color Printer
*IMGC06	IBM 4312, 4317, 4324 NP with IPDS feature (twinax)
*IMGC06	IBM 4312, 4317, 4324 NP with IPDS feature (LAN)
*IMGA02	IBM 4312, 4317, 4324 NP (ASCII/LAN)
*IMGD02	IBM 4312, 4317, 4324 NP with PostScript (ASCII/LAN)
*IMGC03	IBM InfoPrint 60
*IMGC05	IBM InfoPrint 62 Model 2
*IMGC06	IBM InfoPrint 62 Model 3
*IMGB05	IBM InfoColor <sup>®</sup> 70
*IMGC05	IBM InfoPrint 4000
*IMGC06	IBM InfoPrint 4000 High Resolution
*IMGB02	Lexmark 4039Plus
*IMGD11	Lexmark Optra C Color Printer
*IMGA02	Lexmark Optra E, E+
*IMGD02	Lexmark Optra N
*IMGD02	Lexmark Optra R+, Rx+, Lx+, Lxn+
*IMGD02	Lexmark Optra S Printers
*IMGD05	Lexmark Optra SC Color Printer
*IMGA01	Okidata OL400 LED Page Printer
*IMGA02	Okidata OL800, OL810 LED Page Printers
*IMGB12	QMS 2025, 3225
*IMGD04	QMS Magicolor CX
*IMGB09	Tektronix Phaser 140
*IMGB04	Tektronix Phaser 300
*IMGB05	Tektronix Phaser 400
*IMGB05	Tektronix Phaser 540, 550
*IMGB06	Tektronix Phaser 560
*IMGA01	Xerox 4219/MRP
*IMGA02	Xerox 4220/MRP
*IMGA02	Xerox 4230 DocuPrinter
*IMGA02	Xerox 4512, 4517 Network Printer
*IMGB13	Xerox 4520mp Printer
*IMGD04	Xerox 4700 II Color Document Printer
*IMGB08	Xerox 4915 Color Laser Printer
*IMGB05	Xerox 4920, 4925 Color Laser Printer

## Error Messages

Message ID	Error Message Text
CPF24B4 E	Severe error while addressing parameter list.
CPF3330 E	Necessary resource not available.
CPF333B E	Job system name is not valid.
CPF333C E	Spooled file create date is not valid.
CPF333D E	Spooled file create time is not valid.
CPF333E E	Spooled file create time is not blank.
CPF333F E	Job system name is not blank.
CPF3342 E	Job &5/&4/&3 not found.
CPF3343 E	Duplicate job names found.

Message ID	Error Message Text
CPF3344 E	File &1 number &2 no longer in the system.
CPF335B E	Spooled file create date is not blank.
CPF34B1 E	Output queue &1 in library &2 not found.
CPF33DD E	Maximum number of open spooled files exceeded for this job.
CPF33DE E	Size of internal data for opened spooled file exceeds maximum.
CPF33DF E	Internal data area for opened spooled files destroyed.
CPF3492 E	Not authorized to spooled file.
CPF3C1D E	Length specified in parameter &1 not valid.
CPF3C1E E	Required parameter &1 omitted.
CPF3C21 E	Format name &1 is not valid.
CPF3C33 E	Spooled file number &1 is not valid.
CPF3C39 E	Value for reserved field not valid.
CPF3C40 E	Spooled file &4 not found.
CPF3C41 E	More than one spooled file with same name.
CPF3C90 E	Literal value cannot be changed.
CPF3CF1 E	Error code parameter not valid.
CPF7AAB E	Problem &1 not found.
CPF9801 E	Object &2 in library &3 not found.
CPF9802 E	Not authorized to object &2 in &3.
CPF9803 E	Cannot allocate object &2 in library &3.
CPF9804 E	Object &2 in library &3 damaged.
CPF9810 E	Library &1 not found.
CPF9814 E	Device &1 not found.
CPF9818 E	Object &2 in library &3 not created.
CPF9819 E	Object &2 in library &3 not created.
CPF9820 E	Not authorized to use library &1.
CPF9838 E	User profile storage limit exceeded.
CPF9845 E	Error occurred while opening file &1.
CPF9871 E	Error occurred while processing.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.
CPFA094 E	Path name not specified.
CPFA09C E	Not authorized to object.
CPFA09E E	Object in use.
CPFA09F E	Object damaged.
CPFA0A1 E	An input/output error occurred.
CPFA0A2 E	Information passed to this operation was not valid.
CPFA0A3 E	Path name resolution causes looping.
CPFA0A4 E	Too many open files for process.
CPFA0A5 E	Too many open files.
CPFA0A7 E	Path name too long.
CPFA0AA E	Error occurred while attempting to obtain space.
CPFA0AB E	Object name not a directory.
CPFA0AD E	Function not supported by file system.
CPFA0B1 E	Requested operation not allowed. Access problem.
CPFA0C4 E	Object name not a file.
CPFA0DA E	Object name is a directory.
CPFB530 E	Input object name field not valid.
CPFB531 E	Main storage input image parameter not valid.
CPFB532 E	Output data stream not allowed for operation option.
CPFB533 E	Main storage output not allowed for operation option.
CPFB534 E	API call sequence not valid.
CPFB535 E	Space provided for feedback structure not valid or missing.
CPFB536 E	Handle field in parameter &1 not valid or missing.
CPFB537 E	Error found in parameter &1 at offset &2.

<b>Message ID</b>	<b>Error Message Text</b>
CPFB538 E	Error found in parameter &1 at offset &2.
CPFB539 E	Error found in parameter &1 at offset &2.
CPFB53A E	Error found in parameter &1 at offset &2.
CPFB53B E	Output change not allowed for multipage request.
CPFB53C E	Main storage output overlaps main storage input.
CPFB540 E	Device &1 for spooled file output not valid.
CPFB54A E	Image configuration &1 is logically damaged.
CPFB555 E	Compression quality not valid for compression type.
CPFB556 E	Compression type not valid.
CPFB557 E	Error occurred compressing the data stream.
CPFB558 E	Error occurred decompressing the data stream.
CPFB565 E	Input data stream not valid.
CPFB566 E	Data read not complete.
CPFB567 E	Input image too small.
CPFB568 E	Attribute value not supported for data stream format.
CPFB569 E	Main storage allocation failure occurred.
CPFB56A E	LZW minimum-code-length field not valid.
CPFB56B E	Error opening input object.
CPFB56C E	Error opening output object.
CPFB56D E	Color reduction not allowed.
CPFB56E E	Gray-scale reduction not allowed.
CPFB56F E	Resolution quality reduction not allowed.
CPFB570 E	Decimation table error occurred.
CPFB571 E	Error occurred writing AFP output.
CPFB572 E	Output data stream not supported.
CPFB573 E	File type &1 not supported.
CPFB574 E	Error reading GIF block.
CPFB575 E	Image filter error.
CPFB576 E	General data stream I/O error occurred.
CPFB577 E	Spooled file I/O error occurred.
CPFB578 E	Unexpected end of file detected in input data stream.
CPFB579 E	Output buffer too small.
CPFB57A E	Source photometric interpretation not supported.
CPFB57E E	An error occurred while transforming data.
CPFB57F E	Combination of input and output data stream formats not supported.
CPFB5C9 E	Operation would suspend the process.
CPFB5CA E	Input/Output error in file &1.
CPFB5CB E	System resource not available.
CPFB5CC E	File is read only.
CPFB5CD E	Function ended due to unexpected error.
CPFB5CE E	Function ended due to unexpected error.
CPFB5CF E	Function ended due to unexpected error.
CPIB530 I	Unable to delete temporary file &2 in library &1.
CPIB531 I	Different input data stream attribute recognized.
CPIB532 I	Input resolution not supported.
CPIB533 I	Output resolution not supported.
CPIB534 I	Color palette not found in image data stream header.
CPIB535 I	Unexpected end of file.
CPIB536 I	Unrecognized tag in TIFF file header.
CPIB537 I	Unknown GIF extension detected.
CPIB538 I	GIF end of file not found.
CPIB539 I	Output resolution changed.

API Introduced: V4R2

## Copy AFPDS Resource (QGSCPYRS) API

Required Parameter Group:

1	Qualified user space name	Input	Char(20)
2	Format name	Input	Char(8)
3	Qualified resource name	Input	Char(20)
4	Resource object type	Input	Char(10)
5	Error code	I/O	Char(*)

Default Public Authority: \*USE

Threadsafe: No

The Copy AFPDS Resource (QGSCPYRS) API puts an AFPDS data stream equivalent of the specified AFPDS resource into the specified user space.

### Authorities and Locks

*User Space Authority*

\*CHANGE

*User Space Library Authority*

\*EXECUTE

*Resource Authority*

\*USE

*Library Authority*

\*USE

*User Space Lock*

\*EXCLRD

*Resource Lock*

\*SHRRD

### Required Parameter Group

**Qualified user space name**

INPUT; CHAR(20)

The user space that the copy of the AFPDS resource is inserted into. The first 10 characters contain the user space name and the second 10 characters contain the library name.

You can use these special values for the library name:

\*CURLIB            The job's current library

\*LIBL              The library list

**Format name**

INPUT; CHAR(8)

The format and content of the information returned in the user space.

You must specify the following:

CPYR0100           Information about the resources, and the resource AFPDS data stream.

For more information about the CPYR0100 format, see Format CPYR0100 (page 62).

**Qualified resource name**

INPUT; CHAR(20)

The name of the AFPDS resource to be copied. The first 10 characters contain the resource name and the second 10 characters contain the library name. The user must specify the library name.

The following special values can be used for the library name:

*\*CURLIB*            The job's current library.  
*\*LIBL*                The library list.

**Resource object type**

INPUT; CHAR(10)

The object type of the AFPDS resource.

The following values are allowed:

*\*PAGSEG*            Page segment  
*\*PAGDFN*            Page definition  
*\*OVL*                Overlay  
*\*FORMDF*            Form definition  
*\*FNTRSC*            Font resource

**Error code**

I/O; CHAR(\*)

The structure in which to return error information. For the format of the structure, see Error code parameter.

**Format of the User Space**

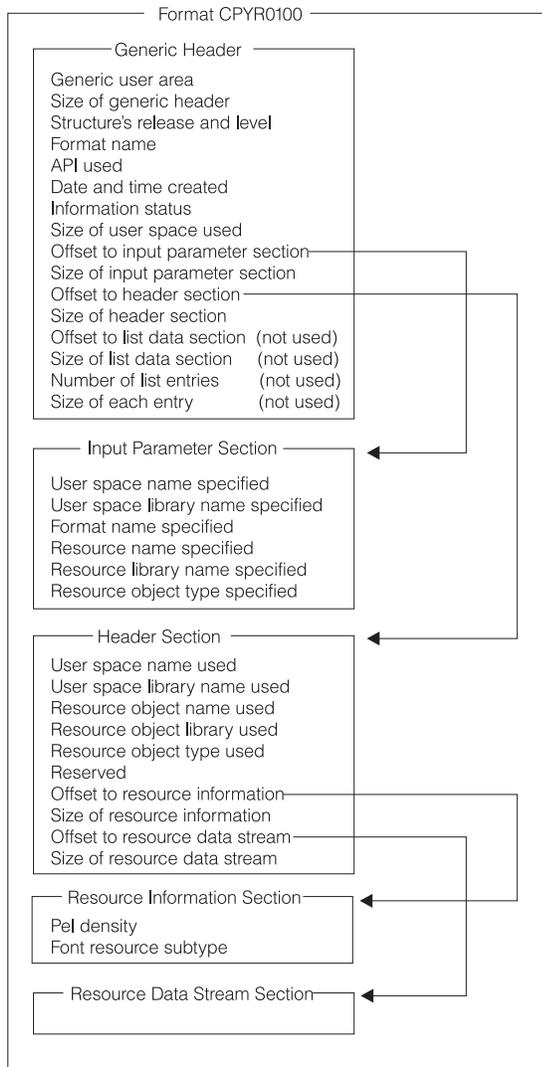
The format of the user space consists of:

- A user area
- A generic header
- An input parameter section
- A header section
- A resource information section
- A resource data stream section

For details about the user area and generic header fields, see User spaces. The organization of the user space after the header section is dependent on format CPYR0100.

Format CPYR0100 (page 62) shows the general structure of the user space for format CPYR0100.

Offset values are calculated from the beginning of the user space.



Format CPYR0100

RBAFX567-0

## Input Parameter Section

For more details about the fields, see "Field Descriptions" on page 63.

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	User space name specified
10	A	CHAR(10)	User space library name specified
20	14	CHAR(8)	Format name specified
28	1C	CHAR(10)	Resource name specified
38	26	CHAR(10)	Resource library name specified
48	30	CHAR(10)	Resource object type specified

## Field Descriptions

**Format name specified.** The format of the user space specified.

**Resource library name specified.** The name of the library name of the AFPDS resource specified.

**Resource name specified.** The name of the AFPDS resource specified.

**Resource object type specified.** The object type of the AFPDS resource specified.

**User space library name specified.** The library name of the user space specified.

**User space name specified.** The name of the user space specified.

## Header Section

The following table shows the header information returned for the CPYR0100 format.

For more details about the fields, see “Field Descriptions.”

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	User space name used
10	A	CHAR(10)	User space library name used
20	14	CHAR(10)	Resource object name used
30	1E	CHAR(10)	Resource object library used
40	28	CHAR(10)	Resource object type used
50	32	CHAR(2)	Reserved
52	34	BINARY(4)	Offset to resource information
56	38	BINARY(4)	Size of resource information
60	3C	BINARY(4)	Offset to resource data stream
64	40	BINARY(4)	Size of resource data stream

## Field Descriptions

**Offset to resource data stream.** Locates the data for the resource data stream returned. The offset value is calculated from the beginning of the user space.

**Offset to resource information.** Locates the data for the resource information returned. The offset value is calculated from the beginning of the user space. This value is zero when the resource object type requested is not \*FNTRSC.

**Reserved.** Reserved for byte alignment.

**Resource object library name used.** The name of the library of the AFPDS resource used to retrieve the resource.

**Resource object name used.** The name of the AFPDS resource used to retrieve the resource.

**Resource object type used.** The object type of the AFPDS resource used to retrieve the resource.

**Size of resource data stream.** The size, in bytes, of the resource data stream section.

**Size of resource information.** The size, in bytes, of the resource information section. This value is zero when the requested resource object type is not \*FNTRSC.

**User space library name used.** The library name of the user space used.

**User space name used.** The name of the user space used.

## Resource Information Section

For more details about the fields, see “Field Descriptions.”

Offset		Type	Field
Dec	Hex		
0	0	CHAR(1)	Pel density
1	1	CHAR(10)	Font resource subtype

## Field Descriptions

**Font resource subtype.** The type of font resource returned in the user space.

Valid values are:

*FNTCHRSET*      Font character set  
*CDEPAG*         Code page  
*CDEFNT*         Coded font

**Pel density.** The pel density of the font resource returned in the user space.

Valid values are:

0      Undefined  
1      240 pel density  
2      300 pel density  
3      Adobe Type 1 outline font.  
4      Adobe Type 0 composite outline font.  
5      600 pel density

## Resource Data Stream Section

The resource data stream section contains the AFPDS data stream of the resource that is to be copied.

## Error Messages

Message ID	Error Message Text
CPF2150 E	Object information function failed.
CPF2151 E	Operation failed for &2 in &1 type *&3.
CPF24B4 E	Severe error while addressing parameter list.
CPF3CF1 E	Error code parameter not valid.
CPF3C21 E	Format name &1 is not valid.
CPF33CF E	Qualified resource name not valid.
CPF34CE E	Value &1 not valid for resource object type.
CPF34C5 E	The AFPDS resource is too large for the user space &1.

Message ID	Error Message Text
CPF811A E	User space &4 in &9 damaged.
CPF9801 E	Object &2 in library &3 not found.
CPF9802 E	Not authorized to object &2 in &3.
CPF9803 E	Cannot allocate object &2 in library &3.
CPF9807 E	One or more libraries in library list deleted.
CPF9808 E	Cannot allocate one or more libraries on library list.
CPF9810 E	Library &1 not found.
CPF9820 E	Not authorized to use library &1.
CPF9830 E	Cannot assign library &1.
CPF9838 E	User profile storage limit exceeded.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V3R7

Top | "Print APIs," on page 1 | APIs by category

---

## Create PDF Map (QPQCRTPM API)

Required Parameter Group:

1	Qualified PDF map name	Input	Char(20)
2	Public authority	Input	Char(10)
3	Text description	Input	Char(50)
4	Error code	I/O	Char(*)

Default Public Authority: \*USE  
Threadsafe: Yes

The Create PDF Map (QPQCRTPM) API creates a PDF map.

The PDF map allows you to define mapping entries (using the Add PDF Map Entry (QPQAPME) API) for spooled files you want to be processed after they have been converted to PDF by PSF/400.

## Authorities and Locks

*Library Authority*

\*READ and \*ADD.

## Required Parameter Group

**Qualified PDF map name**  
INPUT; CHAR(20)

The PDF Map you are creating. The qualified PDF Map name has two parts. The first 10 characters contain the PDF Map name.

The second 10 characters contain the name of the library in which the PDF Map is located.

The following special values are allowed:

\*CURLIB      The job's current library is used to locate the PDF Map. If no library is specified as the current library for the job, the QGPL library is used.

Specify the name of the library where you want to create the PDF Map.

**Public authority**  
INPUT; CHAR(10)

The authority you give to users who do not have specific private or group authority to the PDF map. Once the PDF Map has been created, its public authority stays the same when it is moved to another library or restored from backup media.

The valid values for this parameter are:

<i>*ALL</i>	The user can perform all authorized operations on the PDF map.
<i>Authorization list name</i>	The PDF map is secured by the specified authorization list, and its public authority is set to *AUTL. The specified authorization list must exist on the system when this API is issued. If the list does not exist, the create process fails, and an error message is returned to the application.
<i>*CHANGE</i>	The user has read, add, update, and delete authority for the PDF map and can read the object description.
<i>*EXCLUDE</i>	The user cannot access the PDF map in any way.
<i>*LIBCRTAUT</i>	The public authority for the PDF map is taken from the CRTAUT value for the target library when the object is created. If the CRTAUT value for the library changes later, that change does not affect PDF mapping objects already created. If the CRTAUT value contains an authorization list name and that authorization list secures an object, do not delete the list. If you do, the next time you call this API with the *LIBCRTAUT parameter, it will fail.
<i>*USE</i>	The user can read the object description and contents but cannot change the PDF map.

### Text description

INPUT; CHAR(50)

A brief description of the PDF map.

### Error code

I/O; CHAR(\*)

The structure in which to return error information. For the format of the structure, see Error code parameter.

## Error Messages

Message ID	Error Message Text
CPF2143 E	Cannot allocate object &1 in &2 type *&3.
CPF2144 E	Not authorized to &1 in &2 type *&3.
CPF2283 E	Authorization list &1 does not exist.
CPF3CF1 E	Error code parameter not valid.
CPF3C36 E	Number of parameters, &1, entered for this API was not valid.
CPF5F0A E	PDF map &2 not created.
CPF5F0B D	Value &1 for authority parameter is not valid.
CPF9810 E	Library &1 not found.
CPF9820 E	Not authorized to use library &1.
CPF9830 E	Cannot assign library &1.
CPF9838 E	User profile storage limit exceeded.
CPF9870 E	Object &2 type *&5 already exists in library &3.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V5R3

[Top](#) | [Print APIs](#) | [APIs by category](#)

---

## Delete PDF map object (QPQDLTPM) API

Required Parameter Group:

1	Qualified PDF map object name	Input	Char(20)
2	Error code	I/O	Char(*)

Default Public Authority: \*USE  
Threadsafe: Yes

The Delete PDF map object (QPQDLTPM) API deletes a PDF map object created with the Create PDF map object (QPQCRTPM) API.

The QPQDLTPM API performs the same function as the Delete PDF map object (DLTPDFMAP) command.

## Authorities and Locks

*Library Authority*  
\*EXECUTE

*PDF map object Authority*  
\*OBJEXIST

*PDF map object Lock*  
\*EXCL

## Required Parameter Group

**Qualified PDF map object name**  
INPUT; CHAR(20)

The name of the PDF map object and the name of the library in which it resides. The first 10 characters contain the PDF map object name, and the second 10 characters contain the library name.

You can use these special values for the library name:

*ALL	All libraries
*ALLUSR	All user-defined libraries, plus libraries containing user data and having names starting with Q. For information on the libraries included, see *ALLUSR in Generic library names.
*CURLIB	The job's current library
*LIBL	The library list
*USRLIBL	The user portion of the job's library list

**Error code**  
I/O; CHAR(\*)

The structure in which to return error information. For the format of the structure, see Error code parameter.

## Error Messages

Message ID	Error Message Text
CPF2105 E	Object &1 in &2 type *&3 not found.
CPF2110 E	Library &1 not found.
CPF2113 E	Cannot allocate library &1.
CPF2114 E	Cannot allocate object &1 in &2 type *&3.
CPF2125 E	No objects deleted.
CPF2176 E	Library &1 damaged.
CPF2182 E	Not authorized to library &1.
CPF2189 E	Not authorized to object &1 in &2 type *&3.
CPF24B4 E	Severe error while addressing parameter list.
CPF3C90 E	Literal value cannot be changed.
CPF3CF1 E	Error code parameter not valid.

Message ID	Error Message Text
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V5R3

[Top](#) | [Print APIs](#) | [APIs by category](#)

---

## Extract Writer Status (QSPEXTWI) API

Required Parameter Group:

1	Receiver variable	Output	Char(*)
2	Length of receiver variable	Input	Binary(4)
3	Format name	Input	Char(8)
4	Writer handle	Input	Char(16)
5	Spooled file handle	Input	Char(10)
6	Error code	I/O	Char(*)

Default Public Authority: \*USE  
 Threadsafes: No

The Extract Writer Status (QSPEXTWI) API is used by a driver program to access information about the writer, the spooled file the writer is processing, and to determine if the information for the spooled file has been changed. A call to this API should be made by a driver program after each buffer is sent to the printer.

### Authorities and Locks

None.

### Required Parameter Group

#### Receiver variable

OUTPUT; CHAR(\*)

The receiver variable that receives the information requested. You can specify the size of the area to be smaller than the format requested as long as you specify the length parameter correctly. As a result, the API returns only the data that the area can hold.

#### Length of receiver variable

INPUT; BINARY(4)

The length of the receiver variable provided. The length of receiver variable parameter may be specified up to the size of the receiver variable specified in the user program. If the length of receiver variable parameter specified is larger than the allocated size of the receiver variable specified in the user program, the results are not predictable. The minimum length is 8 bytes.

#### Format name

INPUT; CHAR(8)

The content and format of the writer information being returned.

You must specify one of the following:

*"EXTW0100 Format"* on page 69 Contains the information about the writer and the spooled file it is processing.

#### Writer handle

INPUT; CHAR(16)

The handle to the writer job. This handle is provided to the driver program on a writer call to the driver exit program at initialization time (the initialize option).

### Spooled file handle

INPUT; CHAR(10)

The spooled file handle to the current file which is being processed by the writer. This handle is provided to the driver program on a writer call to the driver exit program (using the process file option).

### Error code

I/O; CHAR(\*)

The structure in which to return error information. For the format of the structure, see Error code parameter.

## EXTW0100 Format

The following table shows the information returned for the EXTW0100 format. For more details about the fields in the following table, see "Field Descriptions."

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Bytes returned
4	4	BINARY(4)	Bytes available
8	8	BINARY(4)	Number of additional copies
12	C	BINARY(4)	Reposition page number
16	10	CHAR(1)	Stop processing at page end
17	11	CHAR(1)	Stop processing at end of copy
18	12	CHAR(1)	Stop processing at page end, allow restart
19	13	CHAR(1)	Stop processing at end of copy, allow restart
20	14	CHAR(1)	Stop processing at page end, file was restarted
21	15	CHAR(1)	File held or deleted

## Field Descriptions

**Bytes available.** The number of bytes of data available to be returned. All available data is returned if enough space is provided.

**Bytes returned.** The number of bytes of data returned.

**File held or deleted.** The file has been held or deleted. This could have been the result of a user executing a Hold Spooled File (HLDSPLF) or Delete Spooled File (DLTSPLF) command while the file was active to the writer. This could also be the result of a user executing a Hold Writer (HLDWTR) command followed by executing a Release Writer (RLSWTR) command with the \*BYPASS option, while the file was active to the writer.

Possible values are:

- 0 The file has not been held or deleted.
- 1 The file has been held or deleted.

**Number of additional copies.** The number of copies requested to be printed. It is possible that the user could change the number of copies to be printed using the Change Spooled File Attributes (CHGSPLFA) command. Therefore the driver program must check the current number of copies. This field has a range of values from 1 to 255. A special value of zero (0) is returned if the number of copies has not changed.

**Reposition page number.** The page where the writer is to reposition the file. This field is set when the writer is restarted and is used with the various stop processing flags.

**Stop processing at end of copy.** Stop processing the file at the end of the current copy. This can be set with an End Writer (ENDWTR) command with the controlled option (\*CNTRLD) specified.

Possible values are:

- 0 The writer has not been ended with the controlled option.
- 1 The writer has been ended with the controlled option. When this API is used within a print driver exit or writer transform exit program, the exit program should finish the current copy of the file and return.

**Stop processing at end of copy, allow restart.** Stop processing the file at the end of the current copy. Allow printing to restart. This is set with the Hold Writer (HLDWTR) command with the controlled option (\*CNTRLD) specified.

Possible values are:

- 0 The writer has not been held with the controlled option.
- 1 The writer has been held with the controlled option. When this API is used within a print driver exit or writer transform exit program, the exit program should finish the current copy of the file and return.

**Stop processing at page end.** Stop processing the file at the end of the current page. This can be set with an End Writer (ENDWTR) command with the page end option (\*PAGEEND) specified.

Possible values are:

- 0 The writer has not been ended with the page end option.
- 1 The writer has been ended with the page end option. When this API is used within a print driver exit or writer transform exit program, the exit program should finish the current page of the file and return.

**Stop processing at page end, allow restart.** Stop processing the file at the end of the current page. Allow printing to restart. This is set with the Hold Writer (HLDWTR) command with the page end option (\*PAGEEND) specified.

Possible values are:

- 0 The writer has not been held with the page end option.
- 1 The writer has been held with the page end option. When this API is used within a print driver exit or writer transform exit program, the exit program should finish the current page of the file and return.

**Stop processing at page end, file was restarted.** The file has restarted printing. It is possible that the user could restart printing the file (CHGSPLFA). Therefore, the print driver program must check the Reposition page number to determine where to restart printing.

Possible values are:

- 0 The file has not restarted printing.
- 1 The file has started printing again.

## Error Messages

Message ID	Error Message Text
CPF24B4 E	Severe error while addressing parameter list.
CPF3CF1 E	Error code parameter not valid.
CPF3C19 E	Error occurred with receiver variable specified.
CPF3C21 E	Format name &1 is not valid.
CPF3C24 E	Length of the receiver variable is not valid.
CPF3C90 E	Literal value cannot be changed.
CPF33CC E	No writer found for specified handle &1.
CPF33CD E	No file found for specified handle &1.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V3R7

[Top](#) | [“Print APIs,” on page 1](#) | [APIs by category](#)

---

## Host Print Transform (QWPZHPTR, QwpzHostPrintTransform) API

Required Parameter Group:

1	Process option	Input	Binary(4)
2	Option specific input information	Input	Char(*)
3	Length of option specific input information	Input	Binary(4)
4	Spooled file data buffer	Input	Char(*)
5	Length of spooled file data buffer	Input	Binary(4)
6	Option specific output information buffer	Output	Char(*)
7	Length of option specific output information buffer	Input	Binary(4)
8	Length of option specific output information available	Output	Binary(4)
9	Transformed data buffer	Output	Char(*)
10	Length of transformed data buffer	Input	Binary(4)
11	Length of transformed data available	Output	Binary(4)
12	Error code	I/O	Char(*)

Default Public Authority: \*USE  
Service Program: QWPZHPT1  
Threadsafe: No

The Host Print Transform (OPM, QWPZHPTR; ILE, QwpzHostPrintTransform) API provides an interface to the host print transform (HPT) function. The host print transform transforms SCS or AFP™ data to ASCII data.

USERASCII spooled files will be converted in the same manner as data streams supplied to the “Convert Image (QIMGCVTI, QimgCvtImg) API” on page 30. If the input data stream is not supported, data is returned as if it were in its final form.

For informational purposes, a sample flow is given to show the interaction between the caller and the Host Print Transform API.

- The Host Print Transform API is called with the initialize (10) process option.

The Host Print Transform API sets any initial information and returns to the caller.

- The first eligible spooled file is selected for processing.

The Host Print Transform API is called with the process file (20) process option. The name and identifier of the spooled file is passed in the input information.

The Host Print Transform API determines if it will transform the spooled file and returns to the caller an indication whether it will transform the spooled file. The end file (40) process option must be used before another process file (20) process option is allowed.

- The Host Print Transform API is called with the transform data (30) process option and the data to be transformed.

You can use the Get Spooled File Data (QSPGETSP) API to get the data.

The Host Print Transform API transforms the data and passes back the transformed data to the caller. This step happens repeatedly until the entire spooled file has been passed to the Host Print Transform API. Alignment data also could be returned with this option.

- The Host Print Transform API also can be called with the maintain state (34) process option or the transform pages (36) process option.

The maintain state process option maintains information about the transforming of the spooled file data, but does not return the transformed data.

The transform pages process option provides additional data at the start and end of each page.

These process options can be intermixed with the transform data process option; however, it is recommended that transitions occur at complete page boundaries.

- The Host Print Transform API is called with the end file (40) process option.

The Host Print Transform API returns to the caller any remaining data to be sent to the printer.

- The caller selects the next eligible spooled file to be transformed.

Steps starting with the process file (20) process option are repeated.

- All transforming completed.

The Host Print Transform API is called with the terminate (50) process option. The Host Print Transform API cleans up any work spaces that it has created and returns to the caller.

## Authorities and Locks

*Device Description Authority*

\*USE

*Device Description Lock*

\*EXCLRD

*Workstation Customization Object Authority*

\*USE

*Workstation Customization Object Library Authority*

\*EXECUTE

## Required Parameter Group

### Process option

INPUT; BINARY(4)

The action for the host print transform to take. The first option should be initialize (10).

Possible values are:

10 (*Initialize*)

The Host Print Transform API sets any initial information with this option.

20 (*Process file*)

A spooled file is available to be transformed. The spooled file name and other information is passed in the option specific input information parameter to the Host Print Transform API. The Host Print Transform API decides whether to transform the data and sets the transform file field in the output information structure. Open time commands are returned in the transformed data buffer for SCS data streams.

30 ( <i>Transform data</i> )	The Host Print Transform API is called when data is available to be transformed. The data to be transformed is passed in the spooled file data buffer parameter. This process is repeated until all data has been processed. The transformed data is returned in the transformed data buffer. If requested, alignment data is also returned. When alignment data is requested, offsets and lengths for the alignment data are returned in the option specific output information buffer parameter.
34 ( <i>Maintain state</i> )	The Host Print Transform API is called when data is available to be transformed, but the transformed data is not returned to the caller. This option is used to maintain the host print transform's state for later processing of a specific page. <b>Note:</b> This option is valid only for SCS and AFPDS data.
36 ( <i>Transform pages</i> )	The Host Print Transform API is called when data is available to be transformed. The data to be processed is passed in the spooled file data buffer parameter. This option is used to provide additional data at the beginning and end of each page. The transformed data is returned in the transformed data buffer. <b>Note:</b> This option is valid only for SCS and AFPDS data.
40 ( <i>End file</i> )	The Host Print Transform API is informed that all data has been processed. No data is passed on this call. The Host Print Transform API may return additional data to the caller as an append to the end of the printed data.
50 ( <i>Terminate</i> )	This option is used even if the Host Print Transform API did not transform any data. All processing is complete. The Host Print Transform API performs cleanup in any user spaces it used. Option 50 may be used after any other process option.

### Option specific input information

INPUT; CHAR(\*)

Information that is provided to the Host Print Transform API. For the format and description of this information, see "Option Specific Input Information" on page 75.

### Length of option specific input information

INPUT; BINARY(4)

The length of the option specific input information specified in the option specific input information parameter. This length must be either 243 bytes, 273 bytes, or 296 bytes.

### Spooled file data buffer

INPUT; CHAR(\*)

The spooled file data to be transformed.

### Length of spooled file data buffer

INPUT; BINARY(4)

The length of the spooled file data specified in the spooled file data buffer parameter.

### Option specific output information buffer

OUTPUT; CHAR(\*)

Information that is returned from the Host Print Transform API to the caller. For the format and description of this information, see "Option Specific Output Information" on page 82.

### Length of option specific output information buffer

INPUT; BINARY(4)

The size, in bytes, of the option specific output information buffer.

### Length of option specific output information available

OUTPUT; BINARY(4)

The length of the option specific output information available. If this length is less than or equal to the length of the option specific output information buffer parameter, all available information has been returned. If this length is greater than the length of the option specific output information buffer parameter, only data up to the length provided is returned in the option specific output information buffer.

**Transformed data buffer**

OUTPUT; CHAR(\*)

Spooled file data that has been transformed by the Host Print Transform API. The data in this buffer can be sent to a printer. Alignment data, if requested, is also returned in the transformed data buffer. The lengths and offsets of the alignment information are returned in the option specific output information buffer parameter. For the format and description of this information, see "Option Specific Output Information" on page 82.

This buffer must be large enough to contain all of the transformed data. As a general rule, this buffer should be twice the size of the length of the spooled file data buffer.

**Length of transformed data buffer**

INPUT; BINARY(4)

The size, in bytes, of the transformed data buffer.

**Length of transformed data available**

OUTPUT; BINARY(4)

The length of the transformed data that is available to be returned in the transformed data buffer. If this length is less than or equal to the length of the transform data buffer, all available information has been returned. If this length is greater than the length of the transformed data buffer, only data up to the length provided was returned in the transformed data buffer. The remaining data is lost. An error message, CPF6DF9, is also returned.

**Error code**

I/O; CHAR(\*)

The structure in which to return error information. For the format of the structure, see Error code parameter.

The following table presents, in the same order, all the parameters previously described. Each entry indicates whether a parameter will be used by the Host Print Transform API. Y indicates yes and N indicates no.

Parameter	Initialize (10)	Process File (20)	Transform Data (30,34,36)	End File (40)	Terminate (50)	Description
Process option	10	20	30,34,36	40	50	Process option
Option specific input information	N	Y	Y	N	N	See description of this parameter.
Length of option specific input information	N	Y	Y	N	N	Length of option specific input information
Spooled file data buffer	N	N	Y	N	N	Contains data only on spooled data to transform option.
Length of spooled file data buffer	N	N	Y	N	N	The length of the spooled file data buffer.
Option specific output information buffer	N	Y	Y	Y	N	See description of this parameter.

Parameter	Initialize (10)	Process File (20)	Transform Data (30,34,36)	End File (40)	Terminate (50)	Description
Length of option specific output information buffer	N	Y	Y	Y	N	See description of this parameter.
Length of option specific output information available	N	Y	Y	Y	N	The number of bytes of option specific output information that is available. If this is greater than the length of the buffer, only information that can fit in the option specific output information buffer will be returned.
Transformed data buffer	N	Y	Y	Y	Y	This buffer can contain data to send to the device before the transformed data (Start of file option) or after the transformed data (End of file option), as well as transformed data (Spooled data to transform option). Alignment information is also returned in this buffer, if requested.
Length of transformed data buffer	N	Y	Y	Y	Y	The size, in bytes, of the transformed data buffer.
Length of transformed data available	N	Y	Y	Y	Y	The number of bytes of transformed data that is available.
Error code	Y	Y	Y	Y	Y	Error reporting.

## Option Specific Input Information

The following table shows the structure for the option specific input information parameter. For more details about the fields in the following table, see "Field Descriptions" on page 76.

Offset		Type	Field
Dec	Hex		
0	0	CHAR(16)	Reserved
16	10	CHAR(10)	Reserved
26	1A	CHAR(10)	Printer device name
36	24	CHAR(20)	Reserved
56	38	CHAR(20)	Reserved
76	4C	CHAR(10)	Reserved
86	56	CHAR(10)	Reserved
96	60	CHAR(16)	Internal job identifier
112	70	CHAR(16)	Internal spooled file identifier
128	80	CHAR(26)	Qualified job name
154	9A	CHAR(10)	Spooled file name
164	A4	BINARY(4)	Spooled file number
168	A8	CHAR(12)	Reserved
180	B4	BINARY(4)	Reserved
184	B8	BINARY(4)	Reserved
188	BC	CHAR(10)	Reserved
198	C6	CHAR(1)	Return alignment data
199	C7	CHAR(5)	Reserved
204	CC	BINARY(4)	Number of complete pages
208	D0	CHAR(10)	Workstation customizing object name
218	DA	CHAR(10)	Workstation customizing object library
228	E4	CHAR(15)	Manufacturer type and model
243	F3	CHAR(10)	Paper source 1
253	FD	CHAR(10)	Paper source 2
263	107	CHAR(10)	Envelope source
273	111	CHAR(1)	Reserved
274	112	CHAR(8)	Job system name
282	11A	CHAR(7)	Spooled file create date
289	121	CHAR(1)	Reserved
290	122	CHAR(6)	Spooled file create time

## Field Descriptions

**Envelope source.** The size of the envelope installed in the envelope source. This field is used only when the value for the printer device name field is \*NONE. If this field is not specified or the value is not valid, the special value of \*MFRTYPMDL is used.

Possible values are:

- \*NONE            There is no envelope source.
- \*MFRTYPMDL    The envelope size suggested by the manufacturer type and model is used.
- \*MONARCH       The envelope is monarch-size (3.875 x 7.5 inches).

<i>*NUMBER9</i>	The envelope is size number 9 (3.875 x 8.875 inches).
<i>*NUMBER10</i>	The envelope is size number 10 (4.125 x 9.5 inches).
<i>*B5</i>	The envelope is size B5 (176 mm x 250 mm).
<i>*C5</i>	The envelope is size C5 (162 mm x 229 mm).
<i>*DL</i>	The envelope is size DL (110 mm x 220 mm).

**Note:** This field is not validity checked when a name is specified in the printer device name field.

**Internal job identifier.** The internal job identifier of the job that owns the spooled file. » The identifier is not valid following an initial program load (IPL). If you attempt to use it after an IPL, an exception occurs. « This field must be blank if a name is given for the qualified job name.

**Internal spooled file identifier.** The internal spooled file identifier of the spooled file being processed. » The identifier is not valid following an initial program load (IPL) or if the spooled file has been moved between auxiliary storage pools (ASPs). If you attempt to use an invalid internal spooled file identifier, an exception occurs. « This field must be blank if a name is given for the spooled file name.

**Job system name.** The name of the system where the job that created the spooled file ran or blank when the spooled file name is \*INT. This field is considered after the job name, user name, job number, spooled file name, and spooled file number field requirements have been met.

The possible values are:

<i>*ONLY</i>	There is one job with the specified job name, user name, job number, spooled file name, spooled file number, spooled file create date, and spooled file create time.
<i>*CURRENT</i>	The job on the current system with the specified job name, user name, job number, spooled file name, spooled file number, spooled file create date, and spooled file create time is used.
<i>*ANY</i>	The job system name is not considered when selecting a spooled file. Use this value when you want the Spooled file create date and Spooled file create time parameters to take precedence over the job system name when selecting a spooled file.
<i>system-name</i>	The name of the system the job that created the spooled file ran on.

When the Length of option specific input information parameter is less than 296 bytes, the API assumes blanks when the spooled file name is \*INT. When spooled file name is not \*INT, the API assumes \*ONLY.

**Manufacturer type and model.** The manufacturer, type, and model for a printer using transform support. This field must be blank when a printer device name is specified.

<i>*WSCST</i>	The workstation customizing object is used to obtain the manufacturer type and model.
<i>type and model</i>	The manufacturer type and model. For a list of valid manufacturer types and models, see the MFRTYPMDL parameter on the Create Device Description (Printer) (CRTDEVPRT) command in the Control language topic collection.
<i>blank</i>	The manufacturer type and model is retrieved from the printer device.

**Number of complete pages.** The number of complete pages that are contained in the spooled file data input buffer.

**Paper source 1.** The size of the paper installed in paper source 1. This field is used only when the value for the printer device name field is \*NONE. If this field is not specified or the value is not valid, the special value of \*MFRTYPMDL is used.

Possible values are:

<i>*NONE</i>	There is no paper source 1 or the paper is manually fed into the printer.
--------------	---

*MFRTYPMDL	The paper size suggested by the manufacturer type and model is used.
*LETTER	The paper for this source is letter-size (8.5 x 11 inches).
*LEGAL	The paper for this source is legal-size (8.5 x 14 inches).
*EXECUTIVE	The paper for this source is executive-size (7.25 x 10.5 inches).
*LEDGER	The paper for this source is ledger-size (8.5 x 17 inches).
*A3	The paper for this source is size A3 (297 mm x 420 mm).
*A4	The paper for this source is size A4 (210 mm x 297 mm).
*A5	The paper for this source is size A5 (148 mm x 210 mm).
*B4	The paper for this source is size B4 (257 mm x 364 mm).
*B5	The paper for this source is size B5 (182 mm x 257 mm).
*CONT80	The paper for this source is 8.0 inches wide and a continuous form.
*CONT132	The paper for this source is 13.2 inches wide and a continuous form.

**Note:** This field is not validity checked when a name is specified in the printer device name field.

**Paper source 2.** The size of the paper installed in paper source 2. This field is used only when the value for the printer device name field is \*NONE. If this field is not specified or the value is not valid, the special value of \*MFRTYPMDL is used.

Possible values are:

*NONE	There is no paper source 2.
*MFRTYPMDL	The paper size suggested by the manufacturer type and model is used.
*LETTER	The paper for this source is letter-size (8.5 x 11 inches).
*LEGAL	The paper for this source is legal-size (8.5 x 14 inches).
*EXECUTIVE	The paper for this source is executive-size (7.25 x 10.5 inches).
*LEDGER	The paper for this source is ledger-size (11 x 17 inches).
*A3	The paper for this source is size A3 (297 mm x 420 mm).
*A4	The paper for this source is size A4 (210 mm x 297 mm).
*A5	The paper for this source is size A5 (148 mm x 210 mm).
*B4	The paper for this source is size B4 (257 mm x 364 mm).
*B5	The paper for this source is size B5 (182 mm x 257 mm).
*CONT80	The paper for this source is 8.0 inches wide and a continuous form.
*CONT132	The paper for this source is 13.2 inches wide and a continuous form.

**Note:** This field is not validity checked when a name is specified in the printer device name field.

**Printer device name.** The name of the printer device.

Possible values are:

*NONE	There is no printer device associated with the transform. Necessary information will be used from the workstation customizing object, the manufacturer type and model, various spooled file attributes, and the paper sources.
<i>name</i>	The name of the printer device.

**Note:** When a device name is specified for this field, the manufacturer type and model must be blank and the work station customizing object name must be \*NONE.

**Qualified job name.** The qualified job name of the job that owns the spooled file.

The qualified job name has three parts:

<i>job name</i>	CHAR(10). A specific job name, or one of the following special values: <ul style="list-style-type: none"> <li>* Only the job that this program is running. The rest of the job name field must be blank.</li> <li>*INT The internal job identifier used to locate the spooled file. The user name and job number must be set to blank.</li> </ul>
<i>user name</i>	CHAR(10). A specific user profile name, or blanks when the job name is * or *INT.
<i>job number</i>	CHAR(6). A specific job number, or blanks when the job name is * or *INT.

**Reserved.** An ignored field.

**Return alignment data.** Whether the transform should return data to the caller that can be used for alignment purposes.

- 0 Do not return alignment information.
- 1 Return alignment information.

**Spooled file create date.** ➤ The date, based on local job time, that the spooled file was created on the system or blank when the spooled file name is \*INT. ⚡ This parameter is considered after the job name, user name, job number, spooled file name, spooled file number, and job system name parameter requirements have been met. The date must be in the CYYMMDD format or one of the following special values:

<i>*ONLY</i>	There is only one spooled file with the specified job name, user name, job number, spooled file name, spooled file number, and job system name.
<i>*LAST</i>	The spooled file with the latest date and time which also has the specified job name, user name, job number, spooled file name, spooled file number, and job system name is used.
<i>date</i>	The date the spooled file was created on the system in the format CYYMMDD. See field Date file opened in API QUSRSPLA under field descriptions for more information about the date format.

When the Length of option specific input information parameter is less than 296 bytes, the API assumes blanks when the spooled file name is \*INT. When spooled file name is not \*INT, the API assumes \*ONLY.

**Spooled file create time.** ➤ The time, in local job time, that the spooled file was created on the system. ⚡ This field must be set to blanks when special values \*LAST or \*ONLY are used for field Spooled file create date. This field must be set if the spooled file create date has a date specified. This field is considered after the job name, user name, job number, spooled file name, spooled file number, job system name, and spooled file create date field requirements have been met.

The time must be in the HHMMSS format or one of the following special values:

<i>*ONLY</i>	There is only one spooled file with the specified job name, user name, job number, spooled file name, spooled file number, job system name, and spooled file create date.
<i>*LAST</i>	The spooled file with the latest time which also has the specified job name, user name, job number, spooled file name, spooled file number, job system name, and spooled file create date is used.
<i>time</i>	The time the spooled file was created on the system in the format HHMMSS. See field Time file opened in API QUSRSPLA under field descriptions for more information about the time format.
<i>blanks</i>	This field must be set to blanks if the spooled file create date field is set to *LAST or *ONLY.

When the Length of option specific input information parameter is less than 296 bytes, the API assumes blanks for this field.

**Spooled file name.** The name of the spooled file being processed.

You can use this special value for the name:

*\*INT* The internal spooled file identifier is used to locate the spooled file.

**Spooled file number.** The number of the spooled file being processed. The valid range is 1 through 999999.

The following special values are supported for this field:

- 0 Only one spooled file from the job has the specified file name. Therefore, the number of the spooled file is not necessary.
- 1 This uses the highest-numbered spooled file with the specified spooled file name.
- 2 The spooled file number is not used to determine which spooled file to process. Use this value when you want the Job system name field or the Spooled file create date and Spooled file create time fields to take precedence over the spooled file number when selecting a spooled file.

**Note:** This field must contain a valid value even if the value for the spooled file name field is *\*INT*.

A value is required because the API performs validity checking on all fields.

**Workstation customizing object library.** The library containing the workstation customizing object.

**Workstation customizing object name (WSCST).** The name of an object that consists of a table of attributes used to customize a given ASCII device. This field must be *\*NONE* when a printer device name is specified.

Possible values are:

- \*NONE* There is no workstation customizing object. The workstation customizing object is obtained from the printer device.
- name* The name of the workstation customizing object, which has been created by the Create Work Station Customizing Object (CRTWSCST) command.

The following table presents, in the same order, all the fields previously described in the option specific input information table. Each entry indicates whether a field will be defined for a particular option. Y indicates yes and N indicates no.

Field	Initialize (10)	Process File (20)	Transform Data (30,34,36)	End File (40)	Terminate (50)	Description
Reserved	N	N	N	N	N	Reserved.
Reserved	N	N	N	N	N	Reserved.
Printer device name	N	Y	N	N	N	Device name of the printer.
Reserved	N	N	N	N	N	Reserved.
Reserved	N	N	N	N	N	Reserved.
Reserved	N	N	N	N	N	Reserved.
Reserved	N	N	N	N	N	Reserved.
Internal job identifier	N	Y	N	N	N	For use with existing spooled file APIs.

Field	Initialize (10)	Process File (20)	Transform Data (30,34,36)	End File (40)	Terminate (50)	Description
Internal spooled file identifier	N	Y	N	N	N	For use with existing spooled file APIs.
Qualified job name	N	Y	N	N	N	Job owning spooled file.
Spooled file name	N	Y	N	N	N	Name of spooled file being processed.
Spooled file number	N	Y	N	N	N	Number of spooled file being processed.
Reserved	N	N	N	N	N	Reserved.
Reserved	N	N	N	N	N	Reserved.
Reserved	N	N	N	N	N	Reserved.
Reserved	N	N	N	N	N	Reserved.
Return alignment information	N	Y	Y	N	N	Return alignment information along with transformed data.
Reserved	N	N	N	N	N	Reserved.
Number of complete pages	N	N	Y	N	N	The number of complete pages in the data to be transformed.
Workstation customizing object name	N	Y	N	N	N	The name of the object that consists of a table of attributes used to customize an ASCII device.
Workstation customizing object library	N	Y	N	N	N	The name of the library containing the workstation customizing object.
Manufacturer type and model	N	Y	N	N	N	The manufacturer, type, and model of the ASCII device using the host print transform function.
Paper source 1	N	Y	N	N	N	The size of the paper in paper source 1.
Paper source 2	N	Y	N	N	N	The size of the paper in paper source 2.
Envelope source	N	Y	N	N	N	The size of the envelope in the envelope source.

## Option Specific Output Information

The following table shows the structure for the option specific output information buffer. This structure is used to pass information from the Host Print Transform API to the caller. For more details about the fields in the following table, see “Field Descriptions.”

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Reserved
4	4	CHAR(1)	Transform file
5	5	CHAR(1)	Pass input data
6	6	CHAR(2)	Reserved
8	8	CHAR(1)	Done transforming file
9	9	CHAR(3)	Reserved
12	C	BINARY(4)	Offset to vertical positioning commands
16	10	BINARY(4)	Length of vertical positioning commands
20	14	BINARY(4)	Offset to print data
24	18	BINARY(4)	Length of print data
28	1C	BINARY(4)	Offset to carriage return commands
32	20	BINARY(4)	Length of carriage return commands
36	24	BINARY(4)	Offset to form feed commands
40	28	BINARY(4)	Length of form feed commands
		CHAR(*)	Carriage return commands
		CHAR(*)	Form feed commands

## Field Descriptions

**Carriage return commands.** The carriage return commands contain the data stream commands, which can be used to do end-of-line processing for alignment purposes.

**Done transforming file.** Indicates the host print transform requires no further data to process the current file. This field should be ignored when the pass input data field is set to 0.

- 0 Continue calling the host print transform with the transform data (30) option.
- 1 The next call to the host print transform will be with the end file (40) option.

**Form feed commands.** The form feed commands contain data stream commands, which can be used to do end-of-page processing for alignment purposes.

**Length of carriage return commands.** The length, in bytes, of the carriage return commands. The carriage return commands are returned in the output specific information.

**Length of form feed commands.** The length, in bytes, of the form feed commands. The form feed commands are returned in the output specific information.

**Length of print data.** The length, in bytes, of the print data for the first line of data on the page. This includes any horizontal positioning that is done in the first line.

**Length of vertical positioning commands.** The length, in bytes, of the vertical positioning commands that are returned in the transformed data buffer.

**Offset to carriage return commands.** Offset to the carriage return commands in the output specific information. The carriage return commands contain the data stream commands that can be used to do end-of-line processing for alignment purposes.

**Offset to form feed commands.** Offset to the form feed commands in the output specific information. The form feed commands contain data stream commands that can be used to do end-of-page processing for alignment purposes.

**Offset to print data.** Offset to the first line of the print data for the page in the transformed data buffer. This first line can be printed for alignment purposes.

**Offset to vertical positioning commands.** Offset to the vertical positioning commands in the transformed data buffer.

**Pass input data.** Whether the caller passes the input data to the host print transform .

Possible values are:

- 0 The caller will pass the input data to the host print transform .
- 1 The caller will not pass the input data to the host print transform . The host print transform will read the data directly from the file.

**Reserved.** An ignored field.

**Transform file.** Whether the spooled file data will be transformed.

- 0 The spooled file data cannot be transformed.  
At least one of the following conditions exists:
  - A transform is not available for the data stream represented in the spooled file.
  - A device description is specified and the device does not use the host print transform function.
- 1 The spooled file data will be transformed.
- 2 The spooled file data is already in its final form.

The following table presents, in the same order, all the fields previously described in the option specific output information table. Each entry indicates whether a field will be used with a particular option. Y indicates yes and N indicates no.

The length and offset fields describe the format of the transformed data buffer when alignment data has been requested.

Field	Initialize (10)	Process File (20)	Transform Data (30,34,36)	End File (40)	Terminate (50)	Description
Reserved	N	N	N	N	N	Reserved

Field	Initialize (10)	Process File (20)	Transform Data (30,34,36)	End File (40)	Terminate (50)	Description
Transform file	N	Y	N	N	N	Informs the caller that the Host Print Transform API can be called with either the process file process option, or the transform data process option.
Pass input data	N	Y	N	N	N	Allows the host print transform to specify whether the caller will provide the input data or the host print transform will read the data directly from the file.
Reserved	N	N	N	N	N	Reserved
Done transforming file	N	N	Y	N	N	For the spooled file currently being processed, it allows the host print transform to stop the caller from calling the host print transform with any additional process option 30 calls.
Reserved	N	N	N	N	N	Reserved
Offset to vertical positioning commands	N	N	Y	N	N	The offset to the vertical positioning commands in the transformed data buffer. When alignment information is requested, vertical positional commands are also returned in the transformed data buffer.

Field	Initialize (10)	Process File (20)	Transform Data (30,34,36)	End File (40)	Terminate (50)	Description
Length of vertical positioning commands	N	N	Y	N	N	The length of the vertical positioning commands that are returned in the transformed data buffer. This can be used in alignment processing.
Offset to print data	N	N	Y	N	N	The offset to the print data for the first line of the page. This can be used for alignment processing.
Length of print data	N	N	Y	N	N	The length of print data for the first line of data for a page. This can be used for alignment processing.
Offset to carriage return commands	N	N	Y	N	N	Offset to the carriage return commands in the output specific information. The carriage return commands are data stream commands that can be used to do end-of-line processing for alignment purposes.
Length of carriage return commands	N	N	Y	N	N	The length of the carriage return commands.
Offset of form feed commands	N	N	Y	N	N	Offset to the form feed commands in the output specific information. The form feed commands are data stream commands that can be used to do end-of-page processing for alignment purposes.

Field	Initialize (10)	Process File (20)	Transform Data (30,34,36)	End File (40)	Terminate (50)	Description
Length of form feed commands	N	N	Y	N	N	The length of the form feed commands.
Carriage return commands	N	N	Y	N	N	The carriage return commands for alignment.
Form feed commands	N	N	Y	N	N	The form feed commands for alignment.

## Error Messages

Message ID	Error Message Text
CPF24B4 E	Severe error while addressing parameter list.
CPF2625 E	Not able to allocate object &1.
CPF2634 E	Not authorized to object &1.
CPF2702 E	Device description &1 not found.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPF3C1D E	Length specified in parameter &1 not valid.
CPF3C33 E	Spooled file number &1 is not valid.
CPF3C40 E	Spooled file &4 not found.
CPF3C41 E	More than one spooled file with same name.
CPF3C42 E	User name or job number is not blank.
CPF3C43 E	Internal job identifier is not valid.
CPF3C44 E	Internal spooled file identifier is not valid.
CPF3C58 E	Job name specified is not valid.
CPF3C90 E	Literal value cannot be changed.
CPF33C9 E	Spooled file name parameter cannot be blank.
CPF3309 E	No files named &1 are active.
CPF3330 E	Necessary resource not available.
CPF333B E	Job system name is not valid.
CPF333C E	Spooled file create date is not valid.
CPF333D E	Spooled file create time is not valid.
CPF333E E	Spooled file create time is not blank.
CPF333F E	Job system name is not blank.
CPF3342 E	Job &5/&4/&3 not found.
CPF3343 E	Duplicate job names found.
CPF3344 E	File &1 number &2 no longer in the system.
CPF335B E	Spooled file create date is not blank.
CPF6DF2 E	AFP data stream in user space &1 not valid.
CPF6DF4 E	Object &2 in library &3 contains unrecognizable data.
CPF6DF5 E	Value &1 not valid for process option parameter.
CPF6DF6 E	Value &1 not valid for return alignment field.
CPF6DF7 E	Option specific input information field values not valid.
CPF6DF8 E	Data cannot be transformed.
CPF6DF9 E	Transformed data buffer too small.
CPF6DFB E	Error occurred while transforming data.
CPF8105 E	Device description &4 damaged.
CPF9801 E	Object &2 in library &3 not found.
CPF9802 E	Not authorized to object &2 in &3.

Message ID	Error Message Text
CPF9803 E	Cannot allocate object &2 in library &3.
CPF9804 E	Object &2 in library &3 damaged.
CPF9820 E	Not authorized to use library &1.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API Introduced: V3R7

Top | "Print APIs," on page 1 | APIs by category

---

## Open List of AFP Information (QGYOLAFP) API

Required Parameter Group:

1	Receiver variable	Output	Char(*)
2	Length of receiver variable	Input	Binary(4)
3	List information	Output	Char(80)
4	Number of records to return	Input	Binary(4)
5	Format name	Input	Char(8)
6	Qualified AFP™ object name	Input	Char(20)
7	Format qualifier	Input	Char(10)
8	Error code	I/O	Char(*)

Default Public Authority: \*USE

Threadsafe: No

The Open List of AFP Information (QGYOLAFP) API lists entries from a font mapping table or list font character set or code page information from a coded font and places the list in a receiver variable. You can access additional records by using the Get List Entries (QGYGTLE) API. On successful completion of the API, a handle is returned in the list information parameter. You may use this handle on subsequent calls to the following APIs:

- Get List Entries (QGYGTLE)
- Find Entry Number in List (QGYFNDE)
- Close List (QGYCLST)

## Authorities and Locks

*Library authority*

\*EXECUTE

*Font table authority*

\*USE

*Font resource authority*

\*USE

*Font table lock*

\*SHRRD

*Font resource lock*

\*SHRRD

## Required Parameter Group

**Receiver variable**

OUTPUT; CHAR(\*)

The variable used to return the number of records requested (given in the number of records to return parameter). The structure of the receiver variable varies depending on which format is requested (format parameter).

#### **Length of Receiver variable**

INPUT; BINARY(4)

The length of the receiver variable provided. This must be large enough to hold at least one record.

#### **List information**

OUTPUT; CHAR(80)

Information about the list of objects that was opened. For a description of the layout of this parameter, see Open list information format.

#### **Number of records to return**

INPUT; BINARY(4)

The number of records in the list to put into the receiver variable.

#### **Format name**

INPUT; CHAR(8)

The format of the information to be returned for the specified object. The valid format names are:

<i>"CFNI0100 Format" on page</i> 90	Information about the specified coded font.
<i>"FCSI0100 Format" on page</i> 90	Information about the specified font character set.
<i>"PHCS0100 Format" on page</i> 90	Information about the specified font table.
<i>"HPCS0100 Format" on page</i> 91	Information about the specified font table.
<i>"HPCP0100 Format" on page</i> 92	Information about the specified font table.
<i>"PPCS0100 Format" on page</i> 92	Information about the specified font table.

#### **Qualified AFP object name**

INPUT; CHAR(20)

The AFP object for which you would like to retrieve information. The qualified object name has two parts. The first 10 characters contain the object name.

You may not use special values for the object name when the format name is PPCS0100.

You must use the following special value for the object name when the format name is FCSI0100:

*\*FNTCHRSET* All coded fonts that contain the font character set specified by the font character set parameter are returned.

You may use one of the following special values for the object name when the format name is PHCS0100:

*\*PHFCS* The printer-resident to host-resident font character set table is to be returned.  
*\*SYSPHFCS* The system printer-resident to host-resident font character set table is to be returned.

You may use one of the following special values for the object name when the format name is HPCS0100:

*\*HPFCS* The host-resident to printer-resident font character set table is to be returned.

*\*SYSHPFCS* The system host-resident to printer-resident font character set table is to be returned.

You may use one of the following special values for the object name when the format name is PHCP0100:

*\*PHCP* The printer-resident to host-resident code page mapping table is to be returned.  
*\*SYSPHCP* The system printer-resident to host-resident code page mapping table is to be returned.

You may use one of the following special values for the object name when the format name is HPCP0100:

*\*HPCP* The host-resident to printer-resident code page mapping table is to be returned.  
*\*SYSHPCP* The system host-resident to printer-resident code page mapping table is to be returned.

You must leave the second 10 characters blank when you use a special value for the AFP object name when the format name is PHCS0100, HPCS0100, PHCP0100, or HPCP0100.

The second 10 characters contain the name of the library in which the object is located when the format name is CFNI0100, PHCS0100, HPCS0100, PHCP0100, HPCP0100, or PPCS0100, and a special value is not specified for the AFP object name.

The following special values are allowed when the format name is CFNI0100, PHCS0100, HPCS0100, PHCP0100, HPCP0100, or PPCS0100:

*\*CURLIB* The job's current library is used to locate the object. If no library is specified as the current library for the job, the QGPL library is used.  
*\*LIBL* The library list is used to locate the object.

The actual library that is used is returned with the object information.

The second 10 characters contain the name of the library searched for coded fonts that contain the font character set specified in the first 10 characters of the format qualifier when the format name is FCSI0100.

The following special values are allowed when the Format name is FCSI0100:

*\*CURLIB* Only the libraries in the current library for the job are searched. If no library is specified as the current library for the job, QGPL is used.  
*\*LIBL* All libraries in the job's library list are searched.  
*\*USRLIBL* Only the libraries listed in the user portion of the job's library list are searched.  
*\*ALL* All libraries in the system, including QSYS, are searched.  
*\*ALLUSR* All user libraries are searched. For information on the libraries included, see *\*ALLUSR* in Generic library names.  
*\*DBCSFNTLIB* The DBCS font libraries QFNT61, QFNT62, QFNT63, QFNT64, and QFNT65 will be searched.

### **Format qualifier**

INPUT; CHAR(10)

The font character set name to search for in the coded font objects when the format name is FCSI0100 and the AFP object name is *\*FNTCHRSET*. All coded fonts that contain the font character set specified by the format qualifier parameter are returned.

This parameter must be blank if the format name is not FCSI0100.

### **Error code**

I/O; CHAR(\*)

The structure in which to return error information. For the format of the structure, see Error code parameter.

## CFNI0100 Format

The following table shows the information returned for the CFNI0100 format.

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	Font character set
10	A	CHAR(10)	Host code page
20	14	CHAR(10)	Resident font
30	1E	CHAR(2)	Reserved

## FCSI0100 Format

The following table shows the information returned for the FCSI0100 format.

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	Coded font name
10	A	CHAR(10)	Coded font library name
20	14	CHAR(50)	Text
70	46	CHAR(2)	Reserved

## Field Descriptions

**Coded font library name.** The library of the coded font being retrieved.

**Coded font name.** The name of the coded font being retrieved.

**Font character set.** The name of a font character set that is contained in the coded font.

**Host code page.** The name of the host resident code page font resource that is contained in the coded font.

**Reserved.** An ignored field.

**Resident font.** Whether the font character set and code page pair are resident in the printer.

The possible values are:

**\*YES** The font character set/code page pair is resident in the printer and does not need to be downloaded by the system to the printer.

**\*NO** The font character set/code page pair is not resident in the printer and needs to be downloaded by the system to the printer.

**Text.** The text that briefly describes the object.

## PHCS0100 Format

The following table shows the information returned for the PHCS0100 format.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Printer font identifier
4	4	CHAR(10)	Width
14	E	CHAR(12)	Attribute
26	1A	CHAR(10)	Graphic character set
36	24	CHAR(10)	Point size
46	2E	CHAR(10)	Host font character
56	38	CHAR(10)	Host font character (1)
66	42	CHAR(10)	Host font character (2)
76	4C	CHAR(10)	Type
86	56	CHAR(2)	Reserved

## HPCS0100 Format

The following table shows the information returned for the HPCS0100 format.

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	Host font character
10	A	CHAR(10)	Type
20	14	BINARY(4)	Printer font identifier
24	18	CHAR(10)	Width
34	22	CHAR(12)	Attribute
46	2E	CHAR(10)	Graphic character set
56	38	CHAR(10)	Point size
66	42	CHAR(2)	Reserved

## PHCP0100 Format

The following table shows the information returned for the PHCP0100 format.

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	Graphic character set
10	A	CHAR(5)	Code page
15	F	CHAR(10)	Host code page
25	19	CHAR(10)	Host code page (1)
35	23	CHAR(10)	Host code page (2)
45	2D	CHAR(3)	Reserved

## HPCP0100 Format

The following table shows the information returned for the HPCP0100 format.

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	Host code page
10	A	CHAR(10)	Graphic character set
20	14	CHAR(5)	Code page
25	19	CHAR(3)	Reserved

## PPCS0100 Format

The following table shows the information returned for the PPCS0100 format.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	From printer font ID
4	4	CHAR(10)	From point size
14	E	CHAR(2)	Reserved
16	10	BINARY(4)	To printer font ID
20	14	CHAR(10)	To point size
30	1E	CHAR(2)	Reserved

## Field Descriptions

**Attribute.** The attributes of the printer-resident font.

The possible attributes are:

- \*NONE* This printer-resident font has no special font attributes.
- \*BOLD* The printer-resident font is a bold font.
- \*ITALIC* The printer-resident font is an italic font.
- \*BOLDITC* The printer-resident font is a bold italic font.
- \*DBLWIDE* The printer-resident font is a double wide font.
- \*ITCDBLWIDE* The printer-resident font is an italic double wide font.
- \*BOLDDBLWIDE* The printer-resident font is a bold double wide font.

**Code page.** The printer-resident code page for which substitution is performed.

**From point size.** The point size of the printer-resident font for which substitution is performed.

The possible values are:

- \*WIDTH* The width of the printer-resident font is used to compute the point size.
- \*NONE* No point size is specified for this font identifier.
- \*ALL* All point sizes for a scalable font will be mapped. If the font is not scalable, then this will be treated the same as *\*NONE*.
- 0.1 to 999.9.* Numeric value of the point size. This value is right justified and padded with leading 0's.

**From printer font ID.** The printer font identifier for which substitution is performed.

**Graphic character set.** The graphic character set for the font identifier.

The possible values are:

*\*SYSVAL*            The graphic character set value specified in the system value QCHRID is used.  
*graphic-character-*    The graphic character set value specified for the font identifier.  
*set*

**Host code page.** The host code page specified for substitution. This value is blank if the qualified AFP object name is \*SYSPHCP.

**Host code page (1).** The first host code page selected for font substitution. If this host code page is not available, PSF/400 checks for the existence of Host Code Page (2). This value is blank if the qualified AFP object name is not \*SYSPHCP.

**Host code page (2).** The second host code page selected for font substitution if the first font code page is not available. A blank entry means that a second host code page is not available. This value is also blank if the qualified AFP object name is not \*SYSPHCP.

**Host font character.** The host font character set specified for font substitution. This value is blank if the qualified AFP object name is \*SYSPHFCS.

**Host font character (1).** The first host font character set selected for font substitution. If this host font character set is not available, PSF/400 checks for the existence of Host font character (2). This value is blank if the qualified AFP object name is not \*SYSPHFCS.

**Host font character (2).** The second host font character set selected for font substitution if the first font character set is not available. A blank entry means that a second host font character set is not available. This value is also blank if the qualified AFP object name is not \*SYSPHFCS.

**Point size.** The point size of the printer-resident font.

The possible values are:

*\*WIDTH*            The width of the printer-resident font is used to compute the point size.  
*\*NONE*             No point size is specified for this font identifier.  
*0.1 to 999.9.*      Numeric value of the point size. This value is right justified and padded with leading 0's.

**Printer font identifier.** The printer font identifier for which substitution is performed.

The possible values are:

*0*                    The mapping of a host-resident to printer-resident font is disabled. (\*NONE)  
*1 to 65535*        The printer font identifier.

**Reserved.** An ignored field.

**To point size.** The point size of the printer resident font identifier.

The possible values are:

*\*NONE:*            No font point size is specified. This should be specified for all non-scalable fonts.

*\*ALL* All point sizes for a scalable font will be mapped. If the font is not scalable, then this will be treated the same as *\*NONE*.

*0.1 to 999.9.* Numeric value of the point size. This value is right justified and padded with leading 0's.

**To printer font ID.** The font identifier of the printer resident font.

**Type.**

The type of host font specified:

*\*OUTLINE* The host-resident font is an outline font.  
*\*RASTER* The host-resident font is a raster font.

**Width.** The width of the printer-resident font.

The possible values are:

*\*PTSIZE* The width for this font identifier is calculated from the point size.  
*\*NONE* No width is specified for this font identifier. *\*NONE* should be specified when mapping an outline font.  
*1 to 32767* The font width for the font identifier. See the Printing topic for information on font widths for printer-resident fonts.

## Error Messages

Message ID	Error Message Text
CPF0001 E	Error found on &1 command.
CPD0172 D	Parameters passed on CALL do not match those required.
CPF2150 E	Object information function failed.
CPF2151 E	Operation failed for &2 in &1 type *&3.
CPF24B4 E	Severe error while addressing parameter list.
CPF3C17 E	Error occurred with input data parameter.
CPF3C19 E	Error occurred with receiver variable specified.
CPF3C1E E	Required parameter &1 omitted.
CPF3C20 E	Error found by program &1.
CPF3C21 E	Format name &1 is not valid.
CPF3C24 E	Length of the receiver variable is not valid.
CPF3C29 E	Object name &1 is not valid.
CPF3C36 E	Number of parameters, &1, entered for this API was not valid.
CPF3C3C E	Value for parameter &1 not valid.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPF8100 E	All CPF81xx messages could be returned. xx is from 01 to FF.
CPF9801 E	Object &2 in library &3 not found.
CPF9802 E	Not authorized to object &2 in &3.
CPF9803 E	Cannot allocate object &2 in library &3.
CPF9808 E	Cannot allocate one or more libraries on library list.
CPF9810 E	Library &1 not found.
CPF9820 E	Not authorized to use library &1.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.
CPF9999 E	Function check. &1 unmonitored by &2 at statement &5, instruction &3.
PQT0107 E	No coded fonts found containing specified font character set.

API introduced: V5R1

---

## Open List of Output Queues (QSPOLOTQ) API

Required Parameter Group:

1	Receiver variable	Output	Char(*)
2	Length of receiver variable	Input	Binary(4)
3	Format of receiver variable	Input	Char(8)
4	List information	Output	Char(80)
5	Filter information	Input	Char(*)
6	Sort information	Input	Char(*)
7	Number of records to return	Input	Binary(4)
8	Error Code	I/O	Char(*)

Threadsafe: Yes

Default Public Authority: \*USE

The Open List of Output Queues (QSPOLOTQ) API generates a list of output queues on the system. The list can include all output queues on the system, all output queues for a specified library list, all output queues on an ASP device, all output queues with a name starting with the same characters, or a specific output queue. The filtered list can then be sorted depending on the value of the sort parameter. When requesting output queue information, if the signed-on user is not authorized to the library of the output queue, information for that output queue is not returned by this API. Upon successful completion of this API, a handle is returned in the list information parameter. You may use this handle on subsequent calls to the following APIs:

- Get List Entries (QGYGTLE)
- Find List Entry (QGYFNDE)
- Close List (QGYCLST)

### Performance Impacts

Sorting on one or more values of output queue name, output queue library name, number of files on an output queue, number of writers started to an output queue, first writer started to the queue, output queue status, description, or ASP name takes more processing power and time.

### Authorities and Locks

*Output Queue Library Authority*  
\*EXECUTE

*Output Queue Lock*  
This API gets an \*EXCLRD lock on the output queue.

### Required Parameter Group

#### Receiver variable

OUTPUT; CHAR(\*)

The variable used to return the output queue information.

#### Length of receiver variable

INPUT; BINARY(4)

The length of the receiver variable.

#### Format of receiver variable

INPUT; CHAR(8)

The format of the output queue information being returned. You can specify the following:

*OOTQ0100* Contains the basic information about the output queue. For more information about the OOTQ0100 format, see “Format of Receiver Variable” on page 98.

#### List information

OUTPUT; CHAR(80)

Information about the list created by this program. For a description of the layout of this parameter, see Format of Open list information.

#### Filter information

INPUT; CHAR(\*)

The information in this parameter is used for filtering the list of output queue information. For more information about the filter information, see “Filter Information.”

#### Sort information

INPUT; CHAR(\*)

Information on what fields within the record of information to sort. See “Format of Sort Information” for a description of the layout of this parameter. Note that when sorting is requested, the entire list has to be built and sorted before any records can be returned.

#### Number of records to return

INPUT; BINARY(4)

The number of records in the list to put into the receiver variable.

If -1 is specified for this parameter, the entire list is built synchronously.

If 0 is specified for this parameter, the entire list is built asynchronously in a server job.

If a positive number of records to return is specified, at least that many records are built synchronously and the remainder are built asynchronously in a server job.

#### Error code

I/O; CHAR(\*)

The structure in which to return error information. For the format of the structure, see Error code parameter.

## Filter Information

The following table shows the format of the filter information parameter. For detailed descriptions of the fields in the table, see Field Descriptions.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Length of filter information
4	4	CHAR(10)	Output queue name
14	E	CHAR(10)	Output queue library name
24	18	CHAR(10)	Auxiliary storage pool (ASP) device name
34	22	CHAR(*)	Reserved

## Format of Sort Information

For more details about the fields in the following table, see “Field Descriptions” on page 97.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Number of keys to sort on
Offsets vary. These fields repeat for each sort key field.		BINARY(4)	Sort key field starting position
		BINARY(4)	Sort key field length
		BINARY(2)	Sort key field data type
		CHAR(1)	Sort order
		CHAR(1)	Reserved

**Note:** If the last three fields (sort key field data type, sort order, and the reserved field) are not used, then they must be set to hexadecimal zeros. This causes all the data to be treated as character data, and it is sorted in ascending order.

## Field Descriptions

**Auxiliary storage pool (ASP) device name.** The name of an ASP device in which storage is allocated for the library that contains the output queue. A specific ASP device name or a special value may be specified. If filtering on an output queue library value of \*CURLIB, \*LIBL, or \*USRLIBL, the ASP device name must not be specified or must be the special value \* or message CPF2173 will be issued.

<i>simple name</i>	A simple name of an ASP device. The ASP device must be in the namespace for the current job or CPF337E will be issued.
*	The ASPs in the thread's library name space.
*SYSBAS	The system ASP (ASP 1) and defined basic user ASPs (ASPs 2-32).
*CURASPGRP	The ASPs in the current thread's ASP group.

**Output queue name.** The output queue about which to retrieve information. A specific output queue name, a generic output queue name, or the special value \*ALL may be specified.

<i>simple name</i>	A simple name of an output queue.
<i>generic name</i>	A generic name for output queues.
*ALL	All output queues are returned.

**Output queue library name.** The library in which the output queue is located. A specific library name or a special value may be specified.

<i>simple name</i>	A simple name of a library.
*ALL	All libraries on the system are searched.
*ALLUSR	All user defined libraries to which the user is authorized are searched. For information on the libraries included, see *ALLUSR in Generic library names.
*CURLIB	The job's current library is searched.
*LIBL	The library list for the job is searched.
*USRLIBL	The libraries in the user portion of the job's library list are searched.

**Length of filter information.** The length of the filter information. An error message is returned if the length of filter information is not set correctly.

**Number of keys to sort on.** The number of fields within the record structure on which to sort. If 0 is specified, the list is not sorted.

**Reserved.** Must be set to hexadecimal zeros.

**Sort key field data type.** Data type of field to sort. See Sort (QLGSORT) API for information on the list of data types available.

**Sort key field length.** The length of the field on which to sort.

**Sort key field starting position.** Within the record of information, the starting position of the field on which to sort.

**Sort order.** Whether the list should be sorted in ascending or descending order according to the key. See Sort (QLGSORT) API for information on the sort order special values.

## Format of Receiver Variable

The following tables describe the order and format of the data returned in the receiver variable.

### OOTQ0100 Format

For more details about the fields in the following table, see "Field Descriptions."

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	Output queue name
10	A	CHAR(10)	Output queue library name
20	14	BINARY(4)	Number of spooled files on output queue
24	18	BINARY(4)	Number of writers started to output queue
28	1C	CHAR(10)	Name of first writer to output queue
38	26	CHAR(1)	Output queue status
39	27	CHAR(3)	Reserved
42	2A	CHAR(50)	Description
92	5C	CHAR(10)	Output queue library auxiliary storage pool (ASP) name

## Field Descriptions

**Description.** The text description for this output queue. The following special value may be returned:

\*BLANK      There is no text description for the output queue.

**Name of first writer started to output queue.** The first active writer producing output from the output queue.

**Output queue library auxiliary storage pool (ASP) name.** The name of the auxiliary storage pool (ASP) device name where storage is allocated for the library that contains the object. The following special values may be returned:

\*N            The name of the ASP device cannot be determined.  
\*SYSBAS     System ASP (ASP 1) or basic user ASPs (ASPs 2-32).

**Output queue library name.** The name of the library in which the output queue is located.

**Output queue name.** The name of the output queue.

**Output queue status.** The current status of the output queue. The possible values are:

- 0 The output queue is currently held. No spooled files can be written from this output queue.
- 1 The output queue is released. Spooled files can be written from this queue.
- 2 The output queue is damaged.

**Number of spooled files on output queue.** The total number of spooled files currently on this output queue. This field is -1 if the output queue is damaged.

**Number of writers started to output queue.** The total number of writers currently producing output from this output queue. This field is -1 if the output queue is damaged.

**Reserved.** A reserved field.

## Error Messages

Message ID	Error Message Text
CPF2173 E	Value for ASPDEV not valid with special value for library.
CPF24B4 E	Severe error while addressing parameter list.
CPF337E E	ASP device &1 not in current ASP group for thread.
CPF3C19 E	Error occurred with receiver variable specified.
CPF3C21 E	Format name &1 is not valid.
CPF3C3B E	Value for &2 for API &1 not valid.
CPF3C90 E	Literal value cannot be changed.
CPF3CF1 E	Error code parameter not valid.
CPF9804 E	Object &2 in library &3 damaged.
CPF9820 E	Not authorized to use library &1.
CPF9830 E	Cannot assign library &1.
CPF9871 E	Error occurred while processing.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.
GUI0002 E	&2 is not valid for length of receiver variable.
GUI0024 E	&1 is not valid for number of keys to sort on.
GUI0025 E	&1 is not valid for sort key field starting position.
GUI0026 E	&1 is not valid for sort key field length.
GUI0027 E	&1 is not valid for number of records to return.
GUI0119 E	Reserved field in sort information not valid.
GUI0120 E	Order field in sort information not valid.
GUI0152 E	&1 is not valid for length of filter information.



API introduced: V6R1 with PTF

[Top](#) | ["Print APIs,"](#) on page 1 | [APIs by category](#)

---

## Open List of PDF Map Information (QPQOLPM) API

Required Parameter Group:

1	Receiver variable	Output	Char(*)
2	Length of receiver variable	Input	Binary(4)
3	List information	Output	Char(80)
4	Number of records to return	Input	Binary(4)
5	Format name	Input	Char(8)
6	Filter information	Input	Char(*)
7	Format of filter information	Input	Char(8)

8	Qualified Map object name	Input	Char(20)
9	Error code	I/O	Char(*)

Default Public Authority: \*USE  
 Threadsafe: Yes

The Open List of PDF Map Information (QPQOLPM) API lists entries from a PDF Map object and places the list in a receiver variable. The size of the receiver variable determines the amount of information returned. You can access additional records by using the Get List Entries (QGYGTLE) API. On successful completion of the API, a handle is returned in the list information parameter. You may use this handle on subsequent calls to the following APIs:

- Get List Entries (QGYGTLE)
- Close List (QGYCLST)

For more information, see Process Open List APIs.

## Authorities and Locks

*Library authority*  
 \*EXECUTE

*PDF Map object authority*  
 \*USE

*PDF Map object lock*  
 \*SHRRD

## Required Parameter Group

### Receiver variable

OUTPUT; CHAR(\*)

The receiver variable that receives the information requested. You can specify the size of the area to be smaller than the format requested as long as you specify the length parameter correctly. As a result, the API returns only the data the area can hold.

### Length of Receiver variable

INPUT; BINARY(4)

The length of the receiver variable provided. This must be large enough to hold at least one record.

### List information

OUTPUT; CHAR(80)

Information about the list of objects that was opened. For a description of the layout of this parameter, see Open list information format.

### Number of records to return

INPUT; BINARY(4)

The number of records in the list to put into the receiver variable.

### Format name

INPUT; CHAR(8)

The format of the information to be returned for the specified object. The valid format names are:

*"PDFM0100 Format" on page 101* Contains abbreviated information about a single entry from the specified Map object. The format is repeated for each entry in the specified Map object. If an entry is segmented, there is just a single entry; there is not an entry for each segment.

*"PDFM0200 Format" on page 103* Contains all information about a single entry from the specified Map object. The format is repeated for each entry in the specified Map object. If an entry is segmented, there is an entry for each segment.

**Filter information**

INPUT; CHAR(\*)

The information in this parameter is used for filtering the list of Map information. You must specify the filter format you are using with the Format of filter information parameter.

**Format of filter information**

INPUT; CHAR(8)

The format of the filter information that can be specified for retrieving entries from a Map object. The valid format name is:

*"FLTR0100 Format" on page 115* Use this format to specify which entries should be listed. You can have all entries, or a subset of the entries in a Map object listed.

**Qualified Map object name**

INPUT; CHAR(20)

The Map object for which you would like to retrieve information. The qualified object name has two parts. The first 10 characters contain the object name.

The second 10 characters contain the name of the library in which the object is located.

The following special values are allowed:

*\*CURLIB* The job's current library is used to locate the object. If no library is specified as the current library for the job, the QGPL library is used.

*\*LIBL* The library list is used to locate the object.

The actual library that is used is returned with the object information.

**Error code**

I/O; CHAR(\*)

The structure in which to return error information. For the format of the structure, see Error code parameter.

**PDFM0100 Format**

The following table shows the information returned for the PDFM0100 format. If necessary, text fields are padded with blanks (X'40'). For more details about the fields in the this table, see "PDFM0100 Format Field Descriptions" on page 102.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Sequence number of entry
4	4	CHAR(10)	Output queue name for input spooled file
14	E	CHAR(10)	Output queue library name for input spooled file
24	18	CHAR(10)	Spooled file name for input spooled file
34	22	CHAR(10)	Job name for input spooled file
44	2C	CHAR(10)	User name for input spooled file
54	36	CHAR(10)	User data for input spooled file
64	40	CHAR(10)	Form type for input spooled file

Offset		Type	Field
Dec	Hex		
74	4A	CHAR(250)	Mail tag for input spooled file
324	144	CHAR(1)	Segmented entry
325	145	CHAR(50)	Text description
375	177	CHAR(1)	Reserved

## PDFM0100 Format Field Descriptions

**Form type for input spooled file.** The type of forms that should be loaded on the printer before this spooled file is printed. This is the value for the input spooled file.

The following special values can be used for the form type for input spooled file:

*\*ALL* The entry is included in the list regardless of the value for form type for filter.

**Job name for input spooled file.** The name of the job that produced the input spooled file.

The following special values can be used for the job name for input spooled file:

*\*ALL* The entry is included in the list regardless of the value for job name for filter.

**Mail tag for input spooled file.** The value of the mail tag for the input spooled file.

The following special values can be used for mail tag for input spooled file:

*\*ALL* The entry is included in the list regardless of the value for mail tag for filter.

**Output queue library name for input spooled file.** The name of the library where the output queue for the input spooled file resides.

If Output queue name for input spooled file has a value of *\*ALL*, than this field is set to blanks.

**Output queue name for input spooled file.** The name of the output queue where the input spooled file resides.

The following special values can be used for the output queue name for input spooled file:

*\*ALL* The entry is included in the list regardless of the value for output queue name for filter.

**Reserved.** Set to a null value.

**Segmented entry.** Indicates whether an entry is a segment.

The following values are used:

1 The entry is a segment.

0 The entry is not a segment.

**Sequence number of entry.** The user-defined sequence number for an entry in a PDF Map object.

**Spooled file name for input spooled file.** The name of the existing spooled file.

The following special values can be used for the spooled file name for input spooled file:

\*ALL        The entry is included in the list regardless of the value for spooled file name for filter.

**Text description.** A brief description of the PDF Map entry.

**User data for input spooled file.** The user-specified data which describe the input spooled file.

The following special values can be used for the user data for input spooled file:

\*ALL        The entry is included in the list regardless of the value for user data for filter.

**User name for input spooled file.** The name of the user that created the spooled file.

The following special values can be used for the user name for input spooled file:

\*ALL        The entry is included in the list regardless of the value for user name for filter.

## PDFM0200 Format

The following table shows the information returned for the PDFM0200 format. If necessary, text fields are padded with blanks (X'40'). For more details about the fields in the this table, see "PDFM0200 Format Field Descriptions" on page 104.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Length of PDF Map entry
4	4	BINARY(4)	Sequence number of entry
8	8	BINARY(4)	Offset to mail information entry
12	C	BINARY(4)	Length of mail information entry
16	10	BINARY(4)	Offset to PDF stream file information
20	14	BINARY(4)	Length of PDF stream file information
24	18	BINARY(4)	Offset to PDF spooled file information
28	1C	BINARY(4)	Length of PDF spooled file information
32	20	BINARY(4)	Offset to AFPDS spooled file information
36	24	BINARY(4)	Length of AFPDS spooled file information
40	28	CHAR(10)	Output queue name for input spooled file
50	32	CHAR(10)	Output queue library name for input spooled file
60	3C	CHAR(10)	Spooled file name for input spooled file
70	46	CHAR(10)	Job name for input spooled file
80	50	CHAR(10)	User name for input spooled file
90	5A	CHAR(10)	User data for input spooled file
100	64	CHAR(10)	Form type for input spooled file
110	6E	CHAR(250)	Mail tag for input spooled file
360	168	CHAR(1)	Segmented entry

Offset		Type	Field
Dec	Hex		
361	169	CHAR(50)	Text description
411	19B	CHAR(1)	Reserved

## PDFM0200 Format Field Descriptions

**Form type for input spooled file.** The type of forms that should be loaded on the printer before this spooled file is printed. This is the value for the input spooled file.

The following special values can be used for the form type for input spooled file:

\*ALL The entry is included in the list regardless of the value for form type for filter.

**Job name for input spooled file.** The name of the job that produced the input spooled file.

The following special values can be used for the job name for input spooled file:

\*ALL The entry is included in the list regardless of the value for job name for filter.

**Length of AFPDS spooled file information.** The size, in bytes, of an AFPDS Spooled File Information entry. If there is an entry (this field has a value greater than 0), there is a maximum of one entry for each non-segmented entry, and one entry for each segment of a segmented entry. For information on the format of an AFPDS Spooled File Information entry, see AFPDS Spooled File Information Entry.

**Length of mail information entry.** The size, in bytes, of a single Mail information entry. If there is an entry (this field has a value greater than 0), there is a maximum of one entry for each non-segmented entry, and one entry for each segment of a segmented entry. For information on the format of a mail information entry, see Mail Information Entry.

**Length of PDF Map entry.** The size, in bytes, of a single PDF Map entry for format PDFM0200. The length includes, in addition to the PDFM0200 format, all entries required to provide the data for one PDF Map entry. There is one PDF Map entry for each non-segmented entry, and one PDF Map entry for each segment of a segmented entry in the Map object. The entries are placed contiguously in storage. For information on the format of a PDF Map entry, see PDFM0200 Format. The number of PDF Map entries returned corresponds to the value in the Records returned field in the Open list information format.

**Length of PDF spooled file information.** The size, in bytes, of a PDF Spooled File Information entry. For information on the format of a PDF Spooled File Information Entry, see PDF Spooled File Information Entry.

**Length of PDF stream file information.** The size, in bytes, of a PDF Stream File Information Entry. This identifies the path where PSF/400 has written the PDF stream file. If there is an entry (this field has a value greater than 0), there is a maximum of one entry for each non-segmented entry, and one entry for each segment of a segmented entry. For information on the format of a stream file entry, see PDF Stream File Information Entry.

**Mail tag for input spooled file.** The value of the mail tag for the input spooled file.

The following special values can be used for mail tag for input spooled file:

\*ALL The entry is included in the list regardless of the value for mail tag for filter.

**Offset to AFPDS spooled file information.** The offset, in bytes, from the beginning of the PDFM0200 format structure, to an AFPDS Spooled File Information entry. If this value is 0, an AFPDS Spooled File Information entry is not specified.

**Offset to mail information entry.** The offset, in bytes, from the beginning of the PDFM0200 format, to a Mail information entry. If this value is 0, a Mail information entry is not specified.

**Offset to PDF spooled file information.** The offset, in bytes, from the beginning of the PDFM0200 format, to a PDF Spooled File Information entry. If this value is 0, a PDF Spooled File information entry is not specified.

**Offset to PDF stream file information.** The offset, in bytes, from the beginning of the PDFM0200 format, to a PDF Stream File Information entry. If this value is 0, a PDF Stream File Information entry is not specified.

**Output queue library name for input spooled file.** The name of the library where the output queue for the input spooled file resides.

If Output queue name for input spooled file has a value of \*ALL, than this field must be set to blanks.

**Output queue name for input spooled file.** The name of the output queue where the input spooled file resides.

The following special values can be used for the output queue name for input spooled file:

\*ALL        The entry is included in the list regardless of the value for output queue name for filter.

**Reserved.** Set to a null value.

**Segmented entry.** Indicates whether an entry is a segment.

The following values are used:

1        The entry is a segment.

0        The entry is not a segment.

**Sequence number of entry.** The user-defined sequence number for an entry in a PDF Map object.

**Spooled file name for input spooled file.** The name of the existing spooled file.

The following special values can be used for the spooled file name for input spooled file:

\*ALL        The entry is included in the list regardless of the value for spooled file name for filter.

**Text description.** A brief description of the PDF Map entry.

**User data for input spooled file.** The user-specified data which describe the input spooled file.

The following special values can be used for the user data for input spooled file:

\*ALL        The entry is included in the list regardless of the value for user data for filter.

**User name for input spooled file.** The name of the user that created the spooled file.

The following special values can be used for the user name for input spooled file:

\*ALL        The entry is included in the list regardless of the value for user name for filter.

## Mail Information Entry

If necessary, text fields are padded with blanks (X'40'). For more details about the fields in the following table, see "Mail Information Entry Field Descriptions" on page 107.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Offset to message text
4	4	BINARY(4)	Length of message text
8	8	BINARY(4)	Offset to subject
12	C	BINARY(4)	Length of subject
16	10	BINARY(4)	CCSID of message text and subject
20	14	BINARY(4)	Offset to ReplyTo e-mail address
24	18	BINARY(4)	Length of ReplyTo e-mail address
28	1C	BINARY(4)	Offset to TO e-mail address entries
32	20	BINARY(4)	Number of TO e-mail address entries
36	24	BINARY(4)	Length of TO e-mail address entry
40	28	BINARY(4)	Offset to CC e-mail address entries
44	2C	BINARY(4)	Number of CC e-mail address entries
48	30	BINARY(4)	Length of CC e-mail address entry
52	34	BINARY(4)	Offset to BCC e-mail address entries
56	38	BINARY(4)	Number of BCC e-mail address entries
60	3C	BINARY(4)	Length of BCC e-mail address entry
64	40	BINARY(4)	Offset to Body of e-mail entries
68	44	BINARY(4)	Number of Body of e-mail entries
72	48	BINARY(4)	Length of Body of e-mail entry
76	4C	BINARY(4)	Offset to attachment e-mail entries
80	50	BINARY(4)	Number of attachment e-mail entries
84	54	BINARY(4)	Length of attachment e-mail entry
88	58	BINARY(4)	Offset to TO e-mail stream file
92	5C	BINARY(4)	Length of TO e-mail stream file
96	60	BINARY(4)	Offset to CC e-mail stream file
100	64	BINARY(4)	Length of CC e-mail stream file
104	68	BINARY(4)	Offset to BCC e-mail stream file
108	6C	BINARY(4)	Length of BCC e-mail stream file
112	70	BINARY(4)	Offset to PDF file for e-mail
116	74	BINARY(4)	Length of PDF file for e-mail
120	78	CHAR(10)	Mail Sender
130	82	CHAR(10)	User password

Offset		Type	Field
Dec	Hex		
140	8C	CHAR(10)	Master password
150	96	CHAR(1)	Encryption level
151	97	CHAR(1)	Print PDF file
152	98	CHAR(1)	Copy PDF file
153	99	CHAR(1)	Change PDF file
154	9A	CHAR(1)	Change Comments in PDF file
155	9B	CHAR(1)	Content Access to PDF file
156	9C	CHAR(1)	Document assembly
157	9D	CHAR(1)	Use stream file for TO e-mail address entries
158	9E	CHAR(1)	Use stream file for CC e-mail address entries
159	9F	CHAR(1)	Use stream file for BCC e-mail address entries

## Mail Information Entry Field Descriptions

**CCSID of message text and subject.** The CCSID associated with the message text and subject.

**Change Comments in PDF file.** Specifies whether users are allowed to add or change comments (annotations) or form fields in the PDF file. This field is applicable for Encryption levels \*40RC4 and \*128RC4.

The following values are used:

*X'F0'* The user is not allowed to change comments in the PDF file.  
*X'F1'* The user is allowed to change comments in the PDF file.

**Change PDF file.** Specifies whether users are allowed to change the PDF file. This field is applicable for Encryption levels \*40RC4 and \*128RC4.

The following values are used:

*X'F0'* The user is not allowed to change the PDF file.  
*X'F1'* The user is allowed to change the PDF file.

**Content Access to PDF file.** Specifies whether content accessibility for the visually impaired is enabled for the PDF file. This field is applicable only for an Encryption level of \*128RC4.

The following values are used:

*X'F0'* Content accessibility for the visually impaired is disabled in the PDF file. .  
*X'F1'* content accessibility for the visually impaired is enabled in the PDF file..

**Copy PDF file.** Specifies whether users are allowed to copy or extract text and graphics. This field is applicable for Encryption levels \*40RC4 and \*128RC4.

The following values are used:

- X'F0'* The user is not allowed to copy and extract text and graphics in the PDF file. The accessibility interface is disabled.
- X'F1'* The user is allowed to copy and extract text and graphics in the PDF file. The accessibility interface is enabled.

**Document assembly of PDF file.** Specifies whether users are allowed to assemble the document (insert, rotate, or delete pages and create bookmarks or thumbnail images). This field is applicable for an Encryption level of \*128RC4 only.

The following values are used:

- X'F0'* The user is not allowed to assemble the document
- X'F1'* The user is allowed to assemble the document

**Encryption level.** The encryption level for the PDF file. If the encryption level is \*40RC4 or \*128RC4, then the different encryption options valid for those levels of security will be returned.

The following special values are used for the encryption level:

- X'F0'* The file is not encrypted.
- X'F1'* The PDF file is created with the 40-bit RC4 encryption level. This level of encryption has a lower level of security that is compatible with Adobe Acrobat Reader 3.X and later versions.
- X'F2'* The PDF file is created with the 128-bit RC4 encryption level. This level of encryption is has a higher level of security but is only compatible with Adobe Acrobat 5.X and later versions.

**Length of attachment e-mail entry.** The size, in bytes, of each entry used for naming an e-mail attachment. Use one entry for each stream file. The entries must be concatenated, as in an array. For information on the format of a stream file entry, see Stream File Entry.

**Length of BCC e-mail address entry.** The size, in bytes, of each E-mail Address entry used for the BCC of an e-mail. For information on the format of an e-mail addresses entry, see E-mail Address Entry.

If Use stream file for BCC e-mail address entries is set to *X'F1'*, then this value will be 0.

**Length of BCC e-mail stream file.** The size, in bytes, of the Stream File entry containing the stream file which lists the BCCs used for an e-mail. For information on the format of an e-mail addresses entry, see Stream File Entry.

If Use stream file for BCC e-mail address entries is set to *X'F0'*, then this value will be 0.

**Length of Body of e-mail entry.** The size, in bytes, of each entry used for naming a stream file placed into the body of the e-mail. Use one entry for each stream file. The entries must be concatenated, as in an array. For information on the format of a stream file entry, see Stream File Entry.

**Length of CC e-mail address entry.** The size, in bytes, of each E-mail Address entry used for the CC of an e-mail. For information on the format of an e-mail addresses entry, see E-mail Address Entry.

If Use stream file for CC e-mail address entries is set to *X'F1'*, then this value will be 0.

**Length of CC e-mail stream file.** The size, in bytes, of the Stream File entry containing the stream file which lists the CCs used for an e-mail. For information on the format of a stream file entry, see Stream File Entry.

If Use stream file for CC e-mail address entries is set to *X'F0'*, then this value will be 0.

**Length of message text.** The size, in bytes, of the message text that gets placed into the body of the e-mail.

The following special values can be used for the message text:

*\*NONE* No message text will be included in the body of the e-mail.

If the length of the message text is 0, then the default message text is obtained from message PQT4133 in message file QPQMSGF.

**Length of PDF file for e-mail** The size, in bytes, of a File Name Entry. This identifies the name to be used for the PDF file produced from a spooled file, that is attached to an e-mail. For information on the format of a file name entry, see File Name Entry.

**Length of ReplyTo to e-mail address.** The size, in bytes, of an E-mail Address entry used for the ReplyTo of an e-mail. For information on the format of an e-mail addresses entry, see E-mail Address Entry.

The following special values can be used for the ReplyTo e-mail address:

*\*MAILSENDER* Replies will be sent to the FROM e-mail address.

**Length of subject.** The size, in bytes, of the text that is used for the subject of the e-mail.

The following special values can be used for the subject:

**Length of subject.** The size, in bytes, of the text that is used for the subject of the e-mail.

The following special values can be used for the subject:

*\*NONE* No subject will be used in the body of the e-mail.

If the length of the subject is 0, then the default subject is obtained from message PQT4133 in message file QPQMSGF.

**Length of TO e-mail address entry.** The size, in bytes, of each E-mail Address entry used for the TO of an e-mail. For information on the format of an e-mail addresses entry, see E-mail Address Entry.

If Use stream file for TO e-mail address entries is set to X'F1', then this value will be 0.

The following special values can be used for the TO e-mail address entry:

*\*SPLF* The e-mail address to be used is specified as an attribute of the spooled file (USRDFNDTA('MAILTAG(e-mail-address)' parameter).

**Length of TO e-mail stream file.** The size, in bytes, of the Stream File entry containing the stream file which lists the TOs used for an e-mail. If the stream file name is surrounded by quotes, use all characters within the quotes as the stream file name. For information on the format of an e-mail addresses entry, see Stream File Entry.

If Use stream file for TO e-mail address entries is set to X'F0', then this value will be 0.

**Master password.** The master password associated with the PDF file to be created. Either an encrypted value or a special value in clear text is returned.

This field is applicable for Encryption levels \*40RC4 and \*128RC4.

The following special value can be used for the master password:

\*NONE No master password is specified for the PDF file.

**Mail Sender.** The user profile from whom the e-mail is being sent.

The following special values can be used for the mail sender:

\*PSFCFG The mail sender specified on the PSF configuration object will be used for the mail sender.

**Number of attachment e-mail entries.** The number of stream files to be attached to the e-mail. This value does not include the spooled file converted to PDF.

**Number of BCC e-mail address entries.** The number of BCC e-mail addresses to be used for sending an e-mail.

If Use stream file for BCC e-mail address entries is set to X'F1', then this value will be 0.

**Number of Body of e-mail entries.** The number of stream files to be placed into the body of the e-mail.

**Number of CC e-mail address entries.** The number of CC e-mail addresses to be used for sending an e-mail.

If Use stream file for CC e-mail address entries is set to X'F1', then this value will be 0.

**Number of TO e-mail address entries.** The number of TO e-mail addresses to be used for sending an e-mail.

If Use stream file for TO e-mail address entries is set to X'F1', then this value will be 0.

If Use stream file for TO e-mail address entries is set to X'F1', then this value will be 0.

**Offset to attachment e-mail entries.** The offset, in bytes, from the beginning of the PDFM0200 format structure, to a list of attachment entries. If this value is 0, a list of attachment entries is not specified.

**Offset to BCC e-mail address entries.** The offset, in bytes, from the beginning of the PDFM0200 format structure, to a list of BCC e-mail address entries. This area contains the BCC e-mail addresses to which the PDF file is sent.

If Use stream file for BCC e-mail address entries is set to X'F1', then this value will be 0.

**Offset to BCC e-mail stream file.** The offset, in bytes, from the beginning of the PDFM0200 format structure, to a file name which contains a list of BCC e-mail address entries.

If Use stream file for BCC e-mail address entries is set to X'F0', then this value will be 0.

**Offset to Body of e-mail entries.** The offset, in bytes, from the beginning of the PDFM0200 format structure, to a list of Stream file entries. If this value is 0, a list of Stream file entries is not specified.

**Offset to CC e-mail address entries.** The offset, in bytes, from the beginning of the PDFM0200 format structure, to a list of CC e-mail address entries. This area contains the CC e-mail addresses to which the PDF file is sent.

If Use stream file for CC e-mail address entries is set to X'F1', then this value will be 0.

**Offset to CC e-mail stream file.** The offset, in bytes, from the beginning of the PDFM0200 format structure, to a file name which contains a list of CC e-mail address entries.

If Use stream file for CC e-mail address entries is set to X'F0', then this value will be 0.

**Offset to message text.** The offset, in bytes, from the beginning of the PDFM0200 format, to the message text which gets placed in to the body of the e-mail. If this value is 0, message text is not specified.

**Offset to PDF file for e-mail** The offset, in bytes, from the beginning of the PDFM0200 format, to the PDF File For E-mail Entry.

**Offset to ReplyTo e-mail address.** The offset, in bytes, from the beginning of the PDFM0200 format, to the E-mail address Entry for a ReplyTo e-mail address. If this value is 0, a ReplyTo e-mail address is not specified.

**Offset to subject.** The offset, in bytes, to the subject of the e-mail. If this value is 0, a subject is not specified.

**Offset to TO e-mail address entries.** The offset, in bytes, from the beginning of the PDFM0200 format, to a list of TO e-mail address entries. This area contains the TO e-mail addresses to which the PDF file is sent.

**Offset to TO e-mail stream file.** The offset, in bytes, from the beginning of the PDFM0200 format, to a file name which contains a list of TO e-mail address entries.

If Use stream file for TO e-mail address entries is set to X'F0', then this value will be 0.

**Print PDF file.** Specifies whether users are allow to print the PDF file. This field is applicable for Encryption levels \*40RC4 and \*128RC4. An additional value is available for an Encryption level of \*128RC4.

X'F0' The user is not allowed to print the PDF file.

X'F1' The user is allowed to print the PDF file.

X'F2' The user is only allowed to print at a low resolution (image). This prevents the user from recreating the PDF file with different security settings. Printing may be slower because each page will be printed as a bit-mapped image. This value can be returned only for an Encryption level of \*128RC4.

**Use stream file for BCC e-mail address entries.** Indicates if a stream file containing BCC e-mail addresses is being used.

The following values are used:

X'F0' A stream file for BCC e-mail address entries is not being used.

X'F1' A stream file for BCC e-mail address entries is being used.

**Use stream file for CC e-mail address entries.** Indicates if a stream file containing CC e-mail addresses is being used.

The following values are used:

X'F0' A stream file for CC e-mail address entries is not being used.

*X'F1'* A stream file for CC e-mail address entries is being used.

**Reserved.** Set to a null value.

**Use stream file for TO e-mail address entries.** Indicates if a stream file containing TO e-mail addresses is being used.

The following values are used:

*X'F0'* A stream file for TO e-mail address entries is not being used.

*X'F1'* A stream file for TO e-mail address entries is being used.

**User password.** The user password associated with the PDF file to be created. Either an encrypted value or a special value in clear text is returned.

This field is applicable for Encryption levels \*40RC4 and \*128RC4.

The following special value can be used for the user password:

\*NONE No user password is specified for the PDF file.

## E-mail Address Entry

This table is used for mapping the TO, CC, BCC and ReplyTo e-mail addresses. If necessary, the field is padded with blanks (X'40'). For more details about the fields that can be used with the following table, see "E-mail Address Entry Field Description."

Offset		Type	Field
Dec	Hex		
0	0	CHAR(80)	E-mail address

## E-mail Address Entry Field Description

**E-mail address.** An e-mail address.

## Stream File Entry

This table is used for mapping the stream files that can be used for the Body, Attachment, TO, CC, and BCC stream files. Use one entry for each stream file. If you are specifying a stream file for the Body, the file extension (for example, txt or htm) must be specified in lower case. Otherwise, the file will be sent as an attachment. If necessary, the field is padded with blanks (X'40'). For more details about the field that can be used with the following table, see "Stream File Entry Field Description" on page 113.

Offset		Type	Field
Dec	Hex		
0	0	CHAR(255)	Stream file name

## Stream File Entry Field Description

**Stream file name.** A path-qualified stream file name.

## PDF Stream File Information Entry

If necessary, the Public Authority field is padded with blanks (X'40'). For more details about the fields in the following table, see "PDF Stream File Information Entry Field Descriptions."

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Offset to PDF file name
4	4	BINARY(4)	Length of PDF file name
8	8	CHAR(10)	Public Authority
18	11	CHAR(2)	Reserved

## PDF Stream File Information Entry Field Descriptions

**Length of PDF file name.** The size, in bytes, of the Stream File entry which contains the PDF file name. For information on the format of a Stream File entry, see Stream File Entry.

**Offset to PDF file name.** The offset, in bytes, from the beginning of the PDFM0200 format, to the Stream File entry for a PDF file.

**Public Authority.** The public authority for the PDF stream file. The default is \*EXCLUDE.

The following special values may be specified:

*EXCLUDE	Users are excluded from accessing the object.
*RWX	Users are given *RWX authority to perform all operations on the object except those limited to the master or controlled by object existence, object management, object alter, and object reference authority. Users can change the object and perform basic functions on the object. *RWX authority provides object operational authority and all the data authorities.
*RX	The users are given *RX authority to perform basic operations on object, such as run a program or display the contents of a file. The user is prevented from changing the object. *RX authority provides object operational authority and read and execute authorities.
*RW	The users are given *RW authority to view the contents of an object and change the contents of an object. *RW authority provides object operational authority and data read, add, update, and delete authorities.
*WX	Users are given *WX authority to change the contents of an object and run a program or search a library or directory. *WX authority provides object operational authority and data add, update, delete, and execute authorities.
*R	Users are given *R authority to view the contents of an object. *R authority provides object operational authority and data read authority.
*W	Users are given *W authority to change the contents of an object. *W authority provides object operational authority and data add, update, and delete authorities.
*X	users are given *X authority to run a program or search a library or directory. *X authority provides object operational authority and data execute authority.

**Reserved.** Set to a null value.

## PDF Spooled File Information Entry

These fields are used for mapping the PDF spooled file information. If necessary, text fields are padded with blanks (X'40'). For more details about the fields in the following table, see "PDF and AFPDS Spooled File Information Entry Field Descriptions."

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	Output queue name for output spooled file
10	A	CHAR(10)	Output queue library name for output spooled file
20	14	CHAR(10)	Spooled file name for output spooled file
30	1E	CHAR(10)	User data for output spooled file
40	28	CHAR(10)	Form type for output spooled file
50	32	CHAR(255)	User defined data for output spooled file
305	13B	CHAR(3)	Reserved

## AFPDS Spooled File Information Entry

These fields are used for mapping the AFPDS spooled file information. If necessary, text fields are padded with blanks (X'40'). For more details about the fields in the following table, see "PDF and AFPDS Spooled File Information Entry Field Descriptions."

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	Output queue name for output spooled file
10	A	CHAR(10)	Output queue library name for output spooled file
20	14	CHAR(10)	Spooled file name for output spooled file
30	1E	CHAR(10)	User data for output spooled file
40	28	CHAR(10)	Form type for output spooled file
50	32	CHAR(255)	User defined data for output spooled file
305	13B	CHAR(3)	Reserved

## PDF and AFPDS Spooled File Information Entry Field Descriptions

**Form type for output spooled file.** The type of forms that should be loaded on the printer before this spooled file is printed. This is the value which will be used when a file is spooled.

The following special values can be used for the form type for output spooled file:

\**SPLF*        The form type of the original spooled file will be used.

**Output queue library name for output spooled file.** The name of the library where the output queue for the file which will be spooled resides.

**Output queue name for output spooled file.** The name of the output queue to which the file to be spooled will be written.

The following special values can be used for the form type for output queue name for output spooled file:

*\*PSFCFG*      The output queue specified in the PSF configuration object will be used when writing the PDF file to spool.

**Reserved.** Set to a null value.

**Spooled file name for output spooled file.** The name which will be used for the file that gets spooled.

The following special values can be used for the spooled file name for output spooled file:

*\*SPLF*      The name of the original spooled file will be used.

**User data for output spooled file.** The user-specified data which will describe the file to be spooled.

The following special values can be used for the user data for output spooled file:

*\*SPLF*      The user data of the original spooled file will be used.

**User defined data for output spooled file.** The user-defined data that will be used for the file to be spooled.

The following special values can be used for the user defined data for output spooled file:

*\*SPLF*      The user defined data of the original spooled file will be used.

## File Name Entry

This table is used for mapping a file name that is not path-qualified. Use this entry for mapping a PDF file for e-mail entry. If necessary, the field is padded with blanks (X'40'). For more details about the field that can be used with the following table, see "File Name Entry Field Description."

Offset		Type	Field
Dec	Hex		
0	0	CHAR(80)	File name

## File Name Entry Field Description

**File name.** A file name which does not contain path information.

## FLTR0100 Format

The following table shows the information provided for the FLTR0100 format. If necessary, use blanks (X'40') as the pad character for text fields. For more details about the fields in the following table, see "FLTR0100 Format Field Descriptions" on page 116.

The minimum length that can be specified for this format, using field Length of filter information, is 8. For any fields not included due to the length specification, a value of \*ALL will be used.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Length of filter information
4	4	BINARY(4)	Sequence number for filter
8	8	CHAR(10)	Output queue name for filter
18	12	CHAR(10)	Output queue library name for filter
28	1A	CHAR(10)	Spooled file name for filter
38	26	CHAR(10)	Job name for filter
48	30	CHAR(10)	User name for filter
58	3A	CHAR(10)	User data for filter
68	44	CHAR(10)	Form type for filter
78	4E	CHAR(250)	Mail tag for filter

## FLTR0100 Format Field Descriptions

**Form type for filter.** The type of forms that should be loaded on the printer before this spooled file is printed or spooled files whose form type matches this value are included in the list. This field can be used in conjunction with any of the other filter values to subset the list of Map entries.

The following special values can be used for the form type for filter:

\*ALL        Map entries are included in the list regardless of the value for form type.

This field must be padded with blanks (X'40').

**Job name for filter.** The name of the job that created the spooled file. This field can be used in conjunction with any of the other filter values to subset the list of Map entries.

The following special values can be used for the form type for filter:

\*ALL        Map entries are included in the list regardless of the value for job name.

**Length of filter information.** The size, in bytes, of the Filter Information Format FLTR0100. The minimum length that can be specified is 8. For any fields not included due to the length specification, a value of \*ALL will be used.

For information on the Filter Information Format, see FLTR0100 Format.

**Mail tag for filter.** The value of the mail tag that matches this value are included in the list.

This field can be used in conjunction with any of the other filter values to subset the list of Map entries.

Here is information on specifying a mail tag for filter:

- Embedded blanks will be used as part of the mail tag.
- If necessary, the field must be padded with blanks (X'40').
- Do not use single quotes around the text string, as they will be interpreted as part of the text string.
- Case is preserved.

The following special values can be used for the mail tag:

*\*ALL* Map entries are included in the list regardless of the value for mail tag.

**Output queue library name for filter.** The name of the library where the output queue resides.

The following special value can be used for the output queue library name when filtering:

*\*CURLIB* The current library for the current job.  
*\*LIBL* The library list of the current job.  
*\*ALL* Map entries are included in the list regardless of the output queue library name. When this value is specified, the output queue library name must not be blank and must not have a value of *\*ALL*.

**Output queue name for filter.** The name of the output queue to filter on or the name of the output queue in which the spooled file is located. At least one output queue or special value must be specified. This field can be used in conjunction with any of the other filter values to subset the list of Map entries.

The following special value is used for the output queue name when filtering:

*\*ALL* Map entries are included in the list regardless of the output queue name. When this value is specified, the output queue library name must be blanks.

This field must be padded with blanks (X'40').

**Sequence number for filter.** The user-defined sequence number on which to filter.

The following values are used:

*X'00000000'* All entries will be searched. If a listed entry is non-segmented, all of the entry's information for the specified format is provided. If a listed entry is segmented, just the base portion of the entry is provided; information on individual segments is not provided. The base portion of a segmented entry consists of the following: Sequence number of entry, Output queue name for input spooled file, Output queue library name for input spooled file, User name for input spooled file, User data for input spooled file, Form type for input spooled file and Segmented entry. Fields not used in a specified format will be set to blanks ('40'X) for character data, 0 for numeric fields and '00'X for reserved fields.

*X'FFFFFFFF'* All entries will be searched. If a listed entry is non-segmented, all of the entry's information for the specified format is provided. If a listed entry is segmented, each segment's information for the specified format is provided.

*sequence-number* The sequence number of an entry. If a listed entry is non-segmented, all of the entry's information for the specified format is provided. If a listed entry is segmented, all information for the specified format for each segment is provided. The values for the filter determine whether all segments or a subset of the segments are returned.

This field can be used in conjunction with any of the other filter values to subset the list of Map entries.

**Spooled file name for filter.** The name of the spooled file to filter on or the name of the spooled file. At least one output queue or special value must be specified. This field can be used in conjunction with any of the other filter values to subset the list of Map entries.

The following special value is used for the Spooled file name when filtering:

*\*ALL* Map entries are included in the list regardless of the spooled file name.

This field must be padded with blanks (X'40').

**User data for filter.** The value of the user data of the Map entry to filter on or the value of the user data. At least one user name or special value must be specified. This field can be used in conjunction with any of the other filter values to subset the list of Map entries.

The following special value is used for the user data when filtering:

\*ALL        Map entries are included in the list regardless of the user data.

This field must be padded with blanks (X'40').

**User name for filter.** The name of the user that created the spooled file or the name of the user to filter on. At least one user name or special value must be specified. This field can be used in conjunction with any of the other filter values to subset the list of Map entries.

The following special value is used for the output queue name when filtering:

\*ALL        Map entries are included in the list regardless of the user name for filter.

This field must be padded with blanks (X'40').

## Error Messages

Message ID	Error Message Text
CPF3C21 E	Format name &1 is not valid.
CPF3C24 E	Length of the receiver variable is not valid.
CPF3C3C E	Value for parameter &1 not valid.
CPF3CF1 E	Error code parameter not valid.
CPD5F01 D	API &5 detected an internal system failure.
CPF5F11 E	Format of filter information name &1 is not valid.
CPF5F12 E	Length of filter information &1 is not valid.
CPF9801 E	Object &2 in library &3 not found.
CPF9802 E	Not authorized to object &2 in &3.
CPF9803 E	Cannot allocate object &2 in library &3.
CPF9810 E	Library &1 not found.
CPF9820 E	Not authorized to use library &1.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V5R3

[Top](#) | ["Print APIs," on page 1](#) | [APIs by category](#)

---

## Open List of Printers (QGYRPTL) API

Required Parameter Group:

Number	Parameter Name	Direction	Length
1	Receiver variable	Output	Char(*)
2	Length of receiver variable	Input	Binary(4)
3	List information	Output	Char(80)
4	Number of records to return	Input	Binary(4)
5	Filter information	Input	Char(*)
6	Format name	Input	Char(8)
7	Error Code	I/O	Char(*)

Default Public Authority: \*USE

Threadsafe: No

The Open List of Printers (QGYRPRTL) API generates a list of printers on the system and information about the printers. The list can include all printers on the system, a specific list of printers, or the printers for a single output queue. The list will be sorted by printer name.

**Note:** The QTEMP library and the system portion of the library list could be different between the main job and the server job when the list is being built asynchronously. If this is a problem, then request that the list be built synchronously.

For more information, see Process Open List APIs.

## Authorities and Locks

None.

## Required Parameter Group

### Receiver variable

OUTPUT; CHAR(\*)

The receiver variable that receives the information requested. You can specify the size of the area to be smaller than the format requested as long as you specify the length parameter correctly. As a result, the API returns only the data that the area can hold.

### Length of receiver variable

INPUT; BINARY(4)

The length of the receiver variable provided. The length of receiver variable parameter may be specified up to the size of the receiver variable specified in the user program. If the length of receiver variable parameter specified is larger than the allocated size of the receiver variable specified in the user program, the results are not predictable. The minimum length is 8 bytes.

### List information

OUTPUT; CHAR(80)

The variable used to return status information about the list of printers that were opened. For a description of the layout of this parameter, see Open list information format.

### Number of records to return

INPUT; BINARY(4)

The number of records in the list to put into the receiver variable after the filtering and sorting is done.

If the value zero is specified, only the list information is returned and no actual list entries are returned.

### Filter information

INPUT; CHAR(\*)

The information in this parameter is used to determine which printers to include in the list. Printers can be filtered based on printer name, output queue name, or both. See "Format of Filter Information" on page 123 for a description of the layout of this parameter.

### Format name

INPUT; CHAR(8)

The format of the printer information being returned.

You must specify one of the following:

*"Format PRTL0100" on page 120*

The basic information about the printer.

“Format PRTL0200”

All the information in format PRTL0100 plus additional information about the output queue and the writer for the printer, and information about the spooled file currently being printed.

### Error code

I/O; CHAR(\*)

The structure in which to return error information. For the format of the structure, see Error code parameter.

## Format of Receiver Variable

The following tables describe the order and format of the data that is returned in the receiver variable.

### Format PRTL0100

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	Device name
10	A	CHAR(50)	Text description
60	3C	BINARY(4)	Overall status

### Format PRTL0200

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	Device name
10	A	CHAR(50)	Text description
60	3C	BINARY(4)	Overall status
64	40	BINARY(4)	Device status
68	44	CHAR(10)	Output queue name
78	4E	CHAR(10)	Output queue library
88	58	CHAR(1)	Output queue status
89	59	CHAR(10)	Writer name
99	63	CHAR(1)	Writer status
100	64	CHAR(1)	Writer started
101	65	CHAR(10)	Form type
111	6F	CHAR(10)	Current file name
121	79	CHAR(10)	Current file user
131	83	CHAR(10)	Current file user-specified data
141	8D	CHAR(1)	Network directory publishing status

## Field Descriptions

**Current file name.** The name of the spooled file currently being processed by the writer. This field is blank when no file is printing.

**Current file user.** The name of the user who owns the spooled file currently being processed by the writer. This field is blank when no file is printing.

**Current file user-specified data.** The user-specified data that describes the file currently being processed by the writer.

**Device name.** The name of the printer device.

**Device status.** The status of the printer device using two fields:

**Note:** Not all of the following values apply to printer devices.

**Status Numeric**

<b>Code (decimal)</b>	<b>Overall Status</b>
0	VARIED OFF
5	AS/36 DISABLED
10	VARY OFF PENDING
20	VARY ON PENDING
30	VARIED ON
40	CONNECT PENDING
50	SIGNON DISPLAY
60	ACTIVE
62	AS/36 ENABLED
63	ACTIVE READER
66	ACTIVE WRITER
70	HELD
75	POWERED OFF
80	RCYPND
90	RCYCNL
95	SYSTEM REQUEST
100	FAILED
103	FAILED READER
106	FAILED WRITER
110	DIAGNOSTIC MODE
111	DAMAGED
112	LOCKED
113	UNKNOWN

**Form type.** The type of form being used to print the spooled file.

Possible values are:

<i>*ALL</i>	The writer is started with the option to print all spooled files of any form type.
<i>*FORMS</i>	The writer is started with the option to print all the spooled files with the same form type before using a different form type.
<i>*STD</i>	The writer is started with the option to print all the spooled files with a form type of <i>*STD</i> .
<i>Form type name</i>	The writer is started with the option to print all the spooled files with the form type you specified.

**Network directory publishing status.** Whether the printer is published in the network directory.

Possible values are:

0	Printer is not published
1	Printer is published

**Output queue library name.** The name of the library that contains the output queue from which spooled files are selected for printing.

**Output queue name.** The name of the output queue from which spooled files are being selected for printing.

**Output queue status.** The status of the output queue from which spooled files are being selected for printing.

Possible values are:

*H* The output queue is held.  
*R* The output queue is released.

**Overall status.** The overall status of the logical printer. Possible values are:

1 Unavailable  
2 Powered off or not yet available  
3 Stopped  
4 Message waiting  
5 Held  
6 Stop (pending)  
7 Hold (pending)  
8 Waiting for printer  
9 Waiting to start  
10 Printing  
11 Waiting for printer output  
12 Connect pending  
13 Powered off  
14 Unusable  
15 Being serviced  
999 Unknown

**Text description.** The text description of the printer device.

**Writer name.** The name of the writer for this printer.

**Writer started.** Indication of whether or not a writer is started for this printer. Possible values are:

0 No writer is started  
1 Writer is started

**Writer status.** The status of the writer for this printer. Possible values are:

*X'01'* Started  
*X'02'* Ended  
*X'03'* On job queue  
*X'04'* Held  
*X'05'* Message waiting

## Format of Filter Information

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Number of printer names
Offsets vary. These fields repeat, in the order listed, for each printer to filter on.		CHAR(10)	Printer name
		CHAR(2)	Reserved
Offset varies.		BINARY(4)	Number of output queues
Offsets vary. These fields repeat, in the order listed, for each output queue to filter on.		CHAR(10)	Output queue
		CHAR(10)	Output queue library

When no filter is to be done, the filter specification must be two BINARY(4) values set to zero.

## Field Descriptions

**Number of output queues.** The total number of output queue names in the output queue name array.

0 Does not filter by output queue  
1-1000 Filters the specified number of output queue names

**Number of printer names.** The total number of printer device names in the printer names array.

0 Does not filter by printer name  
1-1000 Filters the specified number of printer names

**Output queue.** The name of an output queue to be included in the list.

**Output queue library.** The name of the library that contains the output queue.

**Printer name.** The name of a printer to be included in the list.

## Error Messages

Message ID	Error Message Text
CPF24B4 E	Severe error while addressing parameter list.
CPF3C19 E	Error occurred with receiver variable specified.
CPF3C21 E	Format name &1 is not valid.
CPF3C90 E	Literal value cannot be changed.
CPF3CF1 E	Error code parameter not valid.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.
GUI0027 E	&1 is not valid for number of records to return.
GUI0108 E	Filter information is not valid. Reason code &1.

API introduced: V3R6

---

## Publish Print Information (QSPPUBPI) API

Required Parameter Group:

1	Publishing action	Input	Binary(4)
2	Print information	Input	Char(*)
3	Length of Print Information	Input	Binary(4)
4	Information format	Input	Char(8)
5	Error Code	I/O	Char(*)

Default Public Authority: \*USE  
Threadsafe: No

The Publish Print Information (QSPPUBPI) API collects print object information and queues it for publishing in an LDAP directory whenever a print object is published. It also can be used to stop publishing for a print object.

Print publishing is included as part of the configuration for directory services from the System i™ Navigator interface. Before configuring print publishing, the directory to use should be considered carefully to avoid migration problems. Changing the directory can be done from System i Navigator or by using the Change Directory Server Attributes (QgldChgDirSrvA) API.

### Authorities and Locks

*Printer device description authority*

The caller needs both of the following:

- \*IOSYSCFG special authority
- \*CHANGE, \*OBJMGT to the printer device

### Required Parameter Group

#### Publishing action

INPUT; BINARY(4)

The action to be performed for the print object. The following values may be specified:

- 1 Publish a print object
- 2 Stop publishing for a print object
- 3 Stop publishing for a print object and delete the object's information from the directory

#### Print information

INPUT; CHAR(\*)

The formatted print information. For more information, see "Printer Information Formats" on page 125.

#### Length of print information

INPUT; BIN(4)

The total length of the print information provided for the information format specified.

#### Information format

INPUT; CHAR(8)

The type of print information on which the publishing action will be performed. The possible format names are:

*SPPA0100* Printer devices

## Error code

I/O; CHAR(\*)

The structure in which to return error information. For the format of the structure, see Error code parameter.

## Printer Information Formats

The structure of the printer information format is listed below. For a description of each field, see “Field Descriptions.”

### SPPA0100 Format

Offset		Type	Field
Dec	Hex		
0	0	Array of CHAR(10)	Printer device name

This format is used with a printer device name or list of names.

- Action 1 is used to publish a new printer or list of printers, such as may be selected in System i Navigator. This action causes information about each printer to be stored in the directory.
- Action 2 is used to stop publishing a published printer or list of printers. No action is taken if the printer device information was not published previously. This action removes the i5/OS® specific information from the printer’s directory entry, but does not remove the entry from the directory. Future changes to the printer device will not be reflected in the directory.
- Action 3 is used to stop publishing a published printer or list of printers and to remove the directory entry for each printer if one exists. No action is taken if the printer device information was not published previously.

## Field Descriptions

**Printer device name.** The name of a printer device to publish or stop publishing.

## Error Messages

Message ID	Error Message Text
CPF2207 E	Not authorized to use object &1 in library &3 type *&2.
CPF24B4 E	Severe error occurred while addressing parameter list.
CPF2625 E	Not able to allocate object &1.
CPF2702 E	Device Description &2 not found.
CPF34AD E	Value specified in input data is not valid.
CPF34AE E	Printer publishing is not configured or is disabled.
CPF34BA E	Action &1 not valid.
CPF34BB E	Error occurred while publishing printer &1.
CPF3C1D E	Length specified in parameter &1 is not valid.
CPF3C1E E	Required parameter &1 omitted.
CPF3C21 E	Format name &1 is not valid.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPF8105 E	Device Description &4 damaged.
CPF9872 E	Program or service program &1 in library &2 ended.
CPFA314 E	Memory allocation error.

API introduced: V5R1

---

## Remove PDF Map Object Entries (QPQRPM) API

Required Parameter Group:

1	Number of entries removed	Output	Binary(4)
2	Entries removed	Output	Char(*)
3	Length of entries removed	Input	Binary(4)
4	Entry lengths and entry offsets	Output	Array(*) of Char(8)
5	Length of entry lengths and offsets	Input	Binary(4)
6	Qualified PDF Map object name	Input	Char(20)
7	Maximum number of entries	Input	Binary(4)
8	Remove criteria	Input	Char(*)
9	Error code	I/O	Char(*)

Default Public Authority: \*USE  
Threadsafe: Yes

The Remove PDF Map object Entries (QPQRPM) API removes one or more PDF Map object entries that match the values specified on the remove criteria parameter. It returns the number of entries that were removed and, optionally, returns the actual PDF map entries removed.

### Authorities and Locks

*PDF Map object Library Authority*  
\*EXECUTE

*PDF Map object Authority*  
\*CHANGE

*PDF Map object Lock*  
\*SHRUPD

### Required Parameter Group

**Number of entries removed**  
OUTPUT; BINARY(4)

The number of PDF map entries, satisfying the values specified on the remove criteria parameter, that were successfully removed from the PDF Map object. If this field is 0, no entries satisfied the remove criteria. This value can never be greater than the maximum number of entries parameter.

**Entries removed**  
OUTPUT; CHAR(\*)

The actual entries removed. All entries that satisfied the remove criteria parameter and were removed (up to the maximum number of entries parameter) are returned if sufficient space is provided. The API returns only the data that the area can hold.

If you do not want the entries that were removed to be returned in this parameter, specify 0 for the length of entries removed parameter.

Every entry removed causes the number of entries removed parameter to be incremented by 1.

The format of this parameter is the same as the Remove criteria parameter. Refer to "Remove Criteria Format" on page 128 for the layout of this parameter.

**Length of entries removed**  
INPUT; BINARY(4)

The length of the entries removed parameter. If this length is larger than the actual size of the entries removed parameter, the results may not be predictable. The minimum length is 0 or >=

(greater than or equal to) 8 bytes. If 0 is used, the entries removed from the PDF Map are not returned and the bytes returned and the bytes available in the entries removed parameter are not set.

### **Entry lengths and entry offsets**

OUTPUT; ARRAY of CHAR(8)

A data structure that contains entry lengths and entry offsets for all entries that were found that met the remove criteria parameter. An entry length and entry offset exist for every entry returned in the entries removed parameter. These entry lengths and entry offsets are used to parse through the entries removed parameter. If the length of entries removed parameter is 0, this information will not be returned.

The size of the entry lengths and entry offsets parameter should be at least:

$8 + (\text{the maximum number of entries parameter} * 8)$

You must provide enough space in both the entries removed and the entry lengths and offset parameter for this API to return complete information to you.

See “Format for Entry Lengths and Entry Offsets” on page 128 for details on the data structure.

### **Length of entry lengths and entry offsets**

INPUT; BINARY(4)

The length of the entry lengths and entry offsets. If the length is longer than the entry lengths and entry offsets parameter, the results may not be predictable. The minimum length is 8. If the length of entries removed parameter is 0, which means you do not want the entries removed to be returned, this parameter is ignored.

### **Qualified PDF Map object name**

INPUT; CHAR(20)

The PDF Map object from which you want to remove entries, and the library in which it is located. The first 10 characters contain the PDF Map object name, and the second 10 characters contain the library name.

You can use these special values for the library name:

\**CURLIB*            The job’s current library  
\**LIBL*                The library list

### **Maximum number of entries**

INPUT; BINARY(4)

The maximum number of PDF Map object entries to be removed that satisfy the remove criteria. Valid values are 1 through 4095.

### **Remove criteria**

INPUT; CHAR(\*)

The criteria used to find matches in the PDF Map object.

Refer to “Remove Criteria Format” on page 128 for details on the format.

### **Error code**

I/O; CHAR(\*)

The structure in which to return error information. For the format of the structure, see Error code parameter.

## Format for Entry Lengths and Entry Offsets

The following information is returned in the entry lengths and entry offsets parameter. This information is needed to parse through the entries removed parameter. For detailed descriptions of the fields in the table, see "Field Descriptions."

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Bytes returned
4	4	BINARY(4)	Bytes available
<b>Note:</b> The following fields will be repeated. The number of times they are repeated depends on the length of the entry lengths and entry offsets parameter and the number of entries actually removed.			
		BINARY(4)	Entry offset
		BINARY(4)	Entry length

## Remove Criteria Format

The following information is the format of the remove criteria. The same format is also used for what is returned in the Entries removed parameter. For more details about the fields in the following table, see "Field Descriptions."

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Sequence number
14	E	CHAR(10)	Output queue name
24	18	CHAR(10)	Output queue library name
34	22	CHAR(10)	Spooled file name
44	2C	CHAR(10)	Job name
54	36	CHAR(10)	User name
64	40	CHAR(10)	User data
74	4A	CHAR(10)	Form type
84	54	CHAR(250)	Mail tag

## Field Descriptions

**Bytes available.** The length of all data available to return. All available data is returned if enough space is provided.

**Bytes returned.** The length of the data actually returned.

**Entry length.** The length of the entry removed from the PDF Map object. Valid values are 1-2000, depending on how the PDF Map object was created.

**Entry offset.** The number of bytes from the beginning of the immediately preceding entry to the first byte of the entry returned. For the first entry, the offset is the number of bytes from the beginning of the parameter to the first byte of the first entry.

**Form type.** The type of forms that should be loaded on the printer before this spooled file is printed or spooled files whose form type matches this value are included in the list. This field can be used in conjunction with any of the other values in the Remove criteria format to select a list of entries to remove.

The following special values can be used for the form type:

*\*ALL*            Map entries are removed regardless of the value for form type.

**Job name.** The name of the job that created the spooled file. This field can be used in conjunction with any of the other values in the Remove criteria format to select a list of entries to remove.

The following special values can be used for the job name:

*\*ALL*            Map entries are removed regardless of the value for job name.

**Mail tag.** The value of the mail tag for a spooled file or a spooled file segment. This field can be used in conjunction with any of the other values in the Remove criteria format to select a list of entries to remove.

The following special values can be used for mail tag:

*\*ALL*            Map entries are removed regardless of the value for mail tag.

**Output queue library name.** The name of the library where the output queue resides.

The following special values can be used for the output queue library name:

*\*CURLIB*            The current library for the current job.  
*\*LIBL*                The library list of the current job.

**Output queue name.** The name of the output queue in which the spooled file is located. At least one output queue or special value must be specified. This field can be used in conjunction with any of the other values in the Remove criteria format to select a list of entries to remove.

The following special values can be used for the output queue name:

*\*ALL*            Map entries are removed regardless of the value for output queue name.

**Sequence number.** The user-defined sequence number for an entry.

*X'00000000'*            All entries will be examined. If an entry is non-segmented and it matches the remove criteria, then it is removed. If an entry is segmented, only those segments that match the remove criteria will be removed.  
*sequence-number*        The sequence number of an entry. A single entry will be examined. If the entry is non-segmented and it matches the remove criteria, then it is removed. If an entry is segmented, only segments that match the remove criteria will be removed.

This field can be used in conjunction with any of the other values in the Remove criteria format to select a list of entries to remove.

**Spooled file name.** The name of the spooled file. At least one output queue or special value must be specified. This field can be used in conjunction with any of the other values in the Remove criteria format to select a list of entries to remove.

The following special value is used for the output queue name:

\*ALL        Map entries are removed regardless of the spooled file name.

**User data.** The value of the user data. At least one user name or special value must be specified. This field can be used in conjunction with any of the other values in the Remove criteria format to select a list of entries to remove.

The following special value is used for the user data:

\*ALL        Map entries are removed regardless of the user data.

**User name.** The name of the user that created the spooled file. At least one user name or special value must be specified. This field can be used in conjunction with any of the other values in the Remove criteria format to select a list of entries to remove.

The following special value is used for the output queue name:

\*ALL        Map entries are removed regardless of the user name.

## Error Messages

Message ID	Error Message Text
CPF3CF1 E	Error code parameter not valid.
CPF3C70 E	Length of entries removed parameter is not valid.
CPF3C76 E	Length of lengths and offsets of entries &1 is not valid.
CPF3C79 E	Maximum number of entries &1 is not valid.
CPD5F01 D	API &5 detected an internal system failure.
CPF9801 E	Object &2 in library &3 not found.
CPF9802 E	Not authorized to object &2 in &3.
CPF9810 E	Library &1 not found.
CPF9820 E	Not authorized to use library &1.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V5R3

[Top](#) | ["Print APIs," on page 1](#) | [API by category](#)

---

## Retrieve AFP Information (QPQRAFPI) API

Required Parameter Group:

1	Receiver variable	Output	Char(*)
2	Length of receiver variable	Input	Binary(4)
3	Format name	Input	Char(8)
4	Qualified AFP™ object name	Input	Char(20)
5	Format qualifier	Input	Char(10)
6	Error code	I/O	Char(*)

Default Public Authority: \*USE  
Threadsafe: No

The Retrieve AFP Information (QPQRAFPI) API returns specific information about a PSF configuration object, a font table, or a coded font. The size of the receiver variable determines the amount of information returned.

## Authorities and Locks

*Library authority*  
\*EXECUTE

*PSF configuration object authority*  
\*USE

*Font table authority*  
\*USE

*Font resource authority*  
\*USE

*PSF configuration object lock*  
\*SHRRD

*Font table lock*  
\*SHRRD

*Font resource lock*  
\*SHRRD

## Required Parameter Group

**Receiver variable**  
OUTPUT; CHAR(\*)

The receiver variable that receives the information requested. You can specify the size of the area to be smaller than the format requested as long as you specify the length parameter correctly. As a result, the API returns only the data the area can hold.

**Length of receiver variable**  
INPUT; BINARY(4)

The length of the receiver variable provided. The length of the receiver variable parameter may be specified up to the size of the receiver variable specified in the user program. If the length of the receiver variable parameter specified is larger than the allocated size of the receiver variable specified in the user program, the results are not predictable. The minimum length is 8 bytes.

**Format name**  
INPUT; CHAR(8)

The format of the information to be returned for the specified object. The valid format names are:

<i>"PSFI0100 Format" on page 133</i>	Information about the specified PSF Configuration Object.
<i>"CFNI0100 Format" on page 143</i>	Information about the specified coded font.
<i>"FCSI0100 Format" on page 144</i>	Information about the specified font character set.
<i>"FTBI0100 Format" on page 145</i>	Information about the specified font table.
<i>"PHCS0100 Format" on page 146</i>	Information about the specified font table.
<i>"HPCS0100 Format" on page 146</i>	Information about the specified font table.
<i>"PHCP0100 Format" on page 147</i>	Information about the specified font table.
<i>"HPCP0100 Format" on page 147</i>	Information about the specified font table.
<i>"PPCS0100 Format" on page 148</i>	Information about the specified font table.

## Qualified AFP object name

INPUT; CHAR(20)

The AFP object for which you would like to retrieve information. The qualified object name has two parts. The first 10 characters contain the object name.

You may not use special values for the object name when the format name is PSFI0100, CNFI0100, FTBI0100 or PPCS0100.

You must use the following special value for the object name when the format name is FCSI0100:

*\*FNTCHRSET* All coded fonts that contain the font character set specified by the font character set parameter are returned.

You may use one of the following special values for the object name when the format name is FTBI0100:

*\*PHFCS* Information about the printer-resident to host-resident font character set table is to be returned.

*\*SYSPHFCS* Information about the system printer-resident to host-resident font character set table is to be returned.

*\*HPFCS* Information about the host-resident to printer-resident font character set table is to be returned.

*\*SYSHPFCS* Information about the system host-resident to printer-resident font character set table is to be returned.

*\*PHCP* Information about the printer-resident to host-resident code page mapping table is to be returned.

*\*SYSPHCP* Information about the system printer-resident to host-resident code page mapping table is to be returned.

*\*HPCP* Information about the host-resident to printer-resident code page mapping table is to be returned.

*\*SYSHPCP* Information about the system host-resident to printer-resident code page mapping table is to be returned.

You may use one of the following special values for the object name when the format name is PHCS0100:

*\*PHFCS* The printer-resident to host-resident font character set table is to be returned.

*\*SYSPHFCS* The system printer-resident to host-resident font character set table is to be returned.

You may use one of the following special values for the object name when the format name is HPCS0100:

*\*HPFCS* The host-resident to printer-resident font character set table is to be returned.

*\*SYSHPFCS* The system host-resident to printer-resident font character set table is to be returned.

You may use one of the following special values for the object name when the format name is PHCP0100:

*\*PHCP* The printer-resident to host-resident code page mapping table is to be returned.

*\*SYSPHCP* The system printer-resident to host-resident code page mapping table is to be returned.

You may use one of the following special values for the object name when the format name is HPCP0100:

*\*HPCP* The host-resident to printer-resident code page mapping table is to be returned.

\*SYSHPCP The system host-resident to printer-resident code page mapping table is to be returned.

You must leave the second 10 characters blank when you use a special value for the AFP object name when the format name is PHCS0100, HPCS0100, PHCP0100, or HPCP0100.

The second 10 characters contain the name of the library where the object is located when the format name is PSFI0100, CFNI0100, FTBI0100, PHCS0100, HPCS0100, PHCP0100, HPCP0100, or PPCS0100.

The following special values are allowed when the format name is PSFI0100, CFNI0100, FTBI0100, PHCS0100, HPCS0100, PHCP0100, HPCP0100, or PPCS0100:

\*CURLIB The job's current library is used to locate the object. If no library is specified as the current library for the job, the QGPL library is used.  
\*LIBL The library list is used to locate the object.

The actual library that is used is returned with the object information.

The second 10 characters contain the name of the library searched for coded fonts that contain the font character set specified in the first 10 characters of the format qualifier when the format name is FCSI0100.

The following special values are allowed when the format name is FCSI0100:

\*CURLIB Only the libraries in the current library for the job are searched. If no library is specified as the current library for the job, QGPL is used.  
\*LIBL All libraries in the job's library list are searched.  
\*USRLIBL Only the libraries listed in the user portion of the job's library list are searched.  
\*ALL All libraries in the system, including QSYS, are searched.  
\*ALLUSR All user libraries are searched. For information on the libraries included, see \*ALLUSR in Generic library names.  
\*DBCSFNTLIB The DBCS font libraries QFNT61, QFNT62, QFNT63, QFNT64, and QFNT65 will be searched.

### Format qualifier

INPUT; CHAR(10)

The font character set name to search for in the coded font objects when the format name is FCSI0100 and the AFP object name is \*FNTCHRSET. All coded fonts that contain the font character set specified by the format qualifier parameter are returned.

This parameter must be blank if the format name is not FCSI0100.

### Error code

I/O; CHAR(\*)

The structure in which to return error information. For the format of the structure, see Error code parameter.

## PSFI0100 Format

The following table shows the information returned for the PSFI0100 format.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Bytes returned
4	4	BINARY(4)	Bytes available
8	8	CHAR(10)	PSF configuration object name
18	12	CHAR(10)	PSF configuration object library name

Offset		Type	Field
Dec	Hex		
28	1C	CHAR(2)	Reserved
30	1E	CHAR(10)	User resource library
40	28	BINARY(4)	Offset to device resource library list entries
44	2C	BINARY(4)	Number of device resource library list entries
48	30	CHAR(10)	IPDS™ pass through
58	3A	CHAR(10)	Activate release timer
68	44	BINARY(4)	Release timer
72	48	BINARY(4)	Restart timer
76	4C	BINARY(4)	APPC and TCP/IP retry count
80	50	BINARY(4)	Delay between APPC retries
84	54	CHAR(10)	Automatic session recovery
94	5E	CHAR(10)	Automatic session recovery message option
104	68	BINARY(4)	Acknowledgment frequency
108	6C	CHAR(10)	Blank page
118	76	CHAR(10)	Page size control
128	80	CHAR(10)	Resident fonts
138	8A	CHAR(10)	Resource retention
148	94	CHAR(10)	Edge orient
158	9E	CHAR(10)	Use outline fonts
168	A8	BINARY(4)	Offset to PSF-defined options
172	AC	BINARY(4)	Number of PSF-defined options
176	B0	CHAR(10)	Font substitution messages
186	BA	CHAR(10)	Capture host fonts at printer
196	C4	CHAR(10)	Font resolution for formatting
206	CE	CHAR(10)	Font mapping table name
216	D8	CHAR(10)	Font mapping table library
226	E2	CHAR(10)	Cut sheet emulation mode
236	EC	CHAR(10)	Use DBCS simulation fonts
246	F6	CHAR(2)	Reserved
248	F8	BINARY(4)	Printer response timer
252	FC	CHAR(10)	Generate PDF output
262	106	CHAR(10)	Device emulation type
272	110	CHAR(10)	Paper size for drawer 1
282	11A	CHAR(10)	Paper size for drawer 2
292	124	CHAR(10)	Multiple PDF files
302	12E	CHAR(10)	Data queue
312	138	CHAR(10)	Data queue library
322	142	CHAR(10)	Sender of electronic mail
332	14C	CHAR(10)	Mapping program
342	156	CHAR(10)	Mapping program library

Offset		Type	Field
Dec	Hex		
352	160	CHAR(10)	Output queue
362	16A	CHAR(10)	Output queue library
372	174	CHAR(256)	Directory
628	274	BINARY(4)	Directory CCSID
632	278	CHAR(50)	Text
682	2AA	CHAR(10)	Multiple PDF files process option
692	2B4	CHAR(10)	PDF fonts inline
702	2BE	CHAR(2)	Reserved
704	2C0	BINARY(4)	Offset to mail server names
708	2C4	BINARY(4)	Number of mail servers
	2C8	CHAR(10)	Mapping object name
722	2D2	CHAR(10)	Mapping object library
732	2DC	CHAR(80)	PDF administrator
812	32C	BINARY(4)	Offset to additional generate PDF output options
816	330	BINARY(4)	Number of additional generate PDF output options
820	334	CHAR(10)	Save AFP Data
830	33E	CHAR(10)	AFP Output Queue name
840	348	CHAR(10)	AFP Output Queue library
		CHAR(*)	Device resource library list entries (ARRAY of CHAR(10))
		CHAR(*)	PSF-defined option (ARRAY of CHAR(30))
		CHAR(*)	Mail server names. (ARRAY of CHAR(255))
		CHAR(*)	Additional generate PDF output options (ARRAY of CHAR(10))

## Field Descriptions

**Acknowledgment frequency.** The frequency, in pages, with which PSF/400 sends IPDS acknowledgment requests to a printer. Possible values are 1 to 32767.

### Activate release timer.

The point at which the release timer is activated.

The possible values are:

- \**NORDYF*      The release timer is to be activated when there are no spooled files in the printer's output queue with a status of RDY and the last page of the last spooled file processed has printed.
- \**IMMED*      The release timer is to be activated immediately after PSF has successfully linked to the printer.
- \**PRTNORDYF*    The release timer is to be activated:
  - After the writer receives an indication from the printer to release the IPDS dialog,
  - There are no spooled files in the printer's output queue with a status of RDY, and
  - The last page of the last spooled file processed has printed.
- \**PRTIMMED*    The release timer is to be activated immediately after the writer receives an indication from the printer to release the IPDS dialog.

**Additional generate PDF output options (ARRAY of CHAR(10))** If the user would like the spooled file converted to PDF using the IPDS to PDF transform and more than one option used to distribute the associated PDF output.

Possible values are:

*NONE	No PDF generation of the spooled file should be performed
*MAIL	PDF output will be generated and mailed electronically.
*STMF	PDF output will be generated and placed into a stream file in the directory identified in the directory field.
*SPLF	PDF output will be generated and placed on the output queue identified in the output queue field.

**AFP output queue library.** The name of the library which contains the default output queue where AFP data will be spooled.

Possible values are:

*Output-queue-library* Name of output queue library.

**AFP output queue name.** The name of the default output queue where AFP data will be spooled.

Possible values are:

*Output-queue-name* Name of output queue

**APPC and TCP/IP retry count.** The number of times to retry a session start request when attempting to establish a session with a printer.

The possible values are:

1 to 99	The number of retries.
-1	No limit is put on the number of retries. (*NOMAX)

**Automatic session recovery.** Whether PSF/400 will automatically attempt to resume printing when a session has been ended unexpectedly by a device.

The possible values are:

*NO	PSF/400 ends when a session has been unexpectedly ended by a device.
*YES	Automatic session recovery is enabled. PSF/400 attempts to reestablish a session that has been ended unexpectedly by a device.

**Automatic session recovery message option.** Whether an informational message or an inquiry message is sent to the message queue associated with the writer when PSF/400 is performing automatic session recovery. This field is filled in when automatic session recovery is \*YES; otherwise, it is set to blanks.

The possible values are:

*INFO	An informational message is sent to the message queue associated with the writer when PSF/400 is performing automatic session recovery.
-------	---

*\*INQ* An inquiry message is sent to the message queue associated with the writer when PSF/400 is performing automatic session recovery.

**Blank page.** Whether PSF/400 issues a blank page after every separator page and spooled file copy that contains an odd number of pages.

The possible values are:

*\*YES* PSF/400 issues a blank page after every separator page and spooled file copy that contains an odd number of pages.

*\*NO* PSF/400 does not issue a blank page after every separator page and spooled file copy that contains an odd number of pages.

**Bytes available.** The length of all data available to return. All available data is returned if enough space is provided.

**Bytes returned.** The length of the data actually returned.

**Capture host fonts at printer.** Whether the printer should capture host downloaded fonts.

The possible values are:

*\*NO* The printer should not capture host fonts.

*\*YES* The printer should capture host fonts.

**Cut sheet emulation mode.** The degree to which PSF/400 will do size checking of the document when using Cut Sheet Emulation.

The possible values are:

*\*NONE* No checking will be done to verify that the document page will fit on half the continuous forms physical page.

*\*CHKFIRST* The first page of each copy group will be checked to determine if the page will fit on half the continuous forms page.

*\*CHKALL* The front side page will be checked to determine if the page will fit on half the continuous forms page.

**Data queue.** The name of the data queue where PSF will log the IPDS to PDF transform completion notifications. This field is used only when the generate PDF output field has a value other than *\*NONE*; otherwise it is set to blanks.

Possible values are:

*\*NONE* No data queue should be used to log completions.

*PDF-data-queue* The name of the data queue used for logging transformation completions.

**Data queue library.** The name of the library in which the data queue is located. This field is used only when the generate PDF output field has a value other than *\*NONE*; otherwise, it is set to blanks.

**Delay between APPC retries.** The number, between 0 and 999, of seconds PSF/400 pauses after it receives notification that a session start request has failed.

**Device emulation type.** The type of device that the IPDS to PDF transform virtual printer device should emulate. This field is used only when the generate PDF output field has a value other than \*NONE; otherwise it is set to blanks.

Possible values are:

*IP40240	The virtual printer device should emulate an IP40 printer device configured at 240 pels per inch resolution.
*IP40300	The virtual printer device should emulate an IP40 printer device configured at 300 pels per inch resolution.
*4028	The virtual printer device should emulate a 4028 printer device.
*3812	The virtual printer device should emulate a 3812 printer device.

**Device resource library list entry.** A device resource library to be used for searching AFP resources.

The following special value may be returned:

*DFT	The following libraries, if present on the system, are used in searching for AFP resources. <ul style="list-style-type: none"><li>• QFNTCPL</li><li>• QFNT01 - QFNT19</li><li>• QFNT61 - QFNT69</li></ul>
------	---

**Directory.** The directory in which the PDF files will be stored. This field is used only when the generate PDF output field has a value of \*STMF; otherwise, it is set to blanks.

**Directory CCSID.** The CCSID (coded character set ID) of the directory.

The possible values are:

0	There is no directory.
1 to 65533	The CCSID of the directory

**Edge orient.** Whether PSF/400 will rotate the output 270 degrees instead of 90 degrees when the page rotation value of a spooled file is \*COR or \*AUTO.

The possible values are:

*NO	The output remains at its original orientation.
*YES	*COR and *AUTO output of 90 degrees is rotated an additional 180 degrees before printing.

**Font mapping table name.** The name of a printer-resident to printer-resident font table.

The following special value may be returned:

*NONE	No printer-resident to printer-resident font table is specified.
-------	--

**Font mapping table library.** The library in which the printer-resident to printer-resident font table resides.

**Font resolution for formatting.** The resolution PSF/400 should use to print the spooled file when:

- Printing to a multiple resolution printer and
- The printer is configured to report support of multiple resolutions and

- The spooled file does not specify the font metrics and resolution with which to print the spooled file or the font is not available at that resolution.

The possible values are:

<i>*SEARCH</i>	Search the library list for the first occurrence of a host font with a name match.
<i>240</i>	The font resolution used to print the spooled file should be 240 pels per inch.
<i>300</i>	The font resolution used to print the spooled file should be 300 pels per inch.

**Font substitution messages.** Whether PSF/400 will issue messages indicating that a successful font substitution was performed.

The possible values are:

<i>*YES</i>	Messages indicating that a successful font substitution was performed are issued.
<i>*NO</i>	Messages indicating that a successful font substitution was performed are not issued.

**Generate PDF output.** Whether the user would like the spooled file converted to PDF using the IPDS to PDF transform and if so, what to do with the associated PDF output.

Possible values are:

<i>*NONE</i>	No PDF generation of the spooled file should be performed
<i>*MAIL</i>	PDF output will be generated and mailed electronically.
<i>*STMF</i>	PDF output will be generated and placed into a stream file in the directory identified in the directory field.
<i>*SPLF</i>	PDF output will be generated and placed on the output queue identified in the output queue field.

**IPDS pass through.** Whether IPDS pass-through is done for the device.

The possible values are:

<i>*NO</i>	No IPDS pass-through is done.
<i>*YES</i>	IPDS pass-through is to be done for the device for all spooled files that are eligible for IPDS pass-through.

**Mail server names.** The mail server names to use for electronically mailing the resulting PDF file from the IPDS to PDF transform.>

**Mapping object library.** Name of the PDF mapping object library, if one was specified on the PSFCFG object. This is set to blanks (X'40') if no value was provided on the PDFMAPBOBJ parameter of the PSFCFG object.

**Mapping object name.** Name of the PDF mapping object, if one was specified on the PSFCFG object. This is set to blanks (X'40') if no value was provided on the PDFMAPBOBJ parameter of the PSFCFG object.

**Mapping program.** The name of a user-defined mapping program. The mapping program will receive as input the spool file attributes and mail tag information(if specified in spool file). It will return as output the action to be taken (e-mail, store PDF file in Integrated File System (IFS), or respool PDF to specified output queue). It will also return any information needed for the specified action, such as one or more mail addresses to use for electronic mailing of the PDF output when e-mail is specified for the action. If this field is blank, then the action specified in the PSF configuration object will be used.

Possible values are:

<i>*NONE</i>	No mapping program is specified.
<i>*IBMPGM</i>	Use the IBM <sup>®</sup> mapping program provided with i5/OS <sup>®</sup> . When this value is specified, the name of a PDF mapping object must be specified
<i>PDF-mapping-program</i>	The name of the user program to use for mapping.

**Mapping program library.** The name of the library in which the mapping program is located.

**Multiple PDF files.** Specifies if PSF should tell the IPDS to PDF transform to split the PDF output into multiple PDF files at the DDS keyword STRPAGGRP boundaries. This field is used only when the generate PDF output field has a value other than \*NONE; otherwise, it is set to blanks.

The possible values are:

<i>*NO</i>	One PDF file should be generated.
<i>*YES</i>	Multiple PDF files should be created when applicable.

**Multiple PDF files process option.** When Multiple PDF files is yes, specifies how PSF should tell the IPDS to PDF transform to split the PDF output into multiple PDF files at the DDS keyword STRPAGGRP boundaries. This field is used only when the generate PDF output field has a value other than \*NONE; otherwise, it is set to blanks.

The possible values are:

<i>*SPLIT</i>	Multiple PDF output files will be generated at group boundaries.
<i>*INDEX</i>	An index tag or bookmark will be placed at the group boundaries in a single output file. If *INDEX is requested, bookmarks will be labeled according to the group name on the DDS STRPAGGRP keyword. If characters in the group name are not available in the standard PDF encoding they will be presented as a space.

**Number of additional generate PDF output options** The number of additional options to generate PDF.

**Number of device resource library list entries.** The number of resource library list entries returned.

**Number of mail server names** The number of mail server names to use for electronically mailing the resulting PDF file from the IPDS to PDF transform.

**Number of PSF-defined options.** The number of PSF-defined options returned.

**Offset to additional generate PDF output options.** The number of bytes from the start of the PSFI0100 format to the beginning of the additional options to generate PDF.

**Offset to device resource library list entries.** The number of bytes from the start of the PSFI0100 format to the beginning of the first device resource library list entry.

**Offset to mail server names** The number of bytes from the start of the PDFI0100 format to the beginning of the first mail server name to use for electronically mailing the resulting PDF file from the IPDS to PDF transform.

**Offset to PSF-defined options.** The number of bytes from the start of the PSFI0100 format to the beginning of the first PSF-defined option.

**Output queue.** The name of the output queue that should be used when spooling PDF output. This field is used only when the generate PDF output field has a value of \*SPLF; otherwise, it is set to blanks.

Possible values are:

*NONE	No output queue is specified.
<i>output-queue-name</i>	The name of the output queue.

**Output queue library.** The name of the library in which the output queue is located. This field is used only when the generate PDF output field has a value of \*SPLF; otherwise, it is set to blanks.

**Page size control.** Whether the page size (forms) in the printer is set by PSF/400.

The possible values are:

*NO	The page size (forms) in the printer is not set by PSF/400.
*YES	The page size (forms) in the printer is set by PSF/400.

**Paper size for drawer 1.** The paper size to use for drawer 1 during the IPDS to PDF transform process. This field is used only when the generate PDF output field has a value other than \*NONE; otherwise, it is set to blanks.

Possible values are:

*LETTER	Use the dimensions of letter paper.
*LEGAL	Use the dimensions of legal paper.
*STATEMENT	Use the dimensions of statement paper.
*EXECUTIVE	Use the dimensions of executive paper.
*LEDGER	Use the dimensions of ledger paper.
*A5	Use the dimensions of A5 paper.
*A4	Use the dimensions of A4 paper.
*A3	Use the dimensions of A3 paper.
*B5	Use the dimensions of B5 paper.
*B4	Use the dimensions of B5 paper.

**Paper size for drawer 2.** The paper size to use for drawer 2 during the IPDS to PDF transform process. This field is used only when the generate PDF output field has a value other than \*NONE; otherwise, it is set to blanks.

Possible values are:

*LETTER	Use the dimensions of letter paper.
*LEGAL	Use the dimensions of legal paper.
*STATEMENT	Use the dimensions of statement paper.
*EXECUTIVE	Use the dimensions of executive paper.
*LEDGER	Use the dimensions of ledger paper.
*A5	Use the dimensions of A5 paper.
*A4	Use the dimensions of A4 paper.
*A3	Use the dimensions of A3 paper.
*B5	Use the dimensions of B5 paper.
*B4	Use the dimensions of B5 paper.

**PDF administrator.** The e-mail address of an administrator that will receive a notification when one or more portions of a document can not be processed correctly.

Possible values are:

*\*NONE* No PDF administrator e-mail address is specified.  
*PDF administrator* The e-mail address of the PDF administrator who will receive a notification when one or more portions of a document can not be processed correctly.

**PDF fonts inline.** Specifies whether the PDF output generated by the IPDS to PDF transform should carry the necessary fonts inline.

*\*YES* Specifies that the fonts should be carried inline with the PDF output.  
*\*NO* Specifies that the fonts should not be carried inline with the PDF output.

**Printer response timer.** The length of time for which a writer should wait for a response from a printer device.

The possible values are:

*5 to 3600* The amount of time, in seconds, to wait.  
*-1* There is no limit on the length of time a writer will wait for a response from the printer. (\*NOMAX)

**PSF configuration object library.** The library from which the PSF/400 configuration object was retrieved.

**PSF configuration object name.** The name of the PSF/400 configuration object retrieved.

**PSF-defined option.** A value as defined by IBM. One or more values may be made available between releases of i5/OS (OS/400®). If a value is made available, a PTF cover letter will contain the required syntax.

The following special value may be returned:

*\*NONE* No PSF-defined options are specified.

**Release timer.** The amount of time to wait after the last page of the last ready spooled file has printed, before releasing the printer.

The possible values are:

*1 to 1440* The amount of time, in minutes, to wait.  
*0* The printer is not released unless ENDWTR is executed (\*NOMAX).  
*-1* PSF/400 waits 15 seconds before releasing the printer (\*SEC15).  
*-2* PSF/400 waits 30 seconds before releasing the printer (\*SEC30).

**Reserved.** An ignored field.

**Resource retention.** Whether PSF/400 should use resource retention across spooled files.

The possible values are:

*\*YES* PSF/400 stores page segments and overlays in the printer across spooled file boundaries.  
*\*NO* PSF/400 does not store page segments and overlays in the printer across spooled file boundaries.

**Restart timer.** The amount of time to wait before the printer writer attempts to reestablish either a session or dialog.

The possible values are:

*1 to 1440*      The number of minutes to wait.  
*0*                The printer writer attempts to reestablish the session or dialog as soon as a spooled file has a status of RDY (\*IMMED).

**Save AFP data.** The activation of the save AFP data capability.

Possible values are:

\*NO              No AFP Respool allowed.  
\*YES             AFP Respool allowed.

**Sender of electronic mail.** The owner or sender from whom the PDF output file is electronically mailed. This field is used only when the generate PDF output field has a value of \*MAIL; otherwise, it is set to blanks.

The possible values are:

\*SPLFOWN        The owner of the original spooled file should be the sender of the electronically mailed PDF file.  
*mail-file-sender*      The name of a valid user profile to be used as the sender of the electronically mailed PDF file.

**Text.** The text that briefly describes the object.

**Use DBCS simulation fonts.** Whether to use outline DBCS simulation fonts.

The possible values are:

\*NO              No mapping of coded fonts will be done.  
\*YES             All coded font references will be mapped to outline coded font equivalents.

**User resource library.** The user resource library list to be used for searching AFP resources. The user resource library list is searched first and then the device resource library list is searched when attempting to find an AFP resource specified with the spooled file.

The possible values are:

\*JOBLIBL        The job's current library list that created the spooled file is used in searching for AFP resources.  
\*PRTF            The list supplied in the printer file will be used in searching for AFP resources.  
\*CURLIB         The current library for the job that created the spooled file is used in searching for AFP resources.  
\*NONE

No user resource library list is used in searching for AFP resources.

## CFNI0100 Format

The following table shows the information returned for the CFNI0100 format.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Bytes returned
4	4	BINARY(4)	Bytes available
8	8	CHAR(10)	Coded font name
18	12	CHAR(10)	Coded font library name
28	1C	CHAR(50)	Text
78	4E	CHAR(2)	Reserved
80	50	BINARY(4)	Offset to the font character set and code page pairs contained in the coded font
84	54	BINARY(4)	Number of font character set and code page pairs contained in the coded font
88	58	BINARY(4)	Length of a font character set and code page pair
These fields are repeated for each font character set and code page pair contained in the coded font.		CHAR(10)	Font character set
		CHAR(10)	Host code page
		CHAR(10)	Resident font
		CHAR(2)	Reserved

## FCSI0100 Format

The following table shows the information returned for the FCSI0100 format.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Bytes returned
4	4	BINARY(4)	Bytes available
8	8	CHAR(10)	Font character set
18	12	CHAR(2)	Reserved
20	14	BINARY(4)	Offset to the coded font entries
24	18	BINARY(4)	Number of coded fonts
28	1C	BINARY(4)	Length of data returned for each coded font
These fields are repeated for each coded font that contains the font character set specified with the font character set parameter.		CHAR(10)	Coded font name
		CHAR(10)	Coded font library name
		CHAR(50)	Text
		CHAR(2)	Reserved

## Field Descriptions

**Bytes available.** The length of all data available to return. All available data is returned if enough space is provided.

**Bytes returned.** The length of the data actually returned.

**Coded font library name.** The library of the coded font being retrieved.

**Coded font name.** The name of the coded font being retrieved.

**Font character set.** The name of a font character set that is contained in the coded font.

**Host code page.** The name of the host resident code page font resource that is contained in the coded font.

**Length of data returned for each coded font.** The length, in bytes, of the data returned for each coded font that contains the font character set specified with the font character set parameter.

**Length of a font character set and code page pair.** The length, in bytes, of the data returned for each font character set and code page pair.

**Number of coded fonts.** The number of coded fonts that contain the font character set specified with the font character set parameter.

**Number of font character set and code page pairs in the coded font.** The number of font character set and code page pairs that are contained in the coded font.

**Offset to the coded font entries.** The number of bytes from the start of the FCSI0100 format to the beginning of the first coded font entry.

**Offset to the font character set and code page pairs in the coded font.** The number of bytes from the start of the CFNI0100 format to the beginning of the first font character set and code page pair.

**Reserved.** An ignored field.

**Resident font.** Whether the font character set and code page pair is resident in the printer.

The possible values are:

\*YES        The font character set and code page pair is resident in the printer and does not need to be downloaded by the system to the printer.

\*NO         The font character set and code page pair is not resident in the printer and needs to be downloaded by the system to the printer.

**Text.** The text that briefly describes the object.

## FTBI0100 Format

The following table shows the information returned for the FTBI0100 format.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Bytes returned
4	4	BINARY(4)	Bytes available
8	8	CHAR(10)	Font table name
18	12	CHAR(10)	Font table library
28	1C	CHAR(50)	Text
78	4E	CHAR(10)	Font table type
88	58	BINARY(4)	Number of entries in font table
92	5C	BINARY(4)	Maximum length of each font table entry

## PHCS0100 Format

The following table shows the information returned for the PHCS0100 format.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Bytes returned
4	4	BINARY(4)	Bytes available
8	8	CHAR(10)	Font table name
18	12	CHAR(10)	Font table library
28	1C	CHAR(50)	Text
78	4E	CHAR(2)	Reserved
80	50	BINARY(4)	Offset to first entry in font table
84	54	BINARY(4)	Number of entries in font table
88	58	BINARY(4)	Length of font table entry
These fields are repeated for each entry in the font table.		BINARY(4)	Printer font identifier
		CHAR(10)	Width
		CHAR(12)	Attribute
		CHAR(10)	Graphic character set
		CHAR(10)	Point size
		CHAR(10)	Host font character
		CHAR(10)	Host font character (1)
		CHAR(10)	Host font character (2)
		CHAR(10)	Type
CHAR(2)	Reserved		

## HPCS0100 Format

The following table shows the information returned for the HPCS0100 format.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Bytes returned
4	4	BINARY(4)	Bytes available
8	8	CHAR(10)	Font table name
18	12	CHAR(10)	Font table library
28	1C	CHAR(50)	Text
78	4E	CHAR(2)	Reserved
80	50	BINARY(4)	Offset to first entry in font table
84	54	BINARY(4)	Number of entries in font table
88	58	BINARY(4)	Length of font table entry

Offset		Type	Field
Dec	Hex		
These fields are repeated for each entry in the font table.		CHAR(10)	Host font character
		CHAR(10)	Type
		BINARY(4)	Printer font identifier
		CHAR(10)	Width
		CHAR(12)	Attribute
		CHAR(10)	Graphic character set
		CHAR(10)	Point size
		CHAR(2)	Reserved

## PHCP0100 Format

The following table shows the information returned for the PHCP0100 format.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Bytes returned
4	4	BINARY(4)	Bytes available
8	8	CHAR(10)	Font table name
18	12	CHAR(10)	Font table library
28	1C	CHAR(50)	Text
78	4E	CHAR(2)	Reserved
80	50	BINARY(4)	Offset to first entry in font table
84	54	BINARY(4)	Number of entries in font table
88	58	BINARY(4)	Length of font table entry
These fields are repeated for each entry in the font table		CHAR(10)	Graphic character set
		CHAR(5)	Code page
		CHAR(10)	Host code page
		CHAR(10)	Host code page (1)
		CHAR(10)	Host code page (2)
		CHAR(3)	Reserved

## HPCP0100 Format

The following table shows the information returned for the HPCP0100 format.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Bytes returned
4	4	BINARY(4)	Bytes available
8	8	CHAR(10)	Font table name
18	12	CHAR(10)	Font table library

Offset		Type	Field
Dec	Hex		
28	1C	CHAR(50)	Text
78	4E	CHAR(2)	Reserved
80	50	BINARY(4)	Offset to first entry in font table
84	54	BINARY(4)	Number of entries in font table
88	58	BINARY(4)	Length of font table entry
These fields are repeated for each entry in the font table.		CHAR(10)	Host code page
		CHAR(10)	Graphic character set
		CHAR(5)	Code page
		CHAR(3)	Reserved

## PPCS0100 Format

The following table shows the information returned for the PPCS0100 format.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Bytes returned
4	4	BINARY(4)	Bytes available
8	8	CHAR(10)	Font table name
18	12	CHAR(10)	Font table library
28	1C	CHAR(50)	Text
78	4E	CHAR(2)	Reserved
80	50	BINARY(4)	Offset to first entry in font table
84	54	BINARY(4)	Number of entries in font table
88	58	BINARY(4)	Length of font table entry
These fields are repeated for each entry in the font table.		BINARY(4)	From printer font ID
		CHAR(10)	From point size
		CHAR(2)	Reserved
		BINARY(4)	To printer font ID
		CHAR(10)	To Point Size
		CHAR(2)	Reserved

## Field Descriptions

**Attribute.** The attributes of the printer-resident font.

The possible attributes are:

- \**NONE*                This printer-resident font has no special font attributes.
- \**BOLD*                The printer-resident font is a bold font.
- \**ITALIC*              The printer-resident font is an italic font.
- \**BOLDITC*            The printer-resident font is a bold italic font.
- \**DBLWIDE*            The printer-resident font is a double-wide font.

*\*ITCDBLWIDE* The printer-resident font is an italic double-wide font.  
*\*BOLDDBLWIDE* The printer-resident font is a bold double-wide font.

**Bytes available.** The length of all data available to return. All available data is returned if enough space is provided.

**Bytes returned.** The length of the data actually returned.

**Code page.** The printer-resident code page for which substitution is performed.

**Font table library.** The name of the library in which the font table is found. This value is blank when the font table name specified in the qualified AFP object is *\*SYSPHFCS*, *\*SYSHPFCS*, *\*SYSPHCP* or *\*SYSHPCP*.

**Font table name.** The name of the font table retrieved.

When the qualified AFP object is *\*SYSPHFCS*, the font table name is *\*SYSPHFCS*.

When the qualified AFP object is *\*SYSPHCP*, the font table name is *\*SYSPHCP*.

When the qualified AFP object is *\*SYSHPFCS*, the font table name is *\*SYSHPFCS*.

When the qualified AFP object is *\*SYSHPCP*, the font table name is *\*SYSHPCP*.

**Font table type.** The type of font table.

The possible types are:

<i>*PHFCS</i>	The font table is a printer-resident to host-resident font character set table.
<i>*SYSPHFCS</i>	The font table is the system printer-resident to host-resident font character set table.
<i>*PHCP</i>	The font table is a printer-resident to host-resident code page mapping table.
<i>*SYSPHCP</i>	The font table is the system printer-resident to host-resident code page mapping table.
<i>*HPFCS</i>	The font table is a host-resident to printer-resident font character set table.
<i>*SYSHPFCS</i>	The font table is the system host-resident to printer-resident font character set table.
<i>*</i>	The font table is a host-resident to printer-resident code page mapping table.
<i>*SYSHPCP</i>	The font table is the system host-resident to printer-resident code page mapping table.
<i>*PPFCS</i>	The font table is a printer-resident to printer-resident font substitution table.

**From point size.** The point size of the printer-resident font for which substitution is performed.

The possible values are:

<i>*WIDTH</i>	The width of the printer-resident font is used to compute the point size.
<i>*NONE</i>	No point size is specified for this font identifier.
<i>*ALL</i>	All point sizes for a scalable font are mapped. If the font is not scalable, then this is treated the same as <i>*NONE</i> .
<i>0.1 to 999.9.</i>	Numeric value of the point size. This value is right-justified and padded with leading 0's.

**From printer font ID.** The printer font identifier for which substitution is performed.

**Graphic character set.** The graphic character set for the font identifier.

The possible values are:

<i>*SYSVAL</i>	The graphic character set value specified in the system value <i>QCHRID</i> is used.
<i>graphic-character-set</i>	The graphic character set value specified for the font identifier.

**Host code page.** The host code page specified for substitution. This value is blank if the qualified AFP object name is \*SYSPHCP.

**Host code page (1).** The first host code page selected for font substitution. If this host code page is not available, PSF/400 checks for the existence of host code page (2). This value is blank if the qualified AFP object name is not \*SYSPHCP.

**Host code page (2).** The second host code page selected for font substitution if the first font code page is not available. A blank entry means that a second host code page is not available. This value is also blank if the qualified AFP object name is not \*SYSPHCP.

**Host font character.** The host font character set specified for font substitution. This value is blank if the qualified AFP object name is \*SYSPHFCS.

**Host font character (1).** The first host font character set selected for font substitution. If this host font character set is not available, PSF/400 checks for the existence of host font character (2). This value is blank if the qualified AFP object name is not \*SYSPHFCS.

**Host font character (2).** The second host font character set selected for font substitution if the first font character set is not available. A blank entry means that a second host font character set is not available. This value is also blank if the qualified AFP object name is not \*SYSPHFCS.

**Length of font table entry.** Length, in bytes, of each font table entry.

**Maximum length of each font table entry.** Maximum length, in bytes, of each font table entry.

**Number of entries in font table.** The number of font table entries returned.

**Offset to first entry in font table.** The number of bytes from the start of the PHCS0100, HPSC0100, PHCP0100, HPCP0100, or PPCS0100 format to the beginning of the first font table entry.

**Point size.** The point size of the printer-resident font.

The possible values are:

<i>*WIDTH</i>	The width of the printer-resident font is used to compute the point size.
<i>*NONE</i>	No point size is specified for this font identifier.
<i>0.1 to 999.9.</i>	Numeric value of the point size. This value is right-justified and padded with leading 0's.

**Printer font identifier.** The printer font identifier for which substitution is performed.

The possible values are:

<i>0</i>	The mapping of a host-resident to printer-resident font is disabled. (*NONE)
<i>1 to 65535</i>	The printer font identifier.

**Reserved.** An ignored field.

**Text.** The text that briefly describes the object.

**To point size.** The point size of the printer resident font identifier.

The possible values are:

\*NONE: No font point size is specified. This should be specified for all non-scalable fonts.  
 \*ALL All point sizes for a scalable font are mapped. If the font is not scalable, then this is treated the same as \*NONE.  
 0.1 to 999.9. Numeric value of the point size. This value is right-justified and padded with leading 0's.

**To printer font ID.** The font identifier of the printer resident font.

**Type.** The type of host font specified.

\*OUTLINE The host-resident font is an outline font.  
 \*RASTER The host-resident font is a raster font.

**Width.** The width of the printer-resident font.

The possible values are:

\*PTSIZE The width for this font identifier is calculated from the point size.  
 \*NONE No width is specified for this font identifier. \*NONE should be specified when mapping an outline font.  
 1 to 32767 The font width for the font identifier. See the Printing topic for information on font widths for printer-resident fonts.

## Error Messages

Message ID	Error Message Text
CPF3700 E	
CPF0001 E	Error found on &1 command.
CPD0172 D	Parameters passed on CALL do not match those required.
CPF2150 E	Object information function failed.
CPF2151 E	Operation failed for &2 in &1 type *&3.
CPF24B4 E	Severe error while addressing parameter list.
CPF3C17 E	Error occurred with input data parameter.
CPF3C19 E	Error occurred with receiver variable specified.
CPF3C1E E	Required parameter &1 omitted.
CPF3C20 E	Error found by program &1.
CPF3C21 E	Format name &1 is not valid.
CPF3C24 E	Length of the receiver variable is not valid.
CPF3C29 E	Object name &1 is not valid.
CPF3C36 E	Number of parameters, &1, entered for this API was not valid.
CPF3C3C E	Value for parameter &1 not valid.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPF8100 E	All CPF81xx messages could be returned. xx is from 01 to FF.
CPF9801 E	Object &2 in library &3 not found.
CPF9802 E	Not authorized to object &2 in &3.
CPF9803 E	Cannot allocate object &2 in library &3.
CPF9808 E	Cannot allocate one or more libraries on library list.
CPF9810 E	Library &1 not found.
CPF9820 E	Not authorized to use library &1.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.
CPF9999 E	Function check. &1 unmonitored by &2 at statement &5, instruction &3.
PQT0107 E	No coded fonts found containing specified font character set.

API introduced: V5R1

## Retrieve Output Queue Information (QSPROUTQ) API

Required Parameter Group:

1	Receiver variable	Output	Char(*)
2	Length of receiver variable	Input	Binary(4)
3	Format name	Input	Char(8)
4	Qualified output queue name	Input	Char(20)
5	Error code	I/O	Char(*)

Default Public Authority: \*USE

Threadsafe: No

The Retrieve Output Queue Information (QSPROUTQ) API returns information about the parameters used to create the queue, the current status of the queue, and the number of entries on the queue.

### Authorities and Locks

#### *Output Queue Library Authority*

The caller needs \*EXECUTE authority to the output queue library.

#### *Output Queue Authority*

The caller needs one of the following:

- \*READ authority to the output queue.
- Job control special authority (\*JOBCTL) if the output queue is operator controlled (\*OPRCTL(\*YES)).
- Spool control special authority (\*SPLCTL).

#### *Output Queue Lock*

This API needs an \*EXCLRD lock on the output queue.

### Required Parameter Group

#### **Receiver variable**

OUTPUT; CHAR(\*)

The variable that receives the information requested.

#### **Length of receiver variable**

INPUT; BINARY(4)

The length of the receiver variable provided by the receiver variable parameter. The amount of data returned is truncated if the receiver variable is too small. A length of less than 8 is not valid. If the length specified is greater than the actual length, the results may be unpredictable.

#### **Format name**

INPUT; CHAR(8)

The content and format of the queue information being returned.

You must specify one of the following:

*"OUTQ0100 Format" on page 153* Contains the basic information about the output queue and only one of the writers that is started to the queue.

*"OUTQ0200 Format" on page 154* Contains the basic information about the output queue and all of the writers that are started to the queue.

### Qualified output queue name

INPUT; CHAR(20)

The name of the output queue for which information is returned. The first 10 characters contain the queue name and the second 10 characters contain the name of the library in which the queue resides.

The following special values are supported for the library name:

- \*LIBL           The library list used to locate the queue.
- \*CURLIB        The current library for the job is used to locate the queue.

### Error code

I/O; CHAR(\*)

The structure in which to return error information. For the format of the structure, see Error code parameter.

## OUTQ0100 Format

The following table shows the information returned for the OUTQ0100 format. For more details about the fields in the following table, see "Field Descriptions" on page 157.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Bytes returned
4	4	BINARY(4)	Bytes available
8	8	CHAR(10)	Output queue name
18	12	CHAR(10)	Output queue library name
28	1C	CHAR(10)	Order of files on queue
38	26	CHAR(10)	Display any file
48	30	BINARY(4)	Job separators
52	34	CHAR(10)	Operator controlled
62	3E	CHAR(10)	Data queue name
72	48	CHAR(10)	Data queue library name
82	52	CHAR(10)	Authority to check
92	5C	BINARY(4)	Number of files
96	60	CHAR(10)	Output queue status
106	6A	CHAR(10)	Writer job name
116	74	CHAR(10)	Writer job user name
126	7E	CHAR(6)	Writer job number
132	84	CHAR(10)	Writer job status
142	8E	CHAR(10)	Printer device name
152	98	CHAR(50)	Text description
202	CA	CHAR(2)	Reserved
204	CC	BINARY(4)	Number of spooled file pages specified
208	D0	BINARY(4)	Number of writers started to queue
212	D4	BINARY(4)	Writers to autostart
216	D8	CHAR(1)	Remote system name type

Offset		Type	Field
Dec	Hex		
217	D9	CHAR(255)	Remote system name
472	1D8	CHAR(128)	Remote printer queue
600	258	CHAR(10)	Message queue name
610	262	CHAR(10)	Message queue library name
620	26C	BINARY(4)	Connection type
624	270	BINARY(4)	Destination type
628	274	CHAR(1)	VM/MVS™ class
629	275	CHAR(8)	Forms control buffer
637	27D	CHAR(1)	Host print transform
638	27E	CHAR(17)	Manufacturer type and model
655	28F	CHAR(10)	Workstation customizing object name
665	299	CHAR(10)	Workstation customizing object library name
675	2A3	CHAR(1)	Spooled file auxiliary storage pool attribute
676	2A4	BINARY(4)	Offset to maximum spooled file page entries
680	2A8	BINARY(4)	Number of page entries returned
684	2AC	BINARY(4)	Length of each size entry
688	2B0	CHAR(128)	Destination options
816	330	CHAR(1)	Writer type started to queue
817	331	CHAR(1)	Print separator page
818	332	CHAR(255)	Long remote printer queue
1073	431	CHAR(10)	Image configuration
1083	43B	CHAR(10)	Image configuration library
1093	445	CHAR(1)	Network directory publishing status
1094	446	CHAR(2)	Reserved
1096	448	BINARY(4)	Spooled file auxiliary storage pool ID
1100	44C	CHAR(10)	Spooled file auxiliary storage pool device name
		CHAR(*)	Spooled file maximum page entries

## OUTQ0200 Format

The following table shows the information returned for the OUTQ0200 format. For more details about the fields in the following table, see “Field Descriptions” on page 157.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Bytes returned
4	4	BINARY(4)	Bytes available
8	8	CHAR(10)	Output queue name
18	12	CHAR(10)	Output queue library name
28	1C	CHAR(10)	Order of files on queue
38	26	CHAR(10)	Display any file

Offset		Type	Field
Dec	Hex		
48	30	BINARY(4)	Job separators
52	34	CHAR(10)	Operator controlled
62	3E	CHAR(10)	Data queue name
72	48	CHAR(10)	Data queue library name
82	52	CHAR(10)	Authority to check
92	5C	BINARY(4)	Number of files
96	60	CHAR(10)	Output queue status
106	6A	CHAR(50)	Text description
156	9C	BINARY(4)	Number of spooled file pages specified
160	A0	BINARY(4)	Number of writers started to queue
164	A4	BINARY(4)	Writers to autostart
168	A8	CHAR(1)	Remote system name type
169	A9	CHAR(255)	Remote system name
424	1A8	CHAR(128)	Remote printer queue
552	228	CHAR(10)	Message queue name
562	232	CHAR(10)	Message queue library name
572	23C	BINARY(4)	Connection type
576	240	BINARY(4)	Destination type
580	244	CHAR(1)	VM/MVS class
581	245	CHAR(8)	Forms control buffer
589	24D	CHAR(1)	Host print transform
590	24E	CHAR(17)	Manufacturer type and model
607	25F	CHAR(10)	Workstation customizing object name
617	269	CHAR(10)	Workstation customizing object library name
627	273	CHAR(1)	Spooled file auxiliary storage pool attribute
628	274	BINARY(4)	Offset to maximum spooled file page entries
632	278	BINARY(4)	Number of page entries returned
636	27C	BINARY(4)	Length of each size entry
640	280	BINARY(4)	Offset to writer entries
644	284	BINARY(4)	Number of writer entries returned
648	288	BINARY(4)	Length of each writer entry
652	28C	CHAR(128)	Destination options
780	30C	CHAR(1)	Writer type started to queue
781	30D	CHAR(1)	Print separator page
782	30E	CHAR(255)	Long remote printer queue
1037	40D	CHAR(3)	Reserved
1040	410	BINARY(4)	Offset to user-defined options
1044	414	BINARY(4)	Number of user-defined options returned
1048	418	BINARY(4)	Length of each user-defined option entry
1052	41C	BINARY(4)	Offset to user-defined data

Offset		Type	Field
Dec	Hex		
1056	420	BINARY(4)	Bytes of user-defined data returned
1060	424	BINARY(4)	Bytes of user-defined data available
1064	428	CHAR(10)	User-defined object name
1074	432	CHAR(10)	User-defined object library name
1084	43C	CHAR(10)	User object type
1094	446	CHAR(10)	User driver program name
1104	450	CHAR(10)	User driver program library name
1114	45A	CHAR(10)	Data transform program name
1124	464	CHAR(10)	Data transform program library name
1134	46E	CHAR(10)	Image configuration
1144	478	CHAR(10)	Image configuration library
1154	482	CHAR(1)	Network directory publishing status
1155	483	CHAR(1)	Reserved
1156	484	BINARY(4)	Spooled file auxiliary storage pool ID
1160	488	CHAR(10)	Spooled file auxiliary storage pool device name
		CHAR(*)	Spooled file maximum page entries
		CHAR(*)	Writer entries
		CHAR(*)	User-defined option entries
		CHAR(*)	User-defined data

## Format of Spooled File Maximum Page Entry

Each spooled file size entry has the following structure. For more details about the fields in the following table, see “Field Descriptions” on page 157.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Number of pages
4	4	CHAR(8)	Starting time
12	C	CHAR(8)	Ending time

## Format of Writer Entry

Each writer entry from format OUTQ0200 has the following structure. For more details about the fields in the following table, see “Field Descriptions” on page 157.

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	Writer job name
10	A	CHAR(10)	Writer job user name
20	14	CHAR(6)	Writer job number

Offset		Type	Field
Dec	Hex		
26	1A	CHAR(10)	Writer job status
36	24	CHAR(10)	Printer device name

## Format of User-Defined Option Entry

Each user-defined option entry has the following structure. For more details about the fields in the following table, see "Field Descriptions."

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	User-defined option

## Field Descriptions

**Authority to check.** Indicates what type of authorities to the output queue allow the user to control all the files on the queue.

The possible values are:

- \*OWNER Only the owner of the output queue can control all the output files on the queue.
- \*DTAAUT Any user with \*READ, \*ADD, or \*DELETE authority to the output queue can control all output files on the queue.

**Bytes available.** The number of bytes of data available to be returned. All available data is returned if enough space is provided.

**Bytes returned.** The number of bytes of data returned.

**Bytes of user-defined data available.** The number of bytes of user-defined data available to be returned. All available data is returned if enough space is provided.

**Bytes of user-defined data returned.** The number of bytes of user-defined data returned.

**Connection type.** The type of network connection to the remote system.

Values include:

- 0 The destination type is not specified when the remote system is \*NONE.
- 1 The SNADS network is used as the connectivity to the remote system.
- 2 The TCP/IP network is used as the connectivity to the remote system.
- 3 The IPX network is used as the connectivity to the remote system. ❧ This value is no longer supported. ❧
- 4 Reserved.
- 5 A user-defined connectivity is used as the connectivity to the remote system.

**Data queue library name.** The name of the library that contains the data queue.

**Data queue name.** Name of the data queue associated with this output queue. Whenever a spooled file goes into ready status on the output queue, an entry is placed on the data queue.

\*NONE No data queue is associated with this output queue.

**Data transform program library name.** The name of the library that contains the data transform program.

\*LIBL The library list used to locate the data transform program.

\*CURLIB The current library for the job is used to locate the data transform program.

**Data transform program name.** The name of the user-specified data transform program that is used by the driver program.

The following values are supported:

\*NONE No data transform program is specified.

*Data transform program name* The name of the data transform program.

**Destination options.** Destination-dependent options that are specific to a particular implementation of an LPR Print Server.

The possible values are:

\*NONE No destination-dependent options have been specified.

\*NOWAIT The remote writer will not wait for confirmation that the destination system has finished processing the spooled file. >> This value is used only when the connection type is \*SNA. <<

\*USRDFNTXT Use the value for the user-defined text of the user profile when the spooled file was created.

*destination options* The destination-dependent options specified on the output queue.

*blank* This field is blank when the connection type is 0 (\*NONE) or when the connection type is 1 (\*SNA).

**Destination type.** The type of destination system that spooled files on this output queue are being sent to.

The possible values are:

0 The destination type is not specified when the remote system is \*NONE.

1 The destination system is a System i™ product.

2 The destination system is a System i product running OS/400® version 2.

3 This destination system is a System/390® system.

4 The destination system is a PC using Print Services Facility/2.

5 Reserved.

6 The destination system is accessed with Novell NetWare 3.x. >> This value is no longer supported. <<

7 The destination system is accessed with Novell NetWare 4.x. >> This value is no longer supported. <<

-1 The destination system is not a system matching any of the other special values.

**Display any file.** Whether users who have authority to read this output queue can display the output data of any output file on this queue, or only the data in their own files.

\*YES Any user having authority to read the queue can display, copy, or send the data of any file on the queue.

\*NO Users authorized to use the queue can display, copy, or send the output data of their own files only, unless they have some special authority.

*\*OWNER* Only the owner of a file or a user with \*SPLCTL authority can display, copy, send, or move their own spooled files to another output queue.

**Ending time.** The ending time portion of the maximum spooled file page entry. » It specifies the time of day, in local job time, that the maximum spooled file page is to end. « The ending time is given in 24-hour format with a time separator (hh:mm:ss), where hh equals hours, mm equals minutes, and ss equals seconds.

**Forms control buffer.** The forms control buffer for files sent to a VM/MVS host system.

**Note:** This information is valid only when the connection type is 1 and the destination type is 3.

*\*NONE* No FCB is used when sending spooled files.  
*\*USRDATA* The first 8 characters of the user data (USRDATA) spooled file attribute determine the name of the FCB. If the user data is blank, no FCB is used.  
*\*PRTF* The first 8 characters of the printer file used to spool the file determine the name of the FCB.  
*FCB-name* The name of the FCB to be used.

**Host print transform.** Whether to use the host print transform function to transform a spooled file. This field was previously called transform SCS to ASCII.

Values are:

*blank* Used when the connection type is 1 (\*SNA) and the user driver program is \*NONE.  
*0* Do not transform data streams using host print transform.  
*1* Transform data streams using host print transform.

**Image configuration.** The transform services for a variety of image and print data-stream formats. (See the IMGCFG keyword in the *Communications Configuration* manual, SC41-5401. This manual is not included in this release of the i5/OS® Information Center. However, it is available from the IBM®

Publications Center  as a printed hardcopy that you can order or in an online format that you can download at no charge.)

*\*NONE* No image configuration is used when transforming the spooled file before sending.  
*Image-configuration-name* The name of the image configuration.

**Image configuration library.** The library in which the image configuration is located. This field is always set to blanks.

**Job separators.** The number of job separators to be placed at the beginning of the output for each job having spooled file entries on this output queue.

*0-9* The desired number of job separators.  
*-2* No job separators are used; instead a message is sent to the writer's message queue at the end of each job indicating that the output can be removed.

**Length of each size entry.** The length, in bytes, of each spooled file size entry.

**Length of each user-defined option entry.** The length, in bytes, of each user-defined option entry.

**Length of each writer entry.** The length, in bytes, of each writer entry.

**Length of user-defined data.** The length, in bytes, of the user-defined data.

**Long remote printer queue.** The name of the remote printer.

Possible values include:

<i>*USER</i>	The user profile that creates the spooled file will determine the user ID on the remote system that it is sent to.
<i>*SYSTEM</i>	The default system printer on the remote system will determine the printer queue.
<i>printer queue name</i>	The name of the printer queue to which the remote writer will send files on the remote system.
<i>blank</i>	The name of the printer queue will be blank for output queues that are not remote output queues.

**Manufacturer type and model.** The manufacturer, type, and model for a printer using the host print transform function.

<i>*IBM42011</i>	The IBM 4201-1 Proprinter printer is used.
<i>*WSCST</i>	The workstation customizing object is used.
<i>blank</i>	Used when the connection type is 1(*SNA), or when the connection type is 2(*IP) and the host print transform is 0.

**Message queue library name.** The library of the message queue.

**Message queue name.** The name of the message queue to which messages, created by the remote writer started to this output queue, are sent.

<i>message queue name</i>	The message queue to which messages are to be sent.
<i>blank</i>	The message queue is blank when the output queue is not a remote output queue.

**Network directory publishing status.** Whether the output queue is published in the network directory.

Possible values include:

<i>0</i>	Output queue is not published
<i>1</i>	Output queue is published

**Number of files.** The number of spooled files that exist on the output queue.

**Number of pages.** Only files with this number of pages or less will print between the starting and ending times.

**Number of spooled file pages specified.** In a spooled file, the maximum number of pages allowed on the output queue as determined by the output queue definition from the Create Output Queue (CRTOUTQ) command.

**Number of page entries returned.** The number of maximum spooled file page entries returned in this format. Only complete entries will be returned.

**Number of user-defined options returned.** The number of user-defined option entries returned in this format. Only complete entries are returned.

**Number of writer entries returned.** The number of writer entries returned in this format. Depending on the bytes available, this could be less than the number of writers started. Only complete entries will be returned.

**Number of writers started to queue.** The number of printer writers that have been started to this output queue.

**Offset to maximum spooled file page entries.** The offset to the maximum spooled file page entry. A maximum spooled file page entry specifies the requirements a spooled file must meet to be printed from the output queue. The maximum number of pages, the starting time, and the ending time define these requirements.

**Offset to user-defined data.** The offset to information about the user-defined data.

**Offset to user-defined options.** The offset to information about each user-defined option entry. Each entry contains the user-defined option.

**Offset to writer entries.** The offset to information about each writer that is started to the output queue. Each entry contains the writer job name, writer job user name, writer job number, writer job status, and printer device name.

**Operator controlled.** Whether users with job control authority are allowed to manage or control the files on this queue. Users have job control authority if SPCAUT(\*JOBCTL) is specified in their user profile.

The possible values are:

<i>*YES</i>	Users with job control authority can control the queue and make changes to the files on the queue.
<i>*NO</i>	This queue and its entries cannot be controlled or changed by users with job control authority unless they also have some other special authority.

**Order of files on queue.** The order of the spooled files on the output queue.

The possible values are:

<i>*FIFO</i>	The queue is first-in first-out for each file. That is, on the queue, new spooled files are placed after all other spooled files that have the same priority.
<i>*JOBNBR</i>	The queue entries for the spooled files are sorted in priority sequence using the job number (the date and time that the job entered the system) of the job that created the spooled file.

**Output queue library name.** The name of the library that contains the output queue.

**Output queue name.** The name of the output queue.

**Output queue status.** The status of the output queue.

The status may be one of the following values:

<i>RELEASED</i>	The queue is released.
<i>HELD</i>	The queue is held.

**Printer device name.** The name of the printer device.

For format OUTQ0100 the following applies:

- This field is blank when a writer job is not started to this queue.

- This field contains the first printer device when the number of writers started to this output queue is greater than 1.

**Print separator page.** Whether to print a separator page when printing the spooled file on the remote system.

Possible values include:

<i>blank</i>	The connection type is neither 2 (*IP) nor 5 (*USRDFN).
0	Do not print a separator page.
1	Print a separator page.

**Remote printer queue.** The name of the remote printer.

Possible values include:

<i>*USER</i>	The user profile that creates the spooled file will determine the user ID on the remote system that it is sent to.
<i>*SYSTEM</i>	The default system printer on the remote system will determine the printer queue.
<i>printer queue name</i>	The name of the printer queue to which the remote writer will send files on the remote system.
<i>blank</i>	The name of the printer queue will be blank for output queues that are not remote output queues. This field will also be blank if the name of the printer queue will not fit in 128 bytes.

**Remote system name.** The name of the remote system. The format of the name is dependent on the name type. When the remote system name type is 0 (\*NONE) or 1 (\*PASTHR), the remote system name is blank. When the name type is 2 (system name), the remote system name is up to 255 characters. When the name type is 3 (Internet address), the remote system name is a 15-character address in the form nnn.nnn.nnn.nnn, and the remaining characters are blank.

**Remote system name type.** The type of remote system name.

Valid values are:

0	There is no remote system name specified
1	The output is routed to the system from where the pass-through was issued (*PASTHR).
2	The name of the remote system.
3	Internet address

**Reserved.** An ignored field.

**Spooled file auxiliary storage pool attribute.** The auxiliary storage pool (ASP) where the spooled files are to reside.

The possible values are:

0	*SYSTEM: The spooled files reside in the system auxiliary storage pool.
1	*OUTQASP: The spooled files reside in the auxiliary storage pool in which the output queue resides.

**Spooled file auxiliary storage pool device name.** The name of the auxiliary storage pool (ASP) device where the spooled files reside.

The possible values are:

*\*SYSBAS*            The spooled files reside in the system ASP (ASP 1) or one of the defined basic user ASPs (ASPs 2-32).  
*auxiliary storage pool device name*    The name of the auxiliary storage pool device where the spooled files reside.

**Spooled file auxiliary storage pool ID.** The number of the auxiliary storage pool (ASP) where the spooled files reside.

The possible values are:

1                    \*SYSTEM: The spooled files reside in the system auxiliary storage pool.  
2-255                The number of the auxiliary storage pool where the spooled files reside.

**Spooled file maximum page entries.** One maximum page entry is returned for each spooled file maximum page entry defined in the output queue. The number of entries may vary from one output queue to another. See “Format of Spooled File Maximum Page Entry” on page 156 for the format of each entry.

**Starting time.** The starting time portion of the maximum spooled file page array. >> It specifies the time of day, in local job time, that the maximum spooled file page is to start. << The starting time is given in 24-hour format with a time separator (hh:mm:ss), where hh equals hours, mm equals minutes, and ss equals seconds.

**Text description.** Text that briefly describes the output queue.

*\*BLANK*            There is no text description of the job queue.

**User-defined data.** Data defined by the user to be used by user applications or user-specified programs that process spooled files.

**User-defined object library name.** The name of the library that contains the user-defined object.

*\*LIBL*              The library list used to locate the user-defined object.  
*\*CURLIB*            The current library for the job is used to locate the user-defined object.

**User-defined object name.** The name of the user-defined object that is used by user applications or user-specified programs that process spooled files.

The following values are supported:

*\*NONE*              No user-defined object is specified.  
*User-defined object name*    The name of the object defined by the user.

**User-defined option.** The option defined by the user to be used by user applications or user-specified programs that process spooled files. When the number of user-defined options returned is 0, there are no user-defined options.

**User-defined option entries.** One entry is returned for each user-defined option. The number of entries may vary from one output queue to another. See “Format of User-Defined Option Entry” on page 157 for the format of each entry.

**User driver program library name.** The name of the library that contains the user driver program.

*\*LIBL*                The library list used to locate the user driver program.  
*\*CURLIB*            The current library for the job is used to locate the user driver program.

**User driver program name.** The name of the user-specified driver program that is used to process the spooled files on the output queue.

The following values are supported:

*\*NONE*                No user driver program is specified.  
*User driver*            The name of the user-specified driver program.  
*program name*

**User object type.** The type of the user-defined object.

The following values are supported:

*User object type*    The user object type can be data area (\*DTAARA), data queue (\*DTAQ), file (\*FILE), PSF configuration object (\*PSFCFG), user index (\*USRIDX), user queue (\*USRQ), and user space (\*USRSPC).

**Note:** A blank value is returned when the name of the user-defined object is \*NONE.

**VM/MVS class.** The VM/MVS SYSOUT class for distributions sent to a VM host system or to a MVS host system.

Possible values are:

*A-Z, 0-9*            The distribution class.  
*blank*                Used when the connection type is 1(\*SNA).

**Workstation customizing object library name.** The library name for the workstation customizing object. This field is blank when the connection type is 1(\*SNA) and the user driver program is \*NONE.

**Workstation customizing object name.** The name of an object that consists of a table of attributes used to customize a given ASCII device.

Possible values are:

*\*NONE*                There is no workstation customizing object.  
*name*                 The name of the workstation customizing object, which has been created by the Create Work Station Customizing Object (CRTWSCST) command.  
*blank*                Used when the connection type is 1(\*SNA) and the user driver program is \*NONE.

**Writer entries.** One writer entry is returned for each writer that is started to the output queue. The number of writer entries may vary from one output queue to another. See "Format of Writer Entry" on page 156 for the format of each entry.

**Writer job name.** The name of the writer job.

The following applies:

- This field is blank when a writer job is not started to this queue.

- For OUTQ0100, this field contains the first writer job name when the number of writers started to this output queue is greater than 1.

**Writer job number.** The job number associated with the writer job.

The following applies:

- This field is blank when a writer job is not started to this queue.
- For OUTQ0100, this field contains the first writer job number when the number of writers started to this output queue is greater than 1.

**Writer job status.** The status of the writer job.

The following applies:

- This field is blank when a writer job is not started to this queue.
- For OUTQ0100, this field contains the first writer job status when the number of writers started to this output queue is greater than 1.

The status may be one of the following values.

<i>STR</i>	The writer job is started to the output queue.
<i>END</i>	The writer job is ended.
<i>JOBQ</i>	The writer job is on the job queue
<i>HLD</i>	The writer job is held.
<i>MSGW</i>	The writer job is waiting for a message.

**Writer job user name.** The name of the user who started the writer job.

The following applies:

- This field is blank when a writer job is not started to this queue.
- For OUTQ0100, this field contains the first writer job user name when the number of writers started to this output queue is greater than 1.

**Writers to autostart.** The number of remote printer writers to autostart to this output queue at system IPL.

Possible values are:

<i>0</i>	No writers are automatically started to this output queue.
<i>number of writers</i>	The number of writers automatically started to this output queue.

**Writer type started to queue.** The type of writer started to this output queue.

Possible values are:

<i>0</i>	Printer writer
<i>1</i>	Diskette writer
<i>2</i>	Remote writer

## Error Messages

Message ID	Error Message Text
CPF2207 E	Not authorized to use object &1 in library &3 type *&2.
CPF24B4 E	Severe error while addressing parameter list.
CPF3CF1 E	Error code parameter not valid.
CPF3C19 E	Error occurred with receiver variable specified.
CPF3C21 E	Format name &1 is not valid.
CPF3C24 E	Length of the receiver variable is not valid.
CPF3C90 E	Literal value cannot be changed.
CPF3330 E	Necessary resource not available.
CPF3357 E	Output queue &1 in library &2 not found.
CPF8122 E	&8 damage on library &4.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V2R2

[Top](#) | ["Print APIs," on page 1](#) | [APIs by category](#)

---

## Retrieve Printer Attributes (QGYRPRTA) API

Required Parameter Group:

1	Receiver variable	Output	Char(*)
2	Length of receiver variable	Input	Binary(4)
3	Format name	Input	Char(8)
4	Device name	Input	Char(10)
5	Error Code	I/O	Char(*)

Default Public Authority: \*USE  
Threadsafe: No

The Retrieve Printer Attributes (QGYRPRTA) API retrieves attributes about a specific printer device. The output returned varies depending on whether the writer is started or not.

## Authorities and Locks

None.

## Required Parameter Group

### Receiver variable

OUTPUT; CHAR(\*)

The receiver variable that receives the information requested. You can specify the size of the area to be smaller than the format requested as long as you specify the length parameter correctly. As a result, the API returns only the data that the area can hold.

### Length of receiver variable

INPUT; BINARY(4)

The length of the receiver variable provided. The length of receiver variable parameter may be specified up to the size of the receiver variable specified in the user program. If the length of receiver variable parameter specified is larger than the allocated size of the receiver variable specified in the user program, the results are not predictable. The minimum length is 8 bytes.

### Format name

INPUT; CHAR(8)

The format of the printer attributes being returned. The valid format is RPTA0100.

**Device name**

INPUT; CHAR(10)

The name of the printer device for which information should be retrieved.

**Error code**

I/O; CHAR(\*)

The structure in which to return error information. For the format of the structure, see Error code parameter.

**Format of Receiver Variable**

The following table describes the order and format of the data that is returned in the receiver variable.

**RPTA0100 Format**

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Bytes returned
4	4	BINARY(4)	Bytes available
8	8	CHAR(10)	Device name
18	12	CHAR(1)	Writer started
19	13	CHAR(1)	Writer status
20	14	BIN(4)	Device status
24	18	BIN(4)	Overall status
28	1C	CHAR(50)	Text description
78	4E	CHAR(2)	Reserved
80	50	CHAR(10)	Started by user
90	5A	CHAR(1)	Writing status
91	5B	CHAR(1)	Waiting for message status
92	5C	CHAR(1)	Held status
93	5D	CHAR(1)	End pending status
94	5E	CHAR(1)	Hold pending status
95	5F	CHAR(1)	Between files status
96	60	CHAR(1)	Between copies status
97	61	CHAR(1)	Waiting for data status
98	62	CHAR(1)	Waiting for device status
99	63	CHAR(1)	On job queue status
100	64	CHAR(4)	Reserved
104	68	CHAR(10)	Writer job name
114	72	CHAR(10)	Writer job user name
124	7C	CHAR(6)	Writer job number
130	82	CHAR(10)	Printer device type
140	8C	BINARY(4)	Number of separators
144	90	BINARY(4)	Drawer for separators

Offset		Type	Field
Dec	Hex		
148	94	CHAR(10)	Align forms
158	9E	CHAR(10)	Output queue name
168	A8	CHAR(10)	Output queue library name
178	B2	CHAR(1)	Output queue status
179	B3	CHAR(1)	Network directory publishing status
179	B3	CHAR(1)	Reserved
180	B4	CHAR(10)	Form type
190	BE	CHAR(10)	Message option
200	C8	CHAR(10)	Automatically end writer
210	D2	CHAR(10)	Allow direct printing
220	DC	CHAR(10)	Message queue name
230	E6	CHAR(10)	Message queue library name
240	F0	CHAR(2)	Reserved
242	F2	CHAR(10)	Changes take effect
252	FC	CHAR(10)	Next output queue name
262	106	CHAR(10)	Next output queue library name
272	110	CHAR(10)	Next form type
282	11A	CHAR(10)	Next message option
292	124	BINARY(4)	Next file separators
296	128	BINARY(4)	Next separator drawer
300	12C	CHAR(10)	Spooled file name
310	136	CHAR(10)	Job name
320	140	CHAR(10)	User name
330	14A	CHAR(6)	Job number
336	150	BINARY(4)	Spooled file number
340	154	BINARY(4)	Page being written
344	158	BINARY(4)	Total pages
348	15C	BINARY(4)	Copies left to produce
352	160	BINARY(4)	Total copies
356	164	CHAR(10)	Advanced Function Presentation™
366	16E	CHAR(4)	Message key
370	172	CHAR(11)	Reserved
381	17D	CHAR(8)	Job system name
389	185	CHAR(7)	Spooled file create date
396	18C	CHAR(6)	Spooled file create time

## Field Descriptions

**Advanced Function Presentation.** Whether the printer supports Advanced Function Presentation.

\*NO Advanced Function Presentation is not supported  
\*YES Advanced Function Presentation is supported

**Align forms.** The time at which the forms alignment message will be sent.

Possible values are:

\*WTR The writer determines when the message is sent.  
\*FILE Control of the page alignment is specified by each file.

**Allow direct printing.** Whether the printer writer allows the printer to be allocated to a job that prints directly to a printer.

\*NO Direct printing is not allowed  
\*YES Direct printing is allowed.

**Automatically end writer.** When to end the writer if it is to end automatically.

\*NORDYF When no files are ready to print on the output queue from which the writer is selecting files to be printed.  
\*FILEEND When the current spooled file has been printed.  
\*NO The writer will not end, but it will wait for more spooled files.

**Between copies status.** Whether the writer is between copies of a multiple copy spooled file. The possible values are Y (yes) or N (no).

**Between files status.** Whether the writer is between spooled files. The possible values are Y (yes) or N (no).

**Bytes available.** The number of bytes of data available to be returned. All available data is returned if enough space is provided.

**Bytes returned.** The number of bytes of data returned.

**Changes take effect.** The time at which the pending changes to the writer take effect.

Possible values are:

\*NORDYF When all the current eligible files are printed.  
\*FILEEND When the current spooled file is done printing.  
Blank No pending changes to the writer.

**Copies left to produce.** The number of copies left to be printed. This field is set to 0 when no file is printing.

**Device name.** The name of the printer device.

**Device status.** The status of the printer device.

Possible values are:

Status Numeric Code (decimal)	Overall Status
0	VARIED OFF
5	AS/36 DISABLED
10	VARY OFF PENDING
20	VARY ON PENDING
30	VARIED ON
40	CONNECT PENDING
50	SIGNON DISPLAY
60	ACTIVE
62	AS/36 ENABLED
63	ACTIVE READER
66	ACTIVE WRITER
70	HELD
75	POWERED OFF
80	RCYPND
90	RCYCNL
95	SYSTEM REQUEST
100	FAILED
103	FAILED READER
106	FAILED WRITER
110	DIAGNOSTIC MODE
111	DAMAGED
112	LOCKED
113	UNKNOWN

**Note:** Not all of the values apply to printer devices.

**Drawer for separators.** The drawer from which the job and file separator pages are to be taken.

Possible values are:

- 1 \*FILE: The separator page prints from the same drawer that the file is printed from. If you specify a drawer different from the file that contains colored or different type paper, the page separator is more identifiable.
- 2 \*DEV: The separator pages will print from the separator drawer specified in the printer device description.
- 1 The first drawer.
- 2 The second drawer.
- 3 The third drawer.

**End pending status.** Whether an End Writer (ENDWTR) command has been issued for this writer.

Possible values are:

- N* No ENDWTR command was issued.
- I* \*IMMED: The writer ends as soon as its output buffers are empty.
- C* \*CNTRLD: The writer ends after the current copy of the spooled file has been printed.
- P* \*PAGEEND: The writer ends at the end of the page.

**Form type.** The type of form being used to print the spooled file.

Possible values are:

- \*ALL The writer is started with the option to print all spooled files of any form type.

*\*FORMS* The writer is started with the option to print all the spooled files with the same form type before using a different form type.  
*\*STD* The writer is started with the option to print all the spooled files with a form type of *\*STD*.  
*form type name* The writer is started with the option to print all the spooled files with the form type you specified.

**Held status.** Whether the writer is held. The possible values are Y (yes) or N (no).

**Hold pending status.** Whether a Hold Writer (HLDWTR) command has been issued for this writer.

Possible values are:

*N* No HLDWTR command was issued.  
*I* *\*IMMED*: The writer is held as soon as its output buffers are empty.  
*C* *\*CNTRLD*: The writer is held after the current copy of the file has been printed.  
*P* *\*PAGEEND*: The writer is held at the end of the page.

**Job name.** The name of the job that owns the spooled file currently being processed by the writer. This field is blank when no spooled file is printing.

**Job number.** The number of the job that owns the spooled file currently being processed by the writer. This field is blank when no spooled file is printing.

**Job system name.** The name of the system where the job that created the spooled file ran. This field is blank when no spooled file is printing.

**Message key.** The key to the message that the writer is waiting for a reply. This field will be blank when the writer is not waiting for a reply to an inquiry message.

**Message option.** A message option for sending a message to the message queue when this form is finished.

Possible values are:

*\*MSG* A message is sent to the message queue.  
*\*NOMSG* No message is sent to the message queue.  
*\*INFOMSG* An informational message is sent to the message queue.  
*\*INQMSG* An inquiry message is sent to the message queue.

**Message queue library name.** The name of the library that contains the message queue.

**Message queue name.** The name of the message queue that this writer uses for operational messages.

**Network directory publishing status.** Whether the printer is published in the network directory.

Possible values are:

*0* Printer is not published  
*1* Printer is published

**Next file separators.** The next number of separator pages to be printed when the change to the writer takes place.

Possible values are:

-1 \*FILE: The number of separator pages is specified by each file.  
 -10 No pending change to the writer.  
*Number of separators* The number of separator pages to be printed.

**Next form type.** The name of the next form type to be printed.

Possible values are:

\*ALL The writer is changed with the option to print all spooled files of any form type.  
 \*FORMS The writer is changed with the option to print all the spooled files with the same form type before using a different form type.  
 \*STD The writer is changed with the option to print all the spooled files with a form type of \*STD.  
*Form type name* The writer is changed with the option to print all the spooled files with the form type name you specified.  
 Blank No change has been made to this writer.

**Next message option.** The message option for sending a message to the message queue when the next form type is finished.

Possible values are:

\*MSG A message is sent to the message queue.  
 \*NOMSG No message is sent to the message queue.  
 \*INFOMSG An informational message is sent to the message queue.  
 \*INQMSG An inquiry message is sent to the message queue.  
 Blank No change pending.

**Next output queue library name.** The name of the library that contains the next output queue. This field is blank if no changes have been made to the writer.

**Next output queue name.** The name of the next output queue to be processed. This field is blank if no changes have been made to the writer.

**Next separator drawer.** This value indicates the drawer from which to take the separator pages if there is a change to the writer.

Possible values are:

-1 \*FILE: Separator pages print from the same drawer that the spooled file prints from. If you specify a drawer different from the spooled file that contains colored or different type paper, the page separator is more identifiable.  
 -2 \*DEV: Separator pages print from the separator drawer specified in the printer device description.  
 -10 No pending change to the writer.  
 1 The first drawer.  
 2 The second drawer.  
 3 The third drawer.

**Number of separators.** The number of separator pages to be printed.

Possible values are:

-1 \*FILE: The number of separator pages is specified by each file.

*Number of separators*      The number of separator pages to be printed.

**On job queue status.** Whether the writer is on a job queue and, therefore, is not currently running. The possible values are Y (yes) or N (no).

**Output queue library name.** The name of the library that contains the output queue from which spooled files are selected for printing.

**Output queue name.** The name of the output queue from which spooled files are being selected for printing.

**Output queue status.** The status of the output queue from which spooled files are being selected for printing.

Possible values are:

*H*      The output queue is held.  
*R*      The output queue is released.

**Overall status.** The overall status of the logical printer.

Possible values are:

1      Unavailable  
2      Powered off or not yet available  
3      Stopped  
4      Message waiting  
5      Held  
6      Stop (pending)  
7      Hold (pending)  
8      Waiting for printer  
9      Waiting to start  
10     Printing  
11     Waiting for printer output  
12     Connect pending  
13     Powered off  
14     Unusable  
15     Being serviced  
999    Unknown

**Page being written.** The page number in the spooled file that is currently being processed by the writer. The page number shown may be lower or higher than the actual page number being printed because of buffering done by the system. This field is set to 0 when no spooled file is printing.

**Printer device type.** The type of the printer being used to print the spooled file.

Valid values are:

\*SCS                    SNA (Systems Network Architecture) character string  
\*IPDS™                Intelligent Printer Data Stream™

**Reserved.** An ignored field.

**Spooled file create date.** » The date, based on local job time, that the spooled file was created on the system. « The date is in the CYMMDD format as follows:

<i>C</i>	Century, where 0 indicates years 19xx and 1 indicates years 20xx.
<i>YY</i>	Year
<i>MM</i>	Month
<i>DD</i>	Day

This field is blank when no spooled file is printing.

**Spooled file create time** » The time, in local job time, that the spooled file was created on the system. « The time is in the HHMMSS format where:

<i>HH</i>	Hour
<i>MM</i>	Minutes
<i>SS</i>	Seconds

This field is blank when no spooled file is printing.

**Spooled file name.** The name of the spooled file currently being processed by the writer. This field is blank when no spooled file is printing.

**Spooled file number.** The number of the spooled file currently being processed by the writer. This field is set to 0 when no spooled file is printing.

**Started by user.** The name of the user that started the writer.

**Text description.** The text description of the printer device.

**Total copies.** The total number of copies to be printed.

**Total pages.** The total number of pages in the spooled file.

Possible values are:

<i>Number</i>	The number of pages in the spooled file.
0	No spooled file is printing.

**User name.** The name of the user who owns the spooled file currently being processed by the writer. This field is blank when no file is printing.

**Waiting for data status.** Whether the writer has written all the data currently in the spooled file and is waiting for more data.

Possible values are:

<i>N</i>	The writer is not waiting for more data.
<i>Y</i>	The writer has written all the data currently in the spooled file and is waiting for more data. This condition occurs when the writer is producing an open spooled file with SCHEDULE(*IMMED) specified.

**Waiting for device status.** Whether the writer is waiting to get the device from a job that is printing directly to the printer.

<i>N</i>	The writer is not waiting for the device.
----------	---

Y The writer is waiting for the device

**Waiting for message status.** Whether the writer is waiting for a reply to an inquiry message. The possible values are Y (yes) or N (no).

**Writer job name.** The job name of the printer writer.

**Writer job number.** The job number of the printer writer.

**Writer job user name.** The name of the system user.

**Writer started.** Whether or not a writer is started for this printer.

Possible values are:

0 No writer is started  
1 Writer is started

**Writer status.** The status of the writer for this printer.

Possible values are:

X'01' Started  
X'02' Ended  
X'03' On job queue  
X'04' Held  
X'05' Message waiting

**Writing status.** Whether the printer writer is in writing status.

Possible values are:

Y The writer is in writing status.  
N The writer is not in writing status.  
S The writer is writing the file separators.

## Error Messages

Message ID	Error Message Text
CPF24B4 E	Severe error while addressing parameter list.
CPF3C90 E	Literal value cannot be changed.
CPF3CF1 E	Error code parameter not valid.
CPF3313 E	Writer &1 not active nor on job queue.
CPF33C8 E	No writer information returned for printer &1.
CPF9800 E	All CPF98xx messages could be signaled. xx is from 01 to FF.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V3R6

[Top](#) | ["Print APIs," on page 1](#) | [APIs by category](#)

---

## Retrieve Writer Information (QSPRWTRI) API

Required Parameter Group:

1	Receiver variable	Output	Char(*)
2	Length of receiver variable	Input	Binary(4)
3	Format name	Input	Char(8)
4	Printer name	Input	Char(10)
5	Error Code	I/O	Char(*)

Optional Parameter:

6	Writer name	Input	Char(10)
---	-------------	-------	----------

Default Public Authority: \*USE

Threadsafe: No

The Retrieve Writer Information (QSPRWTRI) API retrieves writer information associated with the specified printer or writer. The information retrieved is similar to what can be seen when running the Work with Writer (WRKWTR) command for a particular writer.

Additionally, this API can retrieve information about pending changes to a particular writer. The fields that could contain possible pending changes are:

- Next file separators
- Next form type
- Next message option
- Next output queue library name
- Next output queue name
- Next separator drawer

Changes to these fields result when the Change Writer (CHGWTR) command has been used. The value specified in the Changes take effect field determines when the pending changes take place.

## Authorities and Locks

*Output Queue Lock*

\*EXCLRD

## Required Parameter Group

### Receiver variable

OUTPUT; CHAR(\*)

The receiver variable that receives the information requested. You can specify the size of the area to be smaller than the format requested as long as you specify the length parameter correctly. As a result, the API returns only the data the area can hold.

### Length of receiver variable

INPUT; BINARY(4)

The length of the receiver variable. If the length is larger than the size of the receiver variable, the results are not predictable. The minimum length is 8 bytes.

### Format name

INPUT; CHAR(8)

The content and format of the writer information to be returned.

The following format name can be used:

“WTRI0100 Format”                      Writer information

**Printer name**

INPUT; CHAR(10)

The name of the printer for which writer information is to be returned. The following special value is supported for this parameter:

\*WRITER                      Use the writer name specified on the writer name parameter to return writer information.

**Error code**

I/O; CHAR(\*)

The structure in which to return error information. For the format of the structure, see Error code parameter.

**Optional Parameter**

**Writer name**

INPUT; CHAR(10)

The name of the writer for which writer information is to be returned. The value of this parameter must be set to blanks if the printer name parameter is not set to \*WRITER.

**WTRI0100 Format**

The following information is returned for the WTRI0100 format. See “Field Descriptions” on page 179 for more information.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Bytes returned
4	4	BINARY(4)	Bytes available
8	8	CHAR(10)	Started by user
18	12	CHAR(1)	Writing status
19	13	CHAR(1)	Waiting for message status
20	14	CHAR(1)	Held status
21	15	CHAR(1)	End pending status
22	16	CHAR(1)	Hold pending status
23	17	CHAR(1)	Between files status
24	18	CHAR(1)	Between copies status
25	19	CHAR(1)	Waiting for data status
26	1A	CHAR(1)	Waiting for device status
27	1B	CHAR(1)	On job queue status
28	1C	CHAR(1)	Type of writer
29	1D	CHAR(3)	Reserved
32	20	CHAR(10)	Writer job name
42	2A	CHAR(10)	Writer job user name
52	34	CHAR(6)	Writer job number

Offset		Type	Field
Dec	Hex		
58	3A	CHAR(10)	Printer device type
68	44	BINARY(4)	Number of separators
72	48	BINARY(4)	Drawer for separators
76	4C	CHAR(10)	Align forms
86	56	CHAR(10)	Output queue name
96	60	CHAR(10)	Output queue library name
106	6A	CHAR(1)	Output queue status
107	6B	CHAR(1)	Reserved
108	6C	CHAR(10)	Form type
118	76	CHAR(10)	Message option
128	80	CHAR(10)	Automatically end writer
138	8A	CHAR(10)	Allow direct printing
148	94	CHAR(10)	Message queue name
158	9E	CHAR(10)	Message queue library name
168	A8	CHAR(2)	Reserved
170	AA	CHAR(10)	Changes take effect
180	B4	CHAR(10)	Next output queue name
190	BE	CHAR(10)	Next output queue library name
200	C8	CHAR(10)	Next form type
210	D2	CHAR(10)	Next message option
220	DC	BINARY(4)	Next file separators
224	E0	BINARY(4)	Next separator drawer
228	E4	CHAR(10)	Spooled file name
238	EE	CHAR(10)	Job name
248	F8	CHAR(10)	User name
258	102	CHAR(6)	Job number
264	108	BINARY(4)	Spooled file number
268	10C	BINARY(4)	Page being written
272	110	BINARY(4)	Total pages
276	114	BINARY(4)	Copies left to produce
280	118	BINARY(4)	Total copies
284	11C	CHAR(4)	Message key
288	120	CHAR(1)	Initialize printer
289	121	CHAR(10)	Printer device name
299	12B	CHAR(8)	Job system name
307	133	CHAR(7)	Spooled file create date
314	13A	CHAR(6)	Spooled file create time

## Field Descriptions

**Align forms.** The time at which the forms alignment message will be sent.

Possible values are:

<i>*WTR</i>	The writer determines when the message is sent.
<i>*FILE</i>	Control of the page alignment is specified by each file.
<i>*FIRST</i>	The forms alignment message is only issued for the first spooled file printed.

**Allow direct printing.** Whether the printer writer allows the printer to be allocated to a job that prints directly to a printer. This field is set to blanks for remote writers.

<i>*NO</i>	Direct printing is not allowed
<i>*YES</i>	Direct printing is allowed.

**Automatically end writer.** When to end the writer if it is to end automatically.

<i>*NORDYF</i>	When no files are ready to print on the output queue from which the writer is selecting files to be printed.
<i>*FILEEND</i>	When the current spooled file has been printed.
<i>*NO</i>	The writer will not end, but it will wait for more spooled files.

**Between copies status.** Whether the writer is between copies of a multiple copy spooled file. This field is set to blanks for remote writers. The possible values are Y (yes) or N (no).

**Between files status.** Whether the writer is between spooled files. The possible values are Y (yes) or N (no).

**Bytes available.** Total format data length.

**Bytes returned.** The length of the data returned.

**Changes take effect.** The time at which the pending changes to the writer take effect.

Possible values are:

<i>*NORDYF</i>	When all the current eligible files are printed.
<i>*FILEEND</i>	When the current spooled file is done printing.
<i>blank</i>	No pending changes to the writer.

**Copies left to produce.** The number of copies left to be printed. This field is set to 0 when no file is printing.

**Drawer for separators.** Identifies the drawer from which the job and file separator pages are to be taken.

Possible values are:

-1	<i>*FILE</i> : The separator page prints from the same drawer that the file is printed from. If you specify a drawer different from the file that contains colored or different type paper, the page separator is more identifiable.
-2	<i>*DEV</i> : The separator pages will print from the separator drawer specified in the printer device description.
1	The first drawer.
2	The second drawer.
3	The third drawer.

**End pending status.** Whether an End Writer (ENDWTR) command has been issued for this writer.

Possible values are:

- N* No ENDWTR command was issued.
- I* \*IMMED: The writer ends as soon as its output buffers are empty.
- C* \*CNTRLD: The writer ends after the current copy of the spooled file has been printed.
- P* \*PAGEEND: The writer ends at the end of the page.

**Form type.** The type of form being used to print the spooled file.

Possible values are:

- \*ALL* The writer is started with the option to print all spooled files of any form type.
- \*FORMS* The writer is started with the option to print all the spooled files with the same form type before using a different form type.
- \*STD* The writer is started with the option to print all the spooled files with a form type of \*STD.
- form type name* The writer is started with the option to print all the spooled files with the form type you specified.

**Held status.** Whether the writer is held. The possible values are Y (yes) or N (no).

**Hold pending status.** Whether a Hold Writer (HLDWTR) command has been issued for this writer.

Possible values are:

- N* No HLDWTR command was issued.
- I* \*IMMED: The writer is held as soon as its output buffers are empty.
- C* \*CNTRLD: The writer is held after the current copy of the file has been printed.
- P* \*PAGEEND: The writer is held at the end of the page.

**Initialize printer.** How often to initialize the printer device. This field is set to blanks for remote writers.

Possible values are:

- 0* The writer initializes the printer device when necessary.
- 1* The writer initializes the printer device only before the first spooled file is printed and after device error recovery.
- 2* The writer initializes the printer device before every spooled file is printed.

**Job name.** The name of the job that owns the spooled file currently being processed by the writer. This field is blank when no spooled file is printing.

**Job number.** The number of the job that owns the spooled file currently being processed by the writer. This field is blank when no spooled file is printing.

**Job system name.** The name of the system where the job that created the spooled file, currently being processed by the writer, ran. This field is blank when no spooled file is printing.

**Message key.** A key to the message that the writer is waiting for. This field is blank when the writer is not waiting for a reply to an inquiry message.

**Message option.** Message option for sending a message to the message queue when this form is finished.

Possible values are:

<i>*MSG</i>	A message is sent to the message queue.
<i>*NOMSG</i>	No message is sent to the message queue.
<i>*INFOMSG</i>	An informational message is sent to the message queue.
<i>*INQMSG</i>	An inquiry message is sent to the message queue.

**Message queue library name.** The name of the library that contains the message queue.

**Message queue name.** The name of the message queue that this writer uses for operational messages.

**Next file separators.** The next number of separator pages to be printed when the change to the writer takes place.

Possible values are:

<i>-1</i>	<i>*FILE:</i> The number of separator pages is specified by each file.
<i>-10</i>	No pending change to the writer.
<i>number of separators</i>	The number of separator pages to be printed.

**Next form type.** The name of the next form type to be printed.

Possible values are:

<i>*ALL</i>	The writer is changed with the option to print all spooled files of any form type.
<i>*FORMS</i>	The writer is changed with the option to print all the spooled files with the same form type before using a different form type.
<i>*STD</i>	The writer is changed with the option to print all the spooled files with a form type of <i>*STD</i> .
<i>form type name</i>	The writer is changed with the option to print all the spooled files with the form type name you specified.
<i>blank</i>	No change has been made to this writer.

**Next message option.** The message option for sending a message to the message queue when the next form type is finished.

Possible values are:

<i>*MSG</i>	A message is sent to the message queue.
<i>*NOMSG</i>	No message is sent to the message queue.
<i>*INFOMSG</i>	An informational message is sent to the message queue.
<i>*INQMSG</i>	An inquiry message is sent to the message queue.
<i>blank</i>	No change pending.

**Next output queue library name.** The name of the library that contains the next output queue. This field is blank if no changes have been made to the writer.

**Next output queue name.** The name of the next output queue to be processed. This field is blank if no changes have been made to the writer.

**Next separator drawer.** This value indicates the drawer from which to take the separator pages if there is a change to the writer.

Possible values are:

- 1 \*FILE: Separator pages print from the same drawer that the spooled file prints from. If you specify a drawer different from the spooled file that contains colored or different type paper, the page separator is more identifiable.
- 2 \*DEV: Separator pages print from the separator drawer specified in the printer device description.
- 10 No pending change to the writer.
- 1 The first drawer.
- 2 The second drawer.
- 3 The third drawer.

**Number of separators.** The number of separator pages to be printed.

Possible values are:

- 1 \*FILE: The number of separator pages is specified by each file.
- Number of separators* The number of separator pages to be printed.

**On job queue status.** Whether the writer is on a job queue and, therefore, is not currently running. The possible values are Y (yes) or N (no).

**Output queue library name.** The name of the library that contains the output queue from which spooled files are selected for printing.

**Output queue name.** The name of the output queue from which spooled files are being selected for printing.

**Output queue status.** The status of the output queue from which spooled files are being selected for printing.

Possible values are:

- H* The output queue is held.
- R* The output queue is released.

**Page being written.** The page number in the spooled file that is currently being processed by the writer. The page number shown may be lower or higher than the actual page number being printed because of buffering done by the system. This field is set to 0 when no spooled file is printing.

**Printer device name.** The name of the printer device for which writer information is being returned. This field is set to blanks for remote writers.

**Printer device type.** The type of the printer being used to print the spooled file. This field is set to blanks for remote writers.

Valid values are:

- \*SCS SNA (Systems Network Architecture) character stream
- \*IPDS Intelligent Printer Data Stream™

**Reserved.** An ignored field.

**Spooled file create date.** ➤ The date, based on local job time, that the spooled file was created on the system. ⚡ The date is in the CYYMMDD format as follows:

<i>C</i>	Century, where 0 indicates years 19xx and 1 indicates years 20xx.
<i>YY</i>	Year
<i>MM</i>	Month
<i>DD</i>	Day

This field is blank when no spooled file is printing.

**Spooled file create time.** > The time, in local job time, that the spooled file was created on the system.<< The time is in the HHMMSS format where:

<i>HH</i>	Hour
<i>MM</i>	Minutes
<i>SS</i>	Seconds

This field is blank when no spooled file is printing.

**Spooled file name.** The name of the spooled file currently being processed by the writer. This field is blank when no spooled file is printing.

**Spooled file number.** The number of the spooled file currently being processed by the writer. This field is set to 0 when no spooled file is printing.

**Started by user.** The name of the user that started the writer.

**Total copies.** The total number of copies to be printed.

**Total pages.** The total number of pages in the spooled file.

Possible values are:

<i>number</i>	The number of pages in the spooled file.
<i>0</i>	No spooled file is printing.

**Type of writer.** The type of writer for which writer information is being returned.

Valid values are:

<i>0</i>	Printer writer
<i>1</i>	Remote writer

**User name.** The name of the user who owns the spooled file currently being processed by the writer. This field is blank when no file is printing.

**Waiting for data status.** Whether the writer has written all the data currently in the spooled file and is waiting for more data.

Possible values are:

<i>N</i>	The writer is not waiting for more data.
<i>Y</i>	The writer has written all the data currently in the spooled file and is waiting for more data. This condition occurs when the writer is producing an open spooled file with SCHEDULE(*IMMED) specified.

**Waiting for device status.** Whether the writer is waiting to get the device from a job that is printing directly to the printer. This field is set to blanks for remote writers.

- N The writer is not waiting for the device.
- Y The writer is waiting for the device

**Waiting for message status.** Whether the writer is waiting for a reply to an inquiry message. The possible values are Y (yes) or N (no).

**Writer job name.** The job name of the printer writer.

**Writer job number.** The job number of the printer writer.

**Writer job user name.** The name of the system user.

**Writing status.** Whether the printer writer is in writing status. The

Possible values are:

- Y The writer is in writing status.
- N The writer is not in writing status.
- S The writer is writing the file separators.

## Error Messages

Message ID	Error Message Text
CPF24B4 E	Severe error while addressing parameter list.
CPF3CF1 E	Error code parameter not valid.
CPF3C19 E	Error occurred with receiver variable specified.
CPF3C21 E	Format name &1 is not valid.
CPF3C24 E	Length of the receiver variable is not valid.
CPF3C90 E	Literal value cannot be changed.
CPF33C8 E	No writer information returned for printer &1.
CPF3313 E	Writer &1 not active nor on job queue.
CPF3330 E	Necessary resource not available.
CPF33BB E	Writer name is not valid.
CPF33BC E	No writer information returned for writer &1.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V2R3

[Top](#) | ["Print APIs," on page 1](#) | [APIs by category](#)

---

## Send Writer Message (QSPSNDWM) API

Required Parameter Group:

1	Receiver variable	Output	Char(*)
2	Length of receiver variable	Input	Binary(4)
3	Format name	Input	Char(8)
4	Message identifier	Input	Char(7)
5	Qualified message file name	Input	Char(20)
6	Message data	Input	Char(*)
7	Length of message data	Input	Binary(4)

8	Message type	Input	Char(10)
9	Message wait timeout	Input	Binary(4)
10	Message key	Input	Char(4)
11	Writer handle	Input	Char(16)
12	Coded character set identifier	Input	Binary(4)
13	Error code	I/O	Char(*)

Default Public Authority: \*USE  
 Threadsafe: No

The Send Writer Message (QSPSNDWM) API is used by a driver exit program to send either an informational or inquiry message to the writer's message queue.

When an inquiry message is issued, the writer goes into message wait status and waits the specified number of seconds. When the message is answered, or the timeout value has been reached, the API clears the message wait status and returns to its caller. The answer to the inquiry message is returned through the receiver variable parameter.

When an informational message is issued, the API sets the status of the writer to message wait and returns to the caller. The API's caller must then call the API to clear the message wait status.

## Authorities and Locks

None.

## Required Parameter Group

### Receiver variable

OUTPUT; CHAR(\*)

The receiver variable that receives the information requested. You can specify the size of the area to be smaller than the format requested as long as you specify the length parameter correctly. As a result, the API returns only the data that the area can hold.

### Length of receiver variable

INPUT; BINARY(4)

The length of the receiver variable provided. The length of receiver variable parameter may be specified up to the size of the receiver variable specified in the user program. If the length of receiver variable parameter specified is larger than the allocated size of the receiver variable specified in the user program, the results are not predictable. The minimum length is 8 bytes.

### Format name

INPUT; CHAR(8)

The format of the writer message data returned.

*"SNDM0100 Format" on page 187* Contains the information about the message and its reply.

### Message identifier

INPUT; CHAR(7)

The message identifier of the message to issue.

### Qualified message file name

INPUT; CHAR(20)

The name of the message file in which the message resides. The first 10 characters contain the message file name and the second 10 characters contain the name of the library in which the message file is located. If this field is blank, the system message file (QCPFMSG) is used.

The following special values are supported for the library name:

- \*LIBL*            The library list of the writer job is used to locate the message file.
- \*CURLIB*        The current library for the writer job is used to locate the message file.

#### **Message data**

INPUT; CHAR(\*)

The data to insert in the predefined message's substitution variables

#### **Length of message data**

INPUT; BINARY(4)

The length of the message data in bytes. Value values are 1 through 512 bytes.

#### **Message type**

INPUT; CHAR(10)

The type of the message.

The Values are:

- \*INFO*            The message to be sent is an informational message.
- \*INQ*            The message to be sent is an inquiry message.

#### **Message wait timeout**

INPUT; BINARY(4)

The number of seconds to wait for a reply to an inquiry message. Any number greater than 0 is valid. This parameter is only used for inquiry messages. It is ignored for informational messages. The following special value is supported for this parameter.

- 1            The maximum wait time is be used.

**Note:** This parameter is only used for inquiry messages. It is ignored for informational messages.

#### **Message key**

INPUT; CHAR(4)

The key of the message whose wait status is cleared. This field is blank when a message is being sent.

#### **Writer handle**

INPUT; CHAR(16)

The handle to the writer job. This handle is provided to the driver program on a writer call to the driver exit program during initialization.

#### **Coded character set identifier**

INPUT; BINARY(4)

The coded character set identifier used to convert the message data supplied as replacement text for message fields that correspond to \*CCHAR type fields. The data supplied that does not correspond to a \*CCHAR field is 65535 and is not converted. For more information about \*CCHAR type fields, see the Add Message Description (ADDMSGD) command.

The following values are allowed:

- 0            The message data is assumed in the CCSID of the job running this API. This is the default value if this parameter is not specified.
- 65535        The message data is not be converted.

**CCSID** Specify a valid CCSID that your replacement data or impromptu message is in. Valid values are between 1 and 65535. This API validates the CCSID.

For a list of valid CCSIDs, see the i5/OS® globalization topic collection.

**Error code**

I/O; CHAR(\*)

The structure in which to return error information. For the format of the structure, see Error code parameter.

## SNDM0100 Format

The following table shows the information returned for the SNDM0100 format. For more details about the fields in the following table, see “Field Descriptions.”

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Bytes returned
4	4	BINARY(4)	Bytes available
8	8	CHAR(4)	Message key
12	C	CHAR(16)	Reserved
28	1C	BINARY(4)	Offset to message reply text
32	20	BINARY(4)	Length of message reply text returned
36	24	BINARY(4)	Length of message reply text available
		CHAR(*)	Message reply text

## Field Descriptions

**Bytes available.** The number of bytes of data available to be returned. All available data is returned if enough space is provided.

**Bytes returned.** The number of bytes of data returned.

**Length of message reply text available.** The length of the message reply text that is available. This length is zero (0) for informational messages or when the message wait timeout value was reached.

**Length of message reply text returned.** The length of the message reply text that is returned in the message reply text. This length is zero (0) for informational messages or when the message wait timeout value was reached.

**Message key.** The key of the message issued. This can be used with message handler APIs to perform various functions.

**Message reply text.** The text of the reply received from the inquiry message.

**Offset to message reply text.** The offset to the message reply text, in bytes.

**Reserved.** The field is reserved.

## Error Messages

Message ID	Error Message Text
CPF24A8 E	Value for wait time not valid.
CPF24B3 E	Message type &1 not valid.
CPF24B4 E	Severe error while addressing parameter list.
CPF247E E	CCSID &1 is not valid.
CPF3CF1 E	Error code parameter not valid.
CPF3C19 E	Error occurred with receiver variable specified.
CPF3C1D E	Length specified in parameter &1 not valid.
CPF3C21 E	Format name &1 is not valid.
CPF3C4B E	Value not valid for field &1.
CPF3C90 E	Literal value cannot be changed.
CPF33CC E	No writer found for specified handle &1.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V3R7

[Top](#) | ["Print APIs," on page 1](#) | [APIs by category](#)

---

## Set Writer Status (QSPSETWI) API

Required Parameter Group:

1	Status changes	Input	Char(*)
2	Length of status changes	Input	Binary(4)
3	Format name	Input	Char(8)
4	Writer handle	Input	Char(16)
5	Spooled file handle	Input	Char(10)
6	Error code	I/O	Char(*)

Default Public Authority: \*USE  
Threadsafe: No

The Set Writer Status (QSPSETWI) API is used by a driver exit program to update information about a spooled file that a writer is processing. This information is used on certain spooled file displays. For example, the Work with Spooled File (WRKSPLF) command displays the correct status, current page, and total copies information.

## Authorities and Locks

*Output queue lock*  
\*EXCLRD

The lock is on the output queue on which the spooled file resides.

## Required Parameter Group

**Status changes**  
INPUT; CHAR(\*)

The variable that contains the status information to be updated.

**Length of status changes**  
INPUT; BINARY(4)

The length of the status information provided by the status changes parameter. The amount of data specified can be smaller than the information in the format. However, all the status information may not be set appropriately.

**Format name**

INPUT; CHAR(8)

The format of the file status changes.

*“SETW0100 Format”*

Contains the information about the writer and the spooled file status information to be changed.

**Writer handle**

INPUT; CHAR(16)

The handle to the writer job. This handle is provided to the driver program on a writer call to the driver exit program during initialization.

**Spooled file handle**

INPUT; CHAR(10)

The spooled file handle of the current file for which the information is being set. This handle is provided to the driver program on a writer call to the driver exit program (using the process file option).

**Error code**

I/O; CHAR(\*)

The structure in which to return error information. For the format of the structure, see Error code parameter.

**SETW0100 Format**

The following table shows the information specified in the SETW0100 format. For more details about the fields in the following table, see “Field Descriptions” on page 190.

Offset		Type	Field
Dec	Hex		
0	0	CHAR(1)	Change status
1	1	CHAR(1)	Change current page
2	2	CHAR(1)	Change convert page
3	3	CHAR(1)	Change copies
4	4	CHAR(1)	Change accounting pages
5	5	CHAR(1)	Change accounting lines
6	6	CHAR(1)	Change accounting bytes
7	7	CHAR(5)	Reserved
12	C	BINARY(4)	Status
16	10	BINARY(4)	Current page
20	14	BINARY(4)	Convert page
24	18	BINARY(4)	Copies
28	1C	BINARY(4)	Number of pages for accounting
32	20	BINARY(4)	Number of lines for accounting
26	24	PACKED (15,0)	Number of bytes for accounting

## Field Descriptions

**Change accounting bytes.** Change the number of bytes that have been processed for accounting purposes.

Possible values are:

- 0 Do not change the accounting bytes.
- 1 Change the number of bytes that have been processed for accounting to the value specified in the number of bytes for accounting field.

**Change accounting lines.** Change the number of lines that have been processed for accounting purposes.

Possible values are:

- 0 Do not change the accounting lines.
- 1 Change the number of lines that have been processed for accounting to the value specified in the number of lines for accounting field.

**Change accounting pages.** Change the number of pages that have been processed for accounting purposes.

Possible values are:

- 0 Do not change the accounting pages.
- 1 Change the number of pages that have been processed for accounting to the value specified in the number of pages for accounting field.

**Change convert page.** Change the number of pages that have been converted.

Possible values are:

- 0 Do not change the number of pages that have been converted.
- 1 Change the number of pages that have been converted to the value specified in the convert page field.

**Change copies.** Change the number of copies currently printing.

Possible values are:

- 0 Do not change the number of the copies being printed.
- 1 Change the number of the copies being printed to the value specified in the copies field.

**Change current page.** Change the current page that has just printed.

Possible values are:

- 0 Do not change the current page that has just printed.
- 1 Change the current page that has just printed to the value specified in the current page field.

**Change status.** Change the status of the spooled file being processed by the writer.

Possible values are:

- 0 Do not change the status of the spooled file.
- 1 Change the status of the spooled file to the value specified in the status field.

**Copies.** The current number of copies of the spooled file that have been processed by the driver program. The driver program should call this API with the current number of copies processed after the processing of each copy of the spooled file.

**Convert page.** The number of the pages that have been converted.

**Current page.** The number of the page that is being printed.

**Number of bytes for accounting.** The cumulative number of bytes to be logged for accounting.

**Number of lines for accounting.** The cumulative number of lines to be logged for accounting.

**Number of pages for accounting.** The cumulative number of pages to be logged for accounting.

**Reserved.** Field must be set to blank.

**Status.** The new status of the spooled file.

Possible values are:

- 1 (pending) The spooled file is being converted.
- 2 (writing) The spooled file is being selected by the writer.
- 3 (sending) The spooled file is being sent to a remote system.
- 4 (printing) The spooled file is being printed.
- 5 (separator) The writer is printing separator pages.
- 6 (suspend) The driver is still processing the file.
- 7 (interrupt) The driver is done processing the file.
- 8 (ready) The spooled file is ready for processing.
- 9 (held) The spooled file has been held by the driver.
- 10 (sent) The spooled file has been sent to a remote system.
- 11 (finished) The driver is finished with the spooled file.

**Notes:**

- The driver program should use 6 (suspend) when a Hold Writer (HLDWTR) command with option \*PAGEEND or \*CNTRLD has occurred, and the driver program wishes to continue processing the spooled file when the writer is released (RLSWTR).
- The driver program should use 7 (interrupt) when an interruption of the driver program has occurred during the process file option (20) of the printer driver exit, and the driver program wishes to no longer continue processing the spooled file.

An example scenario using the suspend status:

- The writer calls driver program with the process file (20) process option.
- While the driver program is processing the file, a Hold Writer (HLDWTR) command is issued with option \*PAGEEND or \*CNTRLD. The driver program can periodically check for the holding of the writer using the “Extract Writer Status (QSPEXTWI) API” on page 68.
- The driver program finishes processing the spooled file to the next end of page for the \*PAGEEND option of the Hold Writer (HLDWTR) command, and finishes processing the current copy of the spooled file for the \*CNTRLD option of the Hold Writer (HLDWTR) command.
- The driver program then returns to the writer with the error code field of the option specific output information of the print driver exit set to 10.

- The writer then calls the driver program with the hold writer (40) process option.
- The driver program at this time uses the Set Writer Information (QSPSETWI) API to set the status of the spooled file to suspend. This indicates to the writer that the driver program wishes to continue processing the spooled file when the Release Writer (RLSWTR) command has been issued.
- If a Release Writer (RLSWTR) command is issued, the writer will call the driver program with the reprocess file (21) process option.
- The driver program can use the “Extract Writer Status (QSPEXTWI) API” on page 68 to determine where to continue processing of the spooled file. At this time the driver program would update the status of the spooled file from suspended to whatever is appropriate.
- Processing would continue as normal.

An example scenario using the interrupt status follows:

- The writer calls the driver program with the initialize (10) process option.
- The driver program performs whatever initialization and setup is needed before processing of spooled files begins. At this time the driver program decides to allow interrupts to occur while it is processing a spooled file. The driver program does this by setting the allow interrupt field in the option specific output information of the print driver exit interface to 1 before returning to the writer.
- The writer then will call the driver program with the process file (20) process option when a spooled file becomes eligible for processing.
- While the driver program is processing a spooled file, that spooled file is deleted using the Delete Spooled File (DLTSPLF) command, held using the Hold Spooled File (HLDSPLF) command, or a restart page is specified using the Change Spooled File Attributes (CHGSPLFA) command. Since the driver program allowed interrupts, the driver program loses control to the writer immediately.
- The writer calls the driver program with the reprocess file (21) process option.
- The driver program uses the “Extract Writer Status (QSPEXTWI) API” on page 68 to determine whether a delete, hold, or restart page change has been executed.
- The driver program, upon detecting that a hold or delete operation has occurred, then uses the Set Writer Information (QSPSETWI) API to set the status of the spooled file to interrupt and returns control back to the writer.
- Processing would continue as normal.

## Error Messages

Message ID	Error Message Text
CPF24B4 E	Severe error while addressing parameter list.
CPF3CF1 E	Error code parameter not valid.
CPF3C1D E	Length specified in parameter &1 not valid.
CPF3C19 E	Error occurred with receiver variable specified.
CPF3C21 E	Format name &1 is not valid.
CPF3C90 E	Literal value cannot be changed.
CPF33CC E	No writer found for specified handle &1.
CPF33CD E	No file found for specified handle &1.
CPF34CB E	Value not valid for field &1.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V3R7

[Top](#) | [“Print APIs,” on page 1](#) | [APIs by category](#)

---

## Transform AFP to ASCII (QWPZTAFP) API

Required Parameter Group:

1	Qualified input user space name	Input	Char(20)
2	Length of input data	Input	Binary(4)
3	Qualified output user space name	Input	Char(20)
4	Size of output user space used	Output	Binary(4)
5	Output controls table	Input	Char(*)
6	Length of the output controls table	Input	Binary(4)
7	Error code	I/O	Char(*)

Default Public Authority: \*USE  
Threadsafe: No

The Transform AFP™ to ASCII (QWPZTAFP) API transforms an Advanced Function Printing data stream (AFPDS) into an ASCII data stream. The ASCII data stream can be formatted for IBM®, Hewlett-Packard, or PostScript-capable printers.

The QWPZTAFP API processes the following AFPDS objects:

- Document
- Presentation page
- Presentation text data
- IM1 image data
- IO image (IOCA) commands
- Presentation Text 2 (PT2) text commands
- Overlay resources
- Page segment resources

**Note:** IM1 image data is an object format used in the Advanced Function Presentation™ data stream. It is device-dependent and uses pel-to-pel mapping for presentation.

The QWPZTAFP API does not process the following AFPDS objects:

- Graphic data (GOCA) commands
- Bar code (BCOCA™) commands
- Double-byte character set (DBCS) fonts

AFP commands that are not supported by the QWPZTAFP API are ignored, and no warning or error message is sent to the user.

Fonts and character placement may appear different when printing a transformed data stream on an ASCII printer than if the same data stream (not transformed to ASCII) had been printed on an AFP-capable printer.

**Note:** This API is no longer being enhanced. The “Host Print Transform (QWPZHPTR, QwpzHostPrintTransform) API” on page 71 should be used instead.

## Authorities and Locks

*Library Authority*  
\*USE

*User Space Authority*  
\*CHANGE, \*OBJMGT

User Space Lock  
\*EXCL

## Required Parameter Group

### Qualified input user space name

INPUT; CHAR(20)

The name of the user space that contains the Advanced Function Printer data stream to be transformed. The first 10 characters contain the user space name, and the second 10 characters contain the name of the library where the user space is located. The special values allowed for the library name are \*LIBL and \*CURLIB. Both the user space name and the library name are left-justified.

### Length of input data

INPUT; BINARY(4)

The number of bytes of AFP data to be transformed in the input space.

### Qualified output user space name

INPUT; CHAR(20)

The name of the user space to receive the ASCII data stream. The first 10 characters contain the user space name and the second 10 characters contain the name of the library where the user space is located. The special values allowed for the library name are \*LIBL and \*CURLIB. Both the user space name and the library name are left-justified.

### Size of output user space used

OUTPUT; BINARY(4);

The number of bytes written to the output space.

### Output controls table

INPUT; CHAR(\*)

The format that contains information to control certain aspects of the output data generated by the transform option. See the "Output Controls Table Format" for more information.

### Length of the output controls table

INPUT; BINARY(4)

The length of the output controls table in bytes.

### Error code

I/O; CHAR(\*)

The structure in which to return error information. For the format of the structure, see Error code parameter.

## Output Controls Table Format

The following table shows the format for the fields passed to the QWPZTAFF API to control the generated output. See the "Field Descriptions" on page 195 for more information about the field names in the following table.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Data stream type
4	4	BINARY(4)	Top border size
8	8	BINARY(4)	Left border size
12	C	BINARY(4)	Start page number

Offset		Type	Field
Dec	Hex		
16	10	BINARY(4)	End page number
20	14	CHAR(1)	Set character position
21	15	CHAR(1)	Download fonts

## Field Descriptions

**Data stream type.** A number that corresponds to the type of ASCII data stream to be generated. The following table shows the supported values and the data stream type definition.

### Data Stream Type

Value	Data Stream Type Definition
1	IBM LaserPrinter 4019 in page printer data stream mode
2	IBM LaserPrinter 4029 in page printer data stream mode
3	Hewlett-Packard LaserJet II or compatible (4019 in Hewlett-Packard mode)
4	Hewlett-Packard LaserJet III
5	IBM Personal Page Printer in Hewlett-Packard mode
6	Any printer in PostScript mode
7	Text-only data stream

The default value is 7 for a data stream that is only text.

**Download fonts.** Allowed values are Y (yes) and N (no).

If Y is used, fonts specified in the AFP data stream are mapped by the transform operation to a scalable font, available on the ASCII printer, or to a font contained in the transform, that can be downloaded and used on the printer. This value is recommended if the AFP data stream uses typographic fonts.

If N is used, fonts specified in the AFP data stream are mapped to a resident font on the ASCII printer. No soft fonts are downloaded. This reduces the size of the output created by the transform operation.

The default value is Y.

**Note:** Fonts downloaded in Hewlett-Packard printer control language (PCL) format are in portrait-mode only.

**End page number.** The number of the last page in the input buffer to transform. The following pages are skipped. The special value -1 indicates to transform to the end of the buffer.

The default value is -1.

**Left border size.** The size in 1/300ths of an inch of the left border. The AFP to ASCII Transform API moves the output towards the right of the page by the specified amount. This can be used to force output outside of the unprintable border of an ASCII printer. This may cause output to be pushed off of the right side of the page.

Any value 0 or greater is valid. The default value is 0.

**Set character position.** Allowed values are Y (yes) and N (no).

If Y is used, the transform sets the position of each character on the page. This is done to compensate for any difference in character widths between the font specified in the AFP data stream and the font that is used by the transform operation. Using this option improves the appearance of the output in some instances.

If N is used, the transform only sets the position of a character after an absolute move command in the AFP data stream. This reduces the size of the output created by the transform.

The default value is Y.

**Start page number.** The number of the first page in the input buffer to transform. Previous pages are skipped.

The default value is 1.

**Top border size.** The size, in 1/300ths of an inch, of the top border. The AFP to ASCII Transform API moves the output down the page by the specified amount. You can use this API to force output outside of the unprintable border of an ASCII printer; however, this may cause some information to print off the page.

Any value 0 or greater is valid. The default value is 0.

## Error Messages

Message ID	Error Message Text
CPF24B4 E	Severe error while addressing parameter list.
CPF3C90 E	Literal value cannot be changed.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPF6DF0 E	ASCII data stream too large for user space &1.
CPF6DF1 E	Field number &1 in output controls table not valid.
CPF6DF2 E	AFP data stream in user space &1 not valid.
CPF7B10 E	Length parameter &1 is not valid.
CPF9801 E	Object &2 in library &3 not found.
CPF9802 E	Not authorized to object &2 in &3.
CPF9803 E	Cannot allocate object &2 in library &3.
CPF9810 E	Library &1 not found.
CPF9820 E	Not authorized to use library &1.
CPF9856 E	Service Program &2 for program &3 not found or not available.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API Introduced: V2R3

[Top](#) | ["Print APIs," on page 1](#) | [APIs by category](#)

---

## Spooled File APIs

The spooled file APIs can be used to obtain specific information about spooled files. For example, the spooled file APIs can:

- Return a list of spooled files based on given selection criteria, such as a user or an output queue.
- Provide functions to access a specific spooled file from which the API can return the attributes and data of a spooled file or create a duplicate of a specific spooled file.

The spooled file APIs are useful in writing applications to clean up, save, and restore spooled files.

The spooled file APIs include:

- “Close Spooled File (QSPCLOSP) API” on page 198 (QSPCLOSP) closes a spooled file opened by the Open Spooled File (QSPOPNSP) API or created by the Create Spooled File (QSPCRTSP) API.
- “Create Spooled File (QSPCRTSP) API” on page 198 (QSPCRTSP) creates a spooled file. The attributes for the spooled file are based on the values taken from the spooled file attributes parameter. When this spooled file is created, it does not contain any data.
- “Get Spooled File Data (QSPGETSP) API” on page 249 (QSPGETSP) gets data from an existing spooled file. The existing spooled file must have been opened by the Open Spooled File (QSPOPNSP) API. Data is retrieved from the existing spooled file by buffers (one or more) and stored in a user space. The data in the user space is used as source to the Put Spooled File Data (QSPPUTSP) API, which puts the data in the newly created spooled file.
- “List Spooled File AFPDS Resources (QGSLRSC) API” on page 261 (QGSLRSC) generates a list of the AFPDS resources found in the specified spooled file and returns the list in a user space.
- “List Spooled Files (QUSLSPL) API” on page 279 (QUSLSPL) generates a list of spooled files on the system into a user space. The list can include all spooled files or those of specific users, output queues, form types, or user-specified data values.
- “Move Spooled File (QSPMOVSP) API” on page 296 (QSPMOVSP) moves spooled files from one position to another position within the same output queue or from one output queue to another output queue. The priority and status of the moved spooled file can be affected by the user’s priority level and the status of the spooled file.
- “Open List of Spooled Files (QGYOLSPL) API” on page 308 (QGYOLSPL) generates a list of spooled files on the system. It creates a list similar to the list created by calling the List Spooled Files (QUSLSPL) API. The list can include all spooled files or those of specific users, output queues, form types, user-specified data values, statuses, printers, or specific jobs. The filtered list can then be sorted depending on the value of the sort parameter. You can specify how many list entries to return when calling this API. At a later time, you can request additional entries from the list using the Get List Entries (QGYGTLE) API. You can close this list by using the Close List (QGYCLST) API.
- “Open Spooled File (QSPOPNSP) API” on page 325 (QSPOPNSP) opens an existing spooled file. After the existing spooled file is opened, the Get Spooled File Data (QSPGETSP) API can then get the data and put it in the user space.
- “Put Spooled File Data (QSPPUTSP) API” on page 330 (QSPPUTSP) puts data into a spooled file that was created using the Create Spooled File Data (QSPCRTSP) API. The data put in the spooled file is taken from a user space. The data in the user space can be created by either using the Get Spooled File Data (QSPGETSP) API or a user application.
- “Retrieve Identity of Last Spooled File Created (QSPRILSP) API” on page 333 (QSPRILSP) returns the complete spooled file identity of the last spooled file created for the current job or thread.
- “Retrieve Spooled File Attributes (QUSRSPLA) API” on page 337 (QUSRSPLA) returns specific information about a spooled file into a receiver variable. The size of the receiver variable determines the amount of information returned. You can specify the spooled file for which information is returned either with the internal job and spooled file identifiers, or with a specific job name, spooled file name, and spooled file number.

The spooled file exit program is:

-  “UIM Spooled File User-defined List Action Exit Program” on page 436 (QIBM\_QSP\_SPLF\_LSTACT) adds support for user-defined list actions on spooled files listed through the use of the Work with Spooled Files (WRKSPLF), Work with Output Queue (WRKOUTQ), and Work with Job (WRKJOB) CL commands. 

Top | “Print APIs,” on page 1 | APIs by category

---

## Close Spooled File (QSPCLOSP) API

Required Parameter Group:

1	Spooled file handle	Input	Binary(4)
2	Error code	I/O	Char(*)

Default Public Authority: \*USE  
Threadsafe: No

The Close Spooled File (QSPCLOSP) API closes a spooled file opened by the Open Spooled File (QSPOPNSP) API or created by the Create Spooled File (QSPCRTSP) API. A handle, which identifies the spooled file, is returned by the QSPOPNSP and QSPCRTSP APIs. This handle is used as source to the QSPCLOSP API.

A *spooled file handle* is an internal number that identifies a particular spooled file. The system assigns the handle when an existing spooled file is opened using the QSPOPNSP API or when a new spooled file is created using the QSPCRTSP API. The handle is valid until the spooled file is closed or the job ends.

## Authorities and Locks

None.

## Required Parameter Group

### Spooled file handle

INPUT; BINARY(4)

The handle returned by the QSPOPNSP API or QSPCRTSP API.

### Error code

I/O; CHAR(\*)

The structure in which to return error information. For the format of the structure, see Error code parameter.

## Error Messages

Message ID	Error Message Text
CPF24B4 E	Severe error while addressing parameter list.
CPF3C90 E	Literal value cannot be changed.
CPF3CF1 E	Error code parameter not valid.
CPF33DF E	Internal data area for opened spooled files destroyed.
CPF33D2 E	Spooled file handle not valid.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V2R2

[Top](#) | ["Print APIs,"](#) on page 1 | [APIs by category](#)

---

## Create Spooled File (QSPCRTSP) API

Required Parameter Group:

1	Spooled file handle	Output	Binary(4)
2	Spooled file attributes	Input	Char(*)
3	Error Code	I/O	Char(*)

Default Public Authority: \*EXCLUDE  
Threadsafe: No

The Create Spooled File (QSPCRTSP) API is used to create a new spooled file with attributes based on values taken from the spooled file attributes parameter. The field values of the spooled file attributes can be obtained using the Retrieve Spooled File Attributes (QUSRSPLA) API. The fields can also be defined by a user application.

## Authorities and Locks

### *Special Authority*

\*SPLCTL. This authority is needed if you are creating a spooled file for another user.

### *Output Queue Authority*

\*USE

### *Output Queue Library Authority*

\*EXECUTE

The API authority is set this way so that system administrators can control the use of this API. This API has security implications because you can create a spooled file from the data of another spooled file.

## Required Parameter Group

### **Spooled file handle**

OUTPUT; BINARY(4)

The handle to be used by subsequent put (QSPPUTSP API) and close (QSPCLOSP API) operations to identify the spooled file.

### **Spooled file attributes**

INPUT; CHAR(\*)

Attributes returned into the receiver variable by a prior Retrieve Spooled File Attributes (QUSRSPLA) API call with format SPLA0200 specified. These attributes are used to create the new spooled file.

See “SPLA0200 Format Fields” on page 200.

### **Error code**

I/O; CHAR(\*)

The structure in which to return error information. For the format of the structure, see Error code parameter.

## Considerations Using the QSPCRTSP API

The following special considerations apply when using the QSPCRTSP API.

### System Values

When you use the QSPCRTSP API, all system values must be uppercase for i5/OS<sup>®</sup> to recognize them.

### New Spooled File Ownership

The spooled file that is created by the QSPCRTSP API is spooled under one of two jobs. The job is determined by the user-name field. If the user name is the current user, it is a part of the user’s job and is owned by the user profile that the job was started with.

However, ownership of the new spooled file can be assigned to a different user by specifying a different user profile name in the user-name field. The current user must have \*SPLCTL authority to assign the

spooled file to another user. When this is done, the new spooled file is by the user specified in the user-name field. The new spooled file is then part of a **special system job (QPRTJOB)** that is created for each user.

The user profile for the user name must already exist.

## New Spooled File Placement

The new spooled file is placed on the output queue specified in the output queue name field. If the output queue is to correspond to the output queue of the user profile, the user application can use the Retrieve User Profile (RTVUSRPRF) command or the Retrieve User Information (QSYRUSRI) API to retrieve the output queue name.

In both cases, the spooled file name is the one contained in the spooled file attributes parameter. The spooled file number is the next sequential one available for the job that the spooled file becomes a part of.

In both cases (if you are in the System/36™ environment), the system assigns a new System/36 spooled file identifier and a new internal spooled file identifier.

Many of the fields returned by the Retrieve Spooled File Attributes (QUSRSPLA) are closely interrelated with other fields. Changing these field values can lead to unexpected results.

## Error Messages

Error messages can be CPIxxxx or CPFxxxx. Error messages that start with CPI are informational and do not stop the job from running. These messages are stored in the job log.

Error messages that start with CPF cause the job to stop running. These messages are returned to the application program. The form in which they are returned is determined by the error code parameter.

Error checking on the fields includes:

**Note:** If a default is used, that value is obtained from the device file.

- Length of fields is equal to the length of format SPLA0200.
- Special values for fields

When a value is specified that is not valid, message CPI33E2 is issued and the field default is used.

- Incorrect values for fields

When the field specifies a system name and the name is not valid, message CPF33E2 is issued. In the other cases, message CPI33E2 is issued and the default for that field is taken.

- Incorrect combinations of fields

Message CPF33E2 is issued when the value of one field must correspond to a specific value of another field.

## SPLA0200 Format Fields

The following table presents, in the same order, all of the fields of format SPLA0200. It contains the following entries:

<i>Used</i>	This column indicates whether this field is used in describing a new spooled file.
Y	This field is used in describing a new spooled file. Users creating their own spooled file data streams must set the Y fields.
N	This field is not used in describing a new spooled file.

- Cross-dependency* This column indicates whether this field is dependent on other fields.
- Y This field should be changed with extreme caution. This field is closely interrelated with other fields and could cause unpredictable results if not set correctly. The new value will be reflected in the newly created spooled file.
  - N The field is easily changed, and this field is not related to or dependent on another field. The new value is reflected in the newly created spooled file.
  - N/A This field is not used; therefore, there is no need to change it.

## SPLA0200 Format Field Names and Descriptions

The following table shows the information returned for the SPLA0200 format. For more details about the fields in the following table, see “Field Descriptions” on page 216.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Bytes returned Used: Y Cross-dependency: N
4	4	BINARY(4)	Bytes available Used: N Cross-dependency: N/A
8	8	CHAR(8)	Format name Used: Y Cross-dependency: N
16	10	CHAR(16)	Internal job identifier Used: N Cross-dependency: N/A
32	20	CHAR(16)	Internal spooled file identifier Used: N Cross-dependency: N/A
48	30	CHAR(10)	Job name Used: N Cross-dependency: N/A
58	3A	CHAR(10)	User name Used: Y Cross-dependency: Y
68	44	CHAR(6)	Job number Used: N Cross-dependency: N/A
74	4A	CHAR(10)	Spooled file name Used: Y Cross-dependency: N
84	54	BINARY(4)	Spooled file number Used: N Cross-dependency: N/A
88	58	CHAR(10)	Form type Used: Y Cross-dependency: N
98	62	CHAR(10)	User-specified data Used: Y Cross-dependency: N

Offset		Type	Field
Dec	Hex		
108	6C	CHAR(10)	Status Used: N Cross-dependency: N/A
118	76	CHAR(10)	File available Used: Y Cross-dependency: N
128	80	CHAR(10)	Hold file before written Used: Y Cross-dependency: N
138	8A	CHAR(10)	Save file after written Used: Y Cross-dependency: N
148	94	BINARY(4)	Total pages Used: Y Cross-dependency: N
152	98	BINARY(4)	Page or record being written Used: N Cross-dependency: N/A
156	9C	BINARY(4)	Starting page Used: Y Cross-dependency: N
160	A0	BINARY(4)	Ending page Used: Y Cross-dependency: N
164	A4	BINARY(4)	Last page printed Used: N Cross-dependency: N/A
168	A8	BINARY(4)	Restart printing Used: Y Cross-dependency: N
172	AC	BINARY(4)	Total copies Used: N Cross-dependency: N
176	B0	BINARY(4)	Copies left to produce Used: Y Cross-dependency: N
180	B4	BINARY(4)	Lines per inch Used: Y Cross-dependency: Y
184	B8	BINARY(4)	Characters per inch Used: Y Cross-dependency: Y
188	BC	CHAR(2)	Output priority Used: Y Cross-dependency: N
190	BE	CHAR(10)	Output queue name Used: Y Cross-dependency: N

Offset		Type	Field
Dec	Hex		
200	C8	CHAR(10)	Output queue library name Used: Y Cross-dependency: N
210	D2	CHAR(7)	Date file opened (created) Used: N Cross-dependency: N/A
217	D9	CHAR(6)	Time file opened (created) Used: N Cross-dependency: N/A
223	DF	CHAR(10)	Device file name Used: Y Cross-dependency: Y
233	E9	CHAR(10)	Device file library name Used: Y Cross-dependency: Y
243	F3	CHAR(10)	Program that opened file name Used: Y Cross-dependency: Y
253	FD	CHAR(10)	Program that opened file library name Used: Y Cross-dependency: Y
263	107	CHAR(15)	Accounting code Used: Y Cross-dependency: N
278	116	CHAR(30)	Print text Used: Y Cross-dependency: N
308	134	BINARY(4)	Record length Used: Y Cross-dependency: Y
312	138	BINARY(4)	Maximum records Used: Y Cross-dependency: N
316	13C	CHAR(10)	Device type Used: N Cross-dependency: N
326	146	CHAR(10)	Printer device type Used: Y Cross-dependency: Y
336	150	CHAR(12)	Document name Used: Y Cross-dependency: Y
348	15C	CHAR(64)	Folder name Used: Y Cross-dependency: Y
412	19C	CHAR(8)	System/36 procedure name Used: Y Cross-dependency: N

Offset		Type	Field
Dec	Hex		
420	1A4	CHAR(10)	Print fidelity Used: Y Cross-dependency: Y
430	1AE	CHAR(1)	Replace unprintable characters Used: Y Cross-dependency: N
431	1AF	CHAR(1)	Replacement character Used: Y Cross-dependency: Y
432	1B0	BINARY(4)	Page length Used: Y Cross-dependency: Y
436	1B4	BINARY(4)	Page width Used: Y Cross-dependency: Y
440	1B8	BINARY(4)	Number of separators Used: Y Cross-dependency: N
444	1BC	BINARY(4)	Overflow line number Used: Y Cross-dependency: Y
448	1C0	CHAR(10)	DBCS data Used: Y Cross-dependency: Y
458	1CA	CHAR(10)	DBCS extension characters Used: Y Cross-dependency: Y
468	1D4	CHAR(10)	DBCS shift-out shift-in (SO/SI) spacing Used: Y Cross-dependency: Y
478	1DE	CHAR(10)	DBCS character rotation Used: Y Cross-dependency: Y
488	1E8	BINARY(4)	DBCS characters per inch Used: Y Cross-dependency: Y
492	1EC	CHAR(10)	Graphic character set Used: Y Cross-dependency: Y
502	1F6	CHAR(10)	Code page Used: Y Cross-dependency: Y
512	200	CHAR(10)	Form definition name Used: Y Cross-dependency: Y
522	20A	CHAR(10)	Form definition library name Used: Y Cross-dependency: Y

Offset		Type	Field
Dec	Hex		
532	214	BINARY(4)	Source drawer Used: Y Cross-dependency: Y
536	218	CHAR(10)	Printer font Used: Y Cross-dependency: Y
546	222	CHAR(6)	System/36 spooled file identifier Used: N Cross-dependency: N/A
552	228	BINARY(4)	Page rotation Used: Y Cross-dependency: N
556	22C	BINARY(4)	Justification Used: Y Cross-dependency: N
560	230	CHAR(10)	Print on both sides (duplex) Used: Y Cross-dependency: Y
570	23A	CHAR(10)	Fold records Used: Y Cross-dependency: N
580	244	CHAR(10)	Control character Used: Y Cross-dependency: Y
590	24E	CHAR(10)	Align forms Used: Y Cross-dependency: N
600	258	CHAR(10)	Print quality Used: Y Cross-dependency: N
610	262	CHAR(10)	Form feed Used: Y Cross-dependency: N
620	26C	CHAR(71)	Volumes (array) Used: N Cross-dependency: N/A
691	2B3	CHAR(17)	File label identifier Used: N Cross-dependency: N/A
708	2C4	CHAR(10)	Exchange type Used: N Cross-dependency: N/A
718	2CE	CHAR(10)	Character code Used: N Cross-dependency: N/A
728	2D8	BINARY(4)	Total records Used: Y Cross-dependency: N

Offset		Type	Field
Dec	Hex		
732	2DC	BINARY(4)	Multiple up (pages per side) Used: Y Cross-dependency: N
736	2E0	CHAR(10)	Front overlay name Used: Y Cross-dependency: Y
746	2EA	CHAR(10)	Front overlay library name Used: Y Cross-dependency: Y
756	2F4	PACKED(15,5)	Front overlay offset down Used: Y Cross-dependency: Y
764	2FC	PACKED(15,5)	Front overlay offset across Used: Y Cross-dependency: Y
772	304	CHAR(10)	Back overlay name Used: Y Cross-dependency: Y
782	30E	CHAR(10)	Back overlay library name Used: Y Cross-dependency: Y
792	318	PACKED(15,5)	Back overlay offset down Used: Y Cross-dependency: Y
800	320	PACKED(15,5)	Back overlay offset across Used: Y Cross-dependency: Y
808	328	CHAR(10)	Unit of measure Used: Y Cross-dependency: Y
818	332	CHAR(10)	Page definition name Used: Y Cross-dependency: Y
828	33C	CHAR(10)	Page definition library name Used: Y Cross-dependency: Y
838	346	CHAR(10)	Line spacing Used: Y Cross-dependency: Y
848	350	PACKED(15,5)	Point size Used: Y Cross-dependency: Y
856	358	BINARY(4)	Maximum spooled data record size Used: Y Cross-dependency: Y
860	35C	BINARY(4)	Spooled file buffer size Used: Y Cross-dependency: N

Offset		Type	Field
Dec	Hex		
864	360	CHAR(6)	Spooled file level Used: Y Cross-dependency: N
870	366	ARRAY(4)	Coded font array Used: Y Cross-dependency: Y
886	376	CHAR(10)	Channel mode Used: Y Cross-dependency: Y
896	380	ARRAY(12)	Channel value array Used: Y Cross-dependency: Y
944	3B0	CHAR(8)	Graphics token Used: Y Cross-dependency: Y
952	3B8	CHAR(10)	Record format Used: Y Cross-dependency: Y
962	3C2	CHAR(2)	Reserved
964	3C4	PACKED(15,5)	Height of drawer 1 Used: Y Cross-dependency: Y
972	3CC	PACKED(15,5)	Width of drawer 1 Used: Y Cross-dependency: Y
980	3D4	PACKED(15,5)	Height of drawer 2 Used: Y Cross-dependency: Y
988	3DC	PACKED(15,5)	Width of drawer 2 Used: Y Cross-dependency: Y
996	3E4	BINARY(4)	Number of buffers Used: N Cross-dependency: N/A
1000	3E8	BINARY(4)	Maximum forms width Used: Y Cross-dependency: N
1004	3EC	BINARY(4)	Alternate forms width Used: Y Cross-dependency: Y
1008	3F0	BINARY(4)	Alternate forms length Used: Y Cross-dependency: Y
1012	3F4	BINARY(4)	Alternate lines per inch Used: Y Cross-dependency: Y
1016	3F8	CHAR(2)	System/38™ Text Utility flags Used: Y Cross-dependency: N

Offset		Type	Field
Dec	Hex		
1018	3FA	CHAR(1)	File open Used: Y Cross-dependency: N
1019	3FB	CHAR(1)	Page count estimated Used: Y Cross-dependency: Y
1020	3FC	CHAR(1)	File stopped on page boundary Used: Y Cross-dependency: N
1021	3FD	CHAR(1)	TRC for 1403 Used: Y Cross-dependency: Y
1022	3FE	CHAR(1)	Define characters Used: Y Cross-dependency: Y
1023	3FF	CHAR(1)	Characters per inch changes Used: Y Cross-dependency: Y
1024	400	CHAR(1)	Transparency Used: Y Cross-dependency: Y
1025	401	CHAR(1)	Double-wide characters Used: Y Cross-dependency: Y
1026	402	CHAR(1)	DBCS character rotation commands Used: Y Cross-dependency: Y
1027	403	CHAR(1)	Extended code page Used: Y Cross-dependency: Y
1028	404	CHAR(1)	FFT emphasis Used: Y Cross-dependency: Y
1029	405	CHAR(1)	3812 SCS Used: Y Cross-dependency: Y
1030	406	CHAR(1)	Set Line Density command Used: Y Cross-dependency: Y
1031	407	CHAR(1)	Graphics error actions Used: Y Cross-dependency: Y
1032	408	CHAR(1)	5219 commands Used: Y Cross-dependency: Y
1033	409	CHAR(1)	3812 SCS commands Used: Y Cross-dependency: Y

Offset		Type	Field
Dec	Hex		
1034	40A	CHAR(1)	Field outlining Used: Y Cross-dependency: Y
1035	40B	CHAR(1)	Final form text Used: Y Cross-dependency: Y
1036	40C	CHAR(1)	Bar code Used: Y Cross-dependency: Y
1037	40D	CHAR(1)	Color Used: Y Cross-dependency: Y
1038	40E	CHAR(1)	Drawer change Used: Y Cross-dependency: Y
1039	40F	CHAR(1)	Character ID Used: Y Cross-dependency: Y
1040	410	CHAR(1)	Lines per inch changes Used: Y Cross-dependency: Y
1041	411	CHAR(1)	Font Used: Y Cross-dependency: Y
1042	412	CHAR(1)	Highlight Used: Y Cross-dependency: Y
1043	413	CHAR(1)	Page rotate Used: Y Cross-dependency: Y
1044	414	CHAR(1)	Subscript Used: Y Cross-dependency: Y
1045	415	CHAR(1)	Superscript Used: Y Cross-dependency: Y
1046	416	CHAR(1)	DDS Used: Y Cross-dependency: Y
1047	417	CHAR(1)	Final form feed Used: Y Cross-dependency: Y
1048	418	CHAR(1)	SCS data Used: Y Cross-dependency: Y
1049	419	CHAR(1)	User-generated data stream Used: Y Cross-dependency: Y

Offset		Type	Field
Dec	Hex		
1050	41A	CHAR(1)	Graphics Used: Y Cross-dependency: Y
1051	41B	CHAR(1)	Unrecognizable data Used: Y Cross-dependency: Y
1052	41C	CHAR(1)	ASCII transparency Used: Y Cross-dependency: Y
1053	41D	CHAR(1)	IPDS™ transparent data Used: Y Cross-dependency: N
1054	41E	CHAR(1)	OfficeVision® Used: Y Cross-dependency: N
1055	41F	CHAR(1)	Lines-per-inch (lpi) value not supported Used: Y Cross-dependency: N
1056	420	CHAR(1)	CPA3353 message Used: Y Cross-dependency: N
1057	421	CHAR(1)	Set exception Used: Y Cross-dependency: Y
1058	422	CHAR(1)	Carriage control characters Used: Y Cross-dependency: Y
1059	423	CHAR(1)	Page position Used: Y Cross-dependency: Y
1060	424	CHAR(1)	Character not valid Used: Y Cross-dependency: Y
1061	425	CHAR(1)	Lengths present Used: Y Cross-dependency: Y
1062	426	CHAR(1)	5A present Used: Y Cross-dependency: Y
1063	427	CHAR(1)	Reserved
1064	428	BINARY(4)	Number of font array entries Used: Y Cross-dependency: Y
1068	42C	BINARY(4)	Number of resource library entries Used: N Cross-dependency: N
1072	430	CHAR(1153)	Font equivalence array Used: Y Cross-dependency: Y

Offset		Type	Field
Dec	Hex		
2225	8B1	CHAR(631)	Resource library array Used: N Cross-dependency: N
2856	B28	CHAR(1)	i5/OS-created AFPDS Used: Y Cross-dependency: Y
2857	B29	CHAR(1)	Job character ID specified Used: Y Cross-dependency: Y
2858	B2A	CHAR(1)	S36 CONTINUE-YES Used: Y Cross-dependency: N
2859	B2B	CHAR(10)	Decimal format Used: Y Cross-dependency: N
2869	B35	CHAR(7)	Date file last used Used: N Cross-dependency: N/A
2876	B3C	CHAR(1)	Page groups Used: Y Cross-dependency: Y
2877	B3D	CHAR(1)	Group level index tags Used: Y Cross-dependency: Y
2878	B3E	CHAR(1)	Page level index tags Used: Y Cross-dependency: Y
2879	B3F	CHAR(1)	IPDS pass-through Used: Y Cross-dependency: N
2880	B40	BINARY(4)	Offset to user resource library list Used: Y Cross-dependency: Y
2884	B44	BINARY(4)	Number of user resource library list entries Used: Y Cross-dependency: Y
2888	B48	BINARY(4)	Length of user resource library list entry Used: Y Cross-dependency: Y
2892	B4C	CHAR(2)	Reserved
2894	B4E	CHAR(1)	Corner staple Used: Y Cross-dependency: N
2895	B4F	CHAR(1)	Edge-stitch reference edge Used: Y Cross-dependency: Y
2896	B50	PACKED(15,5)	Offset from edge-stitch reference edge Used: Y Cross-dependency: Y

Offset		Type	Field
Dec	Hex		
2904	B58	BINARY(4)	Edge-stitch number of staples Used: Y Cross-dependency: Y
2908	B5C	BINARY(4)	Offset to edge-stitch staple offset list Used: Y Cross-dependency: Y
2912	B60	BINARY(4)	Number of edge-stitch staple offset entries Used: Y Cross-dependency: Y
2916	B64	BINARY(4)	Length of edge-stitch staple offset entry Used: Y Cross-dependency: Y
2920	B68	CHAR(10)	Font resolution for formatting Used: Y Cross-dependency: N
2930	B72	CHAR(1)	Record format name present in data stream Used: Y Cross-dependency: Y
2931	B73	CHAR(1)	Saddle-stitch reference edge Used: Y Cross-dependency: Y
2932	B74	BINARY(4)	Saddle-stitch number of staples Used: Y Cross-dependency: Y
2936	B78	BINARY(4)	Offset to saddle-stitch staple offset list Used: Y Cross-dependency: Y
2940	B7C	BINARY(4)	Number of saddle-stitch staple offset entries Used: Y Cross-dependency: Y
2944	B80	BINARY(4)	Length of saddle-stitch staple offset entry Used: Y Cross-dependency: Y
2948	B84	PACKED(15,0)	Data stream size Used: N Cross-dependency: N
2956	B8C	BINARY(4)	Offset to library list Used: N Cross-dependency: N
2960	B90	BINARY(4)	Number of library list entries Used: N Cross-dependency: N
2964	B94	BINARY(4)	Length of library list entry Used: N Cross-dependency: N
2968	B98	BINARY(4)	Offset to Internet Print Protocol spooled file attributes Used: N Cross-dependency: N/A

Offset		Type	Field
Dec	Hex		
2972	B9C	BINARY(4)	Offset to Save and Restore spooled file attributes Used: N Cross-dependency: N/A
2976	BA0	BINARY(4)	CCSID of job Used: Y Cross-dependency: N
2980	BA4	CHAR(172)	Reserved
3152	C50	PACKED(15,5)	Front margin offset down Used: Y Cross-dependency: Y
3160	C58	PACKED(15,5)	Front margin offset across Used: Y Cross-dependency: Y
3168	C60	PACKED(15,5)	Back margin offset down Used: Y Cross-dependency: Y
3176	C68	PACKED(15,5)	Back margin offset across Used: Y Cross-dependency: Y
3184	C70	PACKED(15,5)	Length of page Used: Y Cross-dependency: Y
3192	C78	PACKED(15,5)	Width of page Used: Y Cross-dependency: Y
3200	C80	CHAR(10)	Measurement method Used: Y Cross-dependency: Y
3210	C8A	CHAR(1)	Advanced Function Presentation™ (AFP™) resource Used: Y Cross-dependency: Y
3211	C8B	CHAR(10)	Character set name Used: Y Cross-dependency: Y
3221	C95	CHAR(10)	Character set library name Used: Y Cross-dependency: Y
3231	C9F	CHAR(10)	Code page name Used: Y Cross-dependency: Y
3241	CA9	CHAR(10)	Code page library name Used: Y Cross-dependency: Y
3251	CB3	CHAR(10)	Coded font name Used: Y Cross-dependency: Y
3261	CBD	CHAR(10)	Coded font library name Used: Y Cross-dependency: Y

Offset		Type	Field
Dec	Hex		
3271	CC7	CHAR(10)	DBCS-coded font name Used: Y Cross-dependency: Y
3281	CD1	CHAR(10)	DBCS-coded font library name Used: Y Cross-dependency: Y
3291	CDB	CHAR(10)	User-defined file Used: N Cross-dependency: N
3301	CE5	CHAR(10)	Reduce output Used: Y Cross-dependency: Y
3311	CEF	CHAR(1)	Constant back overlay Used: Y Cross-dependency: Y
3312	CF0	BINARY(4)	Output bin Used: Y Cross-dependency: N
3316	CF4	BINARY(4)	CCSID Used: Y Cross-dependency: N
3320	CF8	CHAR(100)	User-defined text Used: Y Cross-dependency: N
3420	D5C	CHAR(8)	System where file originated Used: Y Cross-dependency: N
3428	D64	CHAR(8)	Net ID where file originated Used: Y Cross-dependency: N
3436	D6C	CHAR(10)	User who created file Used: Y Cross-dependency: N
3446	D76	CHAR(2)	Reserved
3448	D78	BINARY(4)	Offset to user-defined options Used: Y Cross-dependency: Y
3452	D7C	BINARY(4)	Number of user-defined options returned Used: Y Cross-dependency: Y
3456	D80	BINARY(4)	Length of each user-defined option entry Used: Y Cross-dependency: Y
3460	D84	CHAR(255)	User-defined data Used: Y Cross-dependency: N
3715	E83	CHAR(10)	User-defined object name Used: Y Cross-dependency: Y

Offset		Type	Field
Dec	Hex		
3725	E8D	CHAR(10)	User-defined object library name Used: Y Cross-dependency: Y
3735	E97	CHAR(10)	User object type Used: Y Cross-dependency: Y
3745	EA1	CHAR(3)	Reserved
3748	EA4	PACKED(15,5)	Character set point size Used: Y Cross-dependency: Y
3756	EAC	PACKED(15,5)	Coded font point size Used: Y Cross-dependency: Y
3764	EB4	PACKED(15,5)	DBCS-coded font point size Used: Y Cross-dependency: Y
3772	EBC	BINARY(4)	Auxiliary storage pool Used: N Cross-dependency: N/A
3776	EC0	BINARY(4)	Spooled file size Used: N Cross-dependency: N/A
3780	EC4	BINARY(4)	Spooled file size multiplier Used: N Cross-dependency: N/A
3784	EC8	BINARY(4)	Internet print protocol job identifier Used: N Cross-dependency: N/A
3788	ECC	CHAR(1)	Spooled file creation security method Used: Y Cross-dependency: N
3789	ECD	CHAR(1)	Spooled file creation authentication method Used: Y Cross-dependency: N
3790	ECE	CHAR(7)	Date writer began processing spooled file Used: N Cross-dependency: N/A
3797	ED5	CHAR(6)	Time writer began processing spooled file Used: N Cross-dependency: N/A
3803	EDB	CHAR(7)	Date writer completed processing spooled file Used: N Cross-dependency: N/A
3810	EE2	CHAR(6)	Time writer completed processing spooled file Used: N Cross-dependency: N/A
3816	EE8	CHAR(8)	Job system name Used: N Cross-dependency: N/A

Offset		Type	Field
Dec	Hex		
3824	EF0	CHAR(10)	Auxiliary storage pool device name Used: N Cross-dependency: N/A
3834	EFA	CHAR(7)	Expiration date Used: Y Cross-dependency: N

## Field Descriptions

**Accounting code.** An identifier assigned by the system to record the resources used to write this spooled file. This can be retrieved for the user owning the spooled file by using the Retrieve User Profile (RTVUSRPRF) command or the Retrieve User Information (QSYRUSRI) API, specifying the ACGCDE parameter.

**Advanced Function Presentation (AFP) resource.** Whether this spooled file refers to AFP resources external to this spooled file, for example, page segments or overlays.

The possible values are:

- Y The spooled file refers to AFP resources external to the spooled file.
- N The spooled file does not refer to AFP resources external to the spooled file.

**Align forms.** Whether a forms alignment message is sent prior to printing the data. The options are \*YES or \*NO.

**Alternate forms length.** The length of the alternate forms in lines. This field applies only to spooled files created by the OfficeVision program and should be set to 0 by an application building this format as opposed to retrieving the fields. The other alternate forms fields should also be set to 0.

**Alternate forms width.** The width of the alternate forms in character positions. This field applies only to spooled files created by the OfficeVision program and should be set to 0 by an application building this format as opposed to retrieving the fields. The other alternate forms fields should also be set to 0.

**Alternate lines per inch.** The lines per inch for the alternate forms. This field applies only to spooled files created by the OfficeVision program and should be set to 0 by an application building this format as opposed to retrieving the fields. The other alternate forms fields should also be set to 0.

**ASCII transparency.** For SCS files, whether ASCII commands are embedded in the ASCII transparency command. The ASCII transparency command is command\_ID/command\_length/ASCII\_data. The command ID is a 1-byte field with the value hex 03. The command length is a 1-byte field that contains the length of the command length field and the ASCII data field.

Valid values are Y (yes) or N (no). This field should be set to N when creating a spooled file as opposed to copying an existing spooled file.

**Auxiliary storage pool.** The number of the auxiliary storage pool (ASP) that the spooled file is stored on.

**Auxiliary storage pool device name.** The name of the auxiliary storage pool (ASP) device that the spooled file is stored on.

**Back margin offset across.** For the back side of a piece of paper, in either inches or centimeters (specified in the unit of measure (UOM) parameter), how far from the left side of the page printing starts. The possible values are:

0-57.79      The offset in inches or centimeters, according to the unit of measure field. The maximum offset is 57.79 centimeters or 22.75 inches.

This field should only be specified for spooled files with device types of \*LINE, \*AFPDSLIN, or \*AFPDS if the spooled file was created on a System i™ platform. See i5/OS-created \*AFPDS field (page 238).

If this field is not specified for one of these valid types, a default value is used.

**Back margin offset down.** For the back side of a piece of paper, in either inches or centimeters (specified in the unit of measure (UOM) parameter), how far down from the top of the page printing starts.

The possible values are:

-1            The back margin offsets are the same as the front margin offsets.  
-2            \*DEV. For printers configured AFP(\*YES), no print border is used for back margin offsets across and down. Otherwise, the offset values are 0.  
0-57.79      The offset in inches or centimeters, according to the unit of measure field. The maximum offset is 57.79 centimeters or 22.75 inches.

This field should only be specified for spooled files with device types of \*LINE, \*AFPDSLIN, or \*AFPDS if the spooled file was created on a System i platform. See i5/OS-created \*AFPDS field (page 238).

If this field is not specified for one of these valid types, a default value is used.

**Back overlay library name.** The name of the library containing the back overlay.

The possible values are:

\*CURLIB      The current library for the job locates the back overlay.  
\*LIBL        The library list locates the back overlay.  
library name      This library is searched for the back overlay.

This field should only be specified for spooled files with device types of \*SCS, \*IPDS, or \*AFPDS if the spooled file was created on a System i platform. See i5/OS-created \*AFPDS field (page 238).

If this field is not specified for one of these valid types, a default value is used.

**Back overlay name.** The name of the **back overlay** (the material that prints on the back side of each page).

The possible values are:

\*FRONTOVL      The back overlay is the same as the front overlay.  
\*NONE            The file does not use a back overlay.  
back overlay name      The name of the back overlay.

If an overlay name is specified, the back overlay library name must also be specified.

This field should only be specified for spooled files with device types of \*SCS, \*IPDS, or \*AFPDS if the spooled file was created on a System i platform. See i5/OS-created \*AFPDS field (page 238).

If this field is not specified for one of these valid types, a default value is used.

**Back overlay offset across.** The offset across from the point of origin where the overlay is printed.

The possible values are:

0-57.79            The offset in inches or centimeters, according to the unit of measure field. The maximum offset is 57.79 centimeters or 22.75 inches.

This field should only be specified for spooled files with device types of \*SCS, \*IPDS, or \*AFPDS if the spooled file was created on a System i platform. See i5/OS-created \*AFPDS field (page 238).

If this field is not specified for one of these valid types, a default value is used.

**Back overlay offset down.** The offset down from the point of origin where the overlay is printed.

The possible values are:

0-57.79            The offset in inches or centimeters, according to the unit of measure field. The maximum offset is 57.79 centimeters or 22.75 inches.

This field should only be specified for spooled files with device types of \*SCS, \*IPDS, or \*AFPDS if the spooled file was created on a System i platform. See i5/OS-created \*AFPDS field (page 238).

If this field is not specified for one of these valid types, a default value is used.

**Bar code.** Whether the spooled file contains bar codes created using the BARCODE keyword in the data description specifications (DDS).

For DEVTYPE(\*IPDS and \*AFPDS), a series of bar code commands is contained in the data stream.

Valid values are Y (yes) or N (no). Set this field to N for all other printer device types.

**Bytes available.** The number of bytes of data available to be returned. All available data is returned if enough space is provided.

**Bytes returned.** The total number of bytes contained in the field format. This must be the length of format SPLA0200 plus the length of the user-defined options.

**Carriage control characters.** Whether the file has machine carriage control characters.

Valid values are Y (yes) or N (no).

**CCSID.** The coded character set identifier (CCSID) used when the text description was created using the Change User Print Information (CHGUSRPTI) command. Possible values are 0 and 1 - 65535, except 65534.

**CCSID of job.** The coded character set identifier (CCSID) of the job when the spooled file was created. The CCSID specified should represent the contents of the spooled file.

The possible values are:

0 The CCSID is unknown. This value should be used if the CCSID of the contents of data to be spooled is not known.  
1-65535, except 65534 The coded character set identifier of the data to be spooled.

**Channel value array.** The array contains 12 channels. Each channel is assigned a number, 1 through 12. Each channel (or number in the array) is assigned a value. That value is the line that the printer skips to before printing starts. For example, if channel 9 had a value of 9, the printer skips to line 9 before printing starts.

When creating a file with data previously spooled and channel values specified, changing the channel values does not affect the newly created spooled file. The channel values used when printing are the ones specified when the data was originally spooled.

**Channel mode.** A variable indicating the channel value mode.

Possible values are:

*\*NORMAL* Use the normal channel values (1 equals skip to line 1, 2 through B equal space one line before printing, C equals skip to overflow line).  
*blank* The channel values specified in the channel value array are used.

**Character code.** Whether the coding for the diskette file is in ASCII or EBCDIC form.

**Character ID.** Whether the character ID can change within the file.

- For DEVTYPE(\*SCS), a Set CGCS through Local ID command is contained in the spooled file.
- For DEVTYPE(\*IPDS), a Load Font Equivalence command is inserted into the Load Font Equivalence table.
- For DEVTYPE(\*AFPDS), a map code font-2 (MCF-2) structured field is contained in the spooled file.

Valid values are Y (yes) or N (no). Set this field to N for all other printer device types.

**Character not valid.** Whether incorrect character errors are reported. This field is valid only when the printer device type (DEVTYPE) field is \*LINE, \*AFPDSLIN, or \*AFPDS.

Valid values are Y (yes) or N (no). This field is set to N for all other printer device types.

**Character set library name.** The name of the library containing the font character set object.

The possible values are:

*\*CURLIB* The current library is searched for the font character set object.  
*\*LIBL* The library list is used to locate the font character set object.  
*library name* This library is searched for the font character set object.

This field should only be specified for spooled files that contain \*AFPDS created on the i5/OS operating system. See i5/OS-created \*AFPDS field (page 238).

If this field is not specified for one of these valid types, a default value is used.

**Character set name.** The name of the font character set object used to print this file.

The possible values are:

*\*FONT*                    The information specified on the font parameter is used instead of the character set and code page.  
*character set name*    The name of the font character set object to use.

This field should only be specified for spooled files that contain \*AFPDS created on the i5/OS operating system. See i5/OS-created \*AFPDS field (page 238).

If this field is not specified for one of these valid types, a default value is used.

**Character set point size.** The point size in which this spooled file's characters (defined by the character set field) should be printed.

The possible values are:

*000.0*                    \*NONE: The character set does not have a point size.  
*point size*              The point size of the character set. The number must be between 000.1 and 999.9.  
  
This field is ignored if the character set is specified as \*FONT.

This field should only be specified for spooled files that contain \*AFPDS created on the i5/OS operating system. See i5/OS-created \*AFPDS field (page 238).

If this field is not specified for one of these valid types, a default value is used.

**Characters per inch.** The number (in tenths) of characters per horizontal inch, defined in the printer file. The value 100 indicates 10 characters per inch. Valid characters per inch include 5, 10, 12, 15, and 16.7. Characters per inch of 13.3, 18, and 20 are valid for double-byte character set devices.

**Characters per inch changes.** Whether the spooled file changes the characters per inch (cpi) within the file.

For DEVTYPE(\*SCS), a Set Character Distance command is contained in the spooled file.

Valid values are Y (yes) or N (no). Set this field to N for all other printer device types.

**Code page.** The mapping of graphic characters to code points for this printer. For \*DEVD, the system gets the code page from the printer device description and this value is ignored. Valid values are 1 through 32767 and \*DEVD.

When a numeric value is specified, it is represented as a character string that is right justified with leading zeros.

**Code page library name.** The name of the library containing the code page used to print this spooled file.

The possible values are:

*\*CURLIB*                The current library is searched for the code page name.  
*\*LIBL*                    The library list is used to locate the code page.  
*library name*            This library is searched for the code page.

This field should only be specified for spooled files that contain \*AFPDS created on the i5/OS operating system. See i5/OS-created \*AFPDS field (page 238).

If this field is not specified for one of these valid types, a default value is used.

**Code page name.** The name of the code page used to print this spooled file.

This field should only be specified for spooled files that contain \*AFPDS created on the i5/OS operating system.. See i5/OS-created \*AFPDS field (page 238).

If this field is not specified for one of these valid types, a default value is used.

**Coded font array.** The name of the fonts used when printing a spooled file with a printer device type (DEVTYPE) of \*LINE or \*AFPDSLIN. The name does not include the 2-character prefix of the coded font name (X0-XG). This array should be blank for all other printer device types.

**Coded font library name.** The name of the library containing the coded font used to print this spooled file.

The possible values are:

<i>*CURLIB</i>	The current library is searched for the coded font.
<i>*LIBL</i>	The library list is used to locate the coded font.
<i>library name</i>	This library is searched for the coded font.

This field should only be specified for spooled files that contain \*AFPDS created on the i5/OS operating system. See i5/OS-created \*AFPDS field (page 238).

If this field is not specified for one of these valid types, a default value is used.

**Coded font name.** The name of the coded font used to print this spooled file. A **coded font** is an AFP resource composed of a font character set and a code page.

The possible values are:

<i>*FNTCHRSET</i>	The values used are the values specified on the character set and code page fields.
<i>coded font name</i>	The name of the coded font used to print this spooled file.

This field should only be specified for spooled files that contain \*AFPDS created on the i5/OS operating system. See i5/OS-created \*AFPDS field (page 238).

If this field is not specified for one of these valid types, a default value is used.

**Coded font point size.** The point size in which this spooled file's characters (defined by the coded font field) should be printed.

The possible values are:

<i>000.0</i>	*NONE: The coded font does not have a point size.
<i>point size</i>	The point size of the coded font. The number must be between 000.1 and 999.9.

This field is ignored if the coded font is specified as \*FNTCHRSET.

This field should only be specified for spooled files that contain \*AFPDS created on the i5/OS operating system. See i5/OS-created \*AFPDS field (page 238).

If this field is not specified for one of these valid types, a default value is used.

**Color.** Whether an IPDS Write Text command or AFPDS presentation text data (PTX) structured field containing a Set Text Color control is contained in the spooled file.

For DEVTYPE(\*IPDS and \*AFPDS), a Set Text Color text control is contained in the spooled file.

Valid values are Y (yes) or N (no). Set this field to N for all other printer device types.

**Constant back overlay.** Whether or not constant back overlay is used. Constant back allows the user to print overlays on the back side of each sheet of media without the need for the print application to generate blank pages. Specifying the constant back function causes the operating system to generate blank pages onto which the back overlay will be printed. The generated pages are called constant forms because no variable data from the user's print application is printed on the pages. If the multiup field is not set to 1, constant back overlay will be set to \*NOCONSTANT. Constant back overlay is ignored if the print on both sides field is set to \*NO.

The possible values are:

- 0 \*NOCONSTANT: The constant back overlay function is not used.
- 1 \*CONSTANT: The constant back overlay function is used. The values for back overlay offset down and back overlay offset across fields are ignored and set to 0.0.

**Control character.** Whether the data to be printed used the American National Standard printer control character.

The possible values are:

- \*NONE No print control characters are passed in the data that is printed.
- \*FCFC The first character of every record is an American National Standard printer control character. The record length field must include one position for forms control code. This value is not valid for externally described printer files.

**Copies left to produce.** The number of copies remaining to be produced on the printer. Valid values are 1 through 255.

**CPA3353 message.** Whether message CPA3353 is issued when this spooled file is printed.

Valid values are Y (yes) or N (no).

**Corner staple.** Specifies the reference corner to be used for a corner staple. A staple is driven into the media at the reference corner. Refer to your printer's documentation for information as to which reference corners are supported.

Valid values are:

- 0 \*NONE. A corner staple is not specified.
- 1 \*DEVD. The default reference corner of the printer device is used.
- 2 \*BOTRIGHT. The bottom right-hand corner of the paper is specified.
- 3 \*TOPRIGHT. The top right-hand corner of the paper is specified.
- 4 \*TOPLEFT. The top left-hand corner of the paper is specified.
- 5 \*BOTLEFT. The bottom left-hand corner of the paper is specified.

**Data stream size.** The number of bytes of data stream for the file.

**Date file last used.** ➤ The date, based on local job time, that the file data was last displayed, copied, printed, or sent; or when the file attributes were last changed. ⏪ The date is in the CYYMMDD format as follows:

- C Century, where 0 indicates years 19xx and 1 indicates years 20xx.

YY           Year  
MM           Month  
DD           Day

**Date file opened (created).** » The date, based on local job time, that the file was opened in the CYYMMDD format where: «

C            Century, where 0 indicates years 19xx and 1 indicates years 20xx.  
YY           Year  
MM           Month  
DD           Day

**Date writer began processing spooled file.** » The date, based on local job time, that a spooling writer started processing the spooled file. «

**Date writer completed processing spooled file.** » The date, based on local job time, that a spooling writer completed processing the spooled file. «

**DBCS character rotation.** Whether this printer file causes the DBCS characters to be rotated 90 degrees counterclockwise before printing. The possible values are \*YES and \*NO. This field should be \*NO when the DBCS field is \*NO.

**DBCS character rotation commands.** Whether the double-byte character set characters are rotated 90 degrees counterclockwise before printing.

- For DEVTYPE(\*SCS), a Set Text Orientation command is contained in the spooled file.
- For DEVTYPE(\*AFPDS), a map coded font-1 (MCF-1) structured field that has the character rotation parameter set to hex 8700 is contained in the spooled file.

Valid values are Y (yes) or N (no). Set this field to N for all other printer device types.

**DBCS characters per inch.** The number of double-byte characters to be printed per inch.

The possible values are:

-1        \*CPI: One-half of the characters-per-inch value.  
-2        \*CONDENSED: 20 double-byte characters per 3 inches.  
5         5 characters per inch.  
6         6 characters per inch.  
10        10 characters per inch.

This field should be -1 when the double-byte character set field is \*NO.

**DBCS-coded font library name.** The name of the library containing the DBCS coded font used to print this spooled file.

The possible values are:

\*CURLIB        The current library is searched for the DBCS-coded font.  
\*LIBL          The library list is used to locate the DBCS-coded font.  
Library name   This library is searched for the DBCS-coded font.

This field should only be specified on a DBCS system, and only for spooled files that will be created with a device type of \*SCS or that will contain i5/OS-created \*AFPDS. See i5/OS-created \*AFPDS field (page 238).

**DBCS-coded font name.** Specifies the name of the DBCS-coded font used to print DBCS data on printers configured as AFP(\*YES).

The possible values are:

\*SYSVAL            The DBCS-coded font specified in the system value is used.  
DBCS-coded font    The name of the DBCS-coded font being used.  
name

This field should only be specified on a DBCS system, and only for spooled files that will be created with a device type of \*SCS or that will contain system-created \*AFPDS. See the i5/OS-created \*AFPDS field (page 238).

**DBCS-coded font point size.** The point size in which this spooled file's DBCS characters (defined by the DBCS-coded font field) should be printed.

The possible values are:

000.0            \*NONE: The DBCS-coded font does not have a point size.  
point size        The point size of the DBCS-coded font. The number must be between 000.1 and 999.9.

This field is ignored if the DBCS-coded font is specified as \*SYSVAL.

This field should only be specified on a DBCS system, and only for spooled files that will be created with a device type of \*SCS or that will contain system-created \*AFPDS. See the i5/OS-created \*AFPDS field (page 238).

**DBCS data.** Whether this spooled file contains double-byte character set (DBCS) data, UCS-2 data, or both. The options are \*YES or \*NO. The other DBCS fields should also be set appropriately.

**DBCS extension characters.** Whether the system is to use the extension character processing function for DBCS data.

The possible values are:

\*YES            The system processes DBCS extension characters.  
\*NO            The system does not process DBCS extension characters; it prints extension characters as the undefined character.

This field should be \*NO when the DBCS data field is \*NO.

**DBCS shift-out shift-in (SO/SI) spacing.** The presentation of shift-out and shift-in characters when printed.

The possible values are:

\*YES            Shift-out and shift-in characters occupy one space.  
\*NO            Shift-out and shift-in characters occupy no space.  
\*RIGHT        Shift-out characters occupy no space and shift-in characters occupy two spaces.

This field should be \*NO when the DBCS data field is \*NO.

**DDS.** Whether the file was constructed with DDS.

Valid values are Y (yes) or N (no). This field should be set to N when the file contains data that has not been spooled before.

**Decimal format.** Whether the decimal format that is used is from the printer file or from the job attribute when the spooled file is generated. The possible values are \*FILE and \*JOB. \*FILE indicates that the decimal format stored in the printer file when the printer file was created is used. \*JOB indicates that the decimal format from the DECFMT job attribute is used.

**Define characters.** Whether the spooled file defines or redefines unique print characters.

For DEVTYPE(\*SCS), a Load Alternate Characters command is contained in the spooled file.

Valid values are Y (yes) or N (no). Set this field to N for all other printer device types.

**Device file library name.** The name of the library that contains the device file. The device file name must also be specified.

**Device file name.** The name of the device file used to create the spooled file. The device file library name must also be specified.

**Device type.** The type of device file for which this file is intended. The possible values are PRINTER, DISKETTE, or TAPE.

**Document name.** The name of the document that was the source of the spooled file. This field is used by the OfficeVision program and must be blank for other spooled files. If the document name is specified, the folder name must also be specified.

**Double-wide characters.** Whether the spooled file prints everything twice as wide as normal.

**Drawer change.** Whether the printer drawer is changed within the spooled file.

- For DEVTYPE(\*SCS), a Page Presentation Media command is contained in the spooled file.
- For DEVTYPE(\*IPDS), an Execute Order Home State-Select Input Media Source command is in the spooled file.
- For DEVTYPE(\*AFPDS), a media map and an invoke media map are in the spooled file.

Valid values are Y (yes) or N (no). Set this field to N for all other printer device types.

**Edge-stitch number of staples.** Specifies the number of staples that are to be applied along the finishing margin.

Valid values are:

0            \*DEVD. Use the device default.  
Specify this value if:

- You specify the offset of each staple in the edge-stitch staple offset list. The printer will automatically determine the number of staples to be used.
- You specify zero (\*DEVD) for number of staple offset entries field. The printer will use its default value for both the number of staples to be used and the offset of each staple.

1-122        Number of staples to be used in the edge-stitch. The number of edge-stitch staple offset entries field must be zero (\*DEVD).

An edge-stitch or saddle-stitch consists of one or more staples driven into the media along a finishing margin. Refer to your printer's documentation for information about whether this field is supported and what values for this field are supported. If this field is not supported, specify a value of \*DEV D.

**Edge-stitch reference edge.** Specifies the edge of the paper to be used as a reference in determining where the edge stitching is to be located.

Valid values are:

- 0 \*NONE. An edge stitch is not specified.
- 1 \*DEV D. The default reference edge used by the printer device is specified.
- 2 \*BOT. The bottom edge of the paper is specified.
- 3 \*RIGHT. The right-hand edge of the paper is specified.
- 4 \*TOP. The top edge of the paper is specified.
- 5 \*LEFT. The left-hand edge of the paper is specified.

An edge-stitch or saddle-stitch consists of one or more staples driven into the media along a finishing margin. Refer to your printer's documentation for information about whether this field is supported and what values for this field are supported. If this field is not supported, specify a value of \*DEV D.

**Edge-stitch staple offset.** The offset from the beginning of this format to the list of the edge-stitch staple offsets section (for more details, see "Edge-Stitch Staple Offset Entry" on page 352).

An edge-stitch or saddle-stitch consists of one or more staples driven into the media along a finishing margin. Refer to your printer's documentation for information about whether this field is supported and what values for this field are supported. If this field is not supported, specify a value of \*DEV D.

**Ending page.** The page at which printing is to end for the spooled file.

0 or 2147483647 indicates the last page. To print the entire spooled file, set this value to 0, 2147483647, or the last page number.

**Exchange type.** The exchange type of the diskette file.

The possible values are:

- \*STD The system allows the exchange type based on the diskette type and sector size.
- \*BASIC The basic exchange type.
- \*H The H exchange type.
- \*I The I exchange type.

**Expiration date.** The date the file will be eligible for removal from the system by the Delete Expired Spooled Files (DLTEXPSPLF) command and blank if the file will not expire. The spooled file will expire at 23:59:59, local system time on the date specified. The date is in the CYYMMDD format as follows:

- C Century, where 0 indicates years 19xx and 1 indicates years 20xx.
- YY Year
- MM Month
- DD Day

**Extended code page.** Whether the extended code page changes within the file.

- For DEVTYPE(\*SCS), a Set CGCS through GCID is contained in the spooled file.
- For DEVTYPE(\*IPDS), a Load Font Equivalence command is inserted into the Load Font Equivalence table.

- For DEVTYPE(\*AFPDS), a map code font-2 (MCF-2) structured field is contained in the spooled file.

Valid values are Y (yes) or N (no).

**FFT emphasis.** Whether the spooled file contains text that is to appear darker than the surrounding text.

For DEVTYPE(\*SCS), Begin Emphasis and End Emphasis commands are contained in the spooled file.

Valid values are Y (yes) or N (no). Set this field to N for all other printer device types.

**Field outlining.** Whether the spooled file outlines fields of data with boxes. If it does, the file must be printed on a printer that supports field outlining.

For DEVTYPE(\*SCS), a Define Grid Line command is contained in the spooled file.

Valid values are Y (yes) or N (no). Set this field to N for all other printer device types.

**File available.** The time this spooled file becomes available to an output device for processing.

The possible values are:

- |          |   |
|----------|---|
| *IMMED   | The spooled file is available as soon as it is opened.                    |
| *FILEEND | The spooled file is available as soon as it is closed.                    |
| *JOBEND  | The spooled file is available when the job that created it has completed. |

When \*JOBEND is specified and the spooled file is being created for another user, the value is changed to \*FILEEND because the job it is assigned to is the special system job QPRTJOB.

When a value is not one of the three listed, the default used is \*FILEEND. In this case the default is not taken from the device file specified.

**File label identifier.** The diskette label used when the system last saved the object.

**File open.** Whether the spooled file is still open when the fields are retrieved. If fields retrieved from an open spooled file are used to create a new spooled file, the create operation fails. Changing this flag to N for a spooled file that is open could cause unpredictable results in a newly created spooled file.

**File stopped on page boundary.** Whether the spooled file stops on a page boundary.

Valid values are Y (yes) or N (no). If the spooled file has not been sent to a printer, set the field to N.

**Final form feed.** Whether the Final Form Feed command is in the printer file.

Valid values are Y (yes) or N (no).

**Final form text.** Whether this spooled file contains various functions that are supported on letter-quality printers.

- For DEVTYPE(\*SCS), a Document Content Architecture (DCA) command is contained in the spooled file. DCA commands include:
  - Required New Line
  - Required Form Feed
  - Indent Tab
  - Set Presentation Page Size
  - Set Horizontal Margins

- Set Vertical Margins
- Release Left Margin
- Set Line Spacing
- Set Single Line Distance
- Justify Text Field Format
- Set Justify Mode
- Set Horizontal Tab Stops
- Set Indent Level
- Set Exception Action
- Set Presentation Color
- Set Spacing Variable
- Begin and End Overstrike
- Begin and End Underscore
- Bell
- Switch
- Repeat
- Tab
- Backspace
- Unit Backspace
- Substitute
- Word Underscore

Valid values are Y (yes) or N (no). Set this field to N for all other printer device types.

**Fold records.** Whether all positions in a record are displayed.

The possible values are:

- \*YES       Records whose length exceeds the page width flow to the following line.
- \*NO        Records whose length exceeds the page width are truncated.

**Folder name.** The name of the folder that contains the document name. This field is used by the OfficeVision program and must be blank for other spooled files. If the folder name is specified, the document name must also be specified.

**Font.** Whether the spooled file uses multiple fonts.

- For DEVTYPE(\*SCS), a Set Font Global command is in the spooled file.
- For DEVTYPE(\*IPDS), a Load Font Equivalence command is contained in the Load Font Equivalence table.
- For DEVTYPE(\*AFPDS), a map coded font-1 (MCF-1) or map coded font-2 (MCF-2) structured field is in the spooled file.

Valid values are Y (yes) or N (no). Set this field to N for all other printer device types.

**Font equivalence array.** The data portion of the Load Font Equivalence (LFE) command for intelligent printer data streams (IPDS). The current maximum is 48. However, for future expansion, the array returns 72 sixteen-character font equivalences. To support that further expansion, the 1153rd character of the variable is set to 1 when more than 72 entries are available. The format of this entry is described in the *IPDS Reference*.

**Font resolution for formatting.** The resolution Print Services Facility/400 (PSF/400) should use to print the spooled file when printing to a multiple resolution printer and either of the following conditions exist:

- The information is not specified in the spooled file data stream.
- The font used in the spooled file could not be found.

The possible values are:

*DEV	PSF/400 is to use the value specified on the font resolution for formatting parameter in the PSF configuration object. The PSF configuration object is specified in the printer device description.
*SEARCH	PSF/400 is to search for a font that will work. The search method is outlined in the Printing topic.
240	PSF/400 is to use only fonts that are 240 pels per inch.
300	PSF/400 is to use only fonts that are 300 pels per inch.

**Form definition library name.** The name of the library that contains the form definition. The special value \*LIBL should be specified when the form definition name field is specified as \*DEV.

This field should only be specified for spooled files with device types of \*USERASCII, \*LINE, \*AFPDSL, \*SCS, \*IPDS, or \*AFPDS if the spooled file was not created on a System i platform. See i5/OS-created \*AFPDS field (page 238).

If this field is not specified for one of these valid types, a default value is used.

**Form definition name.** The name of the form definition to use for printing the spooled file.

The possible values are:

*DEV	The form definition in the printer device description is used. This is the default value.
*IN	The form definition defined in the spooled file data stream is used.
*INL	The form definition defined in the spooled file data stream is used. If a form definition is not found, the form definition in the printer device description is used.
FIDFLT	The form definition defined in the spooled file data stream is used. FIDFLT is the in-line form definition built by the i5/OS operating system for spooled files of printer device type *AFPDSL. When the field i5/OS-created AFPDSL is set to 'Y', the form definition library name field must be set to blanks.
<i>form definition name</i>	The name of the form definition.

This field should only be specified for spooled files with device types of \*USERASCII, \*LINE, \*AFPDSL, \*SCS, \*IPDS, or \*AFPDS if the spooled file was not created on a System i platform. See i5/OS-created \*AFPDSL field (page 238).

If this field is not specified for one of these valid types, a default value is used.

**Form feed.** The form feed attachment used by the printer device file.

The possible values are:

*DEV	The forms are fed into the printer as described in the device description.
*CONT	Continuous forms are used by the printer. The tractor-feed attachment must be put on the printer if this value is specified.
*CUT	Single-cut sheets are used by the printer. Each sheet is manually loaded. The forms alignment message is not sent for cut sheets.
*AUTOCUT	Single-cut sheets are semiautomatically fed into the printer. The sheet-feed attachment must be put on the printer if this value is specified. The forms alignment message is not sent for cut sheets.

\*CONT2           Secondary continuous forms are used by the printer. The secondary tractor-feed attachment must be put on the printer if this value is specified.

**Form type.** The type of form loaded in the printer to print the spooled file.

**Format name.** The name of the format of the fields. The only accepted value is SPLA0200.

**Front margin offset across.** For the front side of a piece of paper it, specifies, in either inches or centimeters (specified in the unit of measure (UOM) parameter), how far in from the left side of the page printing starts.

The possible values are:

0-57.79           The offset in inches or centimeters, according to the unit of measure field. The maximum offset is 57.79 centimeters or 22.75 inches.

This field should only be specified for spooled files with device types of \*LINE, \*AFPDSLIN, or \*AFPDS if the spooled file was created on a System i platform. See i5/OS-created \*AFPDS field (page 238).

If this field is not specified for one of these valid types, a default value is used.

**Front margin offset down.** For the front side of a piece of paper, it specifies, in either inches or centimeters (specified in the unit of measure (UOM) parameter), how far down from the top of the page printing starts.

The possible values are:

-2                \*DEVD. For printers configured AFP(\*YES), no print border is used for front margin offsets across and down. Otherwise, the offset values are 0.

0-57.79           The offset in inches or centimeters, according to the unit of measure field. The maximum offset is 57.79 centimeters or 22.75 inches.

This field should only be specified for spooled files with device types of \*LINE, \*AFPDSLIN, or \*AFPDS if the spooled file was created on a System i platform. See i5/OS-created \*AFPDS field (page 238).

If this field is not specified for one of these valid types, a default value is used.

**Front overlay library name.** The name of the library containing the front overlay.

The possible values are:

\*CURLIB           The current library for the job locates the front overlay.

\*LIBL             The library list locates the front overlay.

library name      This library is searched for the front overlay.

This field should only be specified for spooled files with device types of \*SCS, \*IPDS, or \*AFPDS if the spooled file was created on a System i platform. See i5/OS-created \*AFPDS field (page 238).

If this field is not specified for one of these valid types, a default value is used.

**Front overlay name.** The name of the **front overlay** (the material that prints on the front side of each page).

The possible values are:

*\*NONE*                The file does not use the front overlay.  
*front overlay name* The name of the front overlay.

If an overlay name is specified, the front overlay library name must also be specified.

This field should only be specified for spooled files with device types of \*SCS, \*IPDS, or \*AFPDS if the spooled file was created on a System i platform. See i5/OS-created \*AFPDS field (page 238).

If this field is not specified for one of these valid types, a default value is used.

**Front overlay offset across.** The offset across from the point of origin where the overlay is printed.

The possible values are:

0-57.79            The offset in inches or centimeters, according to the unit of measure field. The maximum offset is 57.79 centimeters or 22.75 inches.

This field should only be specified for spooled files with device types of \*SCS, \*IPDS, or \*AFPDS if the spooled file was created on a System i platform. See i5/OS-created \*AFPDS field (page 238).

If this field is not specified for one of these valid types, a default value is used.

**Front overlay offset down.** The offset down from the point of origin where the overlay is printed.

The possible values are:

0-57.79            The offset in inches or centimeters, according to the unit of measure field. The maximum offset is 57.79 centimeters or 22.75 inches.

This field should only be specified for spooled files with device types of \*SCS, \*IPDS, or \*AFPDS if the spooled file was created on a System i platform. See i5/OS-created \*AFPDS field (page 238).

If this field is not specified for one of these valid types, a default value is used.

**Graphic character set.** The set of graphic characters used when printing the spooled file. For \*DEVDP, the system gets the graphic character set from the printer device description. Valid values are 1 through 32767 and \*DEVDP.

When a numeric value is specified, it is represented as a character string that is right-justified with leading zeros.

**Graphics.** Whether the spooled file contains graphics commands in its data stream.

Valid values are Y (yes) or N (no).

**Graphics error actions.** Whether the file contains graphic error action commands.

For SCS files, the file contains one or more Set Graphic Error Action commands.

Valid values are Y (yes) or N (no). Set this field to N for all other printer device types.

**Graphics token.** The printer type on which the graphics in the spooled file can be printed.

The possible values are:

4214	Data stream contains graphics for a 4214 printer.
4234	Data stream contains graphics for a 4234 printer.
522X	Data stream contains graphics for a 522x printer.
IPDS	Data stream contains graphics for an IPDS printer.

**Group level index tags.** For DEVTYPE(\*AFPDS), whether there are index tags at the group level. Set the field to N for all other printer device types.

**Height of drawer 1.** The height in inches of the paper in drawer 1. This field is set to packed 0 when the spooled file is being created as opposed to being copied. All other drawer fields should also be set to packed 0.

**Height of drawer 2.** The height in inches of the paper in drawer 2. This field is set to packed 0 when the spooled file is being created as opposed to being copied. All other drawer fields should also be set to packed 0.

**Highlight.** Whether the spooled file contains text that is to appear darker than the surrounding text.

- For DEVTYPE(\*SCS), data is overprinted to get the bold effect.
- For DEVTYPE(\*IPDS), a Load Font Equivalence command is contained in the Load Font Equivalence table, which has the bold field set on.
- For DEVTYPE(\*AFPDS), a map coded font-2 (MCF-2) structured field that has the font weight class parameter set to hex 07 is in the spooled file.

Valid values are Y (yes) or N (no). Set this field to N for all other printer device types.

**Hold file before written.** Whether the spooled file is to be held.

The possible values are:

*YES	The file is held.
*NO	The file is not held.

**Internal job identifier.** The internal identifier for the job. Only the i5/OS APIs use this identifier, not any other interface on the system. The identifier is not valid following an initial program load (IPL). If you attempt to use it after an IPL, an exception occurs.

**Internal spooled file identifier.** The value used as input to other programs to improve the performance of locating the spooled file on the system. Only the i5/OS APIs use this identifier, not any other interface on the system. The identifier is not valid following an initial program load (IPL) or if the spooled file has been moved between auxiliary storage pools (ASPs). If you attempt to use an invalid internal spooled file identifier, an exception occurs.

**Internet print protocol job identifier.** The offset from the beginning of this format to the Internet print protocol spooled file attributes section.

**IPDS pass-through.** Specifies whether IPDS pass-through is done for the spooled file.

Valid values are:

0	*DEV. The value specified for the IPDS pass-through (IPDSPASTHR) parameter in the PSF configuration object will be used. The PSF configuration object is specified in the printer device description.
1	*NO. No IPDS pass-through is done.

2 \*YES. IPDS pass-through is to be done if the spooled file is eligible for IPDS pass-through.

Not all SCS or IPDS spooled files are eligible for IPDS pass-through. They may contain special functions that require transform to AFPDS for correct printing. If you specify IPDS pass-through on the printer file, only those spooled files eligible for IPDS pass-through bypass the extra transforms. Those spooled files not eligible for IPDS pass-through will still undergo the transforms to AFPDS and back to IPDS.

IPDS pass-through will not be valid for all PSF/400 supported printers. Any printer (or attachment) that does not support resident fonts cannot support IPDS pass-through. This is because the resident font references in the data stream must be mapped to host fonts that are downloaded to the printer. All IBM® IPDS printers, except for the following, can be supported with IPDS pass-through: 3820, 3825, 3827, 3828, 3829, 3831, 3835, 3900-001 and any printer attached using Print Services Facility™ for the OS/2® distributed print function.

**IPDS transparent data.** Whether the file contains data from System/36 Programming Request for Price Quotations (PRPQs).

Valid values are Y (yes) or N (no). This field should be set to N when creating a spooled file as opposed to copying an existing spooled file.

**Job character ID specified.** Whether the graphic character set and code page of the spooled file is taken from the coded character set identifier (CCSID) of the job. The possible values are Y or N.

**Job name.** The name of the job that owns the spooled file.

**Job number.** The number of the job that owns the spooled file.

**Job system name.** The name of the system where the job that created the spooled file ran.

**Justification.** The percentage that the output is right-justified. The values are 100, 50, or 0.

**Last page printed.** The number of the last printed page in the file if printing ended before the job completed processing.

**Length of each user-defined option entry.** The length, in bytes, of each user-defined option entry.

**Length of page.** The length of a page in rows and columns (\*ROWCOL) or in inches and centimeters (\*UOM).

If this field is specified and the measurement method used is \*ROWCOL, this value should be the same as that specified on the page length field found in this format.

If this field is specified and the measurement method is \*UOM, the actual distance specified must be equivalent to that specified on the page length field found in this format.

If the measurement method is \*ROWCOL, the page length value must be from 1 through 255. If the measurement method is \*UOM, the page length value must be between 0 and 57.79.

**Length of edge-stitch staple offset entry.** The size of each entry in the list. This should always be set to 8 when the number of staple offset entries is not zero.

An edge-stitch or saddle-stitch consists of one or more staples driven into the media along a finishing margin. Refer to your printer's documentation for information about whether this field is supported and what values for this field are supported. If this field is not supported, specify a value of \*DEVD.

**Length of library list entry.** If you need resources to be used with the new spooled file, they must be in libraries contained in your library list at spooled file creation time.

**Length of saddle-stitch staple offset entry.** The size of each entry in the list. This should always be set to 8 when the number of staple offset entries is not zero.

An edge-stitch or saddle-stitch consists of one or more staples driven into the media along a finishing margin. Refer to your printer's documentation for information about whether this field is supported and what values for this field are supported. If this field is not supported, specify a value of \*DEVD.

**Length of user resource library list entry.** The size of each entry in the list. This should always be set to 10 when the user resource library list number is not zero.

**Lengths present.** Whether the 8-byte length information is present in the print text data buffers for format SPLF0200. This field is valid when the printer device type (DEVTYPE) field is \*AFPDS, \*AFPDSLIN, or \*LINE.

Valid values are Y (yes) or N (no). It must be set to Y for \*AFPDSLIN and \*LINE printer device types. \*AFPDS data may or may not have the length information. This field should be set to N for all other printer device types.

**Line spacing.** How a spooled file's line data records are spaced when printed. This information is necessary only when the printer device type (DEVTYPE) parameter on the printer file is \*LINE or \*AFPDSLIN.

The possible values are:

*SINGLE	The printed output is single spaced. If the data contains control characters, they are ignored.
*DOUBLE	The printed output is double spaced. If the data contains control characters, they are ignored.
*TRIPLE	The printed output is triple spaced. If the data contains control characters, they are ignored.
*CTLCHAR	The control characters in the data determine the line spacing. This value should be used when the printer device type is not *LINE or *AFPDSLIN.

**Lines per inch.** The number (in tenths) of lines per vertical inch defined in the printer file. The value 40 indicates 4 lines per inch. Valid lines per inch include 4, 6, 8, 9, and 12. Lines per inch of 3 and 7.5 are valid for double-byte character set devices.

**Lines per inch changes.** Whether the lines per inch (lpi) changes within the spooled file.

- For DEVTYPE(\*SCS), a Set Single Line Density command is contained in the spooled file.
- For DEVTYPE(\*IPDS), a Load Page Descriptor command is contained in the spooled file, which specifies the line-per-inch value
- For DEVTYPE(\*AFPDS), a presentation text description (PTD) structured field that specifies the baseline increment is contained in the spooled file.

Valid values are Y (yes) or N (no). Set this field to N for all other printer device types.

**Lines-per-inch (lpi) value not supported.**

**Maximum forms width.** The maximum width of printer file forms in character positions.

**Maximum records.** The maximum number of records allowed in the spooled file when it is opened. Valid values are 1 through 999999. A special value of 0 indicates \*NOMAX.

**Maximum spooled data record size.** The length of the largest record in the spooled file. For DEVTYPE \*LINE, \*AFPDSLIN, and \*AFPDS spooled files, this length is the length of the largest record in the spooled file. For all other printer device types (DEVTYPE), this length is the same as the spooled file buffer size.

**Measurement method.** The measurement method used for the length of page and width of page fields.

The possible values are:

\*ROWCOL        Uses rows and columns as the units of measure.  
\*UOM            Uses the value specified on the unit of measurement parameter (UOM). UOM is either inches (\*INCH) or centimeters (\*CM).

**Multiple up (pages per side).** The number of logical pages that print on each side of each physical page when this spooled file prints. The possible values are 1, 2, 3, and 4.

This field should only be specified for spooled files with device types of \*SCS, \*IPDS, or \*AFPDS if the spooled file was created on a System i platform. See i5/OS-created \*AFPDS field (page 238).

If this field is not specified for one of these valid types, a default value is used.

**Net ID where file originated.**

**Number of buffers.** The number of buffers in the spooled file.

**Number of edge-stitch staple offset entries.** The number of staple offset entries.

Valid values are:

0                \*DEV. Use the device default. Specify this value if either of the following is specified:  
• A value (1-122) for the edge-stitch number of staples field. The device default for the spacing of each staple will be used.  
• Zero (\*DEV) for the edge-stitch number of staples field. The device default for the spacing of each staple is used.  
1-122           The number of staple offset entries. The edge-stitch number of staples field must be set to \*DEV.

This field specifies the placement of staples along the finishing margin in either inches or centimeters (specified in the unit of measure field). The finishing margin can be thought of as an imaginary line parallel to the edge of the paper where the staples will be placed. The position of the finishing margin relative to the physical edge is specified in the offset from edge-stitch reference edge field.

If you want the staples placed along the left or right edge of the paper, specify the offset of the first staple by measuring from the point where the finishing margin intersects the bottom edge of the paper to where you want the center of the staple to be. For subsequent staples, specify the offset by measuring from the same point (not from the previous staple).

If you want the staples placed along the top or bottom edge of the paper, specify the offset of the first staple by measuring from the point where the finishing margin intersects the left edge of the paper to where you want the center of the staple to be. For subsequent staples, specify the offset by measuring from the same point (not from the previous staple). The possible values for each staple offset are:

0-57.79        The offset in inches or centimeters, according to the Unit of measure (page 246) value. The maximum offset is 57.79 centimeters or 22.75 inches.

An edge-stitch or saddle-stitch consists of one or more staples driven into the media along a finishing margin. Refer to your printer's documentation for information about whether this field is supported and what values for this field are supported. If this field is not supported, specify a value of \*DEV D.

**Number of font array entries.** The number of font equivalence array entries used. The valid values for the field are 0 through 48, where 0 indicates that there are no entries in the array. The field is valid only for spooled files that have the printer device type field equal to \*IPDS. It should be set to zero for all other printer device types.

**Number of library list entries.** If you need resources to be used with the new spooled file, they must be in libraries contained in your library list at spooled file creation time.

**Number of resource library entries.** The number of entries in the resource library array used.

**Number of saddle-stitch staple offset entries.** The number of staple offset entries.

Valid values are:

- 0            \*DEV D. Use the device default. Specify this value either of the following is specified:
- A value for the saddle-stitch number of staples field, which indicates that you want to specify the number of staples to be used. The device default for the spacing of each staple will be used.
  - Zero (\*DEV D) for saddle-stitch number of staples field. The device default for the spacing of each staple is used.
- 1-122        The number of staple offset entries. The saddle-stitch number of staples field must be set to \*DEV D.

This field specifies the placement of staples along the finishing margin in either inches or centimeters (specified in the unit of measure field). The finishing margin can be thought of as an imaginary line parallel to the edge of the paper where the staples will be placed. The position of the finishing margin is the center of the paper relative to the physical edge.

If you want the staples placed parallel to the left edge of the paper, specify the offset of the first staple by measuring from the point where the finishing margin intersects the bottom edge of the paper to where you want the center of the staple to be. For subsequent staples, specify the offset by measuring from the same point (not from the previous staple).

If you want the staples placed parallel to the top edge of the paper, specify the offset of the first staple by measuring from the point where the finishing margin intersects the left edge of the paper to where you want the center of the staple to be. For subsequent staples, specify the offset by measuring from the same point (not from the previous staple).

The possible values for each staple offset are:

- 0-57.79      The offset in inches or centimeters, according to the Unit of measure (page 246) value. The maximum offset is 57.79 centimeters or 22.75 inches.

An edge-stitch or saddle-stitch consists of one or more staples driven into the media along a finishing margin. Refer to your printer's documentation for information about whether this field is supported and what values for this field are supported. If this field is not supported, specify a value of \*DEV D.

**Number of separators.** The number of file separator pages placed at the beginning of each copy. Valid values are 0 through 9, with 0 meaning no separator pages.

**Number of user-defined options returned.** The number of user-defined option entries. Each entry contains one user-defined option.

Valid values are:

- 0 No user-defined options are specified.
- 1-4 The number of user-defined options specified.

**Number of user resource library list entries.** The number of user resource library list entries.

Valid values are:

Each entry contains one user resource library.

- 1 \*DEV.D. The user resource library list in the PSF configuration object specified in the printer device description is to be used to search for AFP resources. If there is no PSF configuration object, then the spooled file's library list is to be used to search for AFP resources.
- 2 \*JOB.LIB.L. The spooled file's library list will be used to search for AFP resources.
- 3 \*CURLIB. The current library for the job that created the spool file is used to search for AFP resources. If no library is specified as the current library for the job, then library QGPL is used.
- 0 \*NONE. There is no user resource library list to be used to search for AFP resources.
- 1-4 Number of user resource libraries.

**OfficeVision.** Whether the spooled file is generated by the OfficeVision licensed program.

Valid values are Y (yes) or N (no). This field should be set to N when creating a spooled file as opposed to copying an existing spooled file.

**Offset from edge-stitch reference edge.** Specifies, in either inches or centimeters (specified in the units of measure field), how far from the edge-stitch reference edge that the edge-stitch is to be placed.

The possible values are:

- 1.0 \*DEV.D. The default edge-stitch offset of the printer device is specified.
- 0-57.79 The offset in inches or centimeters, according to the Unit of measure (page 246) value. The maximum offset is 57.79 centimeters or 22.75 inches.

An edge-stitch or saddle-stitch consists of one or more staples driven into the media along a finishing margin. Refer to your printer's documentation for information about whether this field is supported and what values for this field are supported. If this field is not supported, specify a value of \*DEV.D.

**Offset to edge-stitch staple offset list.** The offset to the information about each staple offset entry. Each entry contains the staple offset for that staple.

**Offset to IPP attributes.** The offset to the IPP specific spooled file attributes. This value is set to 0 if the spooled file was not create through IPP.

**Offset to library list.** If you need resources to be used with the new spooled file, they must be in libraries contained in your library list at spooled file creation time.

**Offset to saddle-stitch staple offset list.** The offset from the beginning of this format to the list of the saddle-stitch staple offsets section (for more details, see "Saddle-Stitch Staple Offset Entry" on page 353).

An edge-stitch or saddle-stitch consists of one or more staples driven into the media along a finishing margin. Refer to your printer's documentation for information about whether this field is supported and what values for this field are supported. If this field is not supported, specify a value of \*DEV.D.

**Offset to Save and Restore spooled file attributes.** The offset to the save and restore specific spooled file attributes.

**Offset to user-defined options.** The offset from the beginning of this format to the list of the user-defined options section (for more details, see “User-Defined Option Entry” on page 352).

**Offset to user resource library list.** The offset from the beginning of this format to the list of user resource library list section (for more details, see “User Resource Library List Entry” on page 352).

**i5/OS-created AFPDS.** Whether the spooled file contains AFPDS data created on i5/OS. Valid values are Y (yes) or N (no).

**Output bin.** The output bin the printer uses for the printed output.

The possible values are:

0	*DEV D: The default output bin of the printer is used.
Output bin	The number of the output bin specified. The number must be between 1 and 65535.

**Output priority.** The priority of the spooled file. The priority ranges from 1 (highest) to 9 (lowest). The priority specified must be within the limits of the owner of the spooled file. The limit can be found by using the Retrieve User Information (QSYRUSRI) API or the Retrieve User Profile (RTVUSRPRF) command, specifying the PTYLM T parameter.

**Output queue library name.** The name of the library that contains the output queue. The OUTQLIB parameter on the RTVUSRPRF can be used to retrieve the library name. A blank library name defaults to \*LIBL.

**Output queue name.** The name of the output queue where the new spooled file is placed. If the output queue specified in the user profile for the owner of the spooled file is to be used, the Retrieve User Profile (RTVUSRPRF) command or the Retrieve User Profile (RTVUSRPRF) API can be used, specifying the OUTQ parameter. A blank output queue name defaults to the printer file.

**Overflow line number.** The last line to be printed before the data being printed overflows to the next page. Valid values are 1 through 255. This value cannot be greater than the value specified in the page length field.

**Page count estimated.** Whether the total pages given is an estimate.

Valid values are Y (yes) or N (no).

The total page count is estimated in the following situations:

- If the spooled file device type is \*AFPDSL I N E.
- If the spooled file device type is \*L I N E and a page definition was used.
- If the spooled file device type is \*AFPDS and the spooled file was not created on a System i platform.
- If the spooled file device type is \*USERASC I I.

**Page definition library name.** The name of the library in which the page definition resides. This information is necessary only when the printer device type (DEVTYPE) parameter is \*L I N E, \*AFPDSL I N E, or \*USERASC I I.

**Page definition name.** The name of the page definition to use for the spooled file. This information is used only when the printer device type (DEVTYPE) parameter is \*LINE, \*AFPDSLIN, or \*USERASCII. The special value of \*NONE indicates no page definition. When a page definition name is specified, the page definition library name must also be specified.

**Page groups.** For DEVTYPE(\*AFPDS), whether there are page groups in the spooled file. Set the field to N for all other printer device types.

**Page length.** The page length (in lines per page) used by the spooled file specified in the device file field. The valid range is row 1 through 255. The value should not exceed the actual length of the page.

**Page level index tags.** For DEVTYPE(\*AFPDS), whether there are index tags at the page level. Set the field to N for all other printer device types.

**Page or record being written.** The page number or record number currently being written. The page number may be lower or higher than the page number actually being printed because of buffering done by the system. The page number shown may be zero if:

- The printer file is routed to a diskette unit.
- The writer is currently printing job or file separators for the file.

**Page position.** Whether page positioning errors are reported.

Valid values are Y (yes) or N (no).

**Page rotate.** Whether the spooled file changes the page rotation to be used within the file.

- For DEVTYPE(\*SCS), a Set Text Orientation command is contained in the spooled file.
- For DEVTYPE(\*IPDS), a Load Page Descriptor command is contained in the spooled file. See the *IPDS Reference* manual, S544-3417, for detailed information about rotating text in a spooled file. This manual is available from the IBM Publications Center .
- For DEVTYPE(\*AFPDS), a presentation text descriptor (PTD) structured field that specifies the page rotation is contained in the spooled file.

Valid values are Y (yes) or N (no). Set this field to N for all other printer device types.

**Page rotation.** The degree of rotation of the text on the page, with respect to the way the form is loaded into the printer.

The possible values are:

- 1 \*AUTO: Computer output reduction is done automatically if the output is too large to fit on the form, regardless of the print quality.
- 2 \*DEV: Page rotation is obtained from the printer device description.
- 3 \*COR: Output created for a form 13.2 inches wide by 11.0 inches long is adjusted to print on a form 11.0 inches wide by 8.5 inches long.
- 0 No rotation is done. Printing starts at the edge of the paper loaded into the printer first and is parallel to that edge.
- 90 Text is rotated 90 degrees clockwise from the 0-degree writing position.
- 180 Text is rotated 180 degrees clockwise from the 0-degree writing position.
- 270 Text is rotated 270 degrees clockwise from the 0-degree writing position.

**Page width.** The page width (in characters per printed line) used by the spooled file specified in the device file field. The valid range is column 1 through 378, although some printers have a page width less than 378. The value should not exceed the actual width of the page used.

**Point size.** The point size in which this spooled file's characters (defined by the printer font field) should be printed. Valid values are 000.1 through 999.9. This field is ignored if the Printer font is specified as 0.0 (\*DEVVD).

**Print fidelity.** The kind of error handling that is performed when printing spooled files with a printer device type of \*AFPDS.

The possible values are:

*CONTENT	The printing overrides errors in the data stream and continues printing with the printers best quality based on the content fidelity.
*ABSOLUTE	The spooled file is printed exactly as intended. Printing is stopped if an error is encountered in the data stream.

This field should be set to \*CONTENT for all non-AFPDS files.

**Print on both sides (duplex).** How the information prints.

The possible values are:

*FORMDF	A user-specified form definition determines how printing is done. This value is used only when the printer device type (on the printer file) is specified as *LINE, *AFPDS, *SCS, *IPDS, or *AFPDSLIN.
*NO	The print appears on only one side of a piece of paper.
*YES	The print is on both sides of the page with the top of each page the same for both sides.
*TUMBLE	The print is on both sides of the page with the top of one printed page at the opposite end from the top of the other printed page.

**Print quality.** The print quality used when printing the data.

The possible values are:

*STD	Standard
*DRAFT	Draft
*NLQ	Near-letter quality
*FASTDRAFT	Pages printed per minute are greater than if *DRAFT is specified.

**Print text.** The text that is printed on separator pages. Print text in the print data stream is not changed.

**Printer device type.** The type of data stream used to represent the spooled file.

The possible values are:

*AFPDS	Advanced Function Presentation data stream
*AFPDSLIN	AFPDS data mixed with 1403 line data
*IPDS	Intelligent printer data stream
*LINE	1403 line data
*SCS	Systems Network Architecture (SNA) character stream
*USERACSI	ASCII data

The data stream of the spooled file must correspond to the printer device type.

**Printer font.** The printer font that is used.

The possible values are:

<i>*DEV</i>	The font defined in the printer device description.
<i>*CPI</i>	A font that has the pitch specified by the characters-per-inch field.
<i>identifier</i>	The numeric font identifier that is used with the printer device file. The font specified should be valid for the printer.

**Program that opened file library name.** The name of the library that contains the program that opened the file.

**Program that opened file name.** The name of the program that opened the spooled file.

**Record format.** The format of the records in the spooled file.

The possible values are:

<i>*FIXED</i>	All records in the spooled file are the same size; that is, the original length specified in the print data for each record is the same for all records.
<i>*VARIABLE</i>	Records in the spooled file vary in size; that is, the original length specified in the print data for each record has at least one different value for records in the spooled file.  For DEVTYPE *USERASCII, *IPDS, and *SCS without ASCII data files are all fixed format. *SCS files with ASCII data are variable length. For DEVTYPE *AFPDS, *AFPDSLIN, and *LINE, data files can be either fixed or variable.

**Record format name present in data stream.** Whether a record format name exists in columns 1 - 10 of a line data record or columns 2 - 11 if the spooled file contains carriage control characters.

For DEVTYPE(\*LINE and \*AFPDSLIN), record formats in a page definition could be used. The possible values are Y or N.

Valid values are Y (yes) or N (no). Set this field to N for all other printer device types.

**Record length.** The length of spooled file records. If the field shows -1 (the special value \*RCDFMT), this is an externally defined spooled file, and the length is included in the spooled file definition. The length includes the extra length of 1 for carriage control if it exists.

**Reduce output.** The manner in which multiple logical pages print on each side of a physical page.

The possible values are:

<i>*TEXT</i>	A smaller font and lpi are used to print multiple logical pages on each side of a physical page. The smaller font and lpi preserves the contents of each logical page. The possible values for the multiple up parameter are 1, 2, and 4.
<i>*NONE</i>	The printer prints multiple logical pages on each side of each physical page. There are no changes to the fonts and lpi values used in each logical page. The possible values for the multiple up parameter are 1, 2, 3, and 4.

**Replace unprintable characters.** Whether characters that cannot be printed are to be replaced with another character.

Valid values are Y (yes) or N (no).

**Replacement character.** The character that replaces any unprintable characters. This field has a value if the replace unprintable characters field specifies Y.

**Reserved.** The field is reserved.

**Resource library array.** If you need resources to be used with the new spooled file, they must be in libraries contained in your library list at spooled file creation time.

**Restart printing.** The number of the page where printing is restarted.

The possible values are:

- 1           \*STRPAGE: The page specified in the starting page field.
- 2           \*ENDPAGE: The last page.
- restart page*   The page number where printing is restarted. This number should be within the page range specified by the spooled file starting and ending page field.

**Saddle-stitch number of staples.** The number of staples that are to be applied along the finishing margin.

Valid values are:

- 0            \*DEVD. Use the device default.
- Specify this value if:
- You specify the offset of each staple in the saddle-stitch staple offset list. The printer will automatically determine the number of staples to be used.
  - You specify zero (\*DEVD) for number of staple offset entries field. The printer will use its default value for both the number of staples to be used and the offset of each staple.
- 1-122       Number of staples to be used in the saddle-stitch. The number of saddle-stitch staple offset entries field must be zero (\*DEVD).

An edge-stitch or saddle-stitch consists of one or more staples driven into the media along a finishing margin. Refer to your printer's documentation for information about whether this field is supported and what values for this field are supported. If this field is not supported, specify a value of \*DEVD.

**Saddle-stitch reference edge.** The edge of the paper to be used as a reference in determining where the saddle stitching is to be located.

Valid values are:

- 0           \*NONE. Saddle stitch is not specified.
- 1           \*DEVD. The default reference edge used by the printer device is specified.
- 2           \*TOP. The top edge of the paper is specified.
- 3           \*LEFT. The left-hand edge of the paper is specified.

An edge-stitch or saddle-stitch consists of one or more staples driven into the media along a finishing margin. Refer to your printer's documentation for information about whether this field is supported and what values for this field are supported. If this field is not supported, specify a value of \*DEVD.

**Save file after written.** Whether the spooled file is saved after it is written.

The possible values are:

- \*YES        Save the spooled file.
- \*NO         Do not save the spooled file.

**SCS data.** Whether the spooled file is created with SCS already in the input data.

Valid values are Y (yes) or N (no).

**Set exception.** Whether the spooled file has the SCS Set Exception Action command in the data stream.

Valid values are Y (yes) or N (no).

**Set Line Density command.** Whether the lines per inch (lpi) changes within the spooled file or is specified with the Set Line Density SCS command.

For DEVTYPE(\*SCS), a Set Line Density command is contained in the spooled file.

Valid values are Y (yes) or N (no). Set this field to N for all other printer device types.

**Source drawer.** The drawer to be used when the form feed field is set to \*AUTOCUT.

The possible values are:

- 1-255        The drawer number.
- 1         \*E1, the envelope drawer.
- 2         \*FORMDF, indicating that the file uses a user-specified form definition. This value is used only when the printer device type (DEVTYPE) field has been set to \*LINE, \*AFPDS, or \*AFPDSLIN.

**Spooled file buffer size.** The length of the buffer being written. Sizes supported are 512 and 4079. A buffer size of 512 is recommended for spooled files that have a schedule field of \*IMMED. A buffer size of 4079 is recommended for spooled files that have a schedule field of \*FILEEND or \*JOBEND. If the original spooled file was spooled with a buffer size of 4079, changing the buffer size to 512 causes a function check to occur.

**Spooled file creation authentication method.** The authentication method used when creating the spooled file.

- 0         \*NONE No authentication method was used.
- 1         \*REQUESTER The requesting user was used to authenticate the request.
- 2         \*BASIC The client requested the user to sign on with a user ID and password to authenticate the request.
- 3         \*CERTIFICATE The request was authenticated through the use of certificates.

**Spooled file creation security method.** The security method used when creating the spooled file.

Valid values are:

- 0         \*NONE No security method was used.
- 1         \*SSL3 The SSL3 security method was used.

**Spooled file level.** The level of the spooled file in Version, Release, and Modification level format (VxRxMx). The following is a list of the spooled file levels and the size in bytes required for the SPLFA0200 structure.

The length in bytes required for the different spooled file levels does not include the space required to specify any user-defined options, user resource libraries, edge-stitch staple offsets, or saddle-stitch staple offsets.

- 3320        V2R3M0
- 3446        V3R1M0, V3R2M0, and V3R6M0
- 3745        V3R7M0, V4R1M0, and V4R1M4

3772	V4R2M0
3784	V4R3M0, V4R4M0, and V4R5M0
3816	V5R1M0
3824	V5R2M0 and V5R3M0
3841	➤ V5R4M0 and V6R1M0 ◀

**Spooled file name.** The name of the spooled file whose information is retrieved.

**Spooled file number.** The spooled file number of the specified file.

**Spooled file size.** The size of the spooled file. This number needs to be multiplied by the spooled file multiplier field to get the size of the spooled file in number of bytes. This size is the data stream size plus the "overhead" storage to store the spooled file's data stream.

**Spooled file size multiplier.** The value to multiply the spooled file size by to get the size of the spooled file in bytes.

**Starting page.** The page at which printing is to start for the spooled file.

Special values include:

-1	Use the ending page value.
0	Printing starts on page 1.

**Status.** The status of the file is one of the following values:

*CLOSED	The program completely processed the file, but SCHEDULE(*JOBEND) was specified, and the job that produced the file has not yet finished.
*DEFERRED	The spooled file has been deferred from printing.
*SENDING	The spooled file is being sent or has been sent to a remote system.
*HELD	The file is held.
*MESSAGE	The file has a message requiring a reply or an action.
*OPEN	The file was not completely processed and is not ready for a writer.
*PENDING	The file is pending to be printed.
*PRINTER	The complete file was sent to the printer, but a print complete status was not sent back.
*READY	The file is available to be written.
*SAVED	The file was written and is saved. This file remains saved until it is released.
*WRITING	The writer is writing the file on a diskette or a printer device.

**Subscript.** Whether the spooled file contains subscript.

- For DEVTYPE(\*SCS), a Subscript command is contained in the spooled file.
- For DEVTYPE(\*IPDS and \*AFPDS), a Temporary Baseline Move text control is contained in the spooled file.

Valid values are Y (yes) or N (no). Set this field to N for all other printer device types.

**Superscript.** Whether the spooled file contains superscript.

- For DEVTYPE(\*SCS), a Superscript command is contained in the spooled file.
- For DEVTYPE(\*IPDS and \*AFPDS), a Temporary Baseline Move text control is contained in the spooled file.

Valid values are Y (yes) or N (no). Set this field to N for all other printer device types.

**System where file originated.** The name of the system where the spooled file data originated. If the creating system was a System i product running OS/400® version 2 or earlier, the address of the user that sent the file is displayed.

When the name of the system where this spooled file was created cannot be determined, the receiving system name is used. For example, a file sent using SNDNETSPLF DTAFMT(\*RCDDATA).

**System/36 procedure name.** The name of the procedure running when the spooled file is created.

**System/36 spooled file identifier.** The 6-character identifier assigned to the spooled file.

**System/38 text utility flags.** The flag for advanced functions. This field is set to hexadecimal zeros when a spooled file is being created as opposed to copying an existing file.

**S36 CONTINUE-YES.** Whether the file is created as if the System/36 environment OCL statement "// PRINTER CONTINUE-YES" is in effect, as described under the QUSRSPLA API. The possible values are Y or N. See the corresponding field description under the QUSRSPLA API.

The value of this field in format SPLA0200 should be left undisturbed for most applications. For user-built attributes, a value of N (no) is generally recommended, unless the user has specific reasons for specifying Y (yes).

**Time file opened (created).** » The time, in local job time, that the file was opened in the HHMMSS format where: «

HH	Hour
MM	Minutes
SS	Seconds

**Time writer began processing spooled file.** » The time, in local job time, that a spooling writer started processing the spooled file. «

**Time writer completed processing spooled file.** » The time, in local job time, that a spooling writer completed processing the spooled file. «

**Total copies.** Total copies to be produced for this spooled file.

**Total pages.** The number of pages this printer file contains.

**Total records.** The number of data records. This field is only used when the device type (DEVTYPE) parameter of the printer file is \*AFPDS, \*LINE, or \*AFPDSLIN.

**Transparency.** Whether the SCS Transparency command is used in the spooled file. It is set to N for all other device type files, such as IPDS.

For DEVTYPE(\*SCS), a TRANSPARENT SCS command is contained in the spooled file.

Valid values are Y (yes) or N (no).

**TRC for 1403.** Whether the file contains 1403 line data with table-reference characters (TRC). This field is only used for device types (DEVTYPE) \*LINE and \*AFPDSLIN.

Valid values are Y (yes) or N (no). Set this field to N for all other printer device types.

**Unit of measure.** The unit of measure to use for specifying distances. The unit of measure is used with certain parameters of the printer file. Page definitions and form definitions are examples of these parameters.

The possible values are:

\**CM*           Centimeters  
\**INCH*        Inches

**Unrecognizable data.** Whether the file contains SCS commands that are not valid because of one of the following:

- Command is not recognized
- Command data is not valid
- Command is recognized, but not supported

Valid values are Y (yes) or N (no). Set this field to N for all other printer device types.

**User name.** The user name of the owner of the new spooled file being created. If the user name is the current user, the spooled file is created as part of the current job. If the user name is someone other than the current user, the spooled file is created as part of the special system job QPRTJOB for the user name specified. The current user must have \*SPLCTL authority.

**User object type.** The type of the user-defined object. Possible values follow:

- Data area (\*DTAARA)
- Data queue (\*DTAQ)
- File (\*FILE)
- User index (\*USRIDX)
- Print Services Facility (PSF) configuration object (\*PSFCFG)
- User queue (\*USRQ)
- User space (\*USRSPC)

This field should be set to blanks when the user-defined object name is set to \*NONE.

**User-defined data.** The data defined by the user to be used by user applications or user-specified programs that process spooled files. The special value of \*NONE should be used when there is no user-defined data.

**User-defined file.** Whether the spooled file was created by using an API.

The possible values are:

\**YES*           The spooled file was created using the QSPCRTSP API.  
\**NO*            The spooled file was created by normal system functions.

**User-defined object library name.** The library that contains the user-defined object. This field should be set to blanks when the user-defined object name field is set to \*NONE.

**User-defined object name.** The name of the user-defined object that is used by user applications or user-specified programs that process spooled files. The special value of \*NONE should be used when there is no user-defined object.

**User-defined text.** User-defined text taken from the interactive user profile when the spooled file was created.

The text is not converted because the text may not be presented to a user. It may be processed by a server type job and converted at a later date.

**User-generated data stream.** Whether the data stream has been validated by a system program on the System i platform when the data was spooled, for example, the OfficeVision program.

Valid values are Y (yes) or N (no). This field should be set to Y (data stream not validated) when opening a spooled file that contains data that has not been spooled, for example, OfficeVision spooled files.

**User-specified data.** The 10 characters of user-specified data that describe the file.

**User who created file.** The name of the user profile that created this file. If the spooled file was created on a System i platform running OS/400 version 2 or earlier, the user id that sent the file is displayed.

When the name of the user profile that created this file cannot be determined, the name of the receiving user profile is used. For example, a file sent using SNDNETSPLF DTAFMT(\*RCDDATA).

**Volumes (array).** The diskette volumes used for saving the object. The variable returns a maximum of 10 six-character volumes. The volume IDs begin in character positions 1, 8, 15, 22, 29, 36, 43, 50, 57, and 64. If more than 10 volumes are used, a 1 is returned in the 71st character of the variable; otherwise, the 71st character is blank. The variable is returned blank if the object was saved to a save file, or if it was never saved.

**Width of drawer 1.** The width in inches of the paper in drawer 1. This field is set to packed 0 when the spooled file is being created as opposed to being copied. All other drawer fields should also be set to packed 0.

**Width of drawer 2.** The width in inches of the paper in drawer 2. This field is set to packed 0 when the spooled file is being created as opposed to being copied. All other drawer fields should also be set to packed

**Width of page.** The width of a page in rows and columns (\*ROWCOL) or in inches and centimeters (\*UOM). The width of a page is determined by the value specified on the PAGESIZE parameter of the printer file.

If this field is specified and the measurement method used is \*ROWCOL, this value should be the same as that specified on the page width field found in this format.

If this field is specified and the measurement method is \*UOM, this value must be equivalent to that specified on the page width field found in this format.

If the measurement method is \*ROWCOL, the page width value must be from 1 through 378. If the measurement method is \*UOM, the page width value must be between 0 and 57.79.

**3812 SCS.** Whether the spooled file contains fonts supported only on the 3812 printer.

For DEVTYPE(\*SCS), a Set Coded Font Local command with a global ID of greater than 255 is contained in the spooled file.

Valid values are Y (yes) or N (no). Set this field to N for all other printer device types.

**3812 SCS commands.** Whether the file contains commands that are not valid on the 3812 printer.

- An incorrect Set Character Set Global ID command is found by validity checking the code page with what is valid for the 3812 printer.
- An incorrect Set Character Set Local ID command is found by validity checking the code page with what is valid for the 3812 printer.
- A Set Presentation Color command

Valid values are Y (yes) or N (no). Set this field to N for all other printer device types.

**5A present.** Whether the hex 5A constant is present in all AFPDS data for the file (in format SPLF0200). This field is valid when the printer device type (DEVTYPE) field is \*AFPDS.

Valid values are Y (yes) or N (no). This field should be set to N for all other printer device types.

**5219 commands.** Whether the file contains commands that are not valid on a 5219 printer. The data stream contains one or more of the following SCS commands which, either are not valid on a 5219 or contain parameters that are not valid.

- An incorrect Set Character Set Global ID is found by validity checking the code page with what is valid for the 5219 printer.
- Begin and End Emphasis SCS commands.
- A Set Presentation Page Size command contains a page size value that is not supported.
- A Page Presentation Media command contains a drawer value other than drawer 1 or drawer 2.
- A Set Single Line Distance command contains a distance value that is not supported.
- A Set Single Line Spacing command contains a distance value that is not supported.
- A Justify Text Field Format command contains a value other than 0, 50, or 100 percent.
- A Set Justify Mode Format command contains a value other than 0, 50, or 100 percent.
- A Set Presentation Color command.
- A Begin Overstrike command, which contains an optional CHRID value of other than 0.

Valid values are Y (yes) or N (no).

## Error Messages

Message ID	Error Message Text
CPF24B4 E	Severe error while addressing parameter list.
CPF3CF1 E	Error code parameter not valid.
CPF3C19 E	Error occurred with receiver variable specified.
CPF3C21 E	Format name &1 is not valid.
CPF3C90 E	Literal value cannot be changed.
CPF33DD E	Maximum number of open spooled files exceeded for this job.
CPF33DE E	Size of internal data for opened spooled file exceeds maximum.
CPF33DF E	Internal data area for opened spooled files destroyed.
CPF33E0 E	Incomplete set of attributes provided.
CPF33E1 E	Attributes are for an opened spooled file.
CPF33E2 E	Value &1 for spooled file attribute at offset &2 not valid.
CPF34B1 E	Spooled file print data format not supported.
CPF34B2 E	Spooled file input record length is not valid.
CPF9801 E	Object &2 in library &3 not found.
CPF9838 E	User profile storage limit exceeded.
CPF9845 E	Error occurred while opening file &1.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V2R1

---

## Get Spooled File Data (QSPGETSP) API

Required Parameter Group:

1	Spooled file handle	Input	Binary(4)
2	Qualified user space name	Input	Char(20)
3	Format name	Input	Char(8)
4	Ordinal number of buffer to be read	Input	Binary(4)
5	End of open spooled file	Input	Char(10)
6	Error code	I/O	Char(*)

Default Public Authority: \*USE  
Threadsafe: No

The Get Spooled File Data (QSPGETSP) API gets data from an existing spooled file. The existing spooled file must have been opened previously by the Open Spooled File (QSPOPNSP) API. Data is retrieved from the existing spooled file by buffers (one or more) and stored in a user space. The data in the user space is used as source to the Put Spooled File Data (QSPPUTSP) API. The number of buffers returned in the user space is no greater than the value specified on the number of buffers to get parameter on the Open Spooled File (QSPOPNSP) API.

Depending on the data format requested, the actual number of bytes returned for each buffer may vary from the size of the buffer. Format SPFR0300 may return fewer bytes for each buffer than the actual buffer size. Format SPFR0200 may return more bytes because additional sections, such as the buffer information, general information, and page data section are included. The buffer size is in relation to the print data, not the other information sections.

When creating the user space for the spooled file data, an initial user space size that is calculated by adding the size of the generic header section to the product of the buffer size (currently 4079 or 512) and the number of buffers to get provides adequate space for format SPFR0300. However, this may be too small for format SPFR0200. For format SPFR0200, the size of the user space needed is variable based on the number of pages per buffer. Normally, a good estimate for the user space size needed for SPFR0200 would be the number of buffers multiplied by the buffer size plus 500 bytes. The user space is automatically extended, if necessary, to allow all buffers requested to be stored in the user space.

User spaces can be created using the Create User Space (QUSCRTUS) API. The time it takes to create a user space is decreased if you specify to initialize the space to hexadecimal zeros.

The maximum size of a user space is 16,776,704 bytes. If the data requested exceeds this maximum size, message CPF3CAA is issued. The user space contains accurate information for the number of buffers returned. To continue processing, create another user space and call the QSPGETSP API again with the new user space. Another alternative would be to return fewer buffers on each call to the QSPGETSP API.

**Note:** The size of the buffer (currently 512 or 4079 bytes) is determined by the buffer size of the existing spooled file being worked with. The spooled file buffer size is an attribute of the spooled file that can be returned using format SPLA0200 with the QUSRSPLA API.

## Authorities and Locks

*User Space Authority*  
\*CHANGE

*Library Authority*  
\*EXECUTE

*Output Queue Authority*  
\*USE

*User Space Lock*  
\*EXCLRD

## Required Parameter Group

### Spooled file handle

INPUT; BINARY(4)

The handle returned by the Open Spooled File (QSPOPNSP) API

### Qualified user space name

INPUT; CHAR(20)

The name of the user space that is to receive the buffer of spooled information. The first 10 characters contain the user space name and the second 10 characters contain the name of the library in which the user space is located. The special values allowed for the library name are \*LIBL and \*CURLIB. Both user space name and library name are left-justified. If no library is specified as the current library of the job, QGPL is used.

### Format name

INPUT; CHAR(8)

The format and the content of the information retrieved from each buffer.  
The possible values are:

<i>SPFR0100</i>	Information about the print data stored in the buffer.
<i>SPFR0200</i>	Information about the print data stored in the buffer and the print data.
<i>SPFR0300</i>	Print data for the buffer or multiple buffers.

### Ordinal number of buffer to be read

INPUT; BINARY(4)

The buffer of the spooled file that is read first (or next).

Any number greater than zero is valid. When a specific buffer is requested, only that buffer is returned.

The following special value is supported for this parameter:

-1 Reads the next buffer in the spooled file.

To read sequentially, starting at a specific buffer, the first API call should specify the specific buffer, with subsequent calls of the API specifying the special value -1 (read next). The first API call returns only the specific buffer requested. The subsequent read operations, with -1 specified for the next buffer, return the number of buffers specified on the open operation. When reading an entire spooled file, the special value -1 should be used.

Performance degradation could happen if a specific buffer is always used.

### End of open spooled file

INPUT; CHAR(10)

How to handle the situation where the spooled file has not been closed (by this job or another job) and the requested data has not yet been written by the application program.

The values supported for this parameter are:

*WAIT	The API waits until the requested data is available or until the spooled file is closed. If the spooled file is closed without the requested data becoming available, the action defined by *ERROR is done.
-------	---

\**ERROR* The API returns either an error code or an error message based on what was specified for the error code parameter.

**Error code**

I/O; CHAR(\*)

The structure in which to return error information. For the format of the structure, see Error code parameter.

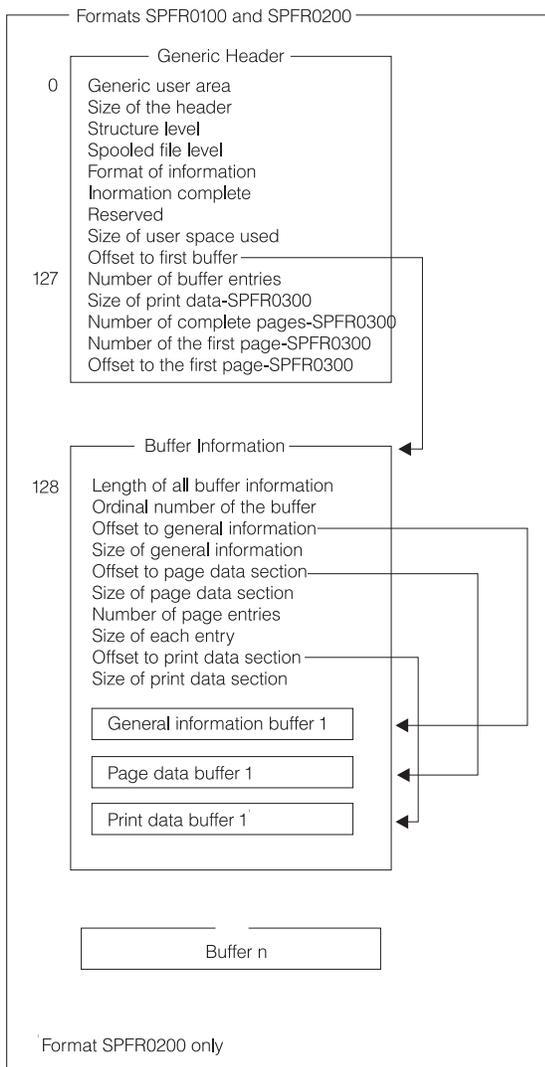
**Format of the User Space**

The organization of the user space is dependent on the format (SPFR0100, SPFR0200, or SPFR0300) used.

Formats SPFR0100 and SPFR0200 (page 251) show the general structure of the user space for formats SPFR0100 and SPFR0200. If SPFR0100 is used, no print data is returned.

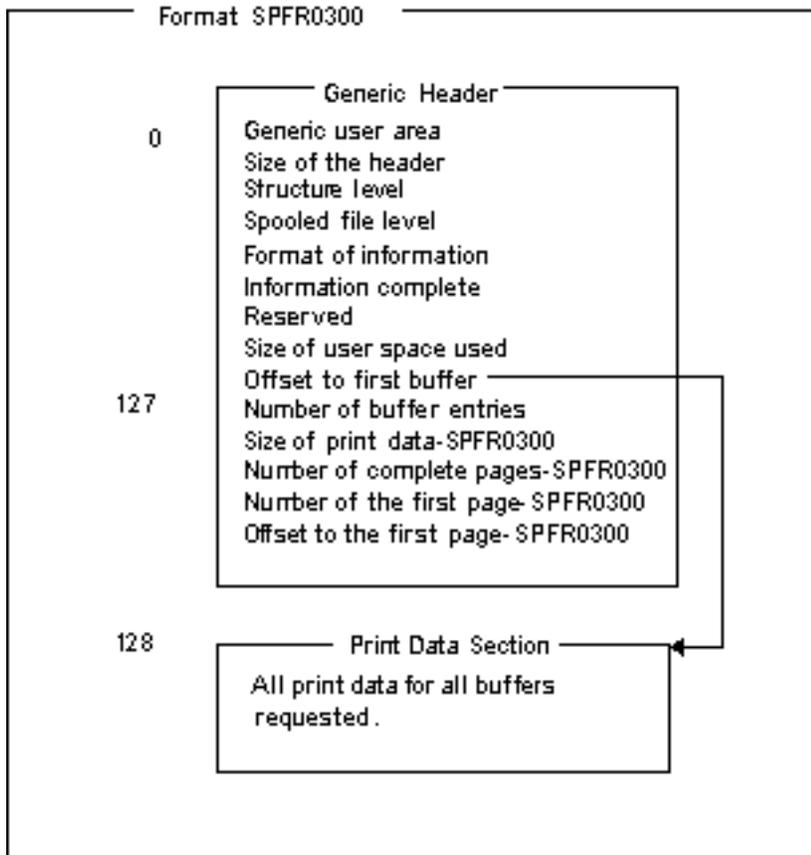
Format SPFR0300 (page 252) shows the structure for format SPFR0300. If SPFR0300 is used, only the print data is returned, and print data for the buffers requested is returned in one print data section.

Offset values are calculated from the beginning of the user space.



RBAFX568-0

## Format SPFR0300



## Generic Header Section

The following table shows the generic header information returned for the SPFR0100, SPFR0200, and SPFR0300 formats.

For more details about the fields, see “Field Descriptions” on page 253.

Offset		Type	Field
Dec	Hex		
0	0	CHAR(64)	Generic user area
64	40	BINARY(4)	Size of header
68	44	CHAR(4)	Structure level
72	48	CHAR(6)	Spooled file level
78	4E	CHAR(8)	Format of the information returned
86	56	CHAR(1)	Information complete indicator
87	57	CHAR(1)	Reserved
88	58	BINARY(4)	Size of user space used
92	5C	BINARY(4)	Offset to first buffer
96	60	BINARY(4)	Number of buffers requested
100	64	BINARY(4)	Number of buffers returned

Offset		Type	Field
Dec	Hex		
104	68	BINARY(4)	Size of print data
108	6C	BINARY(4)	Number of complete pages
112	70	BINARY(4)	Number of first page
116	74	BINARY(4)	Offset to first page
120	78	CHAR(8)	Reserved

## Field Descriptions

**Format of the information returned.** The SPFR0100, SPFR0200, or SPFR0300 format of the data present.

For input to the QSPPUTSP API, this field should contain SPFR0200.

**Generic user area.** Provided for the user's application program.

**Information complete indicator.** Used to indicate if all information requested has been supplied.

- I* Incomplete information. An interruption causes the user space to contain incomplete information about a buffer or buffers. The information for the last buffer is incomplete.
- P* Partial and accurate information. Partial information is returned when the maximum user space was used and not all of the buffers requested were read. Information is complete for the number of buffers returned.
- C* Complete and accurate information. All the buffers requested are read and returned. Complete also applies to the case where the end of file was reached on a closed spooled file, but the number of buffers requested was not reached.

For input to the QSPPUTSP API, this field should contain a value of C or P.

**Number of buffers requested.** The number of buffer entries requested for each read operation.

This is the number specified on the open operation. However, if a specific buffer is requested by the read operation, this value is only one buffer.

**Number of buffers returned.** The number of buffer entries present in the user space.

As output from the QSPGETSP API, this number may be equal to or less than the number of buffers requested. The number may be less because the end of a closed spooled file is reached or the maximum user space has been used before the requested buffer count has been processed.

As input to the QSPPUTSP API, this is the number of buffer entries present in the user space. If this data is built by the user, rather than returned by the QSPGETSP API, the user should specify the proper number of buffer entries being provided to the QSPPUTSP API.

**Number of complete pages.** The number of complete pages contained in the print data.

This field is applicable to format SPFR0300 only.

**Number of first page.** The number of the first page in the print data section of format SPFR0300. The field is zero when no page is starting in the print data returned or when the format is SPFR0100 and SPFR0200.

**Offset to first buffer.** The location of the data for the first buffer returned. The offset value is from the beginning of the user space. For formats SPFR0100 and SPFR0200, this is the offset to the first buffer information section. For format SPFR0300, this is the offset to the print data section.

**Offset to first page.** The location of the data for the beginning of the first page in the print data section of format SPFR0300. This field is zero if the number of first page field is zero or the format is SPFR0100 or SPFR0200.

**Reserved.** Used to align a 4-byte boundary.

**Size of header.** The size of the header section, excluding the generic user area, of format SPFR0100, SPFR0200, or SPFR0300 in bytes.

**Size of print data.** The size of the print data of format SPFR0300 in bytes. This is zero for formats SPFR0100 and SPFR0200.

**Size of user space used.** The number of bytes used (returned) in the user space. This includes the generic header section, all buffer information sections, all general information sections, all page data sections, and all print data sections that are present in the user space.

**Spooled file level.** The level of the spooled file in Version, Release, Modification level format.

**Structure level.** The level of the structure of the user space. This should always be 0200.

## Buffer Information Section

The following buffer information is returned by the QSPGETSP API. For more details about the fields in the following table, see "Field Descriptions."

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Length of all buffer information
4	4	BINARY(4)	Ordinal number of the buffer
8	8	BINARY(4)	Offset to general information section
12	C	BINARY(4)	Size of general information section
16	10	BINARY(4)	Offset to page data section
20	14	BINARY(4)	Size of page data section
24	18	BINARY(4)	Number of page entries
28	1C	BINARY(4)	Size of page entry
32	20	BINARY(4)	Offset to print data section
36	24	BINARY(4)	Size of print data section

## Field Descriptions

**Length of all buffer information.** The size of all information pertaining to this buffer. This includes the sizes of the buffer information section, general information section, page data section, and print data section.

**Number of page entries.** The number of page entries in the page data section.

**Offset to general information section.** The location of the general information section. The offset value is from the beginning of the user space.

**Offset to page data section.** The location of the page data information section. The offset value is from the beginning of the user space.

**Offset to print data section.** The location of the print data section. The offset value is from the beginning of the user space.

**Ordinal number of the buffer.** The ordinal number of the buffer returned.

**Size of general information section.** The size of the general information section in bytes.

**Size of page data section.** The size of the page data section in bytes.

**Size of page entry.** The size of each page entry.

**Size of print data section.** The size of the print data section in bytes. This is zero (0) for format SPFR0100 because this format does not return this information. For data not previously spooled (creating as opposed to copying) the print data buffer section cannot exceed the size of the spooled file buffer minus the sum of the sizes of the following:

- Constant of 24
- Size of page data section

## General Information Section

The general information section of formats SPFR0100 and SPFR0200 has the following structure.

For more details about the fields in the following table, see “Field Descriptions” on page 256.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Nonblank lines in buffer
4	4	BINARY(4)	Nonblank lines in first page
8	8	BINARY(4)	Buffer number of error information
12	C	BINARY(4)	Offset to error recovery information
16	10	BINARY(4)	Size of print data
20	14	CHAR(10)	State
30	1E	CHAR(1)	Last page continues
31	1F	CHAR(1)	Advanced print function file
32	20	CHAR(1)	LAC command array in buffer
33	21	CHAR(1)	Any buffer had LAC
34	22	CHAR(1)	Error recovery information contains LAC
35	23	CHAR(1)	Error recovery information
36	24	CHAR(1)	Zero pages
37	25	CHAR(1)	Load font
38	26	CHAR(1)	IPDS™ data
39	27	CHAR(5)	Reserved

## Field Descriptions

Format SPFR0200 is the format required by the Put Spooled File Data (QSPPUTSP) API. When a spooled file is being copied, the information is retrieved with the QSPGETSP API. When the spooled file is created by a user application, all information in the format must be accurately provided.

**Advanced print function file.** Whether this spooled file was created by the IBM® Advanced Function Printing™ Utilities licensed program. The value is N if the device type is \*SCS.

**Any buffer had LAC.** Whether any buffer had an SCS Load Alternate Characters (LAC) command since last error recovery information was saved. Specify Y for yes and N for no. This should be set to N for all spooled files with a DEVTYPE other than \*SCS.

**Buffer number of error information.** The number of the buffer containing the error recovery information. This should be zero when there is no buffer containing error recovery information or for all spooled files with a device type other than SCS.

**Error recovery information.** Whether there is error recovery information in this buffer for SCS commands. This error recovery information allows a file to be repositioned and print only some pages of a file, making sure the characters print using the proper code point. Information about the proper code points is stored in LAC commands in the error recovery information. This information is stored for each page that contains SCS commands that can cause characters to print differently based on the code point specified. Specify Y for yes and N for no. This should be set to N for all spooled files with a device type other than SCS.

**Error recovery information contains LAC.** Error recovery information for SCS commands at the end of this page includes the LAC command found later in the buffer. Specify Y for yes and N for no. This should be set to N for all spooled files with a DEVTYPE other than SCS.

**IPDS data.** Whether the buffer contains IPDS data. Specify Y for yes and N for no. Buffers of IPDS data may exist in SCS spooled files.

**LAC command array in buffer.** An LAC command exists in this buffer. Specify Y for yes and N for no. This should be set to N for all spooled files with a DEVTYPE other than SCS.

**Last page continues.** The last page in this buffer continues in the next buffer when the value is Y.

**Load font.** Indicates whether a new load font equivalence entry was added to the table for this buffer. Specify Y for yes and N for no. This should be set to N for all spooled files with a device type other than IPDS.

**Nonblank lines in buffer.** The number of nonblank lines in the buffer.

**Nonblank lines in first page.** The number of nonblank lines in the first page ending in the buffer.

**Offset to error recovery information.** Offset to the error recovery information in this buffer. This field is 0 for all spooled files with a DEVTYPE other than SCS.

**Reserved.** Reserved for byte alignment.

**Size of print data.** The number of bytes of print data.

**State.** The state the buffer ends in if the data is IPDS.

\*HOME Home state

\*PAGE Page state

*GRAPHICS	Graphics
*PAGETRANS	Page transparency
*HOMETRANS	Home transparency
blank	All non-IPDS files

**Zero pages.** Whether there are zero (0) pages in the spooled file. Specify Y for yes and N for no.

## Page Data Section

Each entry of the page data section of format SPFR0100 and SPFR0200 has the following structure. For more details about the fields in the following table, see “Field Descriptions.”

**Note:** One page data section entry is returned for each page that starts in a returned buffer. The number of pages within a buffer may vary depending on their sizes. Therefore, the number of page data section entries may also vary from buffer to buffer.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Text data start
4	4	BINARY(4)	Any data start
8	8	BINARY(4)	Page offset

## Field Descriptions

**Any data start.** The number of the first line where user data can start on this page. This count includes all data to be printed.

**Page offset.** The location of the start of this page. The offset value is from the beginning of the print data.

**Text data start.** Number of the first line where user data can start on this page. This count includes text only.

## Print Data Section

The print data section of format SPFR0200 and SPFR0300 has the following structure. For more details about the fields in the following tables, see “Field Descriptions” on page 259.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(*)	Print data

Print data for format SPFR0300 contains all the print data for all of the buffers returned.

The format of the print text data is dependent on the printer device type.

For printer device types of \*AFPDS, \*AFPDSLIN, and \*LINE, the data for a given format may span buffers. The first buffer returned will have one of the following formats:

- Printer device type of \*LINE with the control character field set to \*NONE, the TRC field set to N, and the record format name present in the data stream set to N.

Original length	Trimmed length	Line number	DBCS flag	RSRVD	Line data
-----------------	----------------	-------------	-----------	-------	-----------

- Printer device type of \*LINE with the control character field set to \*NONE, the TRC field set to N, and the record format name present in the data stream set to Y.

Original length	Trimmed length	Line number	DBCS flag	RSRVD	Record format	Line data
-----------------	----------------	-------------	-----------	-------	---------------	-----------

- Printer device type of \*LINE with the control character field set to \*FCFC, the TRC field set to N, and the record format name present in the data stream set to N.

Original length	Trimmed length	Line number	DBCS flag	RSRVD	CC	Line data
-----------------	----------------	-------------	-----------	-------	----	-----------

- Printer device type of \*LINE with the control character field set to \*FCFC, the TRC field set to N, and the record format name present in the data stream set to Y.

Original length	Trimmed length	Line number	DBCS flag	RSRVD	CC	Record format	Line data
-----------------	----------------	-------------	-----------	-------	----	---------------	-----------

- Printer device type of \*LINE with the control character field set to \*NONE, the TRC field set to Y, and the record format name present in the data stream set to N.

Original length	Trimmed length	Line number	DBCS flag	RSRVD	TRC	Line data
-----------------	----------------	-------------	-----------	-------	-----	-----------

- Printer device type of \*LINE with the control character field set to \*FCFC, the TRC field set to Y, and the record format name present in the data stream set to N.

Original length	Trimmed length	Line number	DBCS flag	RSRVD	CC	TRC	Line data
-----------------	----------------	-------------	-----------	-------	----	-----	-----------

- Printer device type of \*AFPDS with the control character field set to \*FCFC. The TRC field does not apply and should be set to N. The original length field is Y, and the 5A field is Y.

Original length	Trimmed length	Line number	DBCS flag	RSRVD	5A	AFPDS data
-----------------	----------------	-------------	-----------	-------	----	------------

- Printer device type of \*AFPDS with the control character field set to \*FCFC. The TRC field does not apply and should be set to N. The original length field is Y, and the 5A attribute is N.

Original length	Trimmed length	Line number	DBCS flag	RSRVD	AFPDS data
-----------------	----------------	-------------	-----------	-------	------------

- Printer device type of \*AFPDS with the control character field set to \*FCFC. The TRC field does not apply and should be set to N. The original length field is N, and the 5A field is Y.

5A	AFPDS data
----	------------

- Printer device type of \*AFPDS with the control character field set to \*FCFC. The TRC field does not apply and should be set to N. The original length field is N, and the 5A field is N.

AFPDS data

- Printer device type of \*AFPDSLIN with the control character field set to \*FCFC, the original length field set to Y, the 5A field set to Y, and the TRC field set to N.  
\*AFPDSLIN spooled files must have control characters and lengths.

Original length	Trimmed length	Line number	DBCS flag	RSRVD	CC	Line data
Original length	Trimmed length	Line number	DBCS flag	RSRVD	5A	AFPDS data

- Printer device type of \*AFPDSLIN with the control character field set to \*FCFC, the original length field set to Y, the 5A field set to Y, and the TRC field set to Y.  
\*AFPDSLIN spooled files must have control characters and lengths.

Original length	Trimmed length	Line number	DBCS flag	RSRVD	CC	TRC	Line data
Original length	Trimmed length	Line number	DBCS flag	RSRVD	5A	AFPDS data	

For a printer device type of \*IPDS, the buffer has the following format:

IPDS data

For information about the buffer format of printer device type SCS, as well as other device types, see the Printing topic collections.

## Field Descriptions

**AFPDS data.** CHAR(\*): A variable length field describing the structured fields that reside in the data stream. See the *Data Stream and Object Architectures: MO:DCA™ Reference* manual, SC31-6802, for a

description of these structured fields. This manual is available from the IBM Publications Center  .  
When the length fields are present, only one structured field can be contained in the AFPDS data.

**CC.** CHAR(1): Carriage control.

All codes are in hexadecimal notation. The carriage control characters control writing, spacing, and skipping operations as the data is being formatted. It may or may not precede the line data. See the Printing topic collections for further information about the valid carriage control codes.

When control characters exist, they can be one of two types: American National Standard printer control characters or machine control characters.

**DBCS flag.** CHAR(1): For DBCS files, whether this page starts in IGC mode. This is determined by the leftmost bit of this field. This should be zero for all lines except the first one on the page.

**IPDS data.** CHAR(\*): A variable length field describing the structured fields that reside in the data stream. See the *IPDS Reference* manual, S544-3417, for a description of these structures. This manual is available from the IBM Publications Center  .

**Line data.** CHAR(\*): A variable length character field composed of the actual characters to print.

**Line number.** BINARY(2): The line number this page starts on. This should be zero for all lines except the first one on the page.

**Original length.** BINARY(2): The original length of the record before trailing blanks were trimmed. This length does not include the first 8 bytes of the record.

**Record format.** CHAR(10): The name of the record format found in the page definition specified on the spooled file's page definition attribute that describes how the record of line data is to be formatted.

**RSRVD.** CHAR(1): Reserved field.

**TRC.** CHAR(1): A table-reference character.

All codes are in hexadecimal notation. A table-reference character selects a font to be used when printing the line. See the Printing topic collections for further information about the valid table reference codes and how they relate to the CHARS attributes and page definitions. If both CC and TRC are present, CC will be first and the TRC character will be second. The absence or presence of these characters is determined by the control character attribute and TRC attribute.

**Trimmed length.** BINARY(2): The actual or trimmed length of the record. This length does not include the first 8 bytes of the record.

**5A.** CHAR(1): A special hexadecimal constant used to indicate that a structured field follows.

## Restrictions for Print Data Section

Some restrictions apply as to where these record formats can appear in the buffer.

- For \*LINE data records it is required that the original length, trimmed length, control character, TRC character, and record format, if they exist, are not split across buffers.
- For \*AFPDS data records it is required that the original length, trimmed length, hex 5A control character, and the next 3 bytes of data are not split across a buffer boundary.

For other data types the print text format corresponds to the architecture for that data stream.

## Error Messages

Message ID	Error Message Text
CPF24B4 E	Severe error while addressing parameter list.
CPF3CAA E	List is too large for user space &1.
CPF3CF1 E	Error code parameter not valid.
CPF3C21 E	Format name &1 is not valid.
CPF3C90 E	Literal value cannot be changed.
CPF33DF E	Internal data area for opened spooled files destroyed.
CPF33D2 E	Spooled file handle not valid.
CPF33D3 E	Value &1 not valid for buffer to read parameter.
CPF33D4 E	Value &1 not valid for end of open file parameter.
CPF33D5 E	Spooled file not opened for operation requested.
CPF33D6 E	Buffer &1 not available.
CPF33D7 E	Requested number of buffers not returned.
CPF3330 E	Necessary resource not available.
CPF811A E	User space &4 in &9 damaged.
CPF9801 E	Object &2 in library &3 not found.
CPF9802 E	Not authorized to object &2 in &3.
CPF9803 E	Cannot allocate object &2 in library &3.
CPF9807 E	One or more libraries in library list deleted.
CPF9808 E	Cannot allocate one or more libraries on library list.
CPF9810 E	Library &1 not found.
CPF9820 E	Not authorized to use library &1.
CPF9830 E	Cannot assign library &1.

Message ID	Error Message Text
CPF9846 E	Error while processing file &1 in library &2.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V2R1

Top | "Print APIs," on page 1 | APIs by category

---

## List Spooled File AFPDS Resources (QGSLRSC) API

### Required Parameter Group:

1	Qualified user space name	Input	Char(20)
2	Format name	Input	Char(8)
3	Qualified job name	Input	Char(26)
4	Internal job identifier	Input	Char(16)
5	Internal spooled file identifier	Input	Char(16)
6	Spooled file name	Input	Char(10)
7	Spooled file number	Input	Binary(4)
8	Resource name	Input	Char(*)
9	Resource object type	Input	Char(10)
10	Pel density	Input	Char(1)
11	Error code	I/O	Char(*)

### Optional Parameter Group 1:

12	Job system name	Input	Char(8)
13	Spooled file create date	Input	Char(7)
14	Spooled file create time	Input	Char(6)

### Optional Parameter Group 2:

15	Length of resource name	Input	Binary(4)
16	CCSID of resource name	Input	Binary(4)

Default Public Authority: \*USE  
Threadsafe: No

The List Spooled File AFPDS Resources (QGSLRSC) API generates a list of the AFPDS resources found in the specified spooled file or returns the location of a specified AFPDS resource. Either option returns the information in a user space.

## How QGSLRSC Resolves to the AFPDS Resource Libraries

If a spooled file is specified, the spooled file attribute, user resource library list, is used to resolve the location of the AFPDS resources. The following is a list of supported values of the user resource library list and how QGSLRSC uses them:

- \*JOBLIBL. The library list of the job that created the spooled file is searched.
- \*NONE. No libraries associated with the spooled file are searched.
- \*CURLIB. The current library of the job that created the spooled file is searched.
- \*DEVD. Same as \*JOBLIBL.
- 1-4 libraries. The specified libraries are searched.

If the AFPDS resource is not found at this point, then the font libraries QFNTCPL, QFNT01-QFNT19, and QFNT61-QFNT69 are searched.

If \*NONE was specified for the spooled file, then only the font libraries QFNTCPL, QFNT01-QFNT19, and QFNT61-QFNT69 are searched.

## How QGSLRSC resolves to integrated file system imbedded object resources

When an imbedded object resource is specified or encountered within the data stream of the spooled file, the following method is used to resolve to the resource.

- If the resource is qualified with a path name, that path is searched.
- If the resource was not found in a specified path, and the path was the current working directory or the resource was not path qualified, the paths specified in the system-level environment variable QIBM\_AFP\_RESOURCES\_PATH is searched.
- If the resource is not found, and the spooled file resides on an independent disk pool, the */independent-disk-pool-name/QIBM/UserData/OS400/AFPresources* directory on the independent disk pool is searched.
- If the resource is not found, or the spooled file resides on \*SYSBAS, the */QIBM/UserData/OS400/AFPresources* directory on the system ASP is searched.

## How QGSLRSC Resolves to integrated file system font resources

When a font is specified or encountered within the data stream of the spooled file, the Resource Access Table in following directories are searched first. If the font is not found in one of the Resource Access Tables or if there are no Resource access Tables, then the files in the directories are searched.

- */QIBM/UserData/OS400/Fonts/TTFonts*
- */QIBM/ProdData/OS400/Fonts/TTFonts*

## Authorities and Locks

*User Space Authority*

\*CHANGE

*Library Authority*

\*EXECUTE

*Output Queue Authority*

\*USE

*Resource Authority*

\*READ/\*R

*Directory Authority*

\*X

## Spooled File Authority

The requester is authorized to the spooled file if one or more of the following conditions are met:

- The requester is the owner of the spooled file.
- The requester has \*READ authority to the queue on which the spooled file resides, and the queue is specified as DSPDTA(\*YES).
- The requester has \*SPLCTL authority.
- The requester has \*JOBCTL authority, and the output queue on which the spooled file resides is specified as OPRCTL(\*YES).
- The requester has ownership of the queue on which the spooled file resides, and the queue is specified as AUTCHK(\*OWNER).
- The requester has read, add, and delete authorities to the output queue on which the spooled files resides, and the output queue is specified as AUTCHK(\*DTAAUT).

*Spoiled File Lock*

\*LSRD

*User Space Lock*

\*EXCLRD

## Required Parameter Group

### Qualified user space name

INPUT; CHAR(20)

The user space that receives the generated list and the library in which it is located. The first 10 characters contain the user space name and the second 10 characters contain the library name.

You can use these special values for the library name:

\*CURLIB            The job's current library

\*LIBL              The library list

### Format name

INPUT; CHAR(8)

The format of the AFPDS resource list being returned. You must specify one of the following:

*"Format LRSC0100" on page 270*    Contains the name, object type, and location of the AFPDS resource.

*"Format LRSC0200" on page 271*    Contains the library list of the specified spooled file where the AFPDS resources can be found.

*"Format LRSC0300" on page 271*    Searches the library list for the specified resource using the spooled file library list and the font libraries.

*"Format LRSC0400" on page 272*    Contains the name, object type, and location of the AFPDS resource if the resource is a library based object. Contains the file name and path if the resource is not a library based object and is in the integrated file system.

*"Format LRSC0500" on page 277*    Searches the integrated file system directories for the specified resource.

### Qualified job name

INPUT; CHAR(26)

The qualified job name of the job that owns the spooled file.

The qualified job name has three parts:

*job name*            CHAR(10)

A specific job name, or one of the following special values:

\*                    Only the job in which this program is running. The rest of the job name parameter must be blank.

\*INT                The internal job identifier used to locate the job. The user name and job number must be set to blank.

*user name*           CHAR(10)

A specific user profile name, or blank when the job name is asterisk (\*) or \*INT.

*job number*          CHAR(6)

A specific job number, or blank when the job name is asterisk (\*) or \*INT.

### Internal job identifier

INPUT; CHAR(16)

The internal job identifier for the job that owns the spooled file whose AFPDS resources are to be listed, or blank when job name is not \*INT. See “Retrieve Spooled File Attributes (QUSRSPLA) API” on page 337 for valid combinations of parameters for selecting the spooled file.

### Internal spooled file identifier

INPUT; CHAR(16)

The internal spooled file identifier for the spooled file whose AFPDS resources are to be listed, or blank when the spooled file name is not \*INT. See “Retrieve Spooled File Attributes (QUSRSPLA) API” on page 337 for valid combinations of parameters for selecting the spooled file.

### Spooled file name

INPUT; CHAR(10)

The name of the spooled file for which the list of AFPDS resources is returned. You can use these special values for the name:

- |       |  |
|-------|--|
| *INT  | The internal spooled file identifier is used to locate the spooled file.   |
| *NONE | Search the font libraries for the specified resource. When this is specified, the input parameters qualified job name, internal job identifier, internal spooled file identifier, spooled file number, job system name, spooled file create date, and spooled file create time are ignored. This value is only valid with format LRSC0300 or LRSC0500. |

### Spooled file number

INPUT; BINARY(4)

The unique number of the spooled file. The valid range is 1 through 999999.

The following special values are supported for this parameter:

- |    |   |
|----|---|
| 0  | Only one spooled file from the job has the specified file name, so the number of the spooled file is not necessary.   |
| -1 | This uses the highest-numbered spooled file with the specified file name.   |
| -2 | The spooled file number is not used to determine which spooled file to process. Use this value when you want the Job system name parameter or the Spooled file create date and Spooled file create time parameters to take precedence over the spooled file number when selecting a spooled file.<br>This parameter must contain a valid value even if the value for the spooled file name parameter is *INT. |

### Resource name

INPUT; CHAR(\*)

The name of the AFPDS resource to be returned in the list of AFPDS resources.

You must specify one of the following:

- |               |  |
|---------------|--|
| *ALL          | Returns a list of all AFPDS resources found in the specified spooled file. This is valid only for format LRSC0100 or LRSC0400.                     |
| resource name | Returns information for the specified AFPDS resource if it is found in the specified spooled file. The resource name must be one of the following: |

- An i5/OS<sup>®</sup> object name.
- An integrated file system file name. The file name must include the proper extension. The file name must not include a path name.
- An external descriptive font name. An example of an external descriptive font name would be *Times New Roman*. An external descriptive font name can only be specified for format LRSC0500.

For format LRSC0300, the library information is returned for the specified resource.

For format LRSC0500, the path information is returned for the specified resource.

This parameter is ignored when using format LRSC0200.

### Resource object type

INPUT; CHAR(10)

The type of resource object. When the resource object type is \*PATH, optional parameters Length of resource name and CCSID of resource name must be specified.

The valid values for this parameter are:

*ALL	Search for all AFPDS resources in the spooled file. This value is valid for format LRSC0100 and LRSC0400 only.
*FNTRSC	Font resource
*FORMDF	Form definition
*OVL	Overlay
*PAGSEG	Page segment
*PAGDFN	Page definition
*PATH	The resource resides in the integrated file system. This value is valid for formats LRSC0400 and LRSC0500 only. This parameter is ignored when using format LRSC0200.

### Pel density

INPUT; CHAR(1)

The pel density of the font resources that are to be returned in the list of AFPDS resources. This parameter is only used for AFPDS resources of type \*FNTRSC with the attribute FNTCHRSET.

The valid values for this parameter:

1	240 pel density.
2	300 pel density.
3	Adobe Type 1 outline font.
4	Adobe Type 0 composite outline font.
5	600 pel density.

### Error code

I/O; CHAR(\*)

The structure in which to return error information. For the format of the structure, see Error code parameter.

## Optional Parameter Group 1

### Job system name

INPUT; CHAR(8)

The name of the system where the job that created the spooled file ran or blank when the spooled file name is \*INT. This parameter is considered after the job name, user name, job number, spooled file name, and spooled file number parameter requirements have been met.

The following special values are supported for this parameter:

*ONLY	There is one job with the specified job name, user name, job number, spooled file name, spooled file number, spooled file create date, and spooled file create time.
*CURRENT	The job on the current system with the specified job name, user name, job number, spooled file name, spooled file number, spooled file create date, and spooled file create time is used.
*ANY	The job system name is not considered when selecting a spooled file. Use this value when you want the Spooled file create date and Spooled file create time parameters to take precedence over the job system name when selecting a spooled file. If this parameter is omitted, the API assumes blanks when the spooled file name is *INT. When spooled file name is not *INT, the API assumes *ONLY.

### Spooled file create date

INPUT; CHAR(7)

» The date, based on local job time, that the spooled file was created on the system or blank when the spooled file name is \*INT. « This parameter is considered after the job name, user name, job number, spooled file name, spooled file number, and job system name parameter requirements have been met. The date must be in the CYYMMDD format or one of the following special values:

- \*ONLY There is only one spooled file with the specified job name, user name, job number, spooled file name, spooled file number, and job system name.
- \*LAST The spooled file with the latest date and time which also has the specified job name, user name, job number, spooled file name, spooled file number, and job system name is used.

The date format CYYMMDD is defined as follows:

C	Century, where 0 indicates years 19xx and 1 indicates years 20xx.
YY	Year
MM	Month
DD	Day

If this parameter is omitted, the API assumes blanks when the spooled file name is \*INT. When spooled file name is not \*INT, the API assumes \*ONLY.

### Spooled file create time

INPUT; CHAR(6)

» The time, in local job time, that the spooled file was created on the system or blank when the spooled file name is \*INT. « This parameter must be set to blanks when special values \*LAST or \*ONLY are used for parameter Spooled file create date. This parameter must have a value set if a date is specified for parameter Spooled file create date. This parameter is considered after the job name, user name, job number, spooled file name, spooled file number, job system name, and spooled file create date parameter requirements have been met. The time must be in the HHMMSS format or one of the following special values:

- \*ONLY There is only one spooled file with the specified job name, user name, job number, spooled file name, spooled file number, job system name, and create date.
- \*LAST The spooled file with the latest time which also has the specified job name, user name, job number, spooled file name, spooled file number, job system name, and create date is used.

The time format HHMMSS is defined as follows:

HH	Hour
MM	Minutes
SS	Seconds

If this parameter is omitted, the API assumes blanks.

## Optional Parameter Group 2

### Length of resource name

INPUT; Binary(4)

The length of the resource name in number of bytes. Valid values range from 1 to 250. A value larger than 250 will result in an error.

If this parameter is omitted, the API assumes a fixed length of 10 and that the resource name is left justified and blank padded.

## CCSID of resource name

INPUT; Binary(4)

The CCSID that the resource name specified in the resource name parameter is encoded in. The CCSID or the following special value:

0 The current job's default CCSID is used.

If this parameter is omitted, the API assumes CCSID 65535.

## Format of the Generated List

The spooled file AFPDS resource list consists of:

- A user area
- A generic header
- An input parameter section
- A header section
- A list data section

When you retrieve list entry information from a user space, you must use the entry size returned in the generic header as a displacement to the next list entry. The size of each entry may be padded at the end. If you do not use the entry size, the result may not be valid. For examples of how to process lists, see Examples: APIs and exit programs.

**Note:** When the entry size returned in the generic header is set to -1, the entries vary in length. To get to the next entry, the entry length field of the LRSC0400 or LRSC0500 formats must be used to determine the displacement to the next list entry.

## Input Parameter Section

For more information about the fields in the following table, see "Field Descriptions" on page 268.

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	User space name specified
10	A	CHAR(10)	User space library name specified
20	14	CHAR(8)	Format name
28	1C	CHAR(26)	Qualified job name specified
54	36	CHAR(16)	Internal job identifier
70	46	CHAR(16)	Internal spooled file identifier
86	56	CHAR(10)	Spooled file name specified
96	60	BINARY(4)	Spooled file number specified
100	64	CHAR(10)	Resource name specified
110	6E	CHAR(10)	Resource object type specified
120	78	CHAR(1)	Pel density specified
121	79	CHAR(8)	Job system name specified
129	81	CHAR(7)	Spooled file create date specified
136	88	CHAR(6)	Spooled file create time specified
142	8E	Char(2)	Reserved

Offset		Type	Field
Dec	Hex		
144	90	Binary(4)	Displacement to resource name specified
148	94	Binary(4)	Length of resource name specified
152	98	Binary(4)	CCSID of resource name specified
		Char(*)	Long resource name specified

## Field Descriptions

**CCSID of resource name specified.** The CCSID of the resource name parameter value specified on the call to the API. This field is set to 65535 when the CCSID of the resource name parameter is not specified.

**Displacement to resource name specified.** The displacement, in number of bytes, from the beginning of the Input parameter section to the long resource name specified. This resource name is the one that was specified in the call to the API. This field is set to 0 if the resource object type specified is not \*PATH.

**Format name.** The name of the format used to list the AFPDS resources in the user space.

**Internal job identifier.** The internal identifier for the job. Only the i5/OS APIs use this identifier, not any other interface on the system. The identifier is not valid following an initial program load (IPL). If you attempt to use it after an IPL, an exception occurs.

**Internal spooled file identifier.** The input value that other programs use to improve the performance of locating the spooled file on the system. Only the spooled file APIs use this identifier, not any other interface on the system. The identifier is not valid following an initial program load (IPL) or if the spooled file has been moved between auxiliary storage pools (ASPs). If you attempt to use an invalid internal spooled file identifier, an exception occurs.

**Job system name specified.** The name of the system, specified on the call to the API, where the job specified by the qualified job name ran.

**Length of resource name specified.** The length of the resource name parameter value specified on the call to the API. This field is set to 10 when the length of resource name parameter is not specified.

**Long resource name specified.** The name of the AFPDS resource specified in the call to the API.

**Pel density specified.** The pel density specified on the call to the API.

The valid values are:

- 1 240 pel density
- 2 300 pel density
- 3 Adobe Type 1 outline font.
- 4 Adobe Type 0 composite outline font.
- 5 600 pel density.

**Qualified job name specified.** The qualified name of the job specified in the call to the API.

**Resource name specified.** The name of the AFPDS resource specified in the call to the API. This field is blank when the resource object type specified is \*PATH.

**Resource object type specified.** The type of the AFPDS resource specified in the call to the API.

**Spooled file create date specified.** The creation date of the spooled file specified on the call to the API.

**Spooled file create time specified.** The creation time of the spooled file specified on the call to the API.

**Spooled file name specified.** The name of the spooled file specified in the call to the API.

**Spooled file number specified.** The number of the spooled file specified in the call to the API.

**User space library name specified.** The library name of the user space specified in the call to the API.

**User space name specified.** The name of the user space specified in the call to the API.

## Header Section

For more information about the fields in the following table, see “Field Descriptions.”

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	User space name
10	A	CHAR(10)	User space library name
20	14	CHAR(26)	Qualified job name
46	2E	CHAR(10)	Spooled file name
56	38	BINARY(4)	Spooled file number
60	3C	CHAR(8)	Job system name
68	44	CHAR(7)	Spooled file create date
75	4B	CHAR(6)	Spooled file create time
81	51	CHAR(3)	Reserved
84	54	BINARY(4)	CCSID of resource names returned

## Field Descriptions

**CCSID of resource names returned.** The CCSID of the resource names and path names returned for resources located in the integrated file system.

**Job system name.** The name of the system where the job that created the spooled file ran.

**Qualified job name.** The qualified name of the job that owns the spooled files.

The qualified job name has three parts:

<i>job name</i>	CHAR(10) The name of the job that owns the spooled file.
<i>user name</i>	CHAR(10) The name of the user of the job.
<i>job number</i>	CHAR(6) The number of the job that owns the spooled file.

**Spooled file create date.** >> The date, based on local job time, that the spooled file was created on the system. <<

**Spooled file create time.** >> The time, in local job time, that the spooled file was created on the system. <<

**Spooled file name.** The name of the spooled file that the AFPDS resources are to be listed for.

**Spooled file number.** The number of the spooled file that the AFPDS resources are to be listed for.

**User space library name.** The library containing the user space.

**User space name.** The name of the user space that contains the format returned.

## Format LRSC0100

For more information about the fields in the following table, see "Field Descriptions."

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	Resource name
10	A	CHAR(10)	Resource library name
20	14	CHAR(10)	Resource object type
30	1E	CHAR(1)	Where resource was referenced
31	1F	CHAR(1)	Resource availability
32	20	CHAR(10)	Source object
42	2A	CHAR(10)	Source object library
52	34	CHAR(10)	Source object type

## Field Descriptions

**Resource availability.** Whether the resource was found, not found, or the owner of the spooled file was not authorized to the resource or to the resource library.

Possible values are:

- 0 The resource was not available for one of the following reasons:
- The resource could not be found on the system.
  - The user who created the spooled file was not authorized to the resource or to the resource library.
- 1 The resource was found on the system.

**Resource library name.** The name of the library that contains the AFPDS resource. The

possible values are:

- \*INLINE* The resource is defined within the data stream of the spooled file.  
*library\_name* The name of the library that the AFPDS resource was found in.

**Resource name.** The name of the AFPDS resource used in the spooled file.

**Resource object type.** The type of AFPDS resource.

The valid values are:

- \*PAGSEG* Page segment  
*\*PAGDFN* Page definition  
*\*OVL* Overlay

\*FORMDF            Form definition  
 \*FNTRSC            Font resource

**Source object.** The name of the object that the resource was found in.

The possible values are:

\*SPLF                The resource is found within the data stream of the spooled file.  
*object\_name*        The name of the AFPDS resource that the resource object is found in.

**Source object library.** The library that contains the source AFPDS resource object. This field is blank when the source object is \*SPLF.

**Source object type.** The object type of the source AFPDS resource object. This field is blank when the source object is \*SPLF or is one of the following special values:

\*PAGSEG            Page segment  
 \*PAGDFN            Page definition  
 \*OVL                Overlay  
 \*FORMDF            Form definition  
 \*FNTRSC            Font resource

**Where resource was referenced.** Where the resource was located.

Possible values are:

0            The resource was found within the spooled file or another AFPDS resource.  
 1            The resource was found within the spooled file attributes.

## Format LRSC0200

For more information about the fields in the following table, see “Field Descriptions.”

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	Resource library

## Field Descriptions

**Resource library.** A library, from the library list of the specified spooled file, where AFPDS resources can be found.

## Format LRSC0300

For more information about the fields in the following table, see “Field Descriptions” on page 272.

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	Resource name
10	A	CHAR(10)	Resource library name

Offset		Type	Field
Dec	Hex		
20	14	CHAR(10)	Resource object type
30	1E	CHAR(1)	Reserved
31	1F	CHAR(1)	Resource availability

## Field Descriptions

**Resource availability.** Whether the resource was found, not found, or the caller or the owner of the spooled file was not authorized to the resource or to the resource library.

Possible values are:

- 0 The resource was not available for one of the following reasons:
- The resource could not be found on the system.
  - The user who created the spooled file was not authorized to the resource or to the resource library.
  - If \*NONE was specified for the spooled file name, the caller of the API was not authorized to the resource or to the resource library.
- 1 The resource was found on the system.

**Resource library name.** The name of the library that contains the AFPDS resource. The

possible values are:

*library\_name* The name of the library that the AFPDS resource was found in.

**Resource name.** The name of the AFPDS resource used in the spooled file.

**Resource object type.** The type of AFPDS resource.

The valid values are:

\*PAGSEG Page segment  
 \*PAGDFN Page definition  
 \*OVL Overlay  
 \*FORMDF Form definition  
 \*FNTRSC Font resource

## Format LRSC0400

For more information about the fields in the following table, see “Field Descriptions” on page 273.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Length of entry
4	4	BINARY(4)	Displacement to name
8	8	BINARY(4)	Length of name
12	0C	CHAR(8)	Format of name

Offset		Type	Field
Dec	Hex		
		CHAR(*)	Name entry

## Field Descriptions

**Displacement to name.** The displacement, in number of bytes, from the beginning of the entry to the name.

**Format of name.** The format of the name. The possible values are:

*LRSE0100* Name entry is returned in the entry format LRSE0100.

*LRSE0200* Name entry is returned in the entry format LRSE0200

**Length of name.** The length of the name in number of bytes.

**Length of entry.** Length of the entry in number of bytes.

**Name entry.** The resource returned in the format specified in the field Format of name.

## Format LRSE0100

For more information about the fields in the following table, see "Field Descriptions."

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	Resource name
10	A	CHAR(10)	Resource library name
20	14	CHAR(10)	Resource object type
30	1E	CHAR(1)	Where resource was referenced
31	1F	CHAR(1)	Resource availability
32	20	CHAR(10)	Source object
42	2A	CHAR(10)	Source object library
52	34	CHAR(10)	Source object type

## Field Descriptions

**Resource availability.** Whether the resource was found, not found, or the owner of the spooled file was not authorized to the resource or to the resource library.

Possible values are:

0 The resource was not available for one of the following reasons:

- The resource could not be found on the system.
- The user who created the spooled file was not authorized to the resource or to the resource library.

1 The resource was found on the system.

**Resource library name.** The name of the library that contains the AFPDS resource. The

possible values are:

*\*INLINE*            The resource is defined within the data stream of the spooled file.  
*library\_name*        The name of the library that the AFPDS resource was found in.

**Resource name.** The name of the AFPDS resource used in the spooled file.

**Resource object type.** The type of AFPDS resource.

The valid values are:

*\*PAGSEG*            Page segment  
*\*PAGDFN*            Page definition  
*\*OVL*                Overlay  
*\*FORMDF*            Form definition  
*\*FNTRSC*            Font resource

**Source object.** The name of the object that the resource was found in.

The possible values are:

*\*SPLF*                The resource is found within the data stream of the spooled file.  
*object\_name*        The name of the AFPDS resource that the resource object is found in.

**Source object library.** The library that contains the source AFPDS resource object. This field is blank when the source object is *\*SPLF*.

**Source object type.** The object type of the source AFPDS resource object. This field is blank when the source object is *\*SPLF* or is one of the following special values:

*\*PAGSEG*            Page segment  
*\*PAGDFN*            Page definition  
*\*OVL*                Overlay  
*\*FORMDF*            Form definition  
*\*FNTRSC*            Font resource

**Where resource was referenced.** Where the resource was located.

Possible values are:

0            The resource was found within the spooled file or another AFPDS resource.  
1            The resource was found within the spooled file attributes.

## Format LRSE0200

For more information about the fields in the following table, see “Field Descriptions” on page 275.

Offset		Type	Field
Dec	Hex		
0	0	CHAR(1)	Resource availability
1	1	CHAR(3)	Reserved

Offset		Type	Field
Dec	Hex		
4	4	BINARY(4)	Displacement to resource name
8	8	BINARY(4)	Length of resource name
12	0C	BINARY(4)	Displacement to path name
16	10	BINARY(4)	Length of path name
20	14	BINARY(4)	Displacement to source object data
24	18	BINARY(4)	Length of source object data
28	1C	CHAR(8)	Format of source object data
		CHAR(*)	Resource name
		CHAR(*)	Path name
		CHAR(*)	Source object data

## Field Descriptions

**Displacement to path name** The displacement, in number of bytes, from the beginning of the structure to the path name. This field will be set to zero if the resource name contains the descriptive name of a True Type font.

**Displacement to resource name** The displacement, in number of bytes, from the beginning of the structure to the resource name.

**Displacement to source object data** The displacement, in number of bytes, from the beginning of the structure to the source object data.

**Length of path name.** The length of the path name in number of bytes.

**Length of resource name.** The length of the resource name in number of bytes.

**Length of source object data.** The length of the source object data in number of bytes.

**Path name.** The name of the integrated file system path where the resource resided.

**Resource availability.** Whether the resource was found, not found, or the owner of the spooled file was not authorized to the resource, the resource library, or the resource directory.

Possible values are:

- 0 The resource was not available for one of the following reasons:
  - The resource could not be found on the system.
  - The user who created the spooled file was not authorized to the resource, the resource library, or the resource directory.
- 1 The resource was found on the system.

**Resource name.** The name of the AFPDS resource used in the spooled file. For a True Type font, the resource name can be the font's descriptive name.

**Source object data.** The identity of the source object in the format specified in the source object format field.

**Source object format** The format of the source object. The

possible values are:

*LRSS0100* Source object data is returned in the format LRSS0100.  
*LRSS0200* Source object data is returned in the format LRSS0200

## Format LRSS0100

For more information about the fields in the following table, see "Field Descriptions."

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	Source object
10	0A	CHAR(10)	Source object library
20	14	CHAR(10)	Source object type

## Field Descriptions

**Source object.** The name of the object that the resource was found in.

The possible values are:

*\*SPLF* The resource is found within the data stream of the spooled file.  
*object\_name* The name of the AFPDS resource that the resource object is found in.

**Source object library.** The library that contains the source AFPDS resource object. This field is blank when the source object is *\*SPLF*.

**Source object type.** The object type of the source AFPDS resource object. This field is blank when the source object is *\*SPLF* or one of the following special values:

*\*PAGSEG* Page segment  
*\*PAGDFN* Page definition  
*\*OVL* Overlay  
*\*FORMDF* Form definition  
*\*FNTRSC* Font resource

## Format LRSS0200

For more information about the fields in the following table, see "Field Descriptions" on page 277.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Displacement to source object name
4	4	BINARY(4)	Length of source object name
8	8	BINARY(4)	Displacement to source object path name
12	0C	BINARY(4)	Length of source object path name
		CHAR(*)	Source object name

Offset		Type	Field
Dec	Hex		
		CHAR(*)	Source object path name

## Field Descriptions

**Displacement to source object name.** The displacement, in number of bytes, from the beginning of the structure to the source object name.

**Displacement to source object path name.** The displacement, in number of bytes, from the beginning of the structure to the source object path name. This field will be set to zero if the source object name contains the descriptive name of a True Type font.

**Length of source object name.** The length of the source object name in number of bytes.

**Length of source object path name.** The length of the path name in number of bytes.

**Source object name.** The name of the object that the resource was found in. For a True Type font, the source object name can be the font's descriptive name.

**Source object path name.** The name of the integrated file system path where the source object resides.

## Format LRSC0500

For more information about the fields in the following table, see "Field Descriptions."

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Length of entry
4	4	CHAR(1)	Resource availability
5	5	CHAR(3)	Reserved
8	8	BINARY(4)	Displacement to resource name
12	0C	BINARY(4)	Length of resource name
16	10	BINARY(4)	Displacement to path name
20	14	BINARY(4)	Length of path name
		CHAR(*)	Resource name
		CHAR(*)	Path name

## Field Descriptions

**Displacement to path name** The displacement, in number of bytes, from the beginning of the entry to the path name.

**Displacement to resource name** The displacement, in number of bytes, from the beginning of the entry to the resource name.

**Length of entry.** Length of the entry in number of bytes.

**Length of path name.** The length of the path name in number of bytes.

**Length of resource name.** The length of the resource name in number of bytes.

**Path name.** The name of the integrated file system path where the resource resides.

**Resource availability.** Whether the resource was found, not found, or the caller or the owner of the spooled file was not authorized to the resource, the resource library, or the resource directory.

Possible values are:

- 0 The resource was not available for one of the following reasons:
- The resource could not be found on the system.
  - The user who created the spooled file was not authorized to the resource, the resource library, or the resource directory.
  - If \*NONE was specified for the spooled file name, the caller of the API was not authorized to the resource, the resource library, or the resource directory.
- 1 The resource was found on the system.

**Resource name.** The name of the AFPDS resource used in the spooled file.

## Error Messages

Message ID	Error Message Text
CPF247E E	CCSID &1 is not valid.
CPF24B4 E	Severe error while addressing parameter list.
CPF3CF1 E	Error code parameter not valid.
CPF3C20 E	Error found by program &1.
CPF3C21 E	Format name &1 is not valid.
CPF3C33 E	Spooled file number &1 is not valid.
CPF3C40 E	Spooled file &4 not found.
CPF3C41 E	More than one spooled file with same name.
CPF3C42 E	User name or job number is not blank.
CPF3C43 E	Internal job identifier is not valid.
CPF3C44 E	Internal spooled file identifier is not valid.
CPF3C58 E	Job name specified is not valid.
CPF33C9 E	Spooled file name parameter cannot be blank.
CPF33CE E	Pel density specified is not valid.
CPF3330 E	Necessary resource not available.
CPF3336 E	Job &5/&4/&3 no longer in the system.
CPF333B E	Job system name is not valid.
CPF333C E	Spooled file create date is not valid.
CPF333D E	Spooled file create time is not valid.
CPF333E E	Spooled file create time is not blank.
CPF333F E	Job system name is not blank.
CPF3342 E	Job &5/&4/&3 not found.
CPF3343 E	Duplicate job names found.
CPF3350 E	Job &5/&4/&3 no longer in the system.
CPF335B E	Spooled file create date is not blank.
CPF34C4 E	List is too large for user space &1.
CPF34CA E	File &1 number &2 not available.
CPF34CC E	Resources cannot be retrieved from diskette files.
CPF34CD E	Value for resource name parameter not valid.
CPF34CE E	Value &1 not valid for resource object type.
CPF34CF E	User profile of the spooled file owner &1 damaged or not found.

Message ID	Error Message Text
CPF34DB E	File *NONE not valid with format &1.
CPF3492 E	Not authorized to spooled file.
CPF811A E	User space &4 in &9 damaged.
CPF9801 E	Object &2 in library &3 not found.
CPF9802 E	Not authorized to object &2 in &3.
CPF9803 E	Cannot allocate object &2 in library &3.
CPF9807 E	One or more libraries in library list deleted.
CPF9808 E	Cannot allocate one or more libraries on library list.
CPF9810 E	Library &1 not found.
CPF9820 E	Not authorized to use library &1.
CPF9830 E	Cannot assign library &1.
CPF9838 E	User profile storage limit exceeded.
CPF9846 E	Error while processing file &1 in library &2.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V3R7

Top | "Print APIs," on page 1 | APIs by category

---

## List Spooled Files (QUSLSPL) API

### Required Parameter Group:

1	Qualified user space name	Input	Char(20)
2	Format name	Input	Char(8)
3	User name	Input	Char(10)
4	Qualified output queue name	Input	Char(20)
5	Form type	Input	Char(10)
6	User-specified data	Input	Char(10)

### Optional Parameter Group 1:

7	Error Code	I/O	Char(*)
---	------------	-----	---------

### Optional Parameter Group 2:

8	Qualified job name	Input	Char(26)
9	Key for the fields to return	Input	Array(*) of Binary(4)
10	Number of fields to return	Input	Binary(4)

### Optional Parameter Group 3:

11	Auxiliary storage pool	Input	Binary(4)
----	------------------------	-------	-----------

### Optional Parameter Group 4:

12	Job system name	Input	Char(8)
13	Starting spooled file create date	Input	Char(7)
14	Starting spooled file create time	Input	Char(6)
15	Ending spooled file create date	Input	Char(7)
16	Ending spooled file create time	Input	Char(6)

### Optional Parameter Group 5:

17	Auxiliary storage pool device name	Input	Char(10)
----	------------------------------------	-------	----------

Default Public Authority: \*USE  
Threadsafe: No

The List Spooled Files (QUSLSPL) API is similar to the Work with Spooled Files (WRKSPLF) command or the Work with Job (WRKJOB OPTION(\*SPLF)) command. The API generates a list of spooled files on the system and places the list in a user space. The list can include some of the following:

- All spooled files
- Spooled files of specific users or all users
- Spooled files in a specified output queue or in all output queues
- Spooled files for all form types or the standard form type
- Spooled files that have any user-specified data values
- Spooled files generated by a specific job
- Spooled files stored in an auxiliary storage pool.

The generated list replaces any existing information in the user space.

## Authorities and Locks

*User Space Authority*  
\*CHANGE

*Library Authority*  
\*EXECUTE

*User Space Lock*  
\*EXCLRD

### Output Queue Authority

The requester is authorized to the output queue if one or more of the following conditions are met:

- The requester has \*SPLCTL authority.
- The requester has \*JOBCTL authority, and the output queue is specified as OPRCTL(\*YES).
- The requester has \*READ authority to the output queue.

## Required Parameter Group

**Qualified user space name**  
INPUT; CHAR(20)

The user space that receives the generated list, and the library in which it is located. The first 10 characters contain the user space name, and the second 10 characters contain the library name.

You can use these special values for the library name:

\*CURLIB            The job's current library  
\*LIBL              The library list

**Format name**  
INPUT; CHAR(8)

The format of the spooled file information being returned. You must specify the following: >>

*"Format SPLF0100" on page 288*            Contains the internal identifiers for the spooled file and the job that owns the spooled file.  
*"Format SPLF0200" on page 288*            Information that is shown when the list of the spooled files selected is displayed with the Work with Spooled Files (WRKSPLF) command.

"Format SPLF0300" on page 289

Contains the basic information for the spooled file. Entries returned are taken directly from internal indexes and are not validated against existing spooled file and job structures. Retrieving a list of spooled files using this format is faster than using format SPLF0100 or SPLF0200.

SPLF0400 (page "Format SPLF0400" on page 290)

Contains the basic information for the spooled file. This format differs from format SPLF0300 in that it provides dates and times in Coordinated Universal time (UTC) instead of local job time. Entries returned are taken directly from internal indexes and are not validated against existing spooled file and job structures. Retrieving a list of spooled files using this format is faster than using format SPLF0100 or SPLF0200.



### User name

INPUT; CHAR(10)

The name of the user whose spooled files are included in the list. This parameter can be used in conjunction with the output queue and library name, form type, user-specified data, auxiliary storage pool, job system name, starting spooled file create date, starting spooled file create time, ending spooled file create date, and ending spooled file create time parameters to return a partial list of all the spooled files. The list of spooled files returned is sorted by status, output priority, date, and time. It must be blank if the qualified job name parameter is specified.

A value must be specified if the qualified job name parameter is blank or not specified. Otherwise, message CPF34C6 or CPD34C6 is issued.

The possible special values are:

\*ALL                Files owned by all users  
\*CURRENT           Files owned by the current user

### Qualified output queue name

INPUT; CHAR(20)

The name of the output queue whose files are to be included in the list, and the library in which it is located. The first 10 characters contain the output queue name, and the second 10 characters contain the library name. This parameter can be used in conjunction with the user name, form type, user-specified data, auxiliary storage pool, job system name, starting spooled file create date, starting spooled file create time, ending spooled file create date, and ending spooled file create time parameters to return a partial list of all the spooled files. The list of spooled files returned is sorted by status, output priority, date, and time. It must be blank if the qualified job name parameter is specified.

**Note:** A value must be specified if the qualified job name parameter is blank or not specified. Otherwise, message CPF34C6 or CPD34C6 is issued.

You can use this special value for the output queue name:

\*ALL                Files on all output queues. When you use this value, the library name must be blanks.  
Output queue        The name of the output queue.  
name

You can use these special values for the library name:

\*CURLIB            The job's current library  
\*LIBL                The library list

### Form type

INPUT; CHAR(10)

The form type whose files are included in the list. The form type is the value specified on the form type parameter of the printer file. This parameter can be used in conjunction with the user name, qualified output queue name, user-specified data, auxiliary storage pool, job system name, starting spooled file create date, starting spooled file create time, ending spooled file create date, and ending spooled file create time parameters to return a partial list of all the spooled files. The list of spooled files returned is sorted by status, output priority, date, and time. It must be blank if the qualified job name parameter is specified.

The special values supported are:

- \*ALL Files for all form types
- \*STD Only files that specify the standard form type

### User-specified data

INPUT; CHAR(10)

The user-specified data value whose files are to be included in the list. This parameter can be used in conjunction with the user name, qualified output queue name, form type, auxiliary storage pool, job system name, starting spooled file create date, starting spooled file create time, ending spooled file create date, and ending spooled file create time parameters to return a partial list of all the spooled files. The list of spooled files returned is sorted by status, output priority, date, and time. It must be blank if the qualified job name parameter is specified.

The special value supported is:

- \*ALL Files with any user-specified data values

## Optional Parameter Group 1

### Error code

I/O; CHAR(\*)

The structure in which to return error information. For the format of the structure, see Error code parameter. If this parameter is omitted, diagnostic and escape messages are issued to the application.

## Optional Parameter Group 2

### Qualified job name

INPUT; CHAR(26)

The qualified job name of the job whose spooled files are to be included in the list. If the user name, the qualified output queue name, the form type, user-specified data, or auxiliary storage pool is specified, this parameter must be blank. Otherwise, message CPF34C2 is issued. The list of spooled files returned is sorted by spooled file number.

The qualified job name has three parts:

*job name* CHAR(10). A specific job name, or the following special value:

- \* Current running job.

The rest of the qualified job name parameter must be blank.

*user name* CHAR(10). A specific user profile name, or blank when the job name is asterisk (\*).

*job number* CHAR(6). A specific job number, or blank when the job name is asterisk (\*).

If this parameter is omitted, the API assumes all blanks.

### Keys for the fields to return

INPUT; ARRAY(\*) of BINARY(4)

The list of the fields to be returned in the SPLF0200 format. By specifying a set of keys, only the fields whose keys are specified are in the returned format. See "Valid Keys" on page 288 for a list of valid keys. This field is only used for the SPLF0200 format. This field is ignored if the number of keys for fields to return parameter is 0.

#### Number of fields to return

INPUT; BINARY(4)

The number of fields to return in the SPLF0200 format. This indicates how many keys are in the array of the keys for the fields to return parameter. This field is used for the SPLF0200 format only, >> and it must be 0 when the SPLF0100, SPLF0300, or SPLF0400 format is specified. <<

If this parameter is omitted, the API assumes 0.

### Optional Parameter Group 3

#### Auxiliary storage pool (ASP)

INPUT; BINARY(4)

The auxiliary storage pool whose spooled files are to be included in the list. This parameter can be used in conjunction with the user name, qualified output queue name, form type, user-specified data, job system name, starting spooled file create date, starting spooled file create time, ending spooled file create date, and ending spooled file create time parameters to return a partial list of all the spooled files. The list of spooled files returned is sorted by status, output priority, date, and time. It must be blank if the qualified job name parameter is specified.

The valid values are:

- 1 \*BLANK. No auxiliary storage pool is specified.
- 0 \*ALL. Spooled files in all auxiliary storage pools as specified in the Auxiliary storage pool device name parameter.
- 1 \*SYSTEM. Spooled files in the system auxiliary storage pool.
- 2 \*ASPDEV. Spooled files in the auxiliary storage pool device specified on the Auxiliary storage pool device name parameter.
- 2-255 User auxiliary storage pool (2-32), or a primary or a secondary auxiliary storage pool (33-255).

If this parameter is omitted, the API assumes 0 when the qualified job name parameter is blank or not specified. The API assumes -1 if the qualified job name parameter is not blank.

### Optional Parameter Group 4

#### Job system name

INPUT; Char(8)

The name of the system where the job, specified in the qualified job name parameter, was run. This parameter can be used in conjunction with the user name, qualified output queue name, form type, user-specified data, auxiliary storage pool, starting spooled file create date, starting spooled file create time, ending spooled file create date, ending spooled file create time, or qualified job name parameters to return a partial list of all the spooled files. The list of spooled files returned is sorted by status, output priority, date, and time.

The following special values are supported for this parameter:

- \*ALL The returned list is not to be filtered based on job system name.
- \*CURRENT Only spooled files created on the current system are to be returned.
- job-system-name* Only spooled files created on the system specified are to be returned.

If this parameter is omitted, the API assumes \*ALL.

### Starting spooled file create date

INPUT; Char(7)

» The date, based on local job time, that the spooled file was created on the system. « This parameter can be used in conjunction with the user name, qualified output queue name, form type, user-specified data, auxiliary storage pool, job system name, starting spooled file create time, ending spooled file create date, ending spooled file create time, or qualified job name parameters to return a partial list of all the spooled files. The list of spooled files returned is sorted by status, output priority, date, and time. The date must be in the CYYMMDD format or one of the following special values:

<i>*ALL</i>	The returned list is not to be filtered based on spooled file create date and spooled file create time.
<i>*FIRST</i>	All spooled files starting with the earliest create date and time and less than or equal to the ending spooled file create date and time are to be returned.
<i>date</i>	All spooled files with a create date and time equal to or later than the starting spooled file create date and time and less than or equal to the ending spooled file create date and time are to be returned.

The date format CYYMMDD is defined as follows:

<i>C</i>	Century, where 0 indicates years 19xx and 1 indicates years 20xx.
<i>YY</i>	Year
<i>MM</i>	Month
<i>DD</i>	Day

### Starting spooled file create time

INPUT; Char(6)

» The time, in local job time, that the spooled file was created on the system. « This parameter can be used in conjunction with the user name, qualified output queue name, form type, user-specified data, auxiliary storage pool, job system name, starting spooled file create date, ending spooled file create date, ending spooled file create time, or qualified job name parameters to return a partial list of all the spooled files. The list of spooled files returned is sorted by status, output priority, date, and time. This parameter must be set to blanks when special value *\*ALL* or *\*FIRST* is used for parameter Starting spooled file create date. This parameter must have a value set if a date is specified for parameter Starting spooled file create date. The time must be in the HHMMSS format.

The time format HHMMSS is defined as follows:

<i>HH</i>	Hour
<i>MM</i>	Minutes
<i>SS</i>	Seconds

### Ending spooled file create date

INPUT; Char(7)

» The date, based on local job time, that the spooled file was created on the system. « This parameter can be used in conjunction with the user name, qualified output queue name, form type, user-specified data, auxiliary storage pool, job system name, starting spooled file create date, starting spooled file create time, ending spooled file create time, or qualified job name parameters to return a partial list of all the spooled files. The list of spooled files returned is sorted by status, output priority, date, and time. If the Starting spooled file create date parameter is set to *\*ALL*, then this parameter must be set to blanks. If a date has been specified for the Starting spooled file create date parameter, then this parameter must be set to a valid date. The date must be in the CYYMMDD format or one of the following special values:

*\*LAST* All spooled files with a create date and time equal to or later than the starting spooled file create date and time and equal to the last spooled file create date and time are to be returned.

*date* All spooled files with a create date and time equal to or later than the starting spooled file create date and time and less than or equal to the ending spooled file create date and time are to be returned.

The date format CYYMMDD is defined as follows:

<i>C</i>	Century, where 0 indicates years 19xx and 1 indicates years 20xx.
<i>YY</i>	Year
<i>MM</i>	Month
<i>DD</i>	Day

### Ending spooled file create time

INPUT; Char(6)

» The time, in local job time, that the spooled file was created on the system. « This parameter can be used in conjunction with the user name, qualified output queue name, form type, user-specified data, auxiliary storage pool, job system name, starting spooled file create date, starting spooled file create time, ending spooled file create date, or qualified job name parameters to return a partial list of all the spooled files. The list of spooled files returned is sorted by status, output priority, date, and time. This parameter must be set to blanks when special value \*ALL is used for parameter Starting spooled file create date or when special value \*LAST is used for parameter Ending spooled file create date. This parameter must have a value set if a date is specified for parameter Ending spooled file create date. The time must be in the HHMMSS format.

The time format HHMMSS is defined as follows:

<i>HH</i>	Hour
<i>MM</i>	Minutes
<i>SS</i>	Seconds

## Optional Parameter Group 5

### Auxiliary storage pool (ASP) device name

INPUT; Char(10)

The auxiliary storage pool device name from which spooled files are to be included in the list. When 0 or -2 is specified on the Auxiliary storage pool parameter, this parameter can be used in conjunction with the user name, qualified output queue name, form type, user-specified data, job system name, starting spooled file create date, starting spooled file create time, ending spooled file create date, and ending spooled file create time parameters to return a partial list of all the spooled files. The list of spooled files returned is sorted by status, output priority, date, and time.

The valid values are:

<i>*</i>	Files which are found in the ASPs that are currently part of the thread's library name space are returned. This includes the system ASP (ASP 1), all defined basic user ASPs (ASP's 2-32), and if the thread has an ASP group, the primary and secondary ASPs in the thread's ASP group.
<i>*CURASPGRP</i>	Files which are found in the primary and secondary ASPs in the thread's ASP group are selected. If no ASP group is associated with the thread, an error will be issued.
<i>*SYSBAS</i>	Files which are found in the system ASP (ASP 1) and all defined basic user ASPs (ASP's 2-32) are selected.
<i>auxiliary storage pool device name</i>	Files which are found in the specified primary or secondary ASP are selected. Only primary or secondary ASPs which are in the thread's ASP group may be specified. If no ASP group is associated with the thread, an error will be issued.

If this parameter is omitted, the API assumes \*. When the Qualified job name parameter is specified, all spooled files for the job that are found in the ASPs that are currently part of the caller's library name space are returned.

## Format of the Generated List

The spooled file list consists of:

- A user area
- A generic header
- An input parameter section
- A header section
- A list data section

For details about the user area and generic header fields, see User spaces. For details about the remaining items, see the following subtopics. For descriptions of the fields in the list, see "Field Descriptions" on page 290.

When you retrieve list entry information from a user space, you must use the entry size returned in the generic header as a displacement to the next list entry. The size of each entry may be padded at the end. If you do not use the entry size, the result may not be valid.

### Input Parameter Section (Optional Parameter Groups 1 and 2)

**Note:** This input parameter section format is used only when Optional Parameter Group 1 or Optional Parameter Group 2 is specified on the call to the QUSLSPL API. The order of the fields in the input parameter section does not match the order of the parameters on the call to the API.

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	User space name specified
10	A	CHAR(10)	User space library name specified
20	14	CHAR(8)	Format name
28	1C	CHAR(10)	User name specified
38	26	CHAR(10)	Output queue name specified
48	30	CHAR(10)	Output queue library name specified
58	3A	CHAR(10)	Form type
68	44	CHAR(10)	User-specified data
78	4E	CHAR(26)	Qualified job name specified
104	68	BINARY(4)	Number of key fields specified
108	6C	ARRAY(*) of BINARY(4)	Key fields specified

### Input Parameter Section 2 (Optional Parameter Group 3, 4, and 5)

**Note:** This input parameter section format is used only when Optional Parameter Group 3, 4, or 5 is specified on the call to the QUSLSPL API. This input parameter section is used if optional parameter group 3, 4, or 5 is specified on the call to the API. The order of the fields in the input parameter section does not match the order of the parameters on the call to the API.

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	User space name specified

Offset		Type	Field
Dec	Hex		
10	A	CHAR(10)	User space library name specified
20	14	CHAR(8)	Format name
28	1C	CHAR(10)	User name specified
38	26	CHAR(10)	Output queue name specified
48	30	CHAR(10)	Output queue library name specified
58	3A	CHAR(10)	Form type
68	44	CHAR(10)	User-specified data
78	4E	CHAR(26)	Qualified job name specified
104	68	BINARY(4)	Offset to key fields specified
108	6C	BINARY(4)	Number of key fields specified
112	70	BINARY(4)	Auxiliary storage pool specified
116	74	CHAR(8)	Job system name specified
124	7C	CHAR(7)	Starting spooled file create date specified
131	83	CHAR(6)	Starting spooled file create time specified
137	89	CHAR(7)	Ending spooled file create date specified
144	90	CHAR(6)	Ending spooled file create time specified
150	96	CHAR(10)	Auxiliary storage pool device name specified
		ARRAY(*) of BINARY(4)	Key fields specified

## Header Section

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	User name
10	A	CHAR(10)	Output queue name
20	14	CHAR(10)	Output queue library name
30	1E	CHAR(10)	User space name
40	28	CHAR(10)	User space library name
50	32	CHAR(26)	Qualified job name
76	4C	CHAR(8)	Qualified job system name
84	54	CHAR(7)	Starting spooled file create date used
91	5B	CHAR(6)	Starting spooled file create time used
97	61	CHAR(7)	Ending spooled file create date used
104	68	CHAR(6)	Ending spooled file create time used

## Format SPLF0100

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	User name
10	A	CHAR(10)	Output queue name
20	14	CHAR(10)	Output queue library name
30	1E	CHAR(10)	Form type
40	28	CHAR(10)	User-specified data
50	32	CHAR(16)	Internal job identifier
66	42	CHAR(16)	Internal spooled file identifier
82	52	CHAR(2)	Reserved
84	54	BINARY(4)	Auxiliary storage pool

## Format SPLF0200

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Number of fields returned
Offsets vary. These fields repeat, in the order listed, for each key field selected.		BINARY(4)	Length of field information returned
		BINARY(4)	Key field for field returned
		CHAR(1)	Type of data
		CHAR(3)	Reserved
		BINARY(4)	Length of data returned
		CHAR(*)	Data
		CHAR(*)	Reserved

## Valid Keys

Key	Type	Field
201	CHAR(10)	Spooled file name
202	CHAR(10)	Job name
203	CHAR(10)	User name
204	CHAR(6)	Job number
205	BINARY(4)	Spooled file number
206	CHAR(10)	Output queue name
207	CHAR(10)	Output queue library name
208	CHAR(10)	Device
209	CHAR(10)	User-specified data
210	CHAR(10)	Status
211	BINARY(4)	Total pages

Key	Type	Field
212	BINARY(4)	Current page
213	BINARY(4)	Copies left to produce
214	CHAR(10)	Form type
215	CHAR(2)	Priority
216	CHAR(7)	Date file was opened (created)
217	CHAR(6)	Time file was opened (created)
218	CHAR(16)	Internal job identifier
219	CHAR(16)	Internal spooled file identifier
220	CHAR(10)	Device type
221	CHAR(7)	Date file last used
222	BINARY(4)	Spooled file size multiplier
223	BINARY(4)	Size of spooled file
224	BINARY(4)	Auxiliary storage pool
225	CHAR(8)	Job system name
226	CHAR(10)	Auxiliary storage pool device name
» 227	CHAR(7)	Date file was created in UTC
228	CHAR(6)	Time file was created in UTC «

## Format SPLF0300

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	Job name
10	A	CHAR(10)	User name
20	14	CHAR(6)	Job number
26	1A	CHAR(10)	Spooled file name
36	24	BINARY(4)	Spooled file number
40	28	BINARY(4)	Spooled file status
44	2C	CHAR(7)	Date file was opened (created)
51	33	CHAR(6)	Time file was opened (created)
57	39	CHAR(1)	Spooled file schedule
58	3A	CHAR(10)	Spooled file system name
68	44	CHAR(10)	User-specified data
78	4E	CHAR(10)	Form type
88	58	CHAR(10)	Output queue name
98	62	CHAR(10)	Output queue library name
108	6C	BINARY(4)	Auxiliary storage pool
112	70	BINARY(4)	Size of spooled file
116	74	BINARY(4)	Spooled file size multiplier
120	78	BINARY(4)	Total pages

Offset		Type	Field
Dec	Hex		
124	7C	BINARY(4)	Copies left to produce
128	80	CHAR(1)	Priority
129	81	CHAR(3)	Reserved
132	84	BINARY(4)	Internet print protocol job identifier

## Format SPLF0400

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	Job name
10	A	CHAR(10)	User name
20	14	CHAR(6)	Job number
26	1A	CHAR(10)	Spooled file name
36	24	BINARY(4)	Spooled file number
40	28	BINARY(4)	Spooled file status
44	2C	CHAR(7)	Date file was created in UTC
51	33	CHAR(6)	Time file was created in UTC
57	39	CHAR(1)	Spooled file schedule
58	3A	CHAR(10)	Spooled file system name
68	44	CHAR(10)	User-specified data
78	4E	CHAR(10)	Form type
88	58	CHAR(10)	Output queue name
98	62	CHAR(10)	Output queue library name
108	6C	BINARY(4)	Auxiliary storage pool
112	70	BINARY(4)	Size of spooled file
116	74	BINARY(4)	Spooled file size multiplier
120	78	BINARY(4)	Total pages
124	7C	BINARY(4)	Copies left to produce
128	80	CHAR(1)	Priority
129	81	CHAR(3)	Reserved
132	84	BINARY(4)	Internet print protocol job identifier



## Field Descriptions

**Auxiliary storage pool.** The auxiliary storage pool in which the spooled file resides.

Valid values are:

1            \*SYSTEM. System auxiliary storage pool

2-255      The number of the auxiliary storage pool.

**Auxiliary storage pool device name.** The name of the auxiliary storage pool device in which the spooled file resides.

Valid values are:

\*SYSBAS      The spooled file resides in the system ASP (ASP 1) or one of the defined basic user ASPs (ASPs 2-32).

*auxiliary storage pool device name*      The name of the auxiliary storage pool device in which the spooled file resides.

**Auxiliary storage pool device name specified.** The auxiliary storage pool device name specified in the call to the API.

**Auxiliary storage pool specified.** The auxiliary storage pool specified in the call to the API.

**Copies left to produce.** The remaining number of copies to be produced on the writer. This attribute applies to printer device type spooled files only.

**Current page.** The page number or record number currently being written. The page number may be lower or higher than the page number actually being printed because of buffering done by the system. For example, if the spooled file is routed to a diskette unit or the writer is currently printing job separators or file separators for the spooled file, the page number shown may be zero.

**Data.** The data returned for the key field.

**Date file last used.** » The date, based on local job time, that the file was last used in the CYYMMDD format. «

C            Century, where 0 indicates years 19xx and 1 indicates years 20xx.  
YY          Year  
MM          Month  
DD          Day

» Date file was created in UTC. The date, based on Coordinated Universal Time (UTC), that the file was opened in the CYYMMDD format.

C            Century, where 0 indicates years 19xx and 1 indicates years 20xx.  
YY          Year  
MM          Month  
DD          Day

«

**Date file was opened (created).** » The date, based on local job time, that the file was opened in the CYYMMDD format. «

C            Century, where 0 indicates years 19xx and 1 indicates years 20xx.  
YY          Year  
MM          Month  
DD          Day

**Device.** The name of the device on which the spooled file is located. If the spooled file is on an output queue that is the default output queue of a printer, this field contains the name of the output queue. Otherwise, this field is blank.

**Device type.** The type of device for which the spooled file is intended. The possible values are PRINTER or DISKETTE.

**Ending spooled file create date specified.** The spooled file create date specified in the call to the API.

**Ending spooled file create date used.** The spooled file create date used by the API to generate the list of spooled files.

**Ending spooled file create time specified.** The spooled file create time specified in the call to the API.

**Ending spooled file create time used.** The spooled file create time used by the API to generate the list of spooled files.

**Form type.** The type of form to load in the printer to print this file. This attribute applies to printer device type spooled files only.

**Format name.** The format of the retrieved output.

**Job name.** The name of the job that owns the spooled file.

**Job number.** The number of the job that owns the spooled file.

**Job system name.** The name of the system where the job that created the spooled file ran.

**Job system name specified.** The name of the system where the qualified job, specified in the call to the API, was run.

**Key field for field returned.** The identifier of the field returned. See “Valid Keys” on page 288 for a list of valid keys. The fields in the list data section are returned in the order specified by the key fields specified parameter for each spooled file in the list.

**Key fields specified.** The list of fields to return in the SPLF0200 format specified in the call to the API.

**Internal job identifier.** The internal identifier for the job. Only the i5/OS® APIs use this identifier, not any other interface on the system. The identifier is not valid following an initial program load (IPL). If you attempt to use it after an IPL, an exception occurs. This identifier is always valid when used with the internal spooled file identifier as input to i5/OS APIs. For spooled files which have been detached from their jobs or which are on independent disk pools, this value is not valid as input to some i5/OS APIs, such as the Retrieve Job Information (QUSRJOB) API, that take this value without the internal spooled file identifier.

**Internal spooled file identifier.** The input value that other programs use to improve the performance of locating the spooled file on the system. Only the spooled file APIs use this identifier, not any other interface on the system. The identifier is not valid following an initial program load (IPL) » or if the spooled file has been moved between auxiliary storage pools (ASPs). If you attempt to use an invalid internal spooled file identifier, an exception occurs. «

**Internet print protocol job identifier.** The IPP job identifier assigned by the system based on the output queue to which the file was added or moved. This value ranges from 1 to 2147483647 and is not guaranteed to be unique for a given output queue.

**Length of data returned.** The length of the data returned for the field.

**Length of field information returned.** The total length of information returned for this field.

**Number of fields returned.** This is the number of fields returned to the application.

**Number of key fields specified.** The number of fields to return in format SPLF0200 specified in the call to the API.

**Offset to key fields specified.** The offset to the keys of the fields to be returned that is specified in the call to the API.

**Output queue library name specified.** The name of the output queue library specified in the call to the API.

**Output queue library name.** The library where the output queue is located.

**Output queue name specified.** The name of the output queue specified in the call to the API.

**Output queue name.** The name of the output queue in which the file is located.

**Priority.** The priority of the spooled file. The priority ranges from 1 (highest) to 9 (lowest).

**Qualified job name specified.** The qualified name of the job specified in the call to the API.

**Qualified job name.** The qualified name of the job that owns the spooled files.

**Qualified job system name.** The name of the system the qualified job was run on.

**Reserved.** The field is reserved.

**Size of spooled file.** The spooled file size. Multiply this field by the spooled file size multiplier field to get the size of the spooled file in number of bytes. The spooled file size is the data stream size plus the "overhead" storage used to store the spooled file's data stream, which includes the spooled file's attributes.

**Spooled file name.** The name of the spooled file in the list entry.

**Spooled file number.** The number of the spooled file in the list entry.

**Spooled file schedule.** The schedule of the spooled file.

Valid values are:

- 1 The spooled file is schedule immediate. A spooling writer can process the spooled file immediately.
- 2 The spooled file is schedule file end. A spooling writer cannot process the spooled file until it has been closed.
- 3 The spooled file is schedule job end. A spooling writer cannot process the spooled file until the job of the spooled file has ended.

**Spooled file size multiplier.** This value must be multiplied by the size of spooled file field to get the size of the spooled file in number of bytes.

**Spooled file status.** The status of the file. The following list of values is used to describe the status of the file:

- 1 RDY The file is available to be written to an output device by a writer.
- 2 OPN The file has not been processed completely and is not ready to be selected by a writer.

- 3 CLO The file has been processed completely by a program, but SCHEDULE(\*JOBEND) was specified. The job that produced the spooled file has not finished.
- 4 SAV The file has been written and then saved. This file remains saved until it is released.
- 5 WTR This file currently is being produced by the writer on an output device.
- 6 HLD The file has been held.
- 7 MSGW This file has a message that needs a reply or needs an action to be taken.
- 8 PND This file is pending (waiting) to be printed.
- 9 PRT The file has been sent to the printer completely, but the print complete status has not been sent back.
- 10 FIN This spooled file is no longer in the system. These spooled files are included in the list of spooled files only if the qualified job name is specified.
- 11 SND This spooled file is being sent or has been sent to a remote system.
- 12 DFR This spooled file has been deferred from printing.

**Spooled file system name.** The name of the system where the job that created the spooled file ran.

**Starting spooled file create date specified.** The spooled file create date specified in the call to the API.

**Starting spooled file create date used.** The spooled file create date used by the API to generate the list of spooled files.

**Starting spooled file create time specified.** The spooled file create time specified in the call to the API.

**Starting spooled file create time used.** The spooled file create time used by the API to generate the list of spooled files.

**Status.** The status of the file. The following list of values is used to describe the status of the file:

*CLOSED	The file has been completely processed by a program, but SCHEDULE(*JOBEND) was specified. The job that produced the spooled file has not finished.
*DEFERRED	This spooled file has been deferred from printing.
*SENDING	This spooled file is being sent or has been sent to a remote system.
*FINISHED	This spooled file is no longer in the system. These spooled files are included in the list of spooled files only if the qualified job name is specified.
*HELD	The file has been held.
*MESSAGE	This file has a message that needs a reply or needs an action to be taken.
*OPEN	The file has not been completely processed and is not ready to be selected by a writer.
*PENDING	This file is pending (waiting) to be printed.
*PRINTER	The file has been completely sent to the printer, but the print complete status has not been sent back.
*READY	The file is available to be written to an output device by a writer.
*SAVED	The file has been written and then saved. This file remains saved until it is released.
*WRITING	This file is currently being produced by the writer on an output device.

» Time file was created in UTC. The time, in Coordinated Universal Time (UTC), that the file was opened in the HHMMSS format.

HH	Hour
MM	Minute
SS	Second

«

**Time file was opened (created).** » The time, in local job time, that the file was opened in the HHMMSS format. «

<i>HH</i>	Hour
<i>MM</i>	Minute
<i>SS</i>	Second

**Total pages.** The number of pages this printer file contains or the number of records this diskette file contains.

**Type of data.** The type of data returned.

<i>C</i>	The data is returned in character format.
<i>B</i>	The data is returned in binary format

**User name.** The name of the user that owns the spooled file.

**User name specified.** The name of the user specified in the call to the API.

**User space library name.** The library containing the user space.

**User space library name specified.** The library name of the user space specified.

**User space name.** The name of the user space that contains the format returned.

**User space name specified.** The name of the user space specified.

**User-specified data.**

The 10 characters of user-specified data that describe the file.

## Error Messages

Message ID	Error Message Text
CPF24B4 E	Severe error while addressing parameter list.
CPF3CF1 E	Error code parameter not valid.
CPF3C20 E	Error found by program &1.
CPD3C21 D	Format name &1 is not valid.
CPD3C30 D	Library name &1 is not valid for output queue *ALL.
CPD34C6 D	Too few selection criteria specified.
CPF3C21 E	Format name &1 is not valid.
CPF3C30 E	Library name &1 is not valid for output queue *ALL.
CPF3C36 E	Number of parameters, &1, entered for this API was not valid.
CPF3C90 E	Literal value cannot be changed.
CPF33FA E	Auxiliary storage pool specified is not valid.
CPF3330 E	Necessary resource not available.
CPF3336 E	Job &5/&4/&3 no longer in the system.
CPF3342 E	Job &5/&4/&3 not found.
CPF3343 E	Duplicate job names found.
CPF3350 E	Job &5/&4/&3 no longer in the system.
CPF335D E	Job system name is not valid.
CPF335E E	Starting spooled file create date is not valid.
CPF335F E	Starting spooled file create time is not valid.
CPF336C E	Starting spooled file create time is not blank.
CPF336D E	Ending spooled file create date is not valid.
CPF336E E	Ending spooled file create date is not blank.
CPF336F E	Ending spooled file create time is not valid.

Message ID	Error Message Text
CPF337A E	Ending spooled file create time is not blank.
CPF337E E	ASP device &1 not in current ASP group for thread.
CPF338B E	ASP device &1 not allowed with ASP number &2.
CPF34C1 E	Value &1 in keys for fields to return parameter not valid.
CPF34C2 E	Too many selection criteria specified.
CPF34C4 E	List is too large for user space &1.
CPF34C6 E	Too few selection criteria specified.
CPF811A E	User space &4 in &9 damaged.
CPF9801 E	Object &2 in library &3 not found.
CPF9802 E	Not authorized to object &2 in &3.
CPF9803 E	Cannot allocate object &2 in library &3.
CPF9807 E	One or more libraries in library list deleted.
CPF9808 E	Cannot allocate one or more libraries on library list.
CPF9810 E	Library &1 not found.
CPF9820 E	Not authorized to use library &1.
CPF9830 E	Cannot assign library &1.
CPF9833 E	*CURASPGRP or *ASPGRPPRI specified and thread has no ASP group.
CPF9838 E	User profile storage limit exceeded.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API Introduced: V1R3

Top | "Print APIs," on page 1 | APIs by category

---

## Move Spooled File (QSPMOVSP) API

Required Parameter Group:

1	Function information	Input	Char(*)
2	Length of function information	Input	Binary(4)
3	Function information format name	Input	Char(8)
4	Error code	I/O	Char(*)

Default Public Authority: \*USE

Threadsafe: No

The Move Spooled File (QSPMOVSP) API can:

- Move a spooled file to the top of an output queue
- Move a spooled file after another spooled file

The spooled files can be identified by:

- The internal job and spooled file identifiers
- The specific job name, spooled file name, and spooled file number

See Spooled File Identifying Fields (page 306) for these combinations for spooled file identifiers.

Using this API, a spooled file can:

- Change output queues

The output queue of the spooled file is changed when it is moved to the top of another output queue.

The source spooled file moves to the output queue of the target spooled file.

- Change priority

The priority of the spooled file is changed to the requester's highest scheduled priority if the spooled file is moved to the top of an output queue. For example:

Requester's priority limit = 2

	file	priority
New priority = 2 when moved here ----->	A	3
	B	3

The priority of the spooled file is changed to the priority of the target spooled file if the requester's highest schedule priority is equal to or less than that of the spooled file it is moving after. For example:

Requester's priority limit = 2

	file	priority
new priority = 3 when moved here ----->	A	3
	B	5
	C	5

The priority of the spooled file is changed to the requester's highest schedule priority if:

- The requester's priority is greater (lower) than the target spooled file and
- The requester's limit is less (higher) than or equal to the spooled files it is moving ahead of. For example:

Requester's priority limit = 4

	file	priority
new priority = 4 when moved here ----->	C	3
	B	5
	D	5

Requester's priority limit = 4

	file	priority
new priority = 4 when moved here ----->	E	3
	F	4

In this case, the requester must own spooled file F or have special authorities.

- Be held

A spooled file in ready, open, closed, or deferred status is held when it is moved after a spooled file that is held or saved.

- Be released and made ready

A spooled file in held, open, closed, or saved status is made ready when it is moved to the top of an output queue.

A spooled file in held, open, closed, or saved status is made ready when it is moved after a spooled file that is in ready status.

## Restrictions for Movement of Spooled Files

Situations exist in which the QSPMOVSP API does not allow a spooled file to be moved.

These situations are:

- A spooled file with deferred (DFR) status can move after a spooled file with ready (RDY) status on the same output queue. This can occur when the target spooled file is the last spooled file in RDY status on that output queue.
- The spooled file being moved cannot be held as a result of a Hold Job (HLDJOB SPLFILE(\*YES)) command.
- A spooled file selected by a writer cannot be moved.

- A spooled file cannot be moved after a spooled file that is in open status.
- A spooled file cannot be moved after a spooled file that is in closed status, unless both spooled files are part of the same job.
- The destination output queue must have the order of spooled files on the queue defined as \*FIFO. Spooled files cannot be moved to the top of an output queue with the order of spooled files on the output queue defined as \*JOBNBR. When a spooled file is requested to be moved after a spooled file and the target spooled file resides on a job number sequenced output queue, the move will occur. However, the spooled file is positioned by the job number sequence. In this situation, CPI33E2 is written to the job log to indicate the action taken.
- A spooled file cannot be moved after a spooled file that has been selected by a writer (PND, WTR, PRT, SND, or MSGW status) unless the target spooled file is the last spooled file selected by the writer.
- A spooled file cannot be moved after a spooled file in deferred status unless both spooled files have the same status.
- A spooled file cannot be moved from an output queue in an independent disk pool to an output queue in \*SYSBAS if that same spooled file resides in an output queue of another independent disk pool.

» **Note:** If a spooled file is being moved using format MSPF0200 after a file which is on an output queue in a library that is in the process of being renamed, the spooled file could end up in some other position than directly after the spooled file specified. «

## Authorities and Locks

*Output Queue of Spooled File Being Moved*  
\*EXCLRD

*New Output Queue of File Being Moved*  
\*EXCLRD

*Output Queue Library Authority*  
\*EXECUTE

## Output Queue Authorities

When the source spooled file moves to a different output queue and the source output queue is defined with DSPDATA(\*OWNER), the requester must be the owner of the source spooled file or have \*SPLCTL authority.

One or more of the following conditions must be met to move a spooled file on the destination output queue. The destination queue can be either the source output queue or a different output queue.

- The requester has \*SPLCTL authority.
- The requester has \*JOBCTL authority, and the destination output queue is specified as OPRCTL(\*YES).
- The requester has \*READ authority to the output queue.

## Spooled File Authorities

The requester is authorized to the source spooled file if one or more of the following conditions are met:

- The requester is the owner of the spooled file.
- The requester has \*SPLCTL authority.
- The requester has \*JOBCTL authority, and the source output queue is specified as OPRCTL(\*YES).
- The source output queue is defined as AUTCHK(\*OWNER) and the requester has ownership rights to the output queue.
- The source output queue is defined as AUTCHK(\*DTAAUT) and the requester has data rights (read, add, and delete) to the output queue.

## User Profile Highest Schedule Priority

When the requester has no other special authorities, the following are also in effect:

- When moving a spooled file to the top of an output queue and one of the following is true, the priority is set to 1:
  - The requester has \*SPLCTL authority.
  - The requester has \*JOBCTL authority, and the destination output queue is specified as OPRCTL(\*YES).
  - The destination output queue is defined as AUTCHK(\*OWNER) and the requester has ownership rights to the output queue.
  - The destination output queue is defined as AUTCHK(\*DTAAUT) and the requester has data rights (read, add, and delete) to the output queue.
- If a spooled file is moved ahead of a spooled file owned by someone else and both spooled files have the same priority, the requester must have a priority limit of one less than the priority of the spooled file.
- If the requester's priority limit is the same as the spooled files it is to be moved ahead of, all spooled files at that priority level must be owned by the requester.

Special authorities of \*JOBCTL (job control) and \*SPLCTL (spool control) allow the requester to move a spooled file anywhere on an output queue.

## Required Parameter Group

### Function information

INPUT; CHAR(\*)

The information that is associated with the spooled file to be moved and the output queue to which it is to be moved. See "Format of the Function Information" on page 300 for the format of this parameter.

### Length of function information

INPUT; BINARY(4)

The length of the function information in the function information parameter. The length depends on the function format. Each format has a different (but fixed) length as shown in the specific format tables. The valid length for format MSPF0100 is 92 bytes or 114 bytes; the valid length for format MSPF0200 is 144 bytes or 188 bytes. If a valid length is not specified, message CPF3C1D will be issued.

### Function information format name

INPUT; CHAR(8)

The format of the function information that is being provided. The information is provided in the function information parameter.

The formats are:

<i>"MSPF0100 Format" on page 300</i>	Format for the function of *NEXT, which is a move of the source spooled file ahead of all other ready spooled files on the target output queue.
<i>"MSPF0200 Format" on page 300</i>	Format for the function of *AFTER, which is a move of the source spooled file after the target spooled file.

### Error code

I/O; CHAR(\*)

The structure in which to return error information. For the format of the structure, see Error code parameter.

## Format of the Function Information

### MSPF0100 Format

The following table shows the information provided for the MSPF0100 format. For more details about the fields in the following table, see “Field Descriptions” on page 301.

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	Source job name
10	A	CHAR(10)	Source job user name
20	14	CHAR(6)	Source job number
26	18	CHAR(16)	Source internal job identifier
42	2A	CHAR(16)	Source internal spooled file identifier
58	3A	CHAR(10)	Source spooled file name
68	44	BINARY(4)	Source spooled file number
72	48	CHAR(10)	Target output queue name
82	52	CHAR(10)	Target output queue library name
92	5C	CHAR(8)	Source job system name
100	64	CHAR(7)	Source spooled file create date
107	6B	CHAR(1)	Reserved
108	6C	CHAR(6)	Source spooled file create time

### MSPF0200 Format

The following table shows the information provided for the MSPF0200 format. For more details about the fields in the following table, see “Field Descriptions” on page 301.

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	Source job name
10	A	CHAR(10)	Source job user name
20	14	CHAR(6)	Source job number
26	18	CHAR(16)	Source internal job identifier
42	2A	CHAR(16)	Source internal spooled file identifier
58	3A	CHAR(10)	Source spooled file name
68	44	BINARY(4)	Source spooled file number
72	48	CHAR(10)	Target job name
82	52	CHAR(10)	Target job user name
92	5C	CHAR(6)	Target job number
98	62	CHAR(16)	Target internal job identifier
114	72	CHAR(16)	Target internal spooled file identifier
130	82	CHAR(10)	Target spooled file name
140	8C	BINARY(4)	Target spooled file number
144	90	CHAR(8)	Source job system name

Offset		Type	Field
Dec	Hex		
152	98	CHAR(7)	Source spooled file create date
159	9F	CHAR(1)	Reserved
160	A0	CHAR(6)	Source spooled file create time
166	A6	CHAR(8)	Target job system name
174	AE	CHAR(7)	Target spooled file create date
181	B5	CHAR(1)	Reserved
182	B6	CHAR(6)	Target spooled file create time

## Field Descriptions

**Source internal job identifier.** The internal identifier for the job that owns the spooled file to be moved. Only i5/OS® APIs use this identifier; no other interface on the system does.

**Note:** The identifier is not valid following an initial program load (IPL). If you attempt to use it after an IPL, an exception occurs.

This field must be blank when the source job name is given.

Use one of the following APIs to make the identifier available:

- “List Spooled Files (QUSLSPL) API” on page 279
- “Retrieve Spooled File Attributes (QUSRSPLA) API” on page 337

There may be a performance advantage when identifying the source job by its internal identifier.

**Source internal spooled file identifier.** The internal identifier for the spooled file to be moved. Only the i5/OS APIs use this identifier; no other interface on the system does.

**Note:** The identifier is not valid following an initial program load (IPL) or if the spooled file has been moved between auxiliary storage pools (ASPs). If you attempt to use an invalid internal spooled file identifier, an exception occurs.

This field must be blank when a source spooled file name is given.

Use one of the following APIs to make the identifier available:

- “List Spooled Files (QUSLSPL) API” on page 279
- “Retrieve Spooled File Attributes (QUSRSPLA) API” on page 337

There may be a performance advantage when identifying the source spooled file by its internal identifier.

**Source job name.** The name of the job that owns the spooled file to be moved.

The possible values are:

- \* Only the job that this program is running. The job user name, job number, and internal identifier for the source job must be blank.
- \*INT The job to be moved is identified by the internal job identifier.
- Name The name of the job that owns the spooled file to be moved.

**Source job number.** The number of the job that owns the spooled file to be moved. It can optionally be blank when a job name or a job user name is specified. This field must be blank when the source job name is specified as \*INT or \*.

**Source job system name.** The name of the system where the job that created the source spooled file ran or blank when the source spooled file name is \*INT. This field is considered after the source job name, source user name, source job number, source spooled file name, and source spooled file number field requirements have been met.

The possible values are:

*ONLY	There is one job with the specified source job name, source job user name, source job number, source spooled file name, source spooled file number, source spooled file creation date, and source spooled file creation time.
*CURRENT	The job on the current system with the specified source job name, source job user name, source job number, source spooled file name, source spooled file number, source spooled file create date, and source spooled file create time is used.
*ANY	The source job system name is not considered when selecting a spooled file. Use this value when you want the source spooled file create date and source spooled file create time fields to take precedence over the source job system name when selecting a spooled file.
<i>system-name</i>	The name of the system where the job that created the source spooled file ran.

When the Length of function information parameter is less than 114 with format MSPF0100 or less than 188 with format MSPF0200, the API assumes blanks when the source spooled file name is \*INT. When source spooled file name is not \*INT, the API assumes \*ONLY.

**Source job user name.** The user name of the job that owns the spooled file to be moved. It can optionally be blank when a job name is specified. This field must be blank when the source job name is specified as \*INT or \*.

**Source spooled file create date.** » The date, based on local job time, that the source spooled file was created on the system or blank when the spooled file name is \*INT. « This field is considered after the source job name, source job user name, source job number, source spooled file name, source spooled file number, and source job system name field requirements have been met. The date must be in the CYYMMDD format or one of the following special values:

*ONLY	There is only one spooled file with the specified source job name, source user name, source job number, source spooled file name, source spooled file number, and source job system name.
*LAST	The spooled file with the latest date and time which also has the specified source job name, source job user name, source job number, source spooled file name, source spooled file number, and source job system name is used.
<i>date</i>	The date the source spooled file was created on the system in the format CYYMMDD. » See field Date file opened (created) in API QUSRSPLA « under field descriptions for more information about the date format.

When the Length of function information parameter is less than 114 with format MSPF0100 or less than 188 with format MSPF0200, the API assumes blanks when the spooled file name is \*INT. When spooled file name is not \*INT, the API assumes \*ONLY.

**Source spooled file create time.** » The time, in local job time, that the source spooled file was created on the system or blank when the source spooled file name is \*INT. « This field must be set to blanks when special values \*LAST or \*ONLY are used for field source spooled file create date. This field must have a value set if a date is specified for field source spooled file create date. This field is considered after the source job name, source job user name, source job number, source spooled file name, source spooled file number, source job system name, and source spooled file create date field requirements have been met. The time must be in the HHMMSS format or one of the following special values:

<i>*ONLY</i>	There is only one spooled file with the specified source job name, source job user name, source job number, source spooled file name, source spooled file number, source job system name, and source spooled file create date.
<i>*LAST</i>	The spooled file with the latest time which also has the specified source job name, source user name, source job number, source spooled file name, source spooled file number, source job system name, and source spooled file create date is used.
<i>time</i>	The time the source spooled file was created on the system in the format HHMMSS. >> See field Time file opened (created) in API QUSRSPLA << under field descriptions for more information about the time format.
<i>blanks</i>	This field must be set to blanks if the Source spooled file create date field is set to *LAST or *ONLY.

When the Length of function information parameter is less than 114 with format MSPF0100 or less than 188 with format MSPF0200, the API assumes blanks for this field.

**Source spooled file name.** The name of the spooled file to be moved.

The possible values are:

<i>*INT</i>	The source spooled file is identified by the internal spooled file identifier.
<i>Name</i>	The name of the spooled file to be moved.

**Source spooled file number.** The unique number of the spooled file to be moved. The valid range is -1 through 999999 and must be specified as such even if the value for the spooled file name field is \*INT.

The following special values are supported for this field:

0	Only one spooled file from the job has the specified file name; therefore, the number of the spooled file is not necessary.
-1	The highest-numbered spooled file with the specified file name.
-2	The source spooled file number is not used to determine which spooled file to process. Use this value when you want the source job system name field or the source spooled file create date and source spooled file create time fields to take precedence over the source spooled file number when selecting a spooled file.

**Target internal job identifier.** The internal identifier for the job of the spooled file after which the source spooled file is to be moved. Only the i5/OS APIs use this identifier; no other interface on the system does.

**Note:** The identifier is not valid following an initial program load (IPL). If you attempt to use it after an IPL, an exception occurs.

This field must be blank when a job name is given.

Use one of the following APIs to make the identifier available:

- "List Spooled Files (QUSLSPL) API" on page 279
- "Retrieve Spooled File Attributes (QUSRSPLA) API" on page 337

There may be a performance advantage when identifying the target job by its internal identifier.

**Target internal spooled file identifier.** The internal identifier for the spooled file after which the source spooled file is to be moved. Only the i5/OS APIs use this identifier; no other interface on the system does.

**Note:** The identifier is not valid following an initial program load (IPL) or if the spooled file has been moved between auxiliary storage pools (ASPs). If you attempt to use an invalid internal spooled file identifier, an exception occurs.

This field must be blank when a target spooled file name is given.

Use one of the following APIs to make the identifier available:

- “List Spooled Files (QUSLSPL) API” on page 279
- “Retrieve Spooled File Attributes (QUSRSPLA) API” on page 337

There may be a performance advantage when identifying the target spooled file by its internal identifier.

**Target job name.** The name of the job of the spooled file after which the source spooled file is to be moved.

The possible values are:

- \* Only the job that this program is running. The target job user name, target job number, and target internal identifier job must be blank.
- \*INT The target job is identified by the target internal job identifier.
- Name The name of the job that owns the spooled file after which the source spooled file is to be moved.

**Target job number.** The number of the target job. This is the number of the job that owns the spooled file after which the source spooled file is to be moved. It can optionally be blank when a target job name or target job user name is specified. This field must be blank when the target job name is specified as \*INT or \*.

**Target job system name.** The name of the system where the job that created the target spooled file ran or blank when the target spooled file name is \*INT. This field is considered after the target job name, target job user name, target job number, target spooled file name, and target spooled file number field requirements have been met.

The possible values are:

- \*ONLY There is one job with the specified target job name, target job user name, target job number, target spooled file name, target spooled file number, target spooled file creation date, and target spooled file creation time.
- \*CURRENT The job on the current system with the specified target job name, target job user name, and target job number is used.
- \*ANY The target job system name is not considered when selecting a spooled file. Use this value when you want the target spooled file create date and target spooled file create time fields to take precedence over the target job system name when selecting a spooled file.
- system-name The name of the system where the job that created the target spooled file ran.

When the Length of function information parameter is less than 188, the API assumes blanks when the spooled file name is \*INT. When spooled file name is not \*INT, the API assumes \*ONLY.

**Target job user name.** The user name of the target job. It can optionally be blank when a target job name is specified. This field must be blank when the target job name is specified as \*INT or \*.

**Target output queue library name.** The name of the library that contains the output queue.

The possible values are:

- \*LIBL The library list is used to locate the output queue.

<i>*CURLIB</i>	The current library is used to locate the output queue. If no library is specified as the current library for the job, QGPL is used.
<i>Name</i>	The library name.
<i>blank</i>	No output queue library is specified when the output queue name is specified as <i>*SAME</i> .

**Target output queue name.** The name of the output queue to which the source spooled file is to move to the top of.

The possible values are:

<i>*SAME</i>	The spooled file will move to the top of the output queue on which it currently resides.
<i>Name</i>	The name of the specific output queue the spooled file is to move to the top of.

**Target spooled file create date.** » The date, based on local job time, that the target spooled file was created on the system or blank when the target spooled file name is *\*INT*. « This field is considered after the target job name, target job user name, target job number, target spooled file name, target spooled file number, and target job system name field requirements have been met. The date must be in the CYYMMDD format or one of the following special values:

<i>*ONLY</i>	There is only one spooled file with the specified target job name, target job user name, target job number, target spooled file name, target spooled file number, and target job system name.
<i>*LAST</i>	The spooled file with the latest date and time which also has the specified target job name, target job user name, target job number, target spooled file name, target spooled file number, and target job system name is used.
<i>date</i>	The date the target spooled file was created on the system in the format CYYMMDD. » See field Date file opened (created) in API QUSRSPLA « under field descriptions for more information about the date format.

When the Length of function information parameter is less than 188, the API assumes blanks when the target spooled file name is *\*INT*. When target spooled file name is not *\*INT*, the API assumes *\*ONLY*.

**Target spooled file create time.** » The time, in local job time, that the target spooled file was created on the system or blank when the target spooled file name is *\*INT*. « This field must be set to blanks when special values *\*LAST* or *\*ONLY* are used for field target spooled file create date. This field must have a value set if a date is specified for field target spooled file create date. This field is considered after the target job name, target job user name, target job number, target spooled file name, target spooled file number, target job system name, and target spooled file create date field requirements have been met. The time must be in the HHMMSS format or one of the following special values:

<i>*ONLY</i>	There is only one spooled file with the specified target job name, target job user name, target job number, target spooled file name, target spooled file number, target job system name, and target spooled file create date.
<i>*LAST</i>	The spooled file with the latest time which also has the specified target job name, target job user name, target job number, target spooled file name, target spooled file number, target job system name, and target spooled file create date is used.
<i>time</i>	The time the target spooled file was created on the system in the format HHMMSS. » See field Time file opened (created) in API QUSRSPLA « under field descriptions for more information about the time format.
<i>blanks</i>	This field must be set to blanks if the Target spooled file create date field is set to <i>*LAST</i> or <i>*ONLY</i> .

When the Length of function information parameter is less than 188, the API assumes blanks for this field.

**Target spooled file name.** The name of the spooled file after which the source spooled file is to be moved.

The possible values are:

- \*INT*                    The target spooled file is identified by the target internal spooled file identifier.
- Name*                    The name of the spooled file after which the source spooled file is to be moved.

**Target spooled file number.** The unique number of the spooled file after which the source spooled file is to be moved. The valid range is -1 through 999999 and must be specified as such even if the value for the spooled file name field is *\*INT*. The following special values are supported for this field:

- 0                    Only one spooled file from the job has the specified file name; therefor, the number of the spooled file is not necessary.
- 1                    The highest-numbered spooled file with the specified file name.
- 2                    The target spooled file number is not used to determine which spooled file to process. Use this value when you want the target job system name field or the target spooled file create date and target spooled file create time fields to take precedence over the target spooled file number when selecting a spooled file.

## How to Specify Spooled File Identifying Fields

This table illustrates the valid field combinations of qualified job name, internal job identifier, internal spooled file identifier, spooled file name, spooled file number, job system name, spooled file create date, and spooled file create time. The combinations of these fields identify the spooled file to be moved and the spooled file it is to be moved after. For example, when the qualified job name field value is *\**, the internal job identifier must be blank, the internal spooled file identifier must be blank, the actual name of the spooled file must be given, and a valid spooled file number must be given.

Spooled File Identifying Fields

Qualified Job Name			Internal Job Identifier	Internal Spooled File Identifier	Spooled File Name	Spooled File Number	Job System Name	Spooled File Create Date	Spooled File Create Time
Job Name	User Name	Job Number							
Name	Name	Number	Blanks	Blanks	Name	-2 through 999999	Name, *ONLY, *CURRENT, or *ANY	Date, *ONLY, or *LAST	Time, blanks, *ONLY, or *LAST
Name	Name	Blanks	Blanks	Blanks	Name	-2 through 999999	Name, *ONLY, *CURRENT, or *ANY	Date, *ONLY, or *LAST	Time, blanks, *ONLY, or *LAST
Name	Blanks	Blanks	Blanks	Blanks	Name	-2 through 999999	Name, *ONLY, *CURRENT, or *ANY	Date, *ONLY, or *LAST	Time, blanks, *ONLY, or *LAST
*	Blanks	Blanks	Blanks	Blanks	Name	-2 through 999999	Name, *ONLY, *CURRENT, *ANY	Date, *ONLY, or *LAST	Time, blanks, *ONLY, or *LAST
*INT	Blanks	Blanks	Internal job identifier	Internal spooled file identifier	*INT	-1 through 999999	Blanks	Blanks	Blanks

Qualified Job Name			Internal Job Identifier	Internal Spooled File Identifier	Spooled File Name	Spooled File Number	Job System Name	Spooled File Create Date	Spooled File Create Time
Job Name	User Name	Job Number							
*INT	Blanks	Blanks	Internal job identifier	Blanks	Name	-2 through 999999	Name, *ONLY, *CURRENT, or *ANY	Date, *ONLY, or *LAST	Time, blanks, *ONLY, or *LAST  See Notes for additional information.

**Notes:** This parameter combination is not valid when a job has been detached from its spooled files or for a spooled file on an independent disk pool. Use of this combination will result in message CPF3C43.

## Error Messages

Message ID	Error Message Text
CPF2207 E	Not authorized to use object &1 in library &3 type *&2.
CPF24B4 E	Severe error while addressing parameter list.
CPF3CF1 E	Error code parameter not valid.
CPF3C1D E	Length specified in parameter &1 not valid.
CPF3C21 E	Format name &1 is not valid.
CPF3C33 E	Spooled file number &1 is not valid.
CPF3C40 E	Spooled file &4 not found.
CPF3C41 E	More than one spooled file with same name.
CPF3C42 E	User name or job number is not blank.
CPF3C43 E	Internal job identifier is not valid.
CPF3C44 E	Internal spooled file identifier is not valid.
CPF3C58 E	Job name specified is not valid.
CPF3C90 E	Literal value cannot be changed.
CPF33AA E	Target spooled file &1 number &2 in job &5/&4/&3 in open status.
CPF33AB E	Target spooled file &1 number &2 in job &5/&4/&3 in closed status.
CPF33AC E	Spooled file &1 number &2 in job &5/&4/&3 in deferred status.
CPF33AD E	Target spooled file &1 not last spooled file in ready status. Source spooled file not moved.
CPF33AE E	Spooled file &1 number &2 in job &5/&4/&3 not moved.
CPF33AF E	Duplicate spooled file &1 number &2 in job &3/&4/&5 found.
CPF33A6 E	Spooled file &1 selected by writer. Spooled file not moved.
CPF33A8 E	Spooled file &1 specified more than once. Spooled file not moved.
CPF33A9 E	Target spooled file &1 changed output queue. Source spooled file not moved.
CPF33CA E	Output queue &1 in library &2 is not valid.
CPF33CB E	Priority required to move spooled file exceeds user's limit.
CPF33C2 E	Moving spooled files to the top allowed only for output queues with SEQ(*FIFO).
CPF33C3 E	Priority required to move spooled file exceeds user's limit.
CPF33C4 E	Spooled file &1 held by HLDJOB command. Spooled file not moved.
CPF33C5 E	Target spooled file &1 selected by writer. Source spooled file not moved.
CPF33C6 E	Priority required to move file exceeds user's limit.
CPF33C7 E	Cannot move file ahead of other users' files.
CPF33C9 E	Spooled file name parameter cannot be blank.
CPF3309 E	No files named &1 are active.
CPF3330 E	Necessary resource not available.
CPF333B E	Job system name is not valid.
CPF333C E	Spooled file create date is not valid.

Message ID	Error Message Text
CPF333D E	Spooled file create time is not valid.
CPF333E E	Spooled file create time is not blank.
CPF333F E	Job system name is not blank.
CPF3342 E	Job &5/&4/&3 not found.
CPF3343 E	Duplicate job names found.
CPF3344 E	File &1 number &2 no longer in the system.
CPF335B E	Spooled file create date is not blank.
» CPF338C E	Internal spool control file inaccessible. «
CPF3410 E	New output queue &1 in &2 not found.
CPF3492 E	Not authorized to spooled file.
CPF8122 E	&8 damage on library &4.
CPF8128 E	&8 damage on output queue &4 in library &9.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.
CPI33C2 I	Spooled file &1 number &2 in job &5/&4/&3 not moved to position requested.

API introduced: V3R1

Top | "Print APIs," on page 1 | APIs by category

---

## Open List of Spooled Files (QGYOLSPL) API

Required Parameter Group:

1	Receiver variable	Output	Char(*)
2	Length of receiver variable	Input	Binary(4)
3	List information	Output	Char(80)
4	Number of records to return	Input	Binary(4)
5	Sort information	Input	Char(*)
6	Filter information	Input	Char(*)
7	Qualified job name	Input	Char(26)
8	Format of the generated list	Input	Char(8)
9	Error code	I/O	Char(*)

Optional Parameter Group:

10	Format of filter information	Input	Char(8)
----	------------------------------	-------	---------

Default Public Authority: \*USE

Threadsafe: No

The Open List of Spooled Files (QGYOLSPL) API generates a list of spooled files on the system. It creates a list similar to the list created by calling the List Spooled Files (QUSLSPL) API. The list can include all spooled files or those of specific users, output queues, form types, user-specified data values, statuses, printers, or specific jobs. The filtered list can then be sorted depending on the value of the sort parameter. You can specify how many list entries to return when calling this API. At a later time, you can request additional entries from the list using the Get List Entries (QGYGTLE) API. You can close this list by using the Close List (QGYCLST) API.

**Note:** The QTEMP library and the system portion of the library list could be different between the main job and the server job when the list is being built asynchronously. If this is a problem, request that the list be built synchronously.

For more information, see Process Open List APIs.

## Differences between QUSLSPL and QGYOLSPL

- The QUSLSPL API puts the list of spooled files into a user space object provided by the caller. The calling program can directly access the user space storage. The QGYOLSPL API puts the list of spooled files into a temporary space and assigns a handle for the list. The calling program cannot directly access the storage where the list is placed. Instead, the calling program can retrieve entries from the list by passing the assigned list handle to the Get List Entries (QGYGTLE) API.
- The QUSLSPL API does not return control to the calling program until the list of spooled files is completely built in the user space object. The QGYOLSPL API returns control as soon as the initial number of records requested has been placed in the list. If the calling program requests zero records to be returned initially, QGYOLSPL returns immediately after it verifies parameter values, does some initial setup operations, and requests that the asynchronous host server job builds the list.
- The QGYOLSPL has a receiver variable parameter. The receiver variable contains the initial number of records, if any, requested.
- The QGYOLSPL has a list information parameter that contains the status information about the list that was built or is being built.
- The QUSLSPL API allows filtering on only one user profile name. The QGYOLSPL API allows filtering on multiple user profile names.
- The QUSLSPL API allows filtering on only one output queue name. The QGYOLSPL API allows filtering on multiple output queue names.
- The QUSLSPL API does not allow filtering on spooled file status. The QGYOLSPL API allows filtering on multiple spooled file statuses.
- The QUSLSPL API does not allow filtering on printer name. The QGYOLSPL API allows filtering on multiple printer names.
- The QUSLSPL API returns the list of spooled files in only one order. The QGYOLSPL API has a sort information parameter to allow sorting on one or more of the fields in the list.

## Performance Impacts

Filtering on multiple values of user name, qualified output queue name, or user-specified data takes more processing power and time.

Additionally, filtering on a status or a printer device name takes more processing power and time.

## Authorities and Locks

*Output Queue Authority*  
\*USE

*Output Queue Library Authority*  
\*EXECUTE

## Required Parameter Group

### Receiver variable

OUTPUT; CHAR(\*)

The variable used to return the number of records requested (given in the number of records to return parameter) of spooled file information. See “Format of Receiver Variable” on page 311 for a description of the layout of this parameter.

### Length of receiver variable

INPUT; BINARY(4)

The length of the receiver variable.

### List information

OUTPUT; CHAR(80)

Information about the list created by this program. For a description of the layout of this parameter, see Open list information format.

### Number of records to return

INPUT; BINARY(4)

The number of records in the list to put into the receiver variable after filtering and sorting has been done.

If -1 is specified for this parameter the entire list is built synchronously.

If 0 is specified for this parameter the entire list is built asynchronously in a server job.

If a positive number of records to return is specified, at least that many records will be built synchronously and the remainder will be built asynchronously in a server job.

### Sort information

INPUT; CHAR(\*)

Information about what fields within the record of information to sort. See "Format of Sort Information" on page 316 for a description of the layout of this parameter. Note that when sorting is requested, the entire list has to be built and sorted before any records can be returned.

### Filter information

INPUT; CHAR(\*)

The information in this parameter is used for filtering the list of spooled file information. For more details about filtering options, see "Format of Filter Information" on page 314.

### Qualified job name

INPUT; CHAR(26)

The qualified job name of the job whose files are to be included in the list. This parameter can be blank. If blanks are specified, spooled files will be listed for all jobs that match the criteria specified in the filter information parameter.

The qualified job name has three parts:

<i>Job name</i>	CHAR(10). A specific job name or the following special value:  * Current job. The rest of the job name parameter must be blank.
<i>User name</i>	CHAR(10). A specific user profile name or blanks when the job name is *.
<i>Job number</i>	CHAR(6). A specific job number or blanks when the job name is *.

### Format of the generated list

INPUT; CHAR(8)

The format of the spooled file information being returned.

You must specify one of the following:

» "OSPL0100 Format" on page 311	Contains the basic information about the spooled file.
"Format OSPL0200" on page 312	Contains all the information in format OSPL0100 plus additional information about the spooled file.
"Format OSPL0300" on page 313	Contains the basic information about the spooled file. This format differs from format OSPL0100 in that it provides slightly different information with less processing time.
OSPL0400 (page "Format OSPL0400" on page 313)	Contains the basic information about the spooled file. This format differs from format OSPL0300 in that it provides dates and times in Coordinated Universal time (UTC) instead of local job time. «

The OSPL0100 format returns information faster than the OSPL0200 format.

» The OSPL0300 and OSPL0400 formats return information faster than the OSPL0100 format. «

### Error code

I/O; CHAR(\*)

The structure in which to return error information. For the format of the structure, see Error code parameter.

## Optional Parameter Group

### Format of filter information

INPUT; CHAR(8)

The format of the filter information passed in the Filter information parameter.

*“Filter Information Format  
OSPF0100” on page 314*

Defines the format of the filter information when filtering on one or more of the following:

- User name
- Output queue name and library
- Form type
- User data
- Spooled file status
- Printer Device

*“Filter Information Format  
OSPF0200” on page 315*

Defines the format of the filter information when filtering on one or more of the following:

- User name
- Output queue name and library
- Spooled file status
- Printer device
- Form type
- User data
- Job system name
- Starting spooled file creation date
- Starting spooled file creation time
- Ending spooled file creation date
- Ending spooled file creation time

## Format of Receiver Variable

The following tables describe the order and format of the data returned in the receiver variable.

### OSPL0100 Format

For more details about the fields in the following table, see “Field Descriptions” on page 317.

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	Spooled file name
10	A	CHAR(10)	Job name
20	14	CHAR(10)	User name
30	1E	CHAR(6)	Job number
36	24	BINARY(4)	Spooled file number

Offset		Type	Field
Dec	Hex		
40	28	BINARY(4)	Total pages
44	2C	BINARY(4)	Current page
48	30	BINARY(4)	Copies left to print
52	34	CHAR(10)	Output queue name
62	3E	CHAR(10)	Output queue library name
72	48	CHAR(10)	User data
82	52	CHAR(10)	Status
92	5C	CHAR(10)	Form type
102	66	CHAR(2)	Priority
104	68	CHAR(16)	Internal job identifier
120	78	CHAR(16)	Internal spooled file identifier
136	88	CHAR(10)	Device type
146	92	CHAR(2)	Reserved
148	94	BINARY(4)	Offset to OSPL0100 extension
152	98	BINARY(4)	Length of OSPL0100 extension
156	9C	CHAR(4)	Reserved

### Format OSPL0100 extension

For more details about the fields in the following table, see “Field Descriptions” on page 317.

Offset		Type	Field
Dec	Hex		
0	0	CHAR(8)	Job system name
8	8	CHAR(7)	» Date spooled file was opened (created) «
15	F	CHAR(6)	» Time spooled file was opened (created) «
» 21	15	CHAR(7)	Date spooled file was created in UTC
28	12	CHAR(6)	Time spooled file was created in UTC
34	22	CHAR(2)	Reserved «

### Format OSPL0200

For more details about the fields in the following table, see “Field Descriptions” on page 317.

Offset		Type	Field
Dec	Hex		
0	0		Everything in format OSPL0100 (not including extension)
160	A0	CHAR(7)	» Date spooled file was opened (created) «
167	A7	CHAR(6)	» Time spooled file was opened (created) «
173	AD	CHAR(1)	Printer assigned
174	AE	CHAR(10)	Printer name
» 184	B8	CHAR(7)	Date spooled file was created in UTC

Offset		Type	Field
Dec	Hex		
191	BF	CHAR(6)	Time spooled file was created in UTC
197	C5	CHAR(3)	Reserved <<

### Format OSPL0300

For more details about the fields in the following table, see “Field Descriptions” on page 317.

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	Job name
10	A	CHAR(10)	User name
20	14	CHAR(6)	Job number
26	1A	CHAR(10)	Spooled file name
36	24	BINARY(4)	Spooled file number
40	28	BINARY(4)	File status
44	2C	CHAR(7)	>> Date spooled file was opened (created) <<
51	33	CHAR(6)	>> Time spooled file was opened (created) <<
57	39	CHAR(1)	Spooled file schedule
58	3A	CHAR(10)	Job system name
68	44	CHAR(10)	User data
78	4E	CHAR(10)	Spooled file form type
88	58	CHAR(10)	Output queue name
98	62	CHAR(10)	Output queue library name
108	6C	BINARY(4)	Auxiliary storage pool
112	70	BINARY(4)	Size of spooled file
116	74	BINARY(4)	Spooled file size multiplier
120	78	BINARY(4)	Total pages
124	7C	BINARY(4)	Copies left to print
128	80	CHAR(1)	Priority
129	81	CHAR(3)	Reserved
132	84	BINARY(4)	Internet print protocol job identifier

### Format OSPL0400

For more details about the fields in the following table, see “Field Descriptions” on page 317.

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	Job name
10	A	CHAR(10)	User name
20	14	CHAR(6)	Job number
26	1A	CHAR(10)	Spooled file name

Offset		Type	Field
Dec	Hex		
36	24	BINARY(4)	Spooled file number
40	28	BINARY(4)	File status
44	2C	CHAR(7)	Date spooled file was created in UTC
51	33	CHAR(6)	Time spooled file was created in UTC
57	39	CHAR(1)	Spooled file schedule
58	3A	CHAR(10)	Job system name
68	44	CHAR(10)	User data
78	4E	CHAR(10)	Spooled file form type
88	58	CHAR(10)	Output queue name
98	62	CHAR(10)	Output queue library name
108	6C	BINARY(4)	Auxiliary storage pool
112	70	BINARY(4)	Size of spooled file
116	74	BINARY(4)	Spooled file size multiplier
120	78	BINARY(4)	Total pages
124	7C	BINARY(4)	Copies left to print
128	80	CHAR(1)	Priority
129	81	CHAR(3)	Reserved
132	84	BINARY(4)	Internet print protocol job identifier



## Format of Filter Information

### Filter Information Format OSPF0100

For more details about the fields in the following table, see “Field Descriptions” on page 317.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Number of user names
These fields are repeated for each user to filter on. The offsets vary.		CHAR(10)	User name
		CHAR(2)	Reserved
		BINARY(4)	Number of output queue names
These fields are repeated for each output queue to filter on. The offsets vary.		CHAR(10)	Output queue name
		CHAR(10)	Output queue library name
		CHAR(10)	Form type
		CHAR(10)	User-specified data
		BINARY(4)	Number of statuses

Offset		Type	Field
Dec	Hex		
These fields are repeated for each status to filter on. The offsets vary.		CHAR(10)	Spoiled file status
		CHAR(2)	Reserved
		BINARY(4)	Number of printer device names
These fields are repeated for each printer device to filter on. The offsets vary.		CHAR(10)	Printer device name
		CHAR(2)	Reserved

### Filter Information Format OSPF0200

For more details about the fields in the following table, see “Field Descriptions” on page 317.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Length of filter information
4	4	BINARY(4)	Offset to user name entries
8	8	BINARY(4)	Number of user name entries
12	C	BINARY(4)	Length of user name entry
16	10	BINARY(4)	Offset to output queue name entries
20	14	BINARY(4)	Number of output queue name entries
24	18	BINARY(4)	Length of output queue name entry
28	1C	BINARY(4)	Offset to spoiled file status entries
32	20	BINARY(4)	Number of spoiled file status entries
36	24	BINARY(4)	Length of spoiled file status entry
40	28	BINARY(4)	Offset to printer device name entries
44	2C	BINARY(4)	Number of printer device name entries
48	30	BINARY(4)	Length of printer device name entry
52	34	CHAR(10)	Form type
62	3E	CHAR(10)	User-specified data
72	48	CHAR(8)	System name
80	50	CHAR(7)	Starting spoiled file create date
87	57	CHAR(6)	Starting spoiled file create time
93	5D	CHAR(7)	Ending spoiled file create date
100	64	CHAR(6)	Ending spoiled file create time
106	6A	CHAR(4)	Reserved

### User Name Entry

For more details about the fields in the following table, see “Field Descriptions” on page 317.

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	User name

## Output Queue Name Entry

For more details about the fields in the following table, see “Field Descriptions” on page 317.

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	Output queue name
10	A	CHAR(10)	Output queue library name

## Spooled File Status Entry

For more details about the fields in the following table, see “Field Descriptions” on page 317.

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	Spooled file status

## Printer Device Name Entry

For more details about the fields in the following table, see “Field Descriptions” on page 317.

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	Printer device name

## Format of Sort Information

For more details about the fields in the following table, see “Field Descriptions” on page 317.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Number of keys to sort on
Offsets vary. These fields repeat for each sort key field.		BINARY(4)	Sort key field starting position
		BINARY(4)	Sort key field length
		BINARY(2)	Sort key field data type
		CHAR(1)	Sort order
		CHAR(1)	Reserved

**Note:** If the last 3 fields (sort key field data type, sort order, and the reserved field) are not used, then they must be set to hexadecimal zeros. This causes all the data to be treated as character data, and it is sorted in ascending order.

## Field Descriptions

**Auxiliary storage pool.** The auxiliary storage pool in which the spooled file resides.

Valid values are:

1	*SYSTEM. System auxiliary storage pool
2-255	The number of the auxiliary storage pool.

**Copies left to print.** The remaining number of copies to be printed. This attribute applies to printer device type spooled files only.

**Current page.** The page number or record number currently being written. The page number may be lower or higher than the page number actually being printed because of buffering done by the system. For example, if the spooled file is routed to a diskette unit or the writer is currently printing job separators or file separators for the spooled file, the page number shown may be zero.

» **Date spooled file was created in UTC.** The date, based on Coordinated Universal Time (UTC), that the file was opened in the CYYMMDD format.

C	Century, where 0 indicates years 19xx and 1 indicates years 20xx.
YY	Year
MM	Month
DD	Day



» **Date spooled file was opened (created).** The date, based on local job time, that the file was opened in the CYYMMDD format. «

C	Century, where 0 indicates years 19xx and 1 indicates years 20xx.
YY	Year
MM	Month
DD	Day

**Device type.** The type of device for which the spooled file is intended. The possible values are PRINTER or DISKETTE.

**Ending spooled file create date.** » The date, based on local job time, that the spooled file was created on the system. « If the Starting spooled file create date field is set to \*ALL, then this field must be set to blanks. If a date has been specified for the Starting spooled file create date field, then this field must be set to a valid date. The date must be in the CYYMMDD format or one of the following special values:

*LAST	All spooled files with a create date and time equal to or later than the starting spooled file create date are to be returned.
date	All spooled files with a create date and time equal to or later than the starting spooled file create date and time and less than or equal to the ending spooled file create date and time are to be returned.

The date format CYYMMDD is defined as follows:

C	Century, where 0 indicates years 19xx and 1 indicates years 20xx.
YY	Year
MM	Month
DD	Day

**Ending spooled file create time.** » The time, in local job time, that the spooled file was created on the system. « This field must be set to blanks when special value \*ALL is used for field Starting spooled file create date or when special value \*LAST is used for field Ending spooled file create date. This field must have a value set if a date is specified for field Ending spooled file create date. The time must be in the HHMMSS format.

The time format HHMMSS is defined as follows:

HH	Hour
MM	Minutes
SS	Seconds

**File status.** The status of the file. The following list of values is used to describe the status of the file:

- 1 RDY The file is available to be written to an output device by a writer.
- 2 OPN The file has not been processed completely and is not ready to be selected by a writer.
- 3 CLO The file has been processed completely by a program, but SCHEDULE(\*JOBEND) was specified. The job that produced the spooled file has not finished.
- 4 SAV The file has been written and then saved. This file remains saved until it is released.
- 5 WTR This file currently is being produced by the writer on an output device.
- 6 HLD The file has been held.
- 7 MSGW This file has a message that needs a reply or needs an action to be taken.
- 8 PND This file is pending (waiting) to be printed.
- 9 PRT The file has been sent to the printer completely, but the print complete status has not been sent back.
- 10 FIN This spooled file is no longer in the system. These spooled files are included in the list of spooled files only if the qualified job name is specified.
- 11 SND This spooled file is being sent or has been sent to a remote system.
- 12 DFR This spooled file has been deferred from printing.

**Form type.** The type of forms that should be loaded on the printer before this spooled file is printed or spooled files whose form type matches this value are included in the list. This attribute applies to printer device type spooled files only. This field can be used in conjunction with the user, output queue and library, user-specified data, status, device fields and qualified job name parameter to subset the list of all the spooled files.

The following special values can be used for the form type:

*ALL	Spooled files are included in the list regardless of the value for form type.
*STD	Only files that specify the standard form type are included in the list.

**Internal job identifier.** The internal identifier for the job. Only the i5/OS® APIs use this identifier, not any other interface on the system. The identifier is not valid following an initial program load (IPL). If you attempt to use it after an IPL, an exception occurs. This identifier is always valid when used with the internal spooled file identifier as input to i5/OS APIs. For spooled files which have been detached from their jobs or which are on independent disk pools, this value is not valid as input to some i5/OS APIs, such as the Retrieve Job Information (QUSRJOBI) API, that take this value without the internal spooled file identifier.

**Internal spooled file identifier.** The input value that other programs use to improve the performance of locating the spooled file on the system. Only the spooled file APIs use this identifier, not any other interface on the system. The identifier is not valid following an initial program load (IPL) » or if the spooled file has been moved between auxiliary storage pools (ASPs). If you attempt to use an invalid internal spooled file identifier, an exception occurs. «

**Internet print protocol job identifier.** The IPP job identifier assigned by the system based on the output queue to which the file was added or moved. This value ranges from 1 to 2147483647 and is not guaranteed to be unique for a given output queue.

**Job name.** The name of the job that owns the spooled file.

**Job number.** The number of the job that owns the spooled file.

**Job system name.** The name of the system where the job that created the spooled file ran.

**Length of filter information.** The number of bytes of the fixed portion for the filter information provided. If this value is greater than 106, the extra bytes must be set to hexadecimal zeros.

» Length of OSPL0100 extension. The size, in bytes, of the OSPL0100 format extension. «

**Length of output queue name entry.** The size, in bytes, of the output queue name entry. The minimum allowed size is 20. The value of this field will be used by the API to find the next output queue name entry.

**Length of printer device name entry.** The size, in bytes, of the printer device name entry. The minimum allowed size is 10. The value of this field will be used by the API to find the next printer device name entry.

**Length of spooled file status entry.** The size, in bytes, of the spooled file status entry. The minimum allowed size is 10. The value of this field will be used by the API to find the next spooled file status entry.

**Length of user name entry.** The size, in bytes, of the user name entry. The minimum allowed size is 10. The value of this field will be used by the API to find the next user name entry.

**Number of printer device names.** The number of printer device names to filter on. The value specified must be greater than or equal to 1 for filter format OSPF0100. The value specified must be greater than or equal to 0 for filter format OSPF0200. With filter format OSPF0200, a value of 0 (\*ALL) indicates spooled files are included in the list regardless of the printer device name.

Spooled files whose assigned printer device matches one of the printer device names specified are included in the list.

**Number of keys to sort on.** The number of fields within the record structure to sort on. If 0 is specified, the list is not sorted.

**Number of output queue name entries.** The number of output queues to filter on. The value specified must be greater than or equal to 1.

Spooled files whose output queue matches one of the output queues specified are included in the list.

**Number of output queue names.** The number of output queues to filter on. The value specified must be greater than or equal to 0. A value of 0 indicates spooled files are included in the list regardless of the output queue the spooled file resides on.

Spooled files whose output queue matches one of the output queues specified are included in the list.

**Number of spooled file status entries.** The number of spooled file statuses to filter on. The value specified must be greater than or equal to 0. A value of 0 indicates spooled files are included in the list regardless of the status of the spooled file.

Spooled files whose current status matches one of the statuses specified are included in the list.

**Number of statuses.** The number of spooled file statuses to filter on. The value specified must be greater than or equal to 1.

Spooled files whose current status matches one of the statuses specified are included in the list.

**Number of user name entries.** The number of user names to filter on. The value specified must be greater than or equal to 0. A value of 0 indicates spooled files are included in the list regardless of the user name.

Spooled files owned by one of the user profile names specified are included in the list.

**Number of user names.** The number of user names to filter on. The value specified must be greater than or equal to 1.

Spooled files owned by one of the user profile names specified are included in the list.

» Offset to OSPL0100 extension. The offset, in bytes, from the beginning of the structure, to the OSPL0100 format extension. «

**Offset to output queue name entries.** The offset, in bytes, from the beginning of the structure, to one or more output queue name entries.

**Offset to printer device name entries.** The offset, in bytes, from the beginning of the structure, to one or more printer device name entries.

**Offset to spooled file status entries.** The offset, in bytes, from the beginning of the structure, to one or more spooled file status entries.

**Offset to user name entries.** The offset, in bytes, from the beginning of the structure, to one or more user name entries.

**Output queue library name.** The name of the library where the output queue resides.

The following special values can be used for the output queue library name when filtering:

<i>*CURLIB</i>	The current library for the current job.
<i>*LIBL</i>	The library list of the current job.

**Output queue library name.** The library where the output queue is located.

**Output queue name.** The name of the output queue to filter on or the name of the output queue in which the spooled file is located. At least one output queue or special value must be specified. This field can be used in conjunction with the user, form type, user-specified data, status, device fields and qualified job name parameter to subset the list of all the spooled files.

The following special value is used for the output queue name when filtering:

<i>*ALL</i>	Spooled files are included in the list regardless of the output queue. When this value is specified, the output queue library name must be blanks. The field Number of output queue names must be set to 1 when using filtering information format OSPF0100. The field Number of output queue name entries must be set to 0 to signify *ALL when using filtering information format OSPF0200.
-------------	---

**Printer assigned.** Set to 1 when the spooled file is assigned to a specific printer. In this case, the printer name field contains the name of the assigned printer. Set to 2 when the spooled file is assigned to multiple printers. In this case, the printer name is set to blanks. Set to 3 when the spooled file is not assigned to a printer. In this case, the printer name is set to blanks. This attribute applies to printer device type spooled files only.

**Printer device name.** The name of the printer device to filter on. At least one printer device name or special value must be specified. This field can be used in conjunction with the user, output queue and library, form type, user-specified data, status fields and the qualified job name parameter to subset the list of all the spooled files. This field is not valid for format OSPL0100 and will be ignored for format OSPL0300 unless the only filtering requested is for a single printer device name. **Note:** For filtering information format OSPF0100, the first 10 characters contain the printer device, and the next 2 characters are reserved.

The following special value can be used for the device name:

*\*ALL* Spooled files are included in the list regardless of the device name. When this value is specified, the Number of printer device names to filter on must be 1.

**Printer name.** The name of the printer the spooled file has been assigned to print on. This attribute applies to printer device type spooled files only.

**Priority.** The priority of the spooled file. The priority ranges from 1 (highest) to 9 (lowest).

**Reserved.** Reserved field. Must be set to hexadecimal or binary zero.

**Size of spooled file.** The spooled file size. Multiply this field by the spooled file size multiplier field to get the size of the spooled file in number of bytes. The size of the spooled file is the data stream size plus the spooled file's attributes, plus the "overhead" storage used to store the spooled file's data stream.

**Sort key field data type.** Data type of field to sort. Refer to QLG SORT API for information about the list of data types available.

**Sort key field length.** The length of the field to sort on.

**Sort key field starting position.** Within the record of information, the starting position of the field to sort on.

**Sort order.** Used to specify if the list should be sorted in ascending or descending order according to the key.

**Spooled file form type.** The type of form to load in the printer to print this file.

**Spooled file name.** The name of the spooled file in the list entry.

**Spooled file number.** The number of the spooled file in the list entry.

**Spooled file schedule.** The schedule of the spooled file. Valid values are:

- 1 The spooled file is schedule immediate. A spooling writer can process the spooled file immediately.
- 2 The spooled file is schedule file end. A spooling writer cannot process the spooled file until it has been closed.
- 3 The spooled file is schedule job end. A spooling writer cannot process the spooled file until the job of the spooled file has ended.

**Spooled file size multiplier.** The number to multiply the size of spooled file field to get the size of the spooled file in number of bytes.

**Spooled file status.** The spooled file status to filter on.

Refer to the QUSLSPL API for possible values for the spooled file status. At least one status or special value must be specified. This field can be used in conjunction with the user, output queue and library, form type, user-specified data, device fields and the qualified job name parameter to subset the list of all the spooled files.

**Note:** When using filter format OSPF0100, the first 10 characters contain the status, and the next 2 characters are reserved.

The following special values can be used for the spooled file status:

*ALL	Spooled files are included in the list regardless of the current status. The field Number of statuses must be set to 1, when using filter information format OSPF0100. The field Number of spooled file status entries must be set to 0 to signify *ALL when using filter information format OSPF0200.
*CLOSED	The file has been completely processed by a program, but SCHEDULE(*JOBEND) was specified. The job that produced the spooled file has not finished.
*DEFERRED	This spooled file has been deferred from printing.
*SENDING	This spooled file is being sent or has been sent to a remote system.
*FINISHED	This spooled file is no longer in the system. These spooled files are included in the list of spooled files only if the qualified job name is specified.
*HELD	The file has been held.
*MESSAGE	This file has a message that needs a reply or needs an action to be taken.
*OPEN	The file has not been completely processed and is not ready to be selected by a writer.
*PENDING	This file is pending (waiting) to be printed.
*PRINTER	The file has been completely sent to the printer, but the print complete status has not been sent back.
*READY	The file is available to be written to an output device by a writer.
*SAVED	The file has been written and then saved. This file remains saved until it is released.
*WRITING	This file is currently being produced by the writer on an output device.

**Starting spooled file create date.** » The date, based on local job time, that the spooled file was created on the system. « The date must be in the CYYMMDD format or one of the following special values:

*ALL	The returned list is not to be filtered based on spooled file create date and spooled file create time.
*FIRST	All spooled files starting with the earliest create date and time and less than or equal to the ending spooled file create date and time are to be returned.
date	All spooled files with a create date and time equal to or later than the starting spooled file create date and time and less than or equal to the ending spooled file create date and time are to be returned.

The date format CYYMMDD is defined as follows:

YY     Year

MM     Month

DD     Day

C       Century, where 0 indicates years 19xx and 1 indicates years 20xx.

**Starting spooled file create time.** » The time, in local job time, that the spooled file was created on the system. « This field must be set to blanks when special value \*ALL or \*FIRST is used for field Starting spooled file create date. This field must have a value set if a date is specified for field Starting spooled file create date. The time must be in the HHMMSS format.

The time format HHMMSS is defined as follows:

<i>HH</i>	Hour
<i>MM</i>	Minutes
<i>SS</i>	Seconds

**System name.** The name of the system where the job that created the spooled file ran. The following special values are supported for this field:

<i>*ALL</i>	The returned list is not to be filtered based on job system name.
<i>*CURRENT</i>	Only spooled files created on the current system are to be returned.
<i>job-system-name</i>	Only spooled files created on the system specified are to be returned.

» Time spooled file was created in UTC. The time, in Coordinated Universal Time (UTC), that the file was opened in the HHMMSS format.

<i>HH</i>	Hour
<i>MM</i>	Minute
<i>SS</i>	Second

«

» Time spooled file was opened (created). The time, in local job time, that the file was opened in the HHMMSS format. «

<i>HH</i>	Hour
<i>MM</i>	Minute
<i>SS</i>	Second

**Total pages.** The total number of pages or number of records for a particular spooled file.

**User data.** The 10 characters of user-specified data that describe the file.

**User name.** The name of the user that owns the spooled file or the name of the user to filter on. At least one user name or special value must be specified. This field can be used in conjunction with the output queue and library, form type, user-specified data, status, device fields and qualified job name parameter to subset the list of all the spooled files.

**Note:** When using filtering information format OSPF0100, the first 10 characters contain the user name, and the next 2 characters are reserved.

The following special values can be used for the user name to filter on:

<i>*ALL</i>	Spooled files are included in the list regardless of the creating user profile. The field Number of user names must be set to 1 when using filtering format OSPF0100. The field Number of user name entries must be set to 0 to signify <i>*ALL</i> when using filtering format OSPF0200.
<i>*CURRENT</i>	Spooled files owned by the current user profile are included in the list.

**User-specified data.** Only spooled files whose user-specified data or file name match this value are included in the list. This field can be used in conjunction with the user, output queue and library, form type, status, device fields and qualified job name parameter to subset the list of all the spooled files.

The special value supported is:

\*ALL

Spooled files are included in the list regardless of the value for user-specified data.

## Error Messages

Message ID	Error Message Text
CPF24B4 E	Severe error while addressing parameter list.
CPF3330 E	Necessary resource not available.
CPF3342 E	Job &5/&4/&3 not found.
CPF3343 E	Duplicate job names found.
CPF335D E	Job system name is not valid.
CPF335E E	Starting spooled file create date is not valid.
CPF335F E	Starting spooled file create time is not valid.
CPF336C E	Starting spooled file create time is not blank.
CPF336D E	Ending spooled file create date is not valid.
CPF336E E	Ending spooled file create date is not blank.
CPF336F E	Ending spooled file create time is not valid.
CPF337A E	Ending spooled file create time is not blank.
CPF3C19 E	Error occurred with receiver variable specified.
CPF3C21 E	Format name &1 is not valid.
CPF3C30 E	Library name &1 is not valid for output queue *ALL.
CPF3C90 E	Literal value cannot be changed.
CPF3CF1 E	Error code parameter not valid.
CPF9801 E	Object &2 in library &3 not found.
CPF9802 E	Not authorized to object &2 in &3.
CPF9803 E	Cannot allocate object &2 in library &3.
CPF9807 E	One or more libraries in library list deleted.
CPF9808 E	Cannot allocate one or more libraries on library list.
CPF9810 E	Library &1 not found.
CPF9820 E	Not authorized to use library &1.
CPF9830 E	Cannot assign library &1.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.
GUI0002 E	&2 is not valid for length of receiver variable.
GUI0009 E	&1 is not valid for number of statuses.
GUI0011 E	&1 is not valid for number of user names.
GUI0012 E	&1 is not valid for number of output queue names.
GUI0024 E	&1 is not valid for number of keys to sort on.
GUI0025 E	&1 is not valid for sort key field starting position.
GUI0026 E	&1 is not valid for sort key field length.
GUI0027 E	&1 is not valid for number of records to return.
GUI0042 E	&1 is not valid for status.
GUI0078 E	*ALL cannot be specified with another value.
GUI0079 E	*ALL cannot be specified with another value.
GUI0080 E	*ALL cannot be specified with another value.
GUI0081 E	*ALL cannot be specified with another value.
GUI0108 E	Filter information is not valid. Reason code &1.
GUI0121 E	Printer name cannot be specified when format OSPL0100 is requested.

API introduced: V3R6

Top | "Print APIs," on page 1 | APIs by category

---

## Open Spooled File (QSPOPNSP) API

### Required Parameter Group:

1	Spooled file handle returned	Output	Binary(4)
2	Qualified job name	Input	Char(26)
3	Internal job identifier	Input	Char(16)
4	Internal spooled file identifier	Input	Char(16)
5	Spooled file name	Input	Char(10)
6	Spooled file number	Input	Binary(4)
7	Number of buffers to get	Input	Binary(4)
8	Error code	I/O	Char(*)

### Optional Parameter Group:

9	Job system name	Input	Char(8)
10	Spooled file create date	Input	Char(7)
11	Spooled file create time	Input	Char(6)

Default Public Authority: \*USE  
Threadsafe: No

The Open Spooled File (QSPOPNSP) API opens an existing spooled file. After the existing spooled file is opened, the Get Spooled File Data (QSPGETSP) API can then get the data from the file.

## Authorities and Locks

### Spooled File Authorities

The requester is authorized to the spooled file if one or more of the following conditions are met.

- The requester is the owner of the spooled file.
- The requester has \*READ authority to the queue on which the spooled file resides, and the queue is specified as DSPDTA(\*YES).
- The requester has \*SPLCTL authority.
- The requester has \*JOBCTL authority, and the queue on which the spooled file resides is specified as OPRCTL(\*YES).
- The requester has ownership of the queue on which the spooled file resides, and the queue is specified as AUTCHK(\*OWNER).
- The requester has read, add, and delete authorities to the queue on which the spooled file resides, and the queue is specified as AUTCHK(\*DTAAUT).

### Spooled File Lock

\*LSRD

## Required Parameter Group

### Spooled file handle returned

OUTPUT; BINARY(4)

The handle used on subsequent get (QSPGETSP API) and close (QSPCLOSP API) operations to identify the spooled file.

### Qualified job name

INPUT; CHAR(26)

The job that owns the spooled file. The qualified job name has three parts:

<i>job name</i>	CHAR(10).
	A specific job name, or one of the following special values:
	* Only the job that this program is running. The rest of the job name parameter must be blank.
	*INT The internal job identifier used to locate the spooled file. The user name and job number must be set to blank.
<i>user name</i>	CHAR(10)
	A specific user profile name, or blanks when the job name is * or *INT.
<i>job number</i>	CHAR(6).
	A specific job number, or blanks when the job name is * or *INT.

### **Internal job identifier**

INPUT; CHAR(16)

The internal job identifier for the job that owns the spooled file. Use the "Retrieve Spooled File Attributes (QUSRSPLA) API" on page 337 or one of these APIs to make the identifier available: "List Spooled Files (QUSLSPL) API" on page 279.

### **Internal spooled file identifier**

INPUT; CHAR(16)

The internal spooled file identifier for the spooled file. To make the identifier available, use the "Retrieve Spooled File Attributes (QUSRSPLA) API" on page 337 or see "List Spooled Files (QUSLSPL) API" on page 279.

### **Spooled file name**

INPUT; CHAR(10)

The name of the spooled file to be opened. You can use this special value for the name:

\*INT The internal spooled file identifier is used to locate the spooled file.

### **Spooled file number**

INPUT; BINARY(4)

The unique number of the spooled file. The valid range is 1 through 999999.

The following special values are supported for this parameter:

- 0 Only one spooled file from the job has the specified file name, so the number of the spooled file is not necessary.
- 1 This uses the highest-numbered spooled file with the specified file name.
- 2 The spooled file number is not used to determine which spooled file to process. Use this value when you want the Job system name parameter or the Spooled file create date and Spooled file create time parameters to take precedence over the spooled file number when selecting a spooled file.

### **Number of buffers to get**

INPUT; BINARY(4)

How many buffers to get on each call to the QSPGETSP API. Valid numbers include 1 through 8, 16, 24, and 32. Any values greater than 32 must be a multiple of 32. A value greater than 1 should show some performance improvement. A value of 8 may give the best performance when using the QSPGETSP API.

The following special value is supported for this parameter:

-1 Reads the entire spooled file.

If the user space cannot accommodate all buffers requested, as many complete buffers as possible will be returned in the available space. The information complete indicator in the header section of the data returned is then set to P.

If the end of an open spooled file is encountered before the buffer count is reached, the end of open spooled file parameter on the Get Spooled File Data (QSPGETSP) API determines the action taken.

#### **Error code**

I/O; CHAR(\*)

The structure in which to return error information. For the format of the structure, see Error code parameter.

## **Optional Parameter Group**

#### **Job system name**

INPUT; CHAR(8)

The name of the system where the job that created the spooled file ran or blank when the spooled file name is \*INT. This parameter is considered after the job name, user name, job number, spooled file name, and spooled file number parameter requirements have been met.

The following special values are supported for this parameter:

\*ONLY There is one job with the specified job name, user name, job number, spooled file name, spooled file number, spooled file creation date, and spooled file creation time.

\*CURRENT The job on the current system with the specified job name, user name, job number, spooled file create date, and spooled file create time is used.

\*ANY The job system name is not considered when selecting a spooled file. Use this value when you want the Spooled file create date and Spooled file create time parameters to take precedence over the job system name when selecting a spooled file.

If this parameter is omitted, the API assumes blanks when the spooled file name is \*INT. When spooled file name is not \*INT, the API assumes \*ONLY.

#### **Spooled file create date**

INPUT; CHAR(7)

» The date, based on local job time, that the spooled file was created on the system or blank when the spooled file name is \*INT. « This parameter is considered after the job name, user name, job number, spooled file name, spooled file number, and job system name parameter requirements have been met. The date must be in the CYYMMDD format or one of the following special values:

\*ONLY There is only one spooled file with the specified job name, user name, job number, spooled file name, spooled file number, and job system name.

\*LAST The spooled file with the latest date and time which also has the specified job name, user name, job number, spooled file name, spooled file number, and job system name is used.

The date format CYYMMDD is defined as follows:

C Century, where 0 indicates years 19xx and 1 indicates years 20xx.  
YY Year  
MM Month  
DD Day

If this parameter is omitted, the API assumes blanks when the spooled file name is \*INT. When spooled file name is not \*INT, the API assumes \*ONLY.

**Spooled file create time**

INPUT; CHAR(6)

» The time, in local job time, that the spooled file was created on the system or blank when the spooled file name is \*INT. « This parameter must be set to blanks when special values \*LAST or \*ONLY are used for parameter Spooled file create date. This parameter must have a value set if a date is specified for parameter Spooled file create date. This parameter is considered after the job name, user name, job number, spooled file name, spooled file number, job system name, and spooled file create date parameter requirements have been met. The time must be in the HHMMSS format or one of the following special values:

- \*ONLY There is only one spooled file with the specified job name, user name, job number, spooled file name, spooled file number, job system name, and create date.
- \*LAST The spooled file with the latest time which also has the specified job name, user name, job number, spooled file name, spooled file number, job system name, and create date is used.

The time format HHMMSS is defined as follows:

- HH Hour
- MM Minutes
- SS Seconds

If this parameter is omitted, the API assumes blanks.

**How to Select a Spooled File to Be Opened**

This table illustrates the valid parameter combinations of qualified job name, internal job identifier, internal spooled file identifier, spooled file name, job system name, spooled file create date, and spooled file create time. The combinations of these parameters identify the spooled file to be opened. For example, when the qualified job name parameter value is \*, the internal job identifier must be blank, the internal spooled file identifier must be blank, and the actual name and number of the spooled file must be given.

Qualified Job Name			Internal Job Identifier	Internal Spooled File Identifier	Spooled File Name	Spooled File Number	Job System Name	Spooled File Create Date	Spooled File Create Time
Job Name	User Name	Job Number							
Name	Name	Number	Blanks	Blanks	Name	-2 through 999999	Name, *ONLY, *CURRENT, or *ANY	Date, *ONLY, or *LAST	Time, blanks, *ONLY, or *LAST
Name	Name	Blanks	Blanks	Blanks	Name	-2 through 999999	Name, *ONLY, *CURRENT, or *ANY	Date, *ONLY, or *LAST	Time, blanks, *ONLY, or *LAST
*	Blanks	Blanks	Blanks	Blanks	Name	-2 through 999999	Name, *ONLY, *CURRENT or *ANY	Date, *ONLY or *LAST	Time, blanks, *ONLY or *LAST
*INT	Blanks	Blanks	Internal job identifier	Internal spooled file identifier	*INT	-2 through 999999	Blanks	Blanks	Blanks

Qualified Job Name			Internal Job Identifier	Internal Spooled File Identifier	Spooled File Name	Spooled File Number	Job System Name	Spooled File Create Date	Spooled File Create Time
Job Name	User Name	Job Number							
*INT	Blanks	Blanks	Internal job identifier	Blanks	Name	-2 through 999999	Name, *ONLY, *CURRENT, or *ANY	Date, *ONLY, or *LAST	Time, blanks, *ONLY, or *LAST  See Notes for additional information.

**Notes:** This parameter combination is not valid when a job has been detached from its spooled files or for a spooled file on an independent disk pool. Use of this combination will result in message CPF3C43.

## Error Messages

Message ID	Error Message Text
CPF24B4 E	Severe error while addressing parameter list.
CPF3CF1 E	Error code parameter not valid.
CPF3C33 E	Spooled file number &1 is not valid.
CPF3C40 E	Spooled file &4 not found.
CPF3C41 E	More than one spooled file with same name.
CPF3C42 E	User name or job number is not blank.
CPF3C43 E	Internal job identifier is not valid.
CPF3C44 E	Internal spooled file identifier is not valid.
CPF3C58 E	Job name specified is not valid.
CPF3C59 E	Internal identifier is not blanks and job name is not *INT.
CPF3C90 E	Literal value cannot be changed.
CPF33C9 E	Spooled file name parameter cannot be blank.
CPF33DA E	Value &1 not valid for number of buffers to read parameter.
CPF33DB E	Qualified job name parameter not valid with internal spooled file name.
CPF33DC E	Open or create not valid for diskette files.
CPF33DD E	Maximum number of open spooled files exceeded for this job.
CPF33DE E	Size of internal data for opened spooled file exceeds maximum.
CPF33DF E	Internal data area for opened spooled files destroyed.
CPF3309 E	No files named &1 are active.
CPF3330 E	Necessary resource not available.
CPF333B E	Job system name is not valid.
CPF333C E	Spooled file create date is not valid.
CPF333D E	Spooled file create time is not valid.
CPF333E E	Spooled file create time is not blank.
CPF333F E	Job system name is not blank.
CPF3342 E	Job &5/&4/&3 not found.
CPF3343 E	Duplicate job names found.
CPF3344 E	File &1 number &2 no longer in the system.
CPF335B E	Spooled file create date is not blank.
CPF3492 E	Not authorized to spooled file.
CPF9838 E	User profile storage limit exceeded.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V2R1

---

## Put Spooled File Data (QSPPUTSP) API

Required Parameter Group:

1	Spooled file handle	Input	Binary(4)
2	Qualified user space name	Input	Char(20)
3	Error code	I/O	Char(*)

Default Public Authority: \*USE

Threadsafe: No

The Put Spooled File Data (QSPPUTSP) API puts data into a spooled file, which was created using the Create Spooled File (QSPCRTSP) API. The data put in the spooled file is taken from a user space. The data in the user space can be created using the Get Spooled File Data (QSPGETSP) API or can be created by a user application.

Before a buffer is put in a spooled file, a limited validity check is performed on the information in the user space for that buffer. The possible errors that result from the values of the fields in the user space can be classified as follows:

- The value with an error can be substituted by the default value. A CPIxxxx message is issued informing the user of the substitution.
- The value with an error causes a CPFxxxx message to be issued. The message is not issued until the validation of the information is complete or a severe error is encountered and validation cannot continue. This provides the caller with all informational messages as well as the first CPFxxxx message encountered.
- The value with an error causes a CPFxxxx message to be issued immediately.

The buffers must be in the format returned by the Get Spooled File Data (QSPGETSP) API and be in format SPFR0200.

## Authorities and Locks

*User Space Authority*

\*CHANGE

*Library Authority*

\*EXECUTE

*User Space Lock*

\*EXCLRD

## Required Parameter Group

### Spooled file handle

INPUT; BINARY(4)

The handle returned by the QSPCRTSP API.

### Qualified user space name

INPUT; CHAR(20)

The name of the user space that contains the buffer of spooled information. The first 10 characters contain the user space name, and the second 10 characters contains the name of the library in which the user space is located. The special values allowed for the library name are \*LIBL and \*CURLIB. Both entries are left-justified. If no library is specified as the current library of the job, QGPL is used. The format of the user space is the same as that returned by QSPGETSP API. The format specified must be SPFR0200.

To see the format of the user space and how the offset values are calculated, see “Format of the User Space” on page 251.

#### **Error code**

I/O; CHAR(\*)

The structure in which to return error information. For the format of the structure, see Error code parameter.

## **Considerations When Changing or Creating a User Space**

When creating your own spooled files or altering the buffers returned by the QSPGETSP API, incorrect data can be put with the QSPPUTSP API. Although errors are caught by the QSPPUTSP API, not all errors are detected.

If the order of the pages, the number of pages, or the number of lines is incorrect, problems can occur when repositioning the spooled file to print it. Repositioning means trying to start printing at a specific page other than page 1. The problems can be caused by:

- Using the restart printing function
- Changing the starting and ending print pages
- Working with an inquiry error message that allows a user to specify what page of the spooled file to restart printing

There are fields in the buffer section of the user space that, if they contain incorrect values, could cause other functions, such as displaying or copying of spooled files, to work incorrectly. These fields are in the general information section and in the page data section.

## **Fields in the General Information Section**

If the state and IPDS™ data fields contain incorrect or changed values, the Display Spooled File (DSPSPLF) and Copy Spooled File (CPYSPLF) commands can be affected in the following ways:

- State field

When this field is set to \*HOMETRANS or \*PAGETRANS, it indicates that the spooled file contains only IPDS transparent data. This field only affects spooled files with a device type of \*IPDS.

A changed or incorrect state field value causes the DSPSPLF and CPYSPLF commands to work the following way:

- DSPSPLF command

IPDS transparent data cannot be displayed by the DSPSPLF command. As a result, when the created spooled file is displayed, the data in the buffer with the state field set to \*HOMETRANS or \*PAGETRANS is not displayed. Furthermore, message CPI3438 (Intelligent Printer Data Stream™ (IPDS) data not displayed) appears. If the spooled file is made up entirely of buffers with the state field set to \*HOMETRANS or \*PAGETRANS, the spooled file is not displayed. Message CPF3429 (File cannot be displayed or copied) is displayed.

- CPYSPLF command

IPDS transparent data cannot be copied by the CPYSPLF command. If some buffers of a spooled file have a state field set to \*HOMETRANS or \*PAGETRANS, those buffers are not copied to the database member. If the spooled file is made up entirely of buffers with the state field set to \*HOMETRANS or \*PAGETRANS, the spooled file is not copied. Message CPF3429 (File cannot be displayed or copied) is displayed.

- IPDS data field

When this field is set to Y, it indicates that the buffer contains only IPDS data. This field only affects spooled files with a device type of \*SCS.

A changed or incorrect IPDS data field value causes the DSPSPLF and CPYSPLF commands to work the following way:

- DSPSPLF command

IPDS data cannot be displayed using the DSPSPLF command. As a result, when a created spooled file is displayed, the data in the buffer with the IPDS data field set to Y is not displayed.

Furthermore, message CPI3437 (Intelligent printer data stream (IPDS) data not displayed) appears at the bottom of the last screen of the DSPSPLF command. If the spooled file is made up entirely of buffers with the IPDS field set to Y, the spooled file is not displayed. Message CPF3429 (File cannot be displayed or copied) is displayed.

- CPYSPLF command

IPDS data cannot be copied by using the CPYSPLF command. As a result, when a spooled file is copied into the database member, the data in the buffer with the IPDS data field set to Y is not copied. If the spooled file is made up entirely of buffers with the IPDS field set to Y, the spooled file is not copied. Message CPF3429 (File cannot be displayed or copied) is displayed.

## Fields in the Page Data Section

If the text data start and page offset fields contain incorrect or changed values, the Display Spooled File (DSPSPLF) command can be affected in the following ways:

- Text data start

The number of the first line where user data can start on the page. This count includes text only. The text data start field can have an incorrect value when the buffers of the original spooled file are put in the created spooled file in an order other than the original spooled file. This field can also have an incorrect value by simply changing its value to another value higher or lower than was returned when the data was retrieved using the QSPGETSP API. This field affects all device types of spooled files.

A changed or incorrect value for the text data start field causes the DSPSPLF command to work the following way:

- DSPSPLF command

When the text data start field contains an incorrect value, DSPSPLF may issue message CPF33F9 (Error occurred while displaying file X number Y) when attempting to find a particular string in the displayed spooled file. Also, the user is able to see only part of the spooled file. The amount of data the user sees depends on the order the buffers were put. For example, if the first buffer of the original spooled file is the last buffer in the created spooled file, the user only sees the pages in the first buffer of the created spooled file. If the second buffer of the original spooled file was the last buffer in the created spooled file, the user sees the pages in the first and second buffer of the created spooled file. The other buffers are there but not displayed.

- Page offset

The location of the start of this page. The offset value is from the beginning of the print data. This field affects all device types of spooled files.

A changed or incorrect value for the page offset field causes the DSPSPLF command to work the following way:

- DSPSPLF command

When the page offset field contains an incorrect value, DSPSPLF may issue informational message CPI3431 (Line number adjusted). The page information of the individual pages overlaps. The user sees only part of the spooled file because some pages overlap.

## Error Messages

Message ID	Error Message Text
CPF24B4 E	Severe error while addressing parameter list.
CPF3CF1 E	Error code parameter not valid.
CPF3C21 E	Format name &1 is not valid.
CPF3C90 E	Literal value cannot be changed.
CPF33DF E	Internal data area for opened spooled files destroyed.
CPF33D2 E	Spooled file handle not valid.

Message ID	Error Message Text
CPF33D5 E	Spooled file not opened for operation requested.
CPF33F2 E	New page expected at beginning of buffer &1.
CPF33F3 E	Data in buffer &1 exceeds spooled file buffer size.
CPF33F4 E	Beyond end of user space &1 in library &2.
CPF33F6 E	Value in generic header of user space &4 in library &5 not valid.
CPF33F7 E	Value in buffer &6 of user space &4 in library &5 not valid.
CPF9801 E	Object &2 in library &3 not found.
CPF9802 E	Not authorized to object &2 in &3.
CPF9803 E	Cannot allocate object &2 in library &3.
CPF9807 E	One or more libraries in library list deleted.
CPF9808 E	Cannot allocate one or more libraries on library list.
CPF9810 E	Library &1 not found.
CPF9820 E	Not authorized to use library &1.
CPF9830 E	Cannot assign library &1.
CPF9846 E	Error while processing file &1 in library &2.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V2R1

Top | "Print APIs," on page 1 | APIs by category

---

## Retrieve Identity of Last Spooled File Created (QSPRILSP) API

Required Parameter Group:

1	Receiver Variable	Output	Char(*)
2	Length of receiver variable	Input	Binary(4)
3	Format name	Input	Char(8)
4	Error code	I/O	Char(*)

Default Public Authority: \*USE

Threadsafe: Yes

The Retrieve Identity of Last Spooled File Created (QSPRILSP) API returns the complete spooled file identity of the last spooled file created for the current job or thread.

### Authorities and Locks

None

### Required Parameter Group

#### Receiver variable

OUTPUT; CHAR(\*)

The receiver variable that receives the information requested. You can specify the size of the area to be smaller than the format requested as long as you specify the length parameter correctly. As a result, the API returns only the data that the area can hold.

#### Length of receiver variable

INPUT; BINARY(4)

The length of the receiver variable provided. The length of receiver variable parameter may be specified up to the size of the receiver variable specified in the user program. If the length of receiver variable parameter specified is larger than the allocated size of the receiver variable specified in the user program, the results are not predictable. The minimum length is 8 bytes.

**Format name**

INPUT; CHAR(8)

The format of the spooled file identity being returned. The valid format is SPRL0100.

**Error code**

I/O; CHAR(\*)

The structure in which to return error information. For the format of the structure, see Error code parameter.

**Format of Receiver Variable**

The following table describes the order and format of the data that is returned in the receiver variable.

**SPRL0100 Format**

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Bytes returned
4	4	BINARY(4)	Bytes available
8	8	CHAR(10)	Spooled file name
18	12	CHAR(10)	Job name
28	1C	CHAR(10)	User name
38	26	CHAR(6)	Job number
44	2C	BINARY(4)	Spooled file number
48	30	CHAR(8)	Job system name
56	38	CHAR(7)	Spooled file create date
63	3F	CHAR(1)	Reserved
64	40	CHAR(6)	Spooled file create time

**Field Descriptions**

**Bytes available.** The number of bytes of data available to be returned. All available data is returned if enough space is provided.

**Bytes returned.** The number of bytes of data returned.

**Job name.** The name of the job that owns the spooled file.

**Job number.** The number of the job that owns the spooled file.

**Job system name.** The name of the system where the job that created the spooled file ran.

**Spooled file create date.** ➤ The date, based on local job time, that the spooled file was created on the system. ⚡ The date is in the CYYMMDD format as follows:

C            Century, where 0 indicates years 19xx and 1 indicates years 20xx.  
 YY          Year  
 MM          Month  
 DD          Day

**Spooled file create time** >> The time, in local job time, that the spooled file was created on the system. <<  
The time is in the HHMMSS format where:

HH            Hour  
MM            Minutes  
SS            Seconds

**Spooled file name.** The name of the last spooled file created by the current job or thread.

**Spooled file number.** The number of the last spooled file created by the current job or thread.

**User name.** The name of the user who owns the last spooled file created by the current job or thread.

## Error Messages

Message ID	Error Message Text
CPF24B4 E	Severe error while addressing parameter list.
CPF3CF1 E	Error code parameter not valid.
CPF333A E	No spooled file created under the current job or thread.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V5R2

Top | "Print APIs," on page 1 | APIs by category

---

## Retrieve Spool Information (QSPSPLI) API

Required Parameter Group:

1	Receiver variable	Output	Char(*)
2	Length of receiver variable	Input	Binary(4)
3	Format name	Input	Char(8)
4	Auxiliary storage pool (ASP) group name	Input	Char(10)
5	User name	Input	Char(10)
6	Error code	I/O	Char(*)

Default Public Authority: \*USE  
Threadsafe: Yes

The Retrieve Spool Information (QSPSPLI) API returns information about the spooling environment on the system such as the number of spooled files in an ASP group or the number of spooled files owned by a user.

## Authorities and Locks

*Auxiliary Storage Pool (ASP) Device Authority*

\*USE to all ASP devices in an ASP group when a specific ASP group is specified for the ASP group parameter.

## Required Parameter Group

**Receiver variable**  
OUTPUT; CHAR(\*)

The receiver variable that receives the information requested. You can specify the size of the area to be smaller than the format requested as long as you specify the length parameter correctly. As a result, the API returns only the data that the area can hold.

**Length of receiver variable**

INPUT; BINARY(4)

The length of the receiver variable. If the length is larger than the size of the receiver variable, the results are not predictable. The minimum length is 8 bytes.

**Format name**

INPUT; CHAR(8)

The format of the spool information being returned. The valid format is SPLI0100.

**Auxiliary storage pool (ASP) group name**

INPUT; CHAR(10)

The name of an auxiliary storage pool (ASP) group for which spool information is being requested. The status of the ASP group must be 'Available'. If this field is blank, the value \*SYSBAS will be assumed. One of the following special values may be specified:

- \*SYSBAS            The spool information for the system ASP (ASP 1) and defined basic user ASPs (ASPs 2-32) will be returned.
- \*CURASPGRP       The spool information for the ASPs in the current thread's ASP group will be returned.

**User name**

INPUT; CHAR(10)

The name of the user who owns the spooled files to be considered when returning the requested spool information. One of the following special values may be specified:

- \*ALL                Spooled files owned by all users
- \*CURRENT          Spooled files owned by the current user

**Error code**

I/O; CHAR(\*)

The structure in which to return error information. For the format of the structure, see Error code parameter.

**Format of Receiver Variable**

The following table describes the order and format of the data that is returned in the receiver variable.

**SPLI0100 Format**

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Bytes returned
4	4	BINARY(4)	Bytes available
8	8	BINARY(4)	Number of spooled files
12	C	CHAR(10)	Auxiliary storage pool (ASP) group name
22	16	CHAR(10)	User name

## Field Descriptions

**Auxiliary storage pool (ASP) group name.** The name of the auxiliary storage pool (ASP) group where the spooled files exist. This is the name of the primary ASP in an ASP group or the name of an ASP device description. The following special value can be returned:

\*SYSBAS            The spool information for the system ASP (ASP 1) and defined basic user ASPs (ASPs 2-32) was returned.

**Bytes available.** The number of bytes of data available to be returned. All available data is returned if enough space is provided.

**Bytes returned.** The number of bytes of data returned.

**Number of spooled files.** The number of spooled files in the specified auxiliary storage pool (ASP) group for the user name specified.

**User name.** The name of the user that owns the spooled files. The following special value can be returned:

\*ALL                Spooled files owned by all users

## Error Messages

Message ID	Error Message Text
CPF24B4 E	Severe error while addressing parameter list.
CPF3C19 E	Error occurred with receiver variable specified.
CPF3C20 E	Error found by program &1.
CPF3C21 E	Format name &1 is not valid.
CPF3C24 E	Length of the receiver variable is not valid.
CPF338C E	Internal spool control file is not accessible.
CPF339A E	*CURASGRP specified and thread has no ASP group.
CPF3CF1 E	Error code parameter not valid.
CPF9814 E	Device &1 not found.
CPF9825 E	Not authorized to device &1.
CPF9833 E	*CURASGRP or *ASPGRPPRI specified and thread has no ASP group.
CPFB8ED E	Device description &1 not correct for operation.



API introduced: V6R1 with PTF

[Top](#) | ["Print APIs," on page 1](#) | [APIs by category](#)

---

## Retrieve Spooled File Attributes (QUSRSPLA) API

Required Parameter Group:

1	Receiver variable	Output	Char(*)
2	Length of receiver variable	Input	Binary(4)
3	Format name	Input	Char(8)
4	Qualified job name	Input	Char(26)
5	Internal job identifier	Input	Char(16)
6	Internal spooled file identifier	Input	Char(16)

7	Spooled file name	Input	Char(10)
8	Spooled file number	Input	Binary(4)

Optional Parameter Group 1:

9	Error code	I/O	Char(*)
---	------------	-----	---------

Optional Parameter Group 2:

10	Job system name	Input	Char(8)
11	Spooled file create date	Input	Char(7)
12	Spooled file create time	Input	Char(6)

Default Public Authority: \*USE  
 Threadsafes: No

The Retrieve Spooled File Attributes (QUSRSPLA) API returns specific information about a spooled file into a receiver variable. The size of the receiver variable determines the amount of information returned. You can specify the spooled file for which information is returned either with the internal job and spooled file identifiers, or with a specific job name, spooled file name, and spooled file number.

For RPG and COBOL examples using this API, see Examples: APIs and exit programs.

**Note:** These examples also contain a delete old spooled file (DLTOLDSPLF) example.

## Authorities and Locks

None.

## Required Parameter Group

### Receiver variable

OUTPUT; CHAR(\*)

The receiver variable that receives the information requested. You can specify the size of the area to be smaller than the format requested as long as you specify the length parameter correctly. As a result, the API returns only the data the area can hold

### Length of receiver variable

INPUT; BINARY(4)

The length of the receiver variable. If the length is larger than the size of the receiver variable, the results are not predictable. The minimum length is 8 bytes.

### Format name

INPUT; CHAR(8)

The format of the information to be returned for the specified spooled file.

The valid format names are:

*"SPLA0100* Basic spooled file attributes.

*Format" on page*

342

*"SPLA0200* Detailed spooled file attributes needed to print or duplicate the spooled file.

*Format" on page*

346

### Qualified job name

INPUT; CHAR(26)

The job that owns the spooled file.

The qualified job name has three parts:

<i>job name</i>	CHAR(10). A specific job name, or one of the following special values: <ul style="list-style-type: none"><li>* Only the job that this program is running. The rest of the job name parameter must be blank.</li><li>*INT The internal job identifier is used to locate the spooled file. The user name and job number must be set to blank.</li></ul>
<i>user name</i>	CHAR(10). A specific user profile name, or blanks when the job name is * or *INT.
<i>job number</i>	CHAR(6). A specific job number, or blanks when the job name is * or *INT.

### Internal job identifier

INPUT; CHAR(16)

The internal job identifier for the job that owns the spooled file whose attributes are to be retrieved. Use one of these APIs to make the identifier available:

- “List Spooled Files (QUSLSPL) API” on page 279
- List Job (QUSLJOB) API
- Retrieve Job Information (QUSRJOBI) API

### Internal spooled file identifier

INPUT; CHAR(16)

The internal spooled file identifier for the spooled file whose attributes are retrieved. To make the identifier available, see “List Spooled Files (QUSLSPL) API” on page 279.

### Spooled file name

INPUT; CHAR(10)

The name of the spooled file for which you may want to retrieve the attributes.

You can use this special value for the name:

\*INT The internal spooled file identifier is used to locate the spooled file.

### Spooled file number

INPUT; BINARY(4)

The unique number of the spooled file. The valid range is 1 through 999999.

The following special values are supported for this parameter:

- 0 Only one spooled file from the job has the specified file name, so the number of the spooled file is not necessary.
- 1 This uses the highest-numbered spooled file with the specified file name.
- 2 The spooled file number is not used to determine which spooled file to process. Use this value when you want the Job system name parameter or the Spooled file create date and Spooled file create time parameters to take precedence over the spooled file number when selecting a spooled file.

**Note:** This parameter must contain a valid value even if the value for the spooled file name parameter is \*INT.

A value is required because the QUSRSPLA API performs validity checking on all parameters.

## Optional Parameter Group 1

### Error code

I/O; CHAR(\*)

The structure in which to return error information. For the format of the structure, see Error code parameter.

If this parameter is omitted, diagnostic and escape messages are issued to the application.

## Optional Parameter Group 2

### Job system name

INPUT; CHAR(8)

The name of the system where the job that created the spooled file ran or blank when the spooled file name is \*INT. This parameter is considered after the job name, user name, job number, spooled file name, and spooled file number parameter requirements have been met.

The following special values are supported for this parameter:

- \*ONLY* There is one job with the specified job name, user name, job number, spooled file name, spooled file number, spooled file create date, and spooled file create time.
- \*CURRENT* The job on the current system with the specified job name, user name, job number, spooled file name, spooled file number, spooled file create date, and spooled file create time is used.
- \*ANY* The job system name is not considered when selecting a spooled file. Use this value when you want the Spooled file create date and Spooled file create time parameters to take precedence over the job system name when selecting a spooled file.

If this parameter is omitted, the API assumes blanks when the spooled file name is \*INT. When spooled file name is not \*INT, the API assumes \*ONLY.

### Spooled file create date

INPUT; CHAR(7)

» The date, based on local job time, that the spooled file was created on the system or blank when the spooled file name is \*INT. « This parameter is considered after the job name, user name, job number, spooled file name, spooled file number, and job system name parameter requirements have been met. The date must be in the CYYMMDD format or one of the following special values:

- \*ONLY* There is only one spooled file with the specified job name, user name, job number, spooled file name, spooled file number, and job system name.
- \*LAST* The spooled file with the latest date and time which also has the specified job name, user name, job number, spooled file name, spooled file number, and job system name is used.

The date format CYYMMDD is defined as follows:

<i>C</i>	Century, where 0 indicates years 19xx and 1 indicates years 20xx.
<i>YY</i>	Year
<i>MM</i>	Month
<i>DD</i>	Day

If this parameter is omitted, the API assumes blanks when the spooled file name is \*INT. When spooled file name is not \*INT, the API assumes \*ONLY.

### Spooled file create time

INPUT; CHAR(6)

» The time, in local job time, that the spooled file was created on the system. « This parameter must be set to blanks when special values \*LAST or \*ONLY are used for parameter Spooled file create date. This parameter must have a value set if a date is specified for parameter Spooled file create date. This parameter is considered after the job name, user name, job number, spooled file name, spooled file number, job system name, and spooled file create date parameter requirements

have been met. The time must be in the HHMMSS format or one of the following special values:

- \*ONLY*            There is only one spooled file with the specified job name, user name, job number, spooled file name, spooled file number, job system name, and create date.
- \*LAST*            The spooled file with the latest time which also has the specified job name, user name, job number, spooled file name, spooled file number, job system name, and create date is used.

The time format HHMMSS is defined as follows:

- HH*            Hour
- MM*            Minutes
- SS*            Seconds

If this parameter is omitted, the API assumes blanks.

## How to Select a Spooled File to Retrieve Its Attributes

This table illustrates the valid parameter combinations of qualified job name, internal job identifier, internal spooled file identifier, spooled file name, job system name, spooled file create date and spooled file create time. The combinations of these parameters identify the spooled file from which attributes can be retrieved. For example, when the qualified job name parameter value is \*, the internal job identifier must be blank, the internal spooled file identifier must be blank, the actual name of the spooled file must be given, and a valid spooled file number must be given.

Qualified Job Name			Internal Job Identifier	Internal Spooled File Identifier	Spooled File Name	Spooled File Number	Job system name	Spooled file create date	Spooled file create time
Job Name	User Name	Job Number							
Name	Name	Number	Blanks	Blanks	Name	-2 through 999999	Name, *ONLY, *CURRENT, or *ANY	Date, *ONLY, or *LAST	Time, blanks, *ONLY, or *LAST
Name	Blanks	Number	Blanks	Blanks	Name	-2 through 999999	Name, *ONLY, *CURRENT, or *ANY	Date, *ONLY, or *LAST	Time, blanks, *ONLY, or *LAST
Name	Name	Blanks	Blanks	Blanks	Name	-2 through 999999	Name, *ONLY, *CURRENT, or *ANY	Date, *ONLY, or *LAST	Time, blanks, *ONLY, or *LAST
*	Blanks	Blanks	Blanks	Blanks	Name	-2 through 999999	Name, *ONLY, *CURRENT, or *ANY	Date, *ONLY, or *LAST	Time, blanks, *ONLY, or *LAST
*INT	Blanks	Blanks	Internal job identifier	Internal spooled file identifier	*INT	-2 through 999999	Blanks	Blanks	Blanks

Qualified Job Name			Internal Job Identifier	Internal Spooled File Identifier	Spooled File Name	Spooled File Number	Job system name	Spooled file create date	Spooled file create time
Job Name	User Name	Job Number							
*INT	Blanks	Blanks	Internal job identifier	Blanks	Name	-2 through 999999	Name, *ONLY, *CURRENT, or *ANY	Date, *ONLY, or *LAST	Time, blanks, *ONLY, or *LAST  See Note (page 342) for additional information.

**Note:** This parameter combination is not valid when a job has been detached from its spooled files or for a spooled file on an independent disk pool. Use of this combination will result in message CPF3C43 or CPD3C43.

## SPLA0100 Format

The following table shows the information returned for the SPLA0100 format.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Bytes returned
4	4	BINARY(4)	Bytes available
8	8	CHAR(16)	Internal job identifier
24	18	CHAR(16)	Internal spooled file identifier
40	28	CHAR(10)	Job name
50	32	CHAR(10)	User name
60	3C	CHAR(6)	Job number
66	42	CHAR(10)	Spooled file name
76	4C	BINARY(4)	Spooled file number
80	50	CHAR(10)	Form type
90	5A	CHAR(10)	User-specified data
100	64	CHAR(10)	Status
110	6E	CHAR(10)	File available
120	78	CHAR(10)	Hold file before written
130	82	CHAR(10)	Save file after written
140	8C	BINARY(4)	Total pages
144	90	BINARY(4)	Page or record being written
148	94	BINARY(4)	Starting page
152	98	BINARY(4)	Ending page
156	9C	BINARY(4)	Last page printed
160	A0	BINARY(4)	Restart printing
164	A4	BINARY(4)	Total copies
168	A8	BINARY(4)	Copies left to produce

Offset		Type	Field
Dec	Hex		
172	AC	BINARY(4)	Lines per inch
176	B0	BINARY(4)	Characters per inch
180	B4	CHAR(2)	Output priority
182	B6	CHAR(10)	Output queue name
192	C0	CHAR(10)	Output queue library name
202	CA	CHAR(7)	Date file opened (created)
209	D1	CHAR(6)	Time file opened (created)
215	D7	CHAR(10)	Device file name
225	E1	CHAR(10)	Device file library name
235	EB	CHAR(10)	Program that opened file name
245	F5	CHAR(10)	Program that opened file library name
255	FF	CHAR(15)	Accounting code
270	10E	CHAR(30)	Print text
300	12C	BINARY(4)	Record length
304	130	BINARY(4)	Maximum records
308	134	CHAR(10)	Device type
318	13E	CHAR(10)	Printer device type
328	148	CHAR(12)	Document name
340	154	CHAR(64)	Folder name
404	194	CHAR(8)	System/36™ procedure name
412	19C	CHAR(10)	Print fidelity
422	1A6	CHAR(1)	Replace unprintable characters
423	1A7	CHAR(1)	Replacement character
424	1A8	BINARY(4)	Page length
428	1AC	BINARY(4)	Page width
432	1B0	BINARY(4)	Number of separators
436	1B4	BINARY(4)	Overflow line number
440	1B8	CHAR(10)	Multi-byte data
450	1C2	CHAR(10)	DBCS extension characters
460	1CC	CHAR(10)	DBCS shift-out shift-in (SO/SI) spacing
470	1D6	CHAR(10)	DBCS character rotation
480	1E0	BINARY(4)	DBCS characters per inch
484	1E4	CHAR(10)	Graphic character set
494	1EE	CHAR(10)	Code page
504	1F8	CHAR(10)	Form definition name
514	202	CHAR(10)	Form definition library name
524	20C	BINARY(4)	Source drawer
528	210	CHAR(10)	Printer font
538	21A	CHAR(6)	System/36 spooled file identifier
544	220	BINARY(4)	Page rotation

Offset		Type	Field
Dec	Hex		
548	224	BINARY(4)	Justification
552	228	CHAR(10)	Print on both sides (duplex)
562	232	CHAR(10)	Fold records
572	23C	CHAR(10)	Control character
582	246	CHAR(10)	Align forms
592	250	CHAR(10)	Print quality
602	25A	CHAR(10)	Form feed
612	264	CHAR(71)	Volumes (array)
683	2AB	CHAR(17)	File label identifier
700	2BC	CHAR(10)	Exchange type
710	2C6	CHAR(10)	Character code
720	2D0	BINARY(4)	Total records
724	2D4	BINARY(4)	Multiple up (pages per side)
728	2D8	CHAR(10)	Front overlay name
738	2E2	CHAR(10)	Front overlay library name
748	2EC	PACKED(15,5)	Front overlay offset down
756	2F4	PACKED(15,5)	Front overlay offset across
764	2FC	CHAR(10)	Back overlay name
774	306	CHAR(10)	Back overlay library name
784	310	PACKED(15,5)	Back overlay offset down
792	318	PACKED(15,5)	Back overlay offset across
800	320	CHAR(10)	Unit of measure
810	32A	CHAR(10)	Page definition name
820	334	CHAR(10)	Page definition library name
830	33E	CHAR(10)	Line spacing
840	348	PACKED(15,5)	Point size
848	350	PACKED(15,5)	Front margin offset down
856	358	PACKED(15,5)	Front margin offset across
864	360	PACKED(15,5)	Back margin offset down
872	368	PACKED(15,5)	Back margin offset across
880	370	PACKED(15,5)	Length of page
888	378	PACKED(15,5)	Width of page
896	380	CHAR(10)	Measurement method
906	38A	CHAR(1)	Advanced Function Presentation™ (AFP™) resource
907	38B	CHAR(10)	Character set name
917	395	CHAR(10)	Character set library name
927	39F	CHAR(10)	Code page name
937	3A9	CHAR(10)	Code page library name
947	3B3	CHAR(10)	Coded font name
957	3BD	CHAR(10)	Coded font library name

Offset		Type	Field
Dec	Hex		
967	3C7	CHAR(10)	DBCS-coded font name
977	3D1	CHAR(10)	DBCS-coded font library name
987	3DB	CHAR(10)	User-defined file
997	3E5	CHAR(10)	Reduce output
1007	3EF	CHAR(1)	Constant back overlay
1008	3F0	BINARY(4)	Output bin
1012	3F4	BINARY(4)	CCSID
1016	3F8	CHAR(100)	User-defined text
1116	45C	CHAR(8)	System where file created
1124	464	CHAR(8)	ID where file created
1132	46C	CHAR(10)	User who created file
1142	476	CHAR(2)	Reserved
1144	478	BINARY(4)	Offset to user-defined options
1148	47C	BINARY(4)	Number of user-defined options returned
1152	480	BINARY(4)	Length of each user-defined option entry
1156	484	CHAR(255)	User-defined data
1411	583	CHAR(10)	User-defined object name
1421	58D	CHAR(10)	User-defined object library name
1431	597	CHAR(10)	User object type
1441	5A1	CHAR(3)	Reserved
1444	5A4	PACKED(15,5)	Character set point size
1452	5AC	PACKED(15,5)	Coded font point size
1460	5B4	PACKED(15,5)	DBCS-coded font point size
1468	5BC	BINARY(4)	Auxiliary storage pool
1472	5C0	BINARY(4)	Spooled file size
1476	5C4	BINARY(4)	Spooled file size multiplier
1480	5C8	BINARY(4)	Internet print protocol job identifier
1484	5CC	CHAR(1)	Spooled file creation security method
1485	5CD	CHAR(1)	Spooled file creation authentication method
1486	5CE	CHAR(7)	Date writer began processing spooled file
1493	5D5	CHAR(6)	Time writer began processing spooled file
1499	5DB	CHAR(7)	Date writer completed processing spooled file
1506	5E2	CHAR(6)	Time writer completed processing spooled file
1512	5E8	CHAR(8)	Job system name
1520	5F0	CHAR(10)	Auxiliary storage pool device name
1530	5FA	CHAR(7)	Expiration date

## SPLA0200 Format

The following table shows the information returned for the SPLA0200 format. For more details about the fields in the following table, see Field Descriptions.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Bytes returned
4	4	BINARY(4)	Bytes available
8	8	CHAR(8)	Format name
16	10	CHAR(16)	Internal job identifier
32	20	CHAR(16)	Internal spooled file identifier
48	30	CHAR(10)	Job name
58	3A	CHAR(10)	User name
68	44	CHAR(6)	Job number
74	4A	CHAR(10)	Spooled file name
84	54	BINARY(4)	Spooled file number
88	58	CHAR(10)	Form type
98	62	CHAR(10)	User-specified data
108	6C	CHAR(10)	Status
118	76	CHAR(10)	File available
128	80	CHAR(10)	Hold file before written
138	8A	CHAR(10)	Save file after written
148	94	BINARY(4)	Total pages
152	98	BINARY(4)	Page or record being written
156	9C	BINARY(4)	Starting page
160	A0	BINARY(4)	Ending page
164	A4	BINARY(4)	Last page printed
168	A8	BINARY(4)	Restart printing
172	AC	BINARY(4)	Total copies
176	B0	BINARY(4)	Copies left to produce
180	B4	BINARY(4)	Lines per inch
184	B8	BINARY(4)	Characters per inch
188	BC	CHAR(2)	Output priority
190	BE	CHAR(10)	Output queue name
200	C8	CHAR(10)	Output queue library name
210	D2	CHAR(7)	Date file opened (created)
217	D9	CHAR(6)	Time file opened (created)
223	DF	CHAR(10)	Device file name
233	E9	CHAR(10)	Device file library name
243	F3	CHAR(10)	Program that opened file name
253	FD	CHAR(10)	Program that opened file library name
263	107	CHAR(15)	Accounting code
278	116	CHAR(30)	Print text

Offset		Type	Field
Dec	Hex		
308	134	BINARY(4)	Record length
312	138	BINARY(4)	Maximum records
316	13C	CHAR(10)	Device type
326	146	CHAR(10)	Printer device type
336	150	CHAR(12)	Document name
348	15C	CHAR(64)	Folder name
412	19C	CHAR(8)	System/36 procedure name
420	1A4	CHAR(10)	Print fidelity
430	1AE	CHAR(1)	Replace unprintable characters
431	1AF	CHAR(1)	Replacement character
432	1B0	BINARY(4)	Page length
436	1B4	BINARY(4)	Page width
440	1B8	BINARY(4)	Number of separators
444	1BC	BINARY(4)	Overflow line number
448	1C0	CHAR(10)	Multi-byte data
458	1CA	CHAR(10)	DBCS extension characters
468	1D4	CHAR(10)	DBCS shift-out shift-in (SO/SI) spacing
478	1DE	CHAR(10)	DBCS character rotation
488	1E8	BINARY(4)	DBCS characters per inch
492	1EC	CHAR(10)	Graphic character set
502	1F6	CHAR(10)	Code page
512	200	CHAR(10)	Form definition name
522	20A	CHAR(10)	Form definition library name
532	214	BINARY(4)	Source drawer
536	218	CHAR(10)	Printer font
546	222	CHAR(6)	System/36 spooled file identifier
552	228	BINARY(4)	Page rotation
556	22C	BINARY(4)	Justification
560	230	CHAR(10)	Print on both sides (duplex)
570	23A	CHAR(10)	Fold records
580	244	CHAR(10)	Control character
590	24E	CHAR(10)	Align forms
600	258	CHAR(10)	Print quality
610	262	CHAR(10)	Form feed
620	26C	CHAR(71)	Volumes (array)
691	2B3	CHAR(17)	File label identifier
708	2C4	CHAR(10)	Exchange type
718	2CE	CHAR(10)	Character code
728	2D8	BINARY(4)	Total records
732	2DC	BINARY(4)	Multiple up (pages per side)

Offset		Type	Field
Dec	Hex		
736	2E0	CHAR(10)	Front overlay name
746	2EA	CHAR(10)	Front overlay library name
756	2F4	PACKED(15,5)	Front overlay offset down
764	2FC	PACKED(15,5)	Front overlay offset across
772	304	CHAR(10)	Back overlay name
782	30E	CHAR(10)	Back overlay library name
792	318	PACKED(15,5)	Back overlay offset down
800	320	PACKED(15,5)	Back overlay offset across
808	328	CHAR(10)	Unit of measure
818	332	CHAR(10)	Page definition name
828	33C	CHAR(10)	Page definition library name
838	346	CHAR(10)	Line spacing
848	350	PACKED(15,5)	Point size
856	358	BINARY(4)	Maximum spooled data record size
860	35C	BINARY(4)	Spooled file buffer size
864	360	CHAR(6)	Spooled file level
870	366	ARRAY(4)	Coded font array
886	376	CHAR(10)	Channel mode
896	380	ARRAY(12)	Channel value array
944	3B0	CHAR(8)	Graphics token
952	3B8	CHAR(10)	Record format
962	3C2	CHAR(2)	Reserved
964	3C4	PACKED(15,5)	Height of drawer 1
972	3CC	PACKED(15,5)	Width of drawer 1
980	3D4	PACKED(15,5)	Height of drawer 2
988	3DC	PACKED(15,5)	Width of drawer 2
996	3E4	BINARY(4)	Number of buffers
1000	3E8	BINARY(4)	Maximum forms width
1004	3EC	BINARY(4)	Alternate forms width
1008	3F0	BINARY(4)	Alternate forms length
1012	3F4	BINARY(4)	Alternate lines per inch
1016	3F8	CHAR(2)	System/38™ Text Utility flags
1018	3FA	CHAR(1)	File open
1019	3FB	CHAR(1)	Page count estimated
1020	3FC	CHAR(1)	File stopped on page boundary
1021	3FD	CHAR(1)	TRC for 1403
1022	3FE	CHAR(1)	Define characters
1023	3FF	CHAR(1)	Characters per inch changes
1024	400	CHAR(1)	Transparency
1025	401	CHAR(1)	Double-wide characters

Offset		Type	Field
Dec	Hex		
1026	402	CHAR(1)	DBCS character rotation commands
1027	403	CHAR(1)	Extended code page
1028	404	CHAR(1)	FFT emphasis
1029	405	CHAR(1)	3812 SCS
1030	406	CHAR(1)	Set Line Density command
1031	407	CHAR(1)	Graphics error actions
1032	408	CHAR(1)	5219 commands
1033	409	CHAR(1)	3812 SCS commands
1034	40A	CHAR(1)	Field outlining
1035	40B	CHAR(1)	Final form text
1036	40C	CHAR(1)	Bar code
1037	40D	CHAR(1)	Color
1038	40E	CHAR(1)	Drawer change
1039	40F	CHAR(1)	Character ID
1040	410	CHAR(1)	Lines per inch changes
1041	411	CHAR(1)	Font
1042	412	CHAR(1)	Highlight
1043	413	CHAR(1)	Page rotate
1044	414	CHAR(1)	Subscript
1045	415	CHAR(1)	Superscript
1046	416	CHAR(1)	DDS
1047	417	CHAR(1)	Final form feed
1048	418	CHAR(1)	SCS data
1049	419	CHAR(1)	User-generated data stream
1050	41A	CHAR(1)	Graphics
1051	41B	CHAR(1)	Unrecognizable data
1052	41C	CHAR(1)	ASCII transparency
1053	41D	CHAR(1)	IPDS™ transparent data
1054	41E	CHAR(1)	OfficeVision®
1055	41F	CHAR(1)	Lines-per-inch (lpi) value not supported
1056	420	CHAR(1)	CPA3353 message
1057	421	CHAR(1)	Set exception
1058	422	CHAR(1)	Carriage control characters
1059	423	CHAR(1)	Page position
1060	424	CHAR(1)	Character not valid
1061	425	CHAR(1)	Lengths present
1062	426	CHAR(1)	5A present
1063	427	CHAR(1)	Reserved
1064	428	BINARY(4)	Number of font array entries
1068	42C	BINARY(4)	Number of resource library entries

Offset		Type	Field
Dec	Hex		
1072	430	CHAR(1153)	Font equivalence array
2225	8B1	CHAR(631)	Resource library array
2856	B28	CHAR(1)	i5/OS <sup>®</sup> -created AFPDS
2857	B29	CHAR(1)	Job character ID specified
2858	B2A	CHAR(1)	S36 CONTINUE-YES
2859	B2B	CHAR(10)	Decimal format
2869	B35	CHAR(7)	Date file last used
2876	B3C	CHAR(1)	Page groups
2877	B3D	CHAR(1)	Group level index tags
2878	B3E	CHAR(1)	Page level index tags
2879	B3F	CHAR(1)	IPDS pass-through
2880	B40	BINARY(4)	Offset to user resource library list
2884	B44	BINARY(4)	Number of user resource library list entries
2888	B48	BINARY(4)	Length of user resource library list entry
2892	B4C	CHAR(2)	Reserved
2894	B4E	CHAR(1)	Corner staple
2895	B4F	CHAR(1)	Edge-stitch reference edge
2896	B50	PACKED(15,5)	Offset from edge-stitch reference edge
2904	B58	BINARY(4)	Edge-stitch number of staples
2908	B5C	BINARY(4)	Offset to edge-stitch staple offset list
2912	B60	BINARY(4)	Number of edge-stitch staple offset entries
2916	B64	BINARY(4)	Length of edge-stitch staple offset entry
2920	B68	CHAR(10)	Font resolution for formatting
2930	B72	CHAR(1)	Record format name present in data stream
2931	B73	CHAR(1)	Saddle-stitch reference edge
2932	B74	BINARY(4)	Saddle-stitch number of staples
2936	B78	BINARY(4)	Offset to saddle-stitch staple offset list
2940	B7C	BINARY(4)	Number of saddle-stitch staple offset entries
2944	B80	BINARY(4)	Length of saddle-stitch staple offset entry
2948	B84	PACKED(15,0)	Data stream size
2956	B8C	BINARY(4)	Offset to library list
2960	B90	BINARY(4)	Number of library list entries
2964	B94	BINARY(4)	Length of library list entry
2968	B98	BINARY(4)	Offset to Internet Print Protocol spooled file attributes
2972	B9C	BINARY(4)	Offset to Save and Restore spooled file attributes
2976	BA0	BINARY(4)	CCSID of job
2980	BA4	CHAR(172)	Reserved
3152	C50	PACKED(15,5)	Front margin offset down
3160	C58	PACKED(15,5)	Front margin offset across
3168	C60	PACKED(15,5)	Back margin offset down

Offset		Type	Field
Dec	Hex		
3176	C68	PACKED(15,5)	Back margin offset across
3184	C70	PACKED(15,5)	Length of page
3192	C78	PACKED(15,5)	Width of page
3200	C80	CHAR(10)	Measurement method
3210	C8A	CHAR(1)	Advanced Function Presentation (AFP) resource
3211	C8B	CHAR(10)	Character set name
3221	C95	CHAR(10)	Character set library name
3231	C9F	CHAR(10)	Code page name
3241	CA9	CHAR(10)	Code page library name
3251	CB3	CHAR(10)	Coded font name
3261	CBD	CHAR(10)	Coded font library name
3271	CC7	CHAR(10)	DBCS-coded font name
3281	CD1	CHAR(10)	DBCS-coded font library name
3291	CDB	CHAR(10)	User-defined file
3301	CE5	CHAR(10)	Reduce output
3311	CEF	CHAR(1)	Constant back overlay
3312	CF0	BINARY(4)	Output bin
3316	CF4	BINARY(4)	CCSID
3320	CF8	CHAR(100)	User-defined text
3420	D5C	CHAR(8)	System where file originated
3428	D64	CHAR(8)	Net ID where file originated
3436	D6C	CHAR(10)	User who owned file
3446	D76	CHAR(2)	Reserved
3448	D78	BINARY(4)	Offset to user-defined options
3452	D7C	BINARY(4)	Number of user-defined options returned
3456	D80	BINARY(4)	Length of each user-defined option entry
3460	D84	CHAR(255)	User-defined data
3715	E83	CHAR(10)	User-defined object name
3725	E8D	CHAR(10)	User-defined object library name
3735	E97	CHAR(10)	User object type
3745	EA1	CHAR(3)	Reserved
3748	EA4	PACKED(15,5)	Character set point size
3756	EAC	PACKED(15,5)	Coded font point size
3764	EB4	PACKED(15,5)	DBCS-coded font point size
3772	EBC	BINARY(4)	Auxiliary storage pool
3776	EC0	BINARY(4)	Spooled file size
3780	EC4	BINARY(4)	Spooled file size multiplier
3784	EC8	BINARY(4)	Internet print protocol job identifier
3788	ECC	CHAR(1)	Spooled file creation security method
3789	ECD	CHAR(1)	Spooled file creation authentication method

Offset		Type	Field
Dec	Hex		
3790	ECE	CHAR(7)	Date writer began processing spooled file
3797	ED5	CHAR(6)	Time writer began processing spooled file
3803	EDB	CHAR(7)	Date writer completed processing spooled file
3810	EE2	CHAR(6)	Time writer completed processing spooled file
3816	EE8	CHAR(8)	Job system name
3824	EF0	CHAR(10)	Auxiliary storage pool device name
3834	EFA	CHAR(7)	Expiration date

## User-Defined Option Entry

Each user-defined option entry has the following structure. For more details about the fields in the following table, see “Field Descriptions” on page 354.

**Note:** One entry is returned for each user-defined option. The number of entries may vary from one spooled file to another.

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	User-defined option

## User Resource Library List Entry

Each user resource library list entry has the following structure. For more details about the fields in the following table, see “Field Descriptions” on page 354.

**Note:** One entry is returned for each user resource library. The number of entries may vary from one spooled file to another.

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	User resource library

## Edge-Stitch Staple Offset Entry

Each edge-stitch staple offset entry has the structure stated in the following table. For more details about the fields in the following table, see “Field Descriptions” on page 354.

**Note:** One entry is returned for each edge-stitch staple. The number of entries may vary from one spooled file to another.

Offset		Type	Field
Dec	Hex		
0	0	PACKED(15,5)	Edge-stitch staple offset

## Saddle-Stitch Staple Offset Entry

Each saddle-stitch staple offset entry has the following structure. For more details about the fields in the following table, see “Field Descriptions” on page 354.

**Note:** One entry is returned for each saddle-stitch staple. The number of entries may vary from one spooled file to another.

Offset		Type	Field
Dec	Hex		
0	0	PACKED(15,5)	Saddle-stitch staple offset

## Library List Entry

Each library list entry has the following structure. For more details about the fields in the following table, see “Field Descriptions” on page 354.

**Note:** One entry is returned for each library. The number of entries may vary from one spooled file to another.

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	Library name

## Internet Print Protocol Spooled File Attributes.

The Internet Print Protocol (IPP) specific spooled file attributes has the following structure. For more details about the fields in the following table, see “Field Descriptions” on page 354.

**Note:** These attributes are present only if the spooled file was created through IPP.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Length of IPP attributes
4	4	BINARY(4)	CCSID of the IPP attributes
8	8	CHAR(63)	Natural language of IPP attributes
71	47	CHAR(127)	IPP printer name
198	C6	CHAR(255)	IPP job name
453	1C5	CHAR(63)	IPP job name natural language
516	204	CHAR(255)	IPP job-originating user name
771	303	CHAR(63)	IPP job-originating user name natural language

## Save and Restore Spooled File Attributes.

The Save and Restore specific spooled file attributes has the following structure. For more details about the fields in the following table, see “Field Descriptions” on page 354.

**Note:** These attributes are present only if the spooled file was saved or restored.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Length of save and restore attributes
4	4	BINARY(4)	Save sequence number
8	8	CHAR(13)	Spooled file saved date and time
21	15	CHAR(13)	Spooled file restore date and time
34	22	CHAR(71)	Save volume ID
105	69	CHAR(10)	Save command
115	73	CHAR(10)	Save device
125	7D	CHAR(10)	Save file name
135	87	CHAR(10)	Save file library name
145	91	CHAR(17)	Save label

## Field Descriptions

**Accounting code.** An identifier assigned by the system to record the resources used to write this file.

**Advanced Function Presentation (AFP) resource.** Whether this spooled file refers to AFP resources external to this spooled file, for example, page segments or overlays.

The possible values are:

- Y The spooled file refers to AFP resources external to the spooled file.
- N The spooled file does not refer to AFP resources external to the spooled file.

**Align forms.** Whether a forms alignment message is sent prior to printing this file. The options are \*YES or \*NO.

**Alternate forms length.** The length of the alternate forms in lines. This field applies only to files created by the OfficeVision program and should be set to 0 by an application building this format as opposed to retrieving the fields.

**Alternate forms width.** The width of the alternate forms in character positions. This field applies only to files created by the OfficeVision program and should be set to 0 by an application building this format as opposed to retrieving the fields.

**Alternate lines per inch.** The lines per inch for the alternate forms. This field applies only to files created by the OfficeVision program and should be set to 0 by an application building this format as opposed to retrieving the fields.

**ASCII transparency.** For SCS files, whether ASCII commands are embedded in the ASCII transparency command. The ASCII transparency command is command\_ID/command\_length/ASCII\_data. The command ID is a 1-byte field with value hex 03. The command length is a 1-byte field that contains the length of the command length field and the ASCII data field.

Valid values are Y (yes) or N (no).

**Auxiliary storage pool.** The number of the auxiliary storage pool (ASP) that the spooled file is stored on.

The possible values are:

1 System ASP.  
2-255 One of the ASPs.

**Auxiliary storage pool device name.** The name of the auxiliary storage pool (ASP) device that the spooled file is stored on.

The possible values are:

\*SYSBAS The spooled file resides on the system ASP (ASP 1) or one of the defined basic user ASPs (ASP 2-32).  
*auxiliary storage pool device name* The name of the auxiliary storage pool (ASP) device that the spooled file resides on.

**Back margin offset across.** For the back side of a piece of paper, it specifies, in either inches or centimeters (specified in the unit of measure (UOM) field), how far in from the left side of the page printing starts.

The possible values are:

0-57.79 The offset in inches or centimeters, according to the unit of measure value (see Unit of measure (page 381)). The maximum offset is 57.79 centimeters or 22.75 inches.

**Back margin offset down.** For the back side of a piece of paper, it specifies, in either inches or centimeters (specified in the unit of measure (UOM) field), how far down from the top of the page printing starts.

The possible values are:

-1 \*FRONTMGN. The back margin offsets are the same as the front margin offsets.  
-2 \*DEVD. For printers configured AFP(\*YES), no print border is used for back margin offsets across and down. Otherwise, the offset values are 0.  
0-57.79 The offset in inches or centimeters, according to the unit of measure value (see Unit of measure (page 381)). The maximum offset is 57.79 centimeters or 22.75 inches.

**Back overlay library name.** The name of the library containing the back overlay.

The possible values are:

\*CURLIB The current library for the job locates the back overlay.  
\*LIBL The library list locates the back overlay.  
*library name* This library is searched for the back overlay.

**Back overlay name.** The name of the back overlay (the material that prints on the back side of each page).

The possible values are:

\*FRONTOVL The back overlay is the same as the front overlay.  
\*NONE The file does not use a back overlay.  
*back overlay name* The name of the back overlay.

**Back overlay offset across.** The offset across from the point of origin where the overlay is printed.

The possible values are:

0-57.79 The offset in inches or centimeters, according to the unit of measure value (see Unit of measure (page 381)). The maximum offset is 57.79 centimeters or 22.75 inches.

**Back overlay offset down.** The offset down from the point of origin where the overlay is printed.

The possible values are:

0-57.79 The offset in inches or centimeters, according to the unit of measure value (see Unit of measure (page 381)). The maximum offset is 57.79 centimeters or 22.75 inches.

**Bar code.** Whether the spooled file contains bar codes created using the BARCODE keyword in the data description specifications (DDS).

For DEVTYPE(\*IPDS and \*AFPDS), a series of bar code commands is contained in the data stream.

Valid values are Y (yes) or N (no).

**Bytes available.** The number of bytes of data available to be returned. All available data is returned if enough space is provided.

**Bytes returned.** The number of bytes of data returned.

**Carriage control characters.** Whether the file has machine carriage control characters.

Valid values are Y (yes) or N (no).

**CCSID.** The coded character set identifier (CCSID) used when the user-defined text was created using the Change User Print Information (CHGUSRPRTI) command.

**CCSID of IPP attributes.** The coded character set identifier (CCSID) used when the IPP attributes were set for the spooled file. The spooled file attributes affected are:

- IPP printer name
- IPP job name
- IPP job-originating user name

**CCSID of job.** The coded character set identifier (CCSID) of the job when the spooled file was created. This CCSID may or may not represent the contents of the spooled file.

The possible values are:

0 The CCSID is unknown. The CCSID of the job that generated the contents of this spooled file was not saved when the spooled file was created.

1-65535 The coded character set identifier.

**Channel value array.** Contains the skip-to line number associated with the channel value. (The first entry in the array applies to channel 1.) The variable returns a maximum of 12 binary(4) values. Each of the values returned is 0 if the mode is \*NORMAL.

**Channel mode.** A variable indicating the channel value mode.

The possible values are:

*\*NORMAL*            The normal channel values (1 equals skip to line 1, 2 through B equal space one line before printing, C equals Skip to overflow line).  
*blank*                The channel values specified in the channel code array are used.

**Character code.** Whether the coding for the diskette file is in ASCII or EBCDIC form.

**Character ID.** Whether the character ID can change within the file.

- For DEVTYPE(\*SCS), a Set CGCS through Local ID is contained in the spooled file.
- For DEVTYPE(\*IPDS), a Load Font Equivalence command is inserted into the Load Font Equivalence table.
- For DEVTYPE(\*AFPDS), a map code font-2 (MCF-2) structured field is contained in the spooled file.

Valid values are Y (yes) or N (no).

**Character not valid.** Whether incorrect character errors are reported.

Valid values are Y (yes) or N (no).

**Character set library name.** The name of the library containing the font character set object.

The possible values are:

*\*CURLIB*            The current library is searched for the font character set object.  
*\*LIBL*                The library list is used to locate the font character set object.  
*library name*        This library is searched for the font character set object.

**Character set name.** The name of the font character set object used to print this file.

The possible values are:

*\*FONT*                The information specified on the font parameter is used instead of the character set and code page.  
*character set name*    The name of the font character set object to use.

**Character set point size.** The point size in which this file's characters (defined by the character set field) should be printed.

The possible values are:

*000.0*                \*NONE. The character set does not have a point size.  
*000.1-999.9*        The point size of the character set.

**Characters per inch.** The number (in tenths) of characters per horizontal inch, defined in the printer file. The value 100 indicates 10 characters per inch.

**Characters per inch changes.** Whether the spooled file changes the characters per inch (cpi) within the file.

For DEVTYPE(\*SCS), a Set Character Distance command is contained in the spooled file.

Valid values are Y (yes) or N (no).

**Code page.** The mapping of graphic characters to code points for this printer. For \*DEV D, the operating system sends a device default character set value to the printer, which determines the character set to be used.

**Code page library name.** The name of the library containing the code page used to print this spooled file.

The possible values are:

<i>*CURLIB</i>	The current library is searched for the code page name.
<i>*LIBL</i>	The library list is used to locate the code page.
<i>library name</i>	This library is searched for the code page name.

**Code page name.** The name of the code page used to print this spooled file.

Code pages are groups of characters. Within a code page unique hexadecimal identifiers are assigned to each of the characters.

**Coded font array.** The name of the fonts used for printing a line data file. This corresponds to the CHARS parameter of MVS™ and VM printing services. The variable returns a maximum of 4 four-character-coded fonts. The variable is returned blank if no coded fonts are specified. The name specified does not include the two-character prefix of the coded font name (X0 through XG). For more information about the CHARS fields, see the Print Services Facility™/MVS Application Programming Guide or Print Services Facility User's Programmers Guide for VM.

**Coded font library name.** The name of the library containing the coded font used to print this spooled file.

The possible values are:

<i>*CURLIB</i>	The current library is searched for the coded font.
<i>*LIBL</i>	The library list is used to locate the coded font.
<i>library name</i>	This library is searched for the coded font.

**Coded font name.** The name of the coded font used to print this spooled file. A **coded font** is an AFP resource composed of a character set and a code page.

The possible values are:

<i>*FNTCHRSET</i>	The values used are the values specified on the character set name and library name and code page name and library name fields.
<i>coded font name</i>	The name of the coded font used to print this spooled file.

**Coded font point size.** The point size in which this file's characters (defined by the coded font field) should be printed.

The possible values are:

<i>000.0</i>	*NONE. The coded font does not have a point size.
<i>000.1-999.9</i>	The point size of the coded font.

**Color.** Whether an IPDS Write Text command or AFPDS presentation text data (PTX) structured field containing a Set Text Color text control is contained in the spooled file.

For DEVTYPE(\*IPDS and \*AFPDS), a Set Text Color text control is contained in the spooled file.

Valid values are Y (yes) or N (no).

**Constant back overlay.** Whether or not constant back overlay is used. Constant back allows the user to print overlays on the back side of each sheet of media without the need for the print application to generate blank pages. The constant back function causes blank pages to be generated by the operating system onto which the back overlay will be printed. The generated blank pages are called constant forms because no variable data from the user's print application is printed on the pages. Constant back overlay is ignored if the print on both sides field is set to \*NO.

The possible values are:

- 0 \*NOCONSTANT. Constant back overlay is not used.
- 1 \*CONSTANT. Constant back overlay is used.

**Control character.** Whether this printer file uses the American National Standard printer control character.

The possible values are:

- \*NONE No print control characters are passed in the data that is printed.
- \*FCFC The first character of every record is an American National Standard printer control character.

**Copies left to produce.** The remaining number of copies to be produced on the printer. Valid values are 1 through 255.

**CPA3353 message.** Whether message CPA3353 is issued when this file is printed.

Valid values are Y (yes) or N (no).

**Corner staple.** The reference corner to be used for a corner staple. A staple is driven into the media at the reference corner. Refer to your printer's documentation for information as to which reference corners are supported.

Valid values are:

- 0 \*NONE. A corner staple is not specified.
- 1 \*DEV. The default reference corner of the printer device is specified.
- 2 \*BOTRIGHT. The bottom right-hand corner of the paper is specified.
- 3 \*TOPRIGHT. The top right-hand corner of the paper is specified.
- 4 \*TOPLEFT. The top left-hand corner of the paper is specified.
- 5 \*BOTLEFT. The bottom left-hand corner of the paper is specified.

**Data stream size.** The number of bytes of data stream for the file. If the file is still open when the fields are retrieved, the data stream size is the number of bytes of data stream written to the file thus far. If the number of bytes of data stream is zero, possible reasons are there was a problem retrieving the number of data stream bytes or there have not been any bytes of data stream written to the spooled file yet.

**Date file last used.** >> The date, based on local job time, that the file data was last displayed, copied, printed, or sent; << or when the file attributes were last changed. The date is in the CYYMMDD format as follows:

- C Century, where 0 indicates years 19xx and 1 indicates years 20xx.

YY            Year  
MM            Month  
DD            Day

**Date file opened (created).** >> The date, based on local job time, that the file was opened in the CYYMMDD format where: <<

C            Century, where 0 indicates years 19xx and 1 indicates years 20xx.  
YY            Year  
MM            Month  
DD            Day

**Date writer began processing spooled file.** >> The date, based on local job time, that a spooling writer started processing the spooled file. << This field is set to blanks if the file has not been processed by a writer. The date is in the CYYMMDD format as follows:

C            Century, where 0 indicates years 19xx and 1 indicates years 20xx.  
YY            Year  
MM            Month  
DD            Day

**Date writer completed processing spooled file.** >> The date, based on local job time, that a spooling writer completed processing the spooled file << or the date the processing of the spooled file was aborted. This field is set to blanks if the file has not been processed by a writer or the writer has not completed processing the file. The date is in the CYYMMDD format as follows:

C            Century, where 0 indicates years 19xx and 1 indicates years 20xx.  
YY            Year  
MM            Month  
DD            Day

**DBCS character rotation.** Whether this printer file causes the double-byte character set (DBCS) characters to be rotated 90 degrees counterclockwise before printing. The possible values are \*YES and \*NO.

**DBCS character rotation commands.** Whether the double byte character set characters are rotated 90 degrees counterclockwise before printing.

- For DEVTYPE(\*SCS), a Set Text Orientation command is contained in the spooled file.
- For DEVTYPE(\*AFPDS), a map coded font-1 (MCF-1) structured field, that has the character rotation parameter set to hex 8700 in the spooled file.

Valid values are Y (yes) or N (no).

**DBCS characters per inch.** The number of double-byte characters to be printed per inch.

The possible values are:

-1        \*CPI One-half of the characters per inch value.  
-2        \*CONDENSED 20 double-byte characters per 3 inches.  
5        5 characters per inch.  
6        6 characters per inch.  
10       10 characters per inch.

**DBCS-coded font library name.** The name of the library containing the DBCS-coded font.

The possible values are:

*\*CURLIB*            The current library is searched for the DBCS-coded font.  
*\*LIBL*                The library list is used to locate the DBCS-coded font.  
*library name*        This library is searched for the DBCS-coded font.

**DBCS-coded font name.** The name of the DBCS-coded font used to print DBCS-coded data on printers configured as AFP(\*YES).

The possible values are:

*\*SYSVAL*            The DBCS-coded font specified in the system value is used.  
*DBCS-coded font name*    The name of the DBCS-coded font being used.

**DBCS-coded font point size.** The point size in which this file's DCBS characters (defined by the DBCS-coded font field) should be printed.

The possible values are:

*000.0*                \*NONE. The DBCS-coded font does not have a point size.  
*000.1-999.9*        The point size of the DBCS-coded font.

**DBCS extension characters.** Whether the system uses the extension character processing function for DBCS data.

The possible values are:

*\*YES*                The system processes DBCS extension characters.  
*\*NO*                  The system does not process DBCS extension characters; it prints extension characters as the undefined character.

**DBCS shift-out shift-in (SO/SI) spacing.** The presentation of shift-out and shift-in characters when printed.

The possible values are:

*\*YES*                Shift-out and shift-in characters occupy one space.  
*\*NO*                  Shift-out and shift-in characters occupy no space.  
*\*RIGHT*            Shift-out characters occupy no space; shift-in characters occupy 2 spaces.

**DDS.** Whether the file was constructed with DDS.

Valid values are Y (yes) or N (no).

**Decimal format.** Whether the decimal format from the DECFMT job attribute or from the printer file was used when the spooled file was created.

*\*FILE*                The decimal format used was from the printer file.  
*\*JOB*                  The decimal format used was from the DECFMT job attribute.

**Define characters.** Whether the spooled file defines or redefines unique print characters.

For DEVTYPE(\*SCS), a Load Alternate Characters command is contained in the spooled file.

Valid values are Y (yes) or N (no).

**Device file library name.** The name of the library that contains the device file.

**Device file name.** The name of the device file used to create the spooled file.

**Device type.** The type of device file for which this file is intended. The possible values are PRINTER, DISKETTE, or TAPE.

**Document name.** The name of the document that was the source of the spooled file.

**Double-wide characters.** Whether the spooled file prints everything twice as wide as normal.

For DEVTYPE(\*SCS), a Set Font Size command is contained in the data stream.

Valid values are Y (yes) or N (no).

**Drawer change.** Whether the printer drawer is changed within the spooled file.

- For DEVTYPE(\*SCS), a Page Presentation Media command is contained in the spooled file.
- For DEVTYPE(\*IPDS), an Execute Order Home State - Select Input Media Source is in the spooled file.
- For DEVTYPE(\*AFPDS), a media map and an invoke media map are in the spooled file.

Valid values are Y (yes) or N (no).

**Edge-stitch number of staples.** The number of staples in the edge-stitch.

The valid values are:

- 0            \*DEV.D. Use the device default. This value is used if either of the following was specified:
- The offset of each staple is specified in the edge-stitch staple offset list. The printer will automatically determine the number of staples to be used.
  - Zero (\*DEV.D) was specified for the number of edge-stitch staple offset entries field. The printer will use its default value for both the number of staples to be used as well as the offset of each staple.
- 1-122        Number of staples to be used in the edge-stitch.

**Edge-stitch reference edge.** The edge of the paper to be used as a reference in determining where the edge stitching is to be located.

Valid values are:

- 0            \*NONE. No edge stitching is specified.
- 1            \*DEV.D. The default reference edge used by the printer device is specified.
- 2            \*BOT. The bottom edge of the paper is specified.
- 3            \*RIGHT. The right-hand edge of the paper is specified.
- 4            \*TOP. The top edge of the paper is specified.
- 5            \*LEFT. The left-hand edge of the paper is specified.

**Edge-stitch staple offset.** For the spooled file, the placement of staples along the finishing margin in either inches or centimeters (specified in the Unit of measure (page 381) (UOM) field). The finishing

margin can be thought of as an imaginary line parallel to the edge of the paper where the staples will be placed. The position of the finishing margin relative to the physical edge is specified in the offset from edge-stitch reference edge field.

If you want the staples placed along the left or right edge of the paper, specify the offset of the first staple by measuring from the point where the finishing margin intersects the bottom edge of paper to where you want the center of the staple to be. For subsequent staples, specify the offset by measuring from the same point (not from the previous staple).

If you want the staples placed along the top or bottom edge of the paper, specify the offset of the first staple by measuring from the point where the finishing margin intersects the left edge of paper to where you want the center of the staple to be. For subsequent staples, specify the offset by measuring from the same point (not from the previous staple).

The possible values are:

0-57.79      The offset in inches or centimeters, according to the unit of measure value (see Unit of measure (page 381)). The maximum offset is 57.79 centimeters or 22.75 inches.

**Ending page.** The page at which printing is to end for the file. 0 or 2147483647 indicates the last page.

**Exchange type.** The exchange type of the diskette file.

The possible values are:

*STD	The system allows the exchange type based on the diskette type and sector size.
*BASIC	The basic exchange type.
*H	The H exchange type.
*I	The I exchange type.

**Expiration date.** The date the file will be eligible for removal from the system by the Delete Expired Spooled Files (DLTEXPSPLF) command and blank if the file will not expire. The spooled file will expire at 23:59:59, local system time on the date specified. The date is in the CYYMMDD format as follows:

C	Century, where 0 indicates years 19xx and 1 indicates years 20xx.
YY	Year
MM	Month
DD	Day

**Extended code page.** Whether the extended code page changes within the file.

- For DEVTYPE(\*SCS), a Set CGCS through GCID is contained in the spooled file.
- For DEVTYPE(\*IPDS), a Load Font Equivalence command is inserted into the Load Font Equivalence table.
- For DEVTYPE(\*AFPDS), a map code font-2 (MCF-2) structured field is contained in the spooled file.

Valid values are Y (yes) or N (no).

**FFT emphasis.** Whether the spooled file contains text that is to appear darker than the surrounding text.

For DEVTYPE(\*SCS), Begin Emphasis and End Emphasis commands are contained in the spooled file.

Valid values are Y (yes) or N (no).

**Field outlining.** Whether the spooled file outlines fields of data with boxes. If it does, the file must be printed on a printer that supports field outlining.

For DEVTYPE(\*SCS), a Define Grid Line command is contained in the spooled file.

Valid values are Y (yes) or N (no).

**File available.** The time when this file becomes available to an output device for processing.

The possible values are:

<i>*IMMED</i>	The file is available as soon as the file is opened.
<i>*FILEEND</i>	The file is available as soon as the file is closed.
<i>*JOBEND</i>	The file is available when the job that owns the file is completed.

**File label identifier.** The diskette label used when the system last saved the object.

**File open.** Whether the file is still open when fields are retrieved.

Valid values are Y (yes) or N (no).

**File stopped on page boundary.** Whether the file has stopped printing on a page boundary.

Valid values are Y (yes) or N (no).

**Final form feed.** Whether the Final Form Feed command is in the printer file.

Valid values are Y (yes) or N (no).

**Final form text.** Whether this spooled file contains various functions that are supported on letter-quality printers.

- For DEVTYPE(\*SCS), a Document Content Architecture (DCA) command, is contained in the spooled file. DCA commands include:
  - Required New line
  - Required Form Feed
  - Indent Tab
  - Set Presentation Page Size
  - Set Horizontal Margins
  - Set Vertical Margins
  - Release Left Margin
  - Set Line Spacing
  - Set Single Line Distance
  - Justify Text Field Format
  - Set Justify Mode
  - Set Horizontal Tab Stops
  - Set Indent Level
  - Set Exception Action
  - Set Presentation Color
  - Set Spacing Variable
  - Begin and End Overstrike
  - Begin and End Underscore

- Bell
- Switch
- Repeat
- Tab
- Backspace
- Unit Backspace
- Substitute
- Word Underscore

Valid values are Y (yes) or N (no).

**Fold records.** Whether records exceeding the printer forms width are folded (wrapped) to the next line. The possible values are \*YES or \*NO.

**Folder name.** The name of the folder that contains the source document.

**Font.** Whether the spooled file uses multiple fonts.

- For DEVTYPE(\*SCS), a Set Font Global command is in the spooled file.
- For DEVTYPE(\*IPDS), a Load Font Equivalence command is contained in the Load Font Equivalence table.
- For DEVTYPE(\*AFPDS), a map coded font-1 (MCF-1) or map coded font-2 (MCF-2) structured field is in the spooled file.

Valid values are Y (yes) or N (no).

**Font equivalence array.** The data portion of the Load Font Equivalence (LFE) command for intelligent printer data streams (IPDSs). The variable returns a maximum of 72 sixteen-character font equivalence entries. The current maximum is 48. The additional array space is provided for further expansion. If more than 72 font equivalence entries are used, a 1 is returned in the 1153rd character of the variable; otherwise, the 1153rd character is blank. The format of this entry is described in the *IPDS Reference* manual, S544-3417. This manual is available from the IBM® Publications Center .

**Font resolution for formatting.** The resolution Print Services Facility/400 (PSF/400) should use to print the spooled file when printing to a multiple resolution printer and either of the following conditions exist:

- The information is not specified in the spooled file data stream.
- The font used in the spooled file could not be found.

The possible values are:

*DEV	PSF/400 is to use the value specified on the font resolution for formatting parameter in the PSF configuration object. The PSF configuration object is specified in the printer device description.
*SEARCH	PSF is to search for a font that will work. The search method is outlined in the Printer Device Programming manual.
240	PSF/400 is to use only fonts that are 240 pels per inch.
300	PSF/400 is to use only fonts that are 300 pels per inch.

**Form definition library name.** The name of the library that contains the form definition.

**Form definition name.** The name of the form definition to use for this print request or one of the following special values:

<i>*DEV</i>	The form definition in the printer device description will be used.
<i>*INLINE</i>	The form definition defined in the spooled file data stream will be used.
<i>*INLINED</i>	The form definition defined in the spooled file data stream will be used. If a form definition is not found, the form definition in the printer device description will be used.
<i>F1DFLT</i>	The form definition defined in the spooled file data stream will be used. F1DFLT is the in-line form definition built by i5/OS for spooled files of printer device type *AFPDS. When the field i5/OS created AFPDS is set to 'Y', the form definition library name field is set to blanks.

**Form feed.** The manner in which forms feed to the printer.

The possible values are:

<i>*CONT</i>	Continuous forms
<i>*CUT</i>	Manually fed cut forms
<i>*AUTOCUT</i>	Automatically fed cut forms
<i>*DEV</i>	As defined in the device description
<i>*CONT2</i>	Secondary continuous forms

**Form type.** The type of form to be loaded in the printer to print this file.

**Format name.** The name of the format used to return information.

**Front margin offset across.** For the front side of a piece of paper, it specifies, in either inches or centimeters (specified in the unit of measure (UOM) field), how far in from the left side of the page printing starts.

The possible values are:

<i>0-57.79</i>	The offset in inches or centimeters, according to the unit of measure value (see Unit of measure (page 381)). The maximum offset is 57.79 centimeters or 22.75 inches.
----------------	--

**Front margin offset down.** For the front side of a piece of paper, it specifies, in either inches or centimeters (specified in the unit of measure (UOM) field), how far down from the top of the page printing starts.

The possible values are:

<i>-2</i>	<i>*DEV</i> . For printers configured AFP(*YES), no print border is used for front margin offsets across and down. Otherwise, the offset values are 0.
<i>0-57.79</i>	The offset in inches or centimeters, according to the unit of measure value (see Unit of measure (page 381)). The maximum offset is 57.79 centimeters or 22.75 inches.

**Front overlay library name.** The name of the library containing the front overlay.

The possible values are:

<i>*CURLIB</i>	The current library for the job locates the front overlay.
<i>*LIBL</i>	The library list locates the front overlay.
<i>library name</i>	This library is searched for the front overlay.

**Front overlay name.** The name of the front overlay (the material that prints on the front side of each page).

The possible values are:

*\*NONE*                The file does not use the front overlay.  
*front overlay name* The name of the front overlay.

**Front overlay offset across.** The offset across from the point of origin where the overlay is printed.

The possible values are:

*0-57.79*            The offset in inches or centimeters, according to the Unit of measure (page 381) value. The maximum offset is 57.79 centimeters or 22.75 inches.

**Front overlay offset down.** The offset down from the point of origin where the overlay is printed.

The possible values are:

*0-57.79*            The offset in inches or centimeters, according to the Unit of measure (page 381) value. The maximum offset is 57.79 centimeters or 22.75 inches.

**Graphic character set.** The set of graphic characters to be used when printing this file. For *\*DEV*, the system gets the graphic character set from the printer device description.

**Graphics.** Whether the spooled file contains graphics.

Valid values are Y (yes) or N (no).

**Graphics error actions.** Whether the file contains graphic error action commands.

For SCS files, the file contains one or more Set Graphic Error Action commands.

Valid values are Y (yes) or N (no).

**Graphics token.** The printer type on which the graphics in this file can be printed.

The possible values are:

*4214*                Data stream contains graphics for a 4214 printer.  
*4234*                Data stream contains graphics for a 4234 SCS printer.  
*522X*                Data stream contains graphics for a 522X printer.  
*IPDS*                Data stream contains graphics for an IPDS printer.

**Group level index tags.** Whether the spooled file contains group level index tags as indicated by the DDS keyword *DOCIDXTAG* with a tag level of *GROUP*. The possible values are Y (yes) or N (no).

**Height of drawer 1.** The height in inches of the paper in drawer 1. This field is for internal use and should be set to 0 by an application building this format as opposed to retrieving the fields.

**Height of drawer 2.** The height in inches of the paper in drawer 2. This field is for internal use and should be set to 0 by an application building this format as opposed to retrieving the fields.

**Highlight.** Whether the spooled file contains text that is to appear darker than the surrounding text.

- For *DEVTYPE(\*SCS)*, data is overprinted to get the bold effect.

- For DEVTYPE(\*IPDS), a Load Font Equivalence command is contained in the Load Font Equivalence table, which has the bold field set on.
- For DEVTYPE(\*AFPDS), a map coded font-2 (MCF-2) structured field that has the font weight class parameter set to hex 07 is in the spooled file.

Valid values are Y (yes) or N (no).

**Hold file before written.** Whether the file is held. The hold parameter handles this function on a Create Printer File (CRTPRPF), Override Printer File (OVRPRTF), or Change Printer File (CHGPRTF) command.

The possible values are:

- \*YES           The file is held.
- \*NO            The file is not held.

**Internal job identifier.** The internal identifier for the job. Only the i5/OS APIs use this identifier, not any other interface on the system. The identifier is not valid following an initial program load (IPL). If you attempt to use it after an IPL, an exception occurs. This identifier is always valid when used with the internal spooled file identifier as input to i5/OS APIs. For spooled files which have been detached from their jobs or which are on independent disk pools, this value is not valid as input to some i5/OS APIs, such as the Retrieve Job Information (QUSRJOBI) API, that take this value without the internal spooled file identifier.

**Internal spooled file identifier.** The value used as input to other programs to improve the performance of locating the spooled file on the system. Only the i5/OS APIs use this identifier, not any other interface on the system. The identifier is not valid following an initial program load (IPL) or if the spooled file has been moved between auxiliary storage pools (ASPs). If you attempt to use an invalid internal spooled file identifier, an exception occurs.

**Internet print protocol job identifier.** The IPP job identifier assigned by the system based on the output queue to which the file was added or moved. This value ranges from 1 to 2147483647 and is not guaranteed to be unique for a given output queue.

**IPDS pass-through.** Whether IPDS pass-through is done for the spooled file. Not all SCS or IPDS spooled files are eligible for IPDS pass-through. They may contain special functions that require transform to AFPDS for correct printing. IPDS pass-through allows only those spooled files eligible for IPDS pass-through to bypass the extra transforms. Those spooled files not eligible for IPDS pass-through will still undergo the transforms to AFPDS and back to IPDS. IPDS pass-through will not be valid for all PSF/400 supported printers. Any printer (or attachment) that does not support resident fonts cannot support IPDS pass-through. This is because the resident font references in the data stream must be mapped to host fonts that are downloaded to the printer. All IBM IPDS printers, except for the following, can be supported with IPDS pass-through: 3820, 3825, 3827, 3828, 3829, 3831, 3835, 3900-001 and any printer attached using Print Services Facility for the OS/2® distributed print function.

- 0            \*DEV.D. The value specified for IPDS pass-through (IPDSPASTHR) parameter in the PSF configuration object is used. The PSF configuration object is specified in the printer device description.
- 1            \*NO.No IPDS pass-through is done.
- 2            \*YES. IPDS pass-through is to be done if the spooled file is eligible for IPDS pass-through.

**IPDS transparent data.** Whether the file contains data from System/36 PRPQs.

Valid values are Y (yes) or N (no).

**IPP job name.** The IPP job name associated with the creation of the spooled file. This attribute is in the CCSID identified by the CCSID of IPP attributes field.

**IPP job name natural language.** The natural language in which the IPP job name was encoded. This attribute is in the CCSID specified by the CCSID of IPP attributes field. The possible values returned are defined in RFC 1766.

**IPP job-originating user name.** The name of the user that submitted the IPP job request that created the spooled file. This attribute is in the CCSID identified by the CCSID of IPP attributes field.

**IPP job-originating user name natural language.** The natural language in which the IPP job-originating user name was encoded. This attribute is in the CCSID specified by the CCSID of IPP attributes field. The possible values returned are defined in RFC 1766.

**IPP printer name.** The name of the IPP printer that created the spooled file. This attribute is in the CCSID specified in the CCSID of IPP attributes field.

**Job character ID specified.** Whether the graphic character set and code page of the spooled file is taken from the coded character set identifier (CCSID) of the job.

Valid values are Y (yes) or N (no).

**Job name.** The name of the job that owns the spooled file.

**Job number.** The number of the job that owns the spooled file.

**Job system name.** The name of the system where the job that created this spooled file ran.

**Justification.** The percentage that the output is right-justified. The possible values are 100, 50, or 0.

**Last page printed.** The number of the last printed page in the file if printing ended before the job completed processing.

**Length of each user-defined option entry.** The length, in bytes, of each user-defined option entry.

**Length of IPP attributes.** The length, in bytes, of the IPP specific spooled file attributes.

**Length of page.** The length of a page. Units of measurement are specified in the measurement method field.

**Length of edge-stitch staple offset entry.** The length, in bytes, of each edge-stitch staple offset entry.

**Length of library list entry.** The length, in bytes, of each library list entry.

**Length of saddle-stitch staple offset entry.** The length, in bytes, of each saddle-stitch staple offset entry.

**Length of save and restore attributes.** The length, in bytes, of the save and restore specific spooled file attributes.

**Length of user resource library list entry.** The length, in bytes, of each user resource library list entry.

**Lengths present.** Whether the 8-byte length information is present in the print text data buffers for format SPLF0200.

Valid values are Y (yes) or N (no).

**Library name.** The name of a library.

**Line spacing.** How a file's line data records are spaced when printed. This information is returned only for \*LINE and \*AFPDSLIN printer device type files.

The possible values are:

- \*SINGLE Single-spaced
- \*DOUBLE Double-spaced
- \*TRIPLE Triple-spaced
- \*CTLCHAR The control character field determines line spacing. The control character field can have one of these values:
  - \*NONE No print control characters are passed in the data that is printed.
  - \*FCFC *The first character of every record is an American National Standards printer control character.*

**Lines per inch.** The number (in tenths) of lines per vertical inch defined in the printer file. A value of 40 indicates 4 lines per inch.

**Lines per inch changes.** Whether the lines per inch (lpi) changes within the spooled file.

- For DEVTYPE(\*SCS), a Set Single Line Density command is contained in the spooled file.
- For DEVTYPE(\*IPDS), a Load Page Descriptor is contained in the spooled file, which specifies the line-per-inch value
- For DEVTYPE(\*AFPDS), a presentation text description (PTD) structured field that specifies how the baseline LPI value changes, is contained in the spooled file.

Valid values are Y (yes) or N (no).

**Lines-per-inch (lpi) value not supported.** The lines-per-inch (lpi) value is represented in a 1440th-inch value and is not equivalent to 4, 6, 8, 9, or 12. This field applies only to OfficeVision files. It is N for all other spooled files.

Valid values are Y (yes) or N (no).

**Maximum forms width.** The maximum forms width (in positions) as specified on the printer file.

**Maximum records.** The maximum number of records allowed in the file at the time the file was opened. A value of 0 indicates no maximum.

**Maximum spooled data record size.** The length of the largest record in the file. For LINE, AFPDSLIN, and AFPDS files, this length is the length of the largest record in the file. For all other file types, this length is the same as the spooled file buffer size.

**Measurement method.** The measurement method used for the length of page and width of page fields.

The possible values are:

- \*ROWCOL Uses rows and columns as the units of measure.
- \*UOM Uses the value specified on the unit of measurement parameter (UOM). UOM is either inches (\*INCH) or centimeters (\*CM).

**Multi-byte data.** Whether the file can contain double-byte character set (DBCS) data, Unicode data, or both. The options are \*YES or \*NO.

**Multiple up (pages per side).** The number of logical pages that print on each side of each physical page when this file is printed. The possible values are 1, 2, 3, and 4.

**Natural language of IPP attributes.** The natural language in which the IPP attributes were encoded. This attribute is in the CCSID specified in the CCSID of IPP attributes field. The possible values returned are defined in RFC 1766. The spooled file attribute affected is:

- IPP Printer name

**Net ID where file originated.** The network ID of the system where the spooled file data stream was originally generated for this spooled file.

**Number of buffers.** The number of buffers currently in the file. If the file is still open when the fields are retrieved, the number of buffers is the number of buffers written to the file thus far.

**Number of edge-stitch staple offset entries.** The number of edge-stitch staple offset entries returned in this format. Only complete entries are returned. Each edge-stitch staple offset entry represents one staple in the edge-stitch.

The valid values are:

- 0            \*DEVVD. Use the device default. This value is used if either of the following is specified:
- The number of staples for the edge-stitch number of staples field was specified. The device default for the spacing of each staple will be used.
  - Zero (\*DEVVD) was specified for the edge-stitch number of staples field. The device default for the spacing of each staple is used.
- 1-122        Number of edge-stitch staple offset entries.

**Number of font array entries.** The number of font equivalence array entries used. (The maximum number of font equivalence entries is currently 48.)

**Number of library list entries.** The number of entries in the library list.

**Number of resource library entries.** The number of entries in the resource library array used.

**Number of saddle-stitch staple offset entries.** The number of saddle-stitch staple offset entries returned in this format. Only complete entries are returned. Each saddle-stitch staple offset entry represents one staple in the saddle-stitch.

The valid values are:

- 0            \*DEVVD. Use the device default. This value is used if either of the following is specified:
- The number of staples for the saddle-stitch number of staples field was specified. The device default for the spacing of each staple will be used.
  - Zero (\*DEVVD) was specified for the saddle-stitch number of staples field. The device default for the spacing of each staple is used.
- 1-122        Number of saddle-stitch staple offset entries.

**Number of separators.** The number of file separator pages placed at the beginning of each copy of this file.

**Number of user-defined options returned.** The number of user-defined option entries returned in this format. Only complete entries are returned.

**Number of user resource library list entries.** The number of user resource library list entries returned in this format. Only complete entries are returned.

The valid values are:

- 1 \*DEV.D. The user resource library list in the PSF configuration object is to be used to search for AFPDS resources. The PSF configuration object is specified in the printer device description. If there is no PSF configuraton object, the spooled file's library list will be used to search for AFP resources.
- 2 \*JOB.LIB.L. The spooled file's library list will be used to search for AFP resources.
- 3 \*CURLIB. The current library for the job that created the spool file is used to search for AFP resources. If no library is specified as the current library for the job, then library QGPL is used.
- 0 \*NONE. There is no user resource library list to be used to search for AFP resources.
- 1-4 The number of user resource libraries.

**OfficeVision.** Whether the spooled file is generated by the OfficeVision licensed program.

Valid values are Y (yes) or N (no).

**Offset from edge-stitch reference edge.** Specifies, in either inches or centimeters (specified in the Unit of measure (page 381) (UOM) field), how far from the edge-stitch reference edge that the edge-stitch is to be placed.

The possible values are:

- 1.0 \*DEV.D. The default edge-stitch offset of the printer device is specified.
- 0-57.79 The offset in inches or centimeters, according to the unit of measure value (see Unit of measure (page 381)). The maximum offset is 57.79 centimeters or 22.75 inches.

**Offset to edge-stitch staple offset list.** The offset to the information about each staple offset entry. Each entry contains the staple offset for that staple.

**Offset to IPP attributes.** The offset to the IPP specific spooled file attributes. This value is set to 0 if the spooled file was not create through IPP.

**Offset to library list.** The offset to the list of libraries for the job at the time the spooled file was created. Each entry contains the name of a library.

**Offset to saddle-stitch staple offset list.** The offset to the information about each staple offset entry. Each entry contains the staple offset for that staple.

**Offset to Save and Restore spooled file attributes.** The offset to the save and restore specific spooled file attributes. This value is set to 0 if the spooled file was not saved or restored.

**Offset to user-defined options.** The offset to information about each user-defined option entry. Each entry contains the user-defined option.

**Offset to user resource library list.** The offset to the information about each user resource library list entry. Each entry contains the user resource library. When the offset is set to 0, there are no user resource libraries.

**i5/OS-created AFPDS.** Whether the spooled file was created on a System i™ platform with a printer file that has the device type (DEVTYPE) parameter set to \*AFPDS.

**Output bin.** The output bin that the printer uses for the printed output.

The possible values are:

- 0 \*DEV.D. The default output bin of the printer is used.
- Output bin The number of the output bin specified. The number must be between 1 and 65535.

**Output priority.** The priority of the output file. The priority ranges from 1 (highest) to 9 (lowest).

**Output queue library name.** The name of the library that contains the output queue.

**Output queue name.** The name of the output queue where the file is located.

**Overflow line number.** The last line to be printed before the data being printed overflows to the next page.

**Page count estimated.** Whether the total number of pages is actual or an estimated count.

Valid values are Y (yes) or N (no).

The total page count is estimated in the following situations:

- If the spooled file device type is \*AFPDSLIN.
- If the spooled file device type is \*LINE and a page definition was used.
- If the spooled file device type is \*AFPDS and the spooled file was not created on a System i platform.
- If the spooled file device type is \*USERASCII.

**Page definition library name.** The name of the library in which the page definition resides. This information is returned only for \*LINE or \*AFPDSLIN printer device type files.

**Page definition name.** The name of the page definition to use for the file. This information is returned only for \*LINE or \*AFPDSLIN printer device type files.

**Page groups.** Whether the spooled file contains page groups as indicated by the DDS keywords STRPAGGRP and ENDPAGGRP. The possible values are Y (yes) or N (no).

**Page length.** The page length (in lines per page) used by the spooled file. The valid range is row 1 through 255. The value used should not exceed the actual length of the page used.

**Page level index tags.** Whether the spooled file contains page level index tags as indicated by the DDS keyword DOCIDXTAG with a tag level of PAGE. The possible values are Y (yes) or N (no).

**Page or record being written.** The page number or record number currently being written. The page number may be lower or higher than the page number actually being printed because of buffering done by the system. The page number shown may be zero if:

- The printer file is routed to a diskette unit.
- The writer is currently printing job or file separators for the file.

**Page position.** Whether page positioning errors are reported.

Valid values are Y (yes) or N (no).

**Page rotate.** Whether the spooled file changes the page rotation to be used within the file.

- For DEVTYPE(\*SCS), a Set Text Orientation command is contained in the spooled file.
- For DEVTYPE(\*IPDS), a Load Page Descriptor command is contained in the spooled file. See the *IPDS Reference* manual, S544-3417, for detailed information about rotating text in a spooled file. This manual is available from the IBM Publications Center .
- For DEVTYPE(\*AFPDS), a presentation text descriptor (PTD) structured field that specifies the page rotation is contained in the spooled file.

Valid values are Y (yes) or N (no).

**Page rotation.** The degree of rotation of the text on the page, with respect to the way the form is loaded into the printer.

The possible values are:

- 1 \*AUTO: Computer output reduction is done automatically if the output is too large to fit on the form, regardless of the print quality.
- 2 \*DEVN: The operating system sends a device default rotation value to the printer. Page rotation is dependent on the printer's specifications. See your printer or printer emulation documentation to determine how page rotation is affected.
- 3 \*COR: Output created for a form 13.2 inches wide by 11.0 inches long is adjusted to print on a form 11.0 inches wide by 8.5 inches long.
- 0 No rotation is done. Printing starts at the edge loaded into the printer first, and is parallel to that edge.
- 90 Text is rotated 90 degrees clockwise from the 0-degree writing position.
- 180 Text is rotated 180 degrees clockwise from the 0-degree writing position.
- 270 Text is rotated 270 degrees clockwise from the 0-degree writing position.

**Page width.** The page width (in characters per printed line) used by the spooled file. The valid range is column 1 through 378, although some printers have a page width less than 378. The value should not exceed the actual width of the page used.

**Point size.** The point size in which this file's characters (defined by the printer font field) should be printed.

**Print fidelity.** The kind of error handling that is performed when printing.

The possible values are:

- \*ABSOLUTE The file is printed only if it can be printed exactly as specified in the data stream.
- \*CONTENT The printing overrides errors in the data stream and continues printing with the printers best quality based on the content fidelity.

**Print on both sides (duplex).** How the information prints.

The possible values are:

- \*FORMDF The file uses a user-specified form definition. This value is used only for \*LINE, \*AFPDS, and \*AFPDSLIN printer device type files.
- \*NO The printing on the page is on one side only.
- \*YES The printing is on both sides of the page with the top of each page the same for both sides.
- \*TUMBLE The printing is on both sides with the top of one printed page at the opposite end from the top of the other printed page.

**Print quality.** The print quality that is used when printing this output.

The possible values are:

- \*STD Standard
- \*DEVN Device default
- \*DRAFT Draft
- \*NLQ Near-letter quality
- \*FASTDRAFT Prints at a faster speed than \*DRAFT

**Print text.** The text that is printed at the bottom of each page of printed output and on separator pages.

**Printer device type.** The type of data stream used to represent the file.

The possible values are:

<i>*AFPDS</i>	Advanced Function Presentation data stream
<i>*AFPDSL</i>	AFPDS data mixed with 1403 line data
<i>*IPDS</i>	Intelligent printer data stream
<i>*LINE</i>	1403 line data
<i>*SCS</i>	Systems Network Architecture (SNA) character stream
<i>*USERASCII</i>	ASCII data

**Printer font.** The printer font used. If *\*DEV* is shown, the printer uses the font defined in the printer device description. If *\*CPI* is shown, the file is printed with a font that has the pitch specified by the *CPI* (character per inch) field.

**Profile name of spooled file owner.** The name of the user profile that owns the spooled file.

**Program that opened file library name.** The name of the library that contains the program that opened the file.

**Program that opened file name.** The name of the program that opened the spooled file.

**Record format.** The format of the records.

The possible values are:

<i>*FIXED</i>	All records in the file are the same size; that is, the original length field of the print data is the same for all records in the file.
<i>*VARIABLE</i>	Not all records in the file are the same size; that is, the original length field of the print data is not the same for all records in the file.

**Record format name present in data stream.** A record format name exists in columns 1-10 of a line data record or columns 2-11 if spooled file contains print control characters. This is not to be confused with DDS record formats. Valid values are Y (yes) or N (no).

<i>N</i>	A record format name is not present in the line data record.
<i>Y</i>	A record format name is present in the line data record.

**Record length.** The length of the file's records. If the field shows -1 (the special value for *\*RCDFMT*), this is an externally defined file, and the length is included in the file definition. The length includes the extra length of 1 for carriage controls.

**Reduce output.** The manner in which multiple logical pages print on each side of a physical page.

The possible values are:

<i>*TEXT</i>	A smaller font and lpi are used to print multiple logical pages on each side of a physical page. The smaller font and lpi preserves the contents of each logical page. The possible values for the multiple up parameter are 1, 2, and 4.
<i>*NONE</i>	The printer prints multiple logical pages on each side of each of a physical page. There are no changes to the fonts and lpi values used in each logical page. The possible values for the multiple up parameter are 1, 2, 3, and 4.

**Replace unprintable characters.** Whether characters that cannot be printed are to be replaced with another character. The options are Y (yes) or N (no).

**Replacement character.** The character that replaces any unprintable characters. This field has a value if the replace unprintable characters field specifies Y.

**Reserved.** The field is reserved.

**Resource library array.** The library names in the library list to use when the spooled file is printed. The variable returns a maximum of 63 ten-character library names. If more than 63 resource libraries are used, a 1 is returned in the 631st character of the variable; otherwise, the 631st character is blank. Use of this field should be for compatibility reasons only. All new development should use the offset to library list, number of library list entries, and length of library list entry fields to access the spooled file's library list located at the end of format SPLA0200.

**Restart printing.** The number of the page where printing restarts. When you specify a value while the file is printing, the writer stops printing and restarts on the specified page. If the file is not currently printing, this change takes effect after the first copy prints.

The possible values are:

- |                     |  |
|---------------------|--|
| -1                  | *STRPAGE: The starting page specified in the PAGERANGE parameter is the page on which to restart printing. |
| -2                  | *ENDPAGE: Only the last page prints.   |
| -3                  | *NEXT: The next page of the file to print is the page where printing is restarted.                         |
| <i>restart page</i> | The page number you specify is where printing restarts.  |

**Saddle-stitch number of staples.** The number of staples in the saddle-stitch.

The valid values are:

- |       |   |
|-------|---|
| 0     | *DEVD. Use the device default. This value is used if either of the following is specified: <ul style="list-style-type: none"><li>• The offset of each staple is specified in the saddle-stitch staple offset list. The printer automatically determines the number of staples to be used.</li><li>• Zero (*DEVD) was specified for the number of saddle-stitch staple offset entries field. The printer uses its default value for both the number of staples to be used and the offset of each staple.</li></ul> |
| 1-122 | Number of staples to be used in the saddle-stitch.  |

**Saddle-stitch reference edge.** The edge of the paper to be used as a reference in determining where the saddle stitching is to be located.

Valid values are:

- |   |  |
|---|--|
| 0 | *NONE. No saddle stitching is specified.                                   |
| 1 | *DEVD. The default reference edge used by the printer device is specified. |
| 2 | *TOP. The top edge of the paper is specified.                              |
| 3 | *LEFT. The left-hand edge of the paper is specified.                       |

**Saddle-stitch staple offset.** For the spooled file, the placement of staples along the finishing margin in either inches or centimeters (specified in the Unit of measure (page 381) (UOM) field). The finishing margin can be thought of as an imaginary line parallel to the edge of the paper where the staples will be placed. The position of the finishing margin is the center of the paper relative to the physical edge.

If you want the staples placed parallel to the left edge of the paper, specify the offset of the first staple by measuring from the point where the finishing margin intersects the bottom edge of paper to where you want the center of the staple to be. For subsequent staples, specify the offset by measuring from the same point (not from the previous staple).

If you want the staples placed parallel to the top edge of the paper, specify the offset of the first staple by measuring from the point where the finishing margin intersects the left edge of paper to where you want the center of the staple to be. For subsequent staples, specify the offset by measuring from the same point (not from the previous staple).

The possible values are:

0-57.79      The offset in inches or centimeters, according to the unit of measure value (see Unit of measure (page 381)). The maximum offset is 57.79 centimeters or 22.75 inches.

**Save command.** The command used to save the spooled file. The field is blank if the spooled file was not saved.

**Save device.** The type of device to which the spooled file was last saved. The field is \*SAVF if the last save operation was to a save file. The field is \*TAP if the last save operation was to tape. The field is \*OPT if the last save operation was to optical. The field is blank if the spooled file was not saved.

**Save file after written.** Whether this file is to be saved after it is written.

The possible values are:

\*NO            The file is deleted after it has been written.  
\*YES          The file is set to save status after it has been written.

**Save file library name.** The name of the library that contains the save file if the spooled file was saved to a save file. The field is blank if the spooled file was not saved to a save file.

**Save file name.** The name of the save file if the spooled file was saved to a save file. The field is blank if the spooled file was not saved to a save file.

**Save label.** The file label used when the spooled file was saved. This field is blank if the spooled file was not saved to tape or optical. This field corresponds to the value specified for the LABEL or OPTFILE parameter on the command used to save the spooled file.

**Save sequence number.** The tape sequence number assigned when the spooled file was saved on tape. If the spooled file was not saved to tape, the field contains zeros.

**Save volume ID.** The tape or optical volumes that are used for saving the spooled file. This field returns a maximum of 10 six-character volumes. The volume IDs begin in character positions 1, 8, 15, 22, 29, 36, 43, 50, 57, and 64. Each volume ID entry is separated by a single character. If the spooled file was saved in parallel format, the separator character contains a 2 before the first volume in the second media file, a 3 before the third media file, and so on, up to a 0 before the tenth media file. Otherwise, the separator characters are blank. If more than 10 volumes are used and the spooled file was saved in serial format, 1 is returned in the 71st character of this field. If the spooled file was saved in parallel format, a 2 is returned in the 71st character of this field. Otherwise, the 71st character is blank. The field is blank if the spooled file was last saved to a save file or if it was never saved.

**SCS data.** Whether the spooled file is created with SCS already in the input data.

Valid values are Y (yes) or N (no).

**Set exception.** Whether the file has the SCS Set Exception Action (SCS) command in the data stream.

Valid values are Y (yes) or N (no).

**Set Line Density command.** Whether the lines per inch (lpi) changes within the spooled file or is specified with the Set Line Density SCS command.

For DEVTYPE(\*SCS), a Set Line Density command is contained in the spooled file.

Valid values are Y (yes) or N (no).

**Source drawer.** The drawer to be used when the automatically cut form feed option is selected.

The possible values are:

- 1-255 The drawer number.
- 1 \*E1, the envelope drawer.
- 2 \*FORMDF, indicating that the file uses a user-specified form definition. This value is used only for \*LINE, \*AFPDS, \*IPDS, \*SCS, or \*AFPDSLIN printer device type files.

**Spooled file buffer size.** The maximum size of a spooled file buffer. Valid lengths are 512 and 4079.

**Spooled file creation authentication method.** The authentication method used when the spooled file was created. The possible values are:

- 0 \*NONE No authentication method was used.
- 1 \*REQUESTER The requesting user was used to authenticate the request.
- 2 \*BASIC The client requested the user to sign on with a userid and password to authenticate the request.
- 3 \*CERTIFICATE The request was authenticated through the use of certificates.

**Spooled file creation security method.** The security method used when the spooled file was created. The possible values are:

- 0 \*NONE No security method was used.
- 1 \*SSL3 The SSL3 security method was used.

**Spooled file level.** The level of the spooled file in Version, Release, and Modification level format (VxRxMx).

**Spooled file name.** The name of the spooled file whose information is retrieved.

**Spooled file number.** The spooled file number of the specified file.

**Spooled file size.** The spooled file size. Multiply this field by the spooled file size multiplier field to get the size of the spooled file in number of bytes. The spooled file size is the data stream size plus the "overhead" storage used to store the spooled file's data stream.

**Spooled file restored date and time.** ➤ The date and time, in local job time, that the spooled file was last restored. ⏪ This field is blank if the spooled file was not restored. The date and time is in the CYYMMDDHHMMSS format as follows:

- C Century, where 0 indicates years 19xx and 1 indicates years 20xx.
- YY Year
- MM Month

<i>DD</i>	Day
<i>HH</i>	Hour
<i>MM</i>	Minute
<i>SS</i>	Second

**Spooled file saved date and time.** » The date and time, in local job time, that the spooled file was last saved. « This field is blank if the spooled file was not saved. The date and time is in the CYYMMDDHHMMSS format as follows:

<i>C</i>	Century, where 0 indicates years 19xx and 1 indicates years 20xx.
<i>YY</i>	Year
<i>MM</i>	Month
<i>DD</i>	Day
<i>HH</i>	Hour
<i>MM</i>	Minute
<i>SS</i>	Second

**Spooled file size multiplier.** The number to multiply the spooled file size field by to get the spooled file size in number of bytes.

**Starting page.** The page at which printing is to start for the file.

The possible values are:

<i>0</i>	Printing starts on page 1.
<i>-1</i>	Use the ending page value.
<i>1-2147483647</i>	Number of page to start printing from.

**Status.** The status of the file is one of the following values:

<i>*CLOSED</i>	The program completely processed the file, but SCHEDULE(*JOBEND) was specified, and the job that produced the file has not yet finished.
<i>*DEFERRED</i>	The spooled file has been deferred from printing.
<i>*SENDING</i>	The spooled file is being sent or has been sent to a remote system.
<i>*HELD</i>	The file is held.
<i>*MESSAGE</i>	The file has a message requiring a reply or an action.
<i>*OPEN</i>	The file was not completely processed and is not ready for a writer.
<i>*PENDING</i>	The file is pending to be printed.
<i>*PRINTER</i>	The complete file was sent to the printer, but a print complete status was not sent back.
<i>*READY</i>	The file is available to be written.
<i>*SAVED</i>	The file was written and is saved. This file remains saved until it is released.
<i>*WRITING</i>	The writer is writing the file on a diskette or a printer device.

**Subscript.** Whether the spooled file contains subscript.

- For DEVTYPE(\*SCS), a Subscript command is contained in the spooled file.
- For DEVTYPE(\*IPDS and \*AFPDS), a Temporary Baseline Move text control is contained in the spooled file.

Valid values are Y (yes) or N (no).

**Superscript.** Whether the spooled file contains superscript.

- For DEVTYPE(\*SCS), a Superscript command is contained in the spooled file.

- For DEVTYPE(\*IPDS and \*AFPDS), a Temporary Baseline Move text control is contained in the spooled file.

Valid values are Y (yes) or N (no).

**System where file originated.** The name of the system where the spooled file data stream was originally generated for this spooled file. If the creating system was a System i product running OS/400® version 2 or earlier, the address of the user that sent the file is displayed.

When the name of the system where this spooled file was created cannot be determined, the receiving system name is used. For example, a file sent using SNDNETSPLF DTAFMT(\*RCDDATA).

**System/36 procedure name.** The name of the procedure running when the spooled file was created.

**System/36 spooled file identifier.** The 6-character identifier assigned to the spooled file.

The possible formats are:

SPxxxx	SP is constant, and xxxx is a number from 0000 to 9999.
cxxxxx	c is a letter from 'A' to 'Z' and xxxxx is a number from 00000 to 99999.
blanks	The spooled file resides on an independent disk pool and does not have a System/36 spooled file identifier.

**System/38 text utility flags.** The flag for advanced functions.

When you create a file that will contain data that has never been spooled before, set the field to hexadecimal zeros (hex 00).

**S36 CONTINUE-YES.** The file was created with the System/36 environment OCL statement “// PRINTER CONTINUE-YES” in effect. See the System/36 Environment Reference manual, SC41-4731, for more information.

This field should only be changed by i5/OS.

A spooled file created in this manner can change its overflow line number and maximum print position from within its print data stream. When it is printed, the writer needs to know that the page size may have changed and it needs to reset the printer before printing the next spooled file. Otherwise, the next spooled file printed may cause a printer error or data loss. The options are Y (yes) or N (no).

**Time file opened (created).** ➤ The time, in local job time, that the file was opened in the HHMMSS format where: ⏪

HH	Hour
MM	Minutes
SS	Seconds

**Time writer began processing spooled file.** ➤ The time, in local job time, that a spooling writer started processing the spooled file. ⏪ This field is set to blanks if the file has not been processed by a writer. The time is in the HHMMSS format where:

HH	Hour
MM	Minutes
SS	Seconds

**Time writer completed processing spooled file.** » The time, in local job time, that a spooling writer completed processing the spooled file « or the time the processing of the spooled file was aborted. This field is set to blanks if the file has not been processed by a writer or the writer has not completed processing the file. The time is in the HHMMSS format where:

<i>HH</i>	Hour
<i>MM</i>	Minutes
<i>SS</i>	Seconds

**Total copies.** The total number of copies to be produced for this printer file.

**Total pages.** The number of pages this printer file contains.

**Total records.** If the spooled file data stream is \*AFPDS, \*AFPDSLIN, or \*LINE or the spooled file device type is diskette, this field contains the total number of records in the printer file or the diskette file. The field is blank if the file is open.

**Transparency.** Whether the SCS Transparency command is used in the spooled file. It is set to N for all other device type files, such as IPDS.

For DEVTYPE(\*SCS), a TRANSPARENT SCS command is contained in the spooled file.

Valid values are Y (yes) or N (no).

**TRC for 1403.** Whether the file contains 1403 line data with table-reference characters (TRC). This field is only used for device types \*LINE and \*AFPDSLIN.

Valid values are Y (yes) or N (no).

**Unit of measure.** The unit of measure to use for specifying distances. The unit of measure is used with certain parameters of the printer file. Page definitions and form definitions are examples of these parameters.

The possible values are:

*CM	Centimeters
*INCH	Inches

**Unrecognizable data.** Whether the file contains SCS commands that are not valid because of one of the following:

- Command is not recognized
- Command data is not valid
- Command is recognized, but not supported

Valid values are Y (yes) or N (no).

**User name.** The name of the user that owns the spooled file.

**User object type.** The type of the user-defined object.

The following values are supported:

*blanks*                    The user-defined object is \*NONE.

*User object type* The user object type can be:

- \*DTAARA  
Data area
- \*DTAQ Data queue
- \*FILE File
- \*PSFCFG  
PSF configuration object
- \*USRIDX  
User index
- \*USRQ User queue
- \*USRSPC  
User space

**User-defined data.** Data defined by the user to be used by user applications or user-specified programs that process spooled files.

The following values are supported:

\*NONE No user-defined data is specified.  
*User-defined data* Data defined by the user.

**User-defined file.** Whether the spooled file was created by using an API.

The possible values are:

\*YES The spooled file was created using the QSPCRTSP API.  
 \*NO The spooled file was created by normal system functions.

**User-defined object library name.** The name of the library that contains the user-defined object.

**User-defined object name.** The name of the user-defined object that is used by user applications or user-specified programs that process spooled files.

The following values are supported:

\*NONE No user-defined object is specified.  
*User-defined object name* The name of the object defined by the user.

**User-defined option.** The option defined by the user to be used by user applications or user-specified programs that process spooled files.

The possible values are:

\*NONE No user-defined options are specified.  
*User-defined option* The value of a user-defined option.

**User-defined text.** User-defined text taken from the interactive user profile when the spooled file was created.

**User-generated data stream.** Whether the data stream has been validated by a system program on the System i platform when the file was spooled, for example, the OfficeVision program.

Valid values are Y (yes) or N (no).

**User-specified data.** The 10 characters of user-specified data that describe the file.

**User who owned file.** The name of the user profile that owns this file. If the spooled file was created on a System i platform running OS/400 version 2 or earlier, the user id that sent the file is displayed.

When the name of the user profile that owned this file cannot be determined, the name of the receiving user profile is used. For example, a file sent using SNDNETSPLF DTAFMT(\*RCDDATA).

**User resource library.** The name of the user resource library.

**Volumes (array).** The diskette volumes used for saving the object. The variable returns a maximum of 10 six-character volumes. The volume IDs begin in character positions 1, 8, 15, 22, 29, 36, 43, 50, 57, and 64. If more than 10 volumes are used, a 1 is returned in the 71st character of the variable; otherwise, the 71st character is blank. The variable is returned blank if the object was saved to a save file, or if it was never saved.

**Width of drawer 1.** The width in inches of the paper in drawer 1. This field is for internal use and should be set to 0 by an application building this format as opposed to retrieving the fields.

**Width of drawer 2.** The width in inches of the paper in drawer 2. This field is for internal use and should be set to 0 by an application building this format as opposed to retrieving the fields.

**Width of page.** The width of a page. Units of measurement are specified in the measurement method field.

**3812 SCS.** Whether the spooled file contains fonts supported only on the 3812 printer.

For DEVTYPE(\*SCS), a Set Coded Font Local command with a global ID of greater than 255 is contained in the spooled file.

Valid values are Y (yes) or N (no).

**3812 SCS commands.** Whether the file contains commands that are not valid on the 3812 printer.

- An incorrect Set Character Set Global ID is found by validity checking the code page with what is valid for the 3812 printer.
- An incorrect Set Character Set Local ID is found by validity checking the code page with what is valid for the 3812 printer.
- A Set Presentation Color command

Valid values are Y (yes) or N (no).

**5A present.** Whether the 5A hexadecimal constant is present in all AFPDS data for the file (in format SPLF0200).

Valid values are Y (yes) or N (no).

**5219 commands.** Whether the file contains commands that are not valid on a 5219 printer. The data stream contains one or more of the following SCS commands which, either are not valid on a 5219 or contain parameters that are not valid.

- An incorrect Set Character Set Global ID is found by validity checking the code page with what is valid for the 5219 printer.
- Begin and End Emphasis SCS commands.
- A Set Presentation Page Size command contains a page size value that is not supported.
- A Page Presentation Media command contains a drawer value other than drawer 1 or drawer 2.
- A Set Single Line Distance command contains a distance value that is not supported.
- A Set Single Line Spacing command contains a distance value that is not supported.
- A Justify Text Field Format command contains a value other than 0, 50, or 100 percent.
- A Set Justify Mode Format command contains a value other than 0, 50, or 100 percent.
- A Set Presentation Color command.
- A Begin Overstrike command, which contains an optional CHRID value of other than 0.

Valid values are Y (yes) or N (no).

## Error Messages

Message ID	Error Message Text
CPF24B4 E	Severe error while addressing parameter list.
CPF3CF1 E	Error code parameter not valid.
CPF3C19 E	Error occurred with receiver variable specified.
CPF3C20 E	Error found by program &1.
CPD3C21 D	Format name &1 is not valid.
CPD3C24 D	Length of the receiver variable is not valid.
CPD3C40 D	Spooled file number &1 is not valid.
CPD3C42 D	User name or job number is not blank.
CPD3C43 D	Internal job identifier is not valid.
CPD3C44 D	Internal spooled file identifier is not valid.
CPD3C58 D	Job name specified is not valid.
CPD33C9 D	Spooled file name parameter cannot be blank.
CPD335B D	Spooled file create date is not blank.
CPD3360 D	Job system name is not valid.
CPD3361 D	Spooled file create date is not valid.
CPD3362 D	Spooled file create time is not valid.
CPD3363 D	Spooled file create time is not blank.
CPD3364 D	Job system name is not blank.
CPF3C21 E	Format name &1 is not valid.
CPF3C24 E	Length of the receiver variable is not valid.
CPF3C33 E	Spooled file number &1 is not valid.
CPF3C36 E	Number of parameters, &1, entered for this API was not valid.
CPF3C40 E	Spooled file &4 not found.
CPF3C41 E	More than one spooled file with same name.
CPF3C42 E	User name or job number is not blank.
CPF3C43 E	Internal job identifier is not valid.
CPF3C44 E	Internal spooled file identifier is not valid.
CPF3C58 E	Job name specified is not valid.
CPF3C90 E	Literal value cannot be changed.
CPF33C9 E	Spooled file name parameter cannot be blank.
CPF3309 E	No files named &1 are active.
CPF3330 E	Necessary resource not available.
CPF333B E	Job system name is not valid.
CPF333C E	Spooled file create date is not valid.
CPF333D E	Spooled file create time is not valid.
CPF333E E	Spooled file create time is not blank.

Message ID	Error Message Text
CPF333F E	Job system name is not blank.
CPF3342 E	Job &5/&4/&3 not found.
CPF3343 E	Duplicate job names found.
CPF3344 E	File &1 number &2 no longer in the system.
CPF335B E	Spooled file create date is not blank.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API Introduced: V1R3

Top | "Print APIs," on page 1 | APIs by category

---

## PrintManager APIs

PrintManager API Reference  provides reference and use information for the PrintManager™ APIs. It is intended for designers and programmers who want to create print descriptors to use with applications created with the PrintManager product implementation of the PrintManager interface. With the APIs, you can create common print descriptors that contain common information for printing across the supported environments.

Systems Application Architecture: Common Programming Interface PrintManager Reference  defines the SAA® PrintManager interface. It is intended for programmers who write applications that adhere to this definition.

"Print APIs," on page 1 | APIs by category

---

## Exit Programs

These are the Exit Programs for this category.

---

### Exit Program for a Customized Separator Page

Required Parameter Group:

1	Separator data	Output	Char(*)
2	Separator data length	Input	Binary(4)
3	Separator information	Input	Char(*)
4	Separator information length	Input	Binary(4)

QSYSINC Member Name: ESPBLSEP

An exit program can be specified on the separator program parameter of the printer device description. If a separator exit program is specified on the printer device description, the system printer writer calls this program to create the separator page. This separator page is used instead of the system separator page.

## Authorities and Locks

None.

## Required Parameter Group

### Separator data

OUTPUT; CHAR(\*)

The separator data consists of two parts:

- Options that control how the separator data is to be printed
- User-generated separator data

See the “Separator Data Format” for more information.

### Separator data length

INPUT; BINARY(4)

The length of the space used to pass separator data from the exit program.

### Separator information

INPUT; CHAR(\*)

Information that may be used to generate the separator data.

See the “Separator Information Format” on page 390 for more information.

### Separator information length

INPUT; BINARY(4)

The length of the space used to pass separator information to the exit program.

## Separator Data Format

The following table shows the information passed from the exit program to create the customized separator page for the printer writer. The printer writer takes the information and data, and prints the separator page.

See “Field Descriptions” on page 387 for more information.

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	Transform option
10	A	CHAR(2)	Reserved
12	C	BINARY(4)	Page rotation
16	10	BINARY(4)	Page length
20	14	BINARY(4)	Page width
24	18	BINARY(4)	Lines per inch
28	1C	BINARY(4)	Characters per inch
32	20	BINARY(4)	DBCS Characters per inch
36	24	CHAR(10)	DBCS characters rotation
46	2E	CHAR(10)	Page size measurement method
56	38	CHAR(10)	Print quality
66	42	CHAR(10)	Overlay name
76	4C	CHAR(10)	Overlay library name
86	56	CHAR(98)	Reserved
184	B8	BINARY(4)	Length of user-generated data
188	BC	BINARY(4)	Record length of user data
192	C0	CHAR(*)	User-generated separator data

## Field Descriptions

Each field is only valid when the transform option field is specified as \*FCFC. Otherwise, the field is ignored.

**Characters per inch.** The number of characters (in tenths) to be printed per inch.

The possible values are:

50	5 characters per inch
100	10 characters per inch
120	12 characters per inch
133	13.3 characters per inch
150	15 characters per inch
167	16.7 characters per inch
180	18 characters per inch
200	20 characters per inch

Any other value, or a value not supported by the particular printer, will result in 10 characters per inch.

**DBCS character rotation.** Whether double-byte character set (DBCS) characters are to be rotated 90 degrees counterclockwise before printing. The possible values are \*YES and \*NO. The default value is \*NO.

**DBCS characters per inch.** The number of double-byte characters to be printed per inch.

The possible values are:

-1	One-half of the characters-per-inch value
-2	20 double-byte characters per 3 inches
5	5 characters per inch
6	6 characters per inch
10	10 characters per inch

If any other value is used, the characters are printed at one-half of the characters-per-inch (\*CPI) value.

**Length of user-generated data.** The number of bytes of data (contained in the user-generated separator data field) passed from the user exit program.

The length of the user-generated separator data cannot exceed 8096 bytes for printers configured AFP(\*NO). A value greater than 8096 causes the system separator page to be printed.

**Lines per inch.** The number of lines (in tenths) to print per vertical inch.

The possible values are:

30	3 lines per inch
40	4 lines per inch
60	6 lines per inch
75	7.5 lines per inch
80	8 lines per inch
90	9 lines per inch

Any other value, or a value not supported by the particular printer, will result in 6 lines per inch.

This parameter is output from the exit program and is only valid when the Transform Option is specified as \*FCFC. Otherwise this parameter is ignored.

**Page length.** The length of the separator page. The page length can be specified in inches, centimeters, or rows depending on the page size measurement method field.

A zero indicates the page length of the spooled file should be used. A nonzero page length value is valid only when the page width field is also set to a nonzero value. Otherwise, the page length is ignored.

Inches and centimeters are specified in 1/100ths. The value 1100 indicates a page length of 11 inches when the page size measurement method field is \*INCH.

**Page size measurement method.** The measurement method used for the page length and page width fields.

The possible values are:

<i>*INCH</i>	The page size is specified in 1/100ths of an inch.
<i>*CM</i>	The page size is specified in 1/100ths of a centimeter.
<i>*ROWCOL</i>	The page size is specified in row and column format.

**Page width.** The width of the separator page. The page width can be specified in inches, centimeters, or rows depending on the page size measurement method field.

A 0 indicates the page width of the spooled file should be used. A nonzero page width value is valid only when the page length field is also set to a nonzero value. Otherwise, the page width is ignored.

Inches and centimeters are specified in 1/100ths. The value 850 indicates a page width of 8.5 inches when the page size measurement method field is \*INCH.

**Overlay library name.** The name of the library containing the separator overlay.

The possible values are:

<i>*LIBL</i>	The library list of the job that created the file locates the overlay.
<i>library name</i>	This library is searched for the overlay.

If no value is returned from the exit program, \*LIBL is assumed.

**Overlay name.** The name of the separator overlay (the material that prints on the front side of each separator page).

The possible values are:

<i>*NONE</i>	The separator does not use an overlay.
<i>Overlay Name</i>	The name of the overlay to print with the separator.

If no value is returned from the exit program, \*NONE is assumed. Otherwise this parameter is ignored.

**Page rotation.** The degree of rotation of the text on the page, with respect to the way the form is loaded into the printer.

The possible values are:

0	No rotation is done. Printing starts at the edge loaded into the printer first, and is parallel to that edge.
90	Text is rotated 90-degrees clockwise from the 0-degree writing position.
180	Text is rotated 180-degrees clockwise from the 0-degree writing position.
270	Text is rotated 270-degrees clockwise from the 0-degree writing position.

If no valid value is returned from the exit program, 0 is assumed.

**Print quality.** The print quality that is used when printing the separator.

The possible values are:

*STD	Standard
*DRAFT	Draft
*DEV	Print quality determined by printer hardware setting
*FASTDRAFT	Fast draft
*NLQ	Near-letter quality

If no valid value is returned from the exit program, \*DEV is assumed. Otherwise this parameter is ignored.

**Record length of user data.** The length of the \*FCFC format records in the user-generated separator data field.

**Reserved.** An ignored field.

**Transform option.** Whether the user-generated data is to be transformed into commands recognized by the printer.

The possible values are:

\*FCFC The first character of every record is an American National Standards printer control character.

The supported control characters and their functions follow:

- + Prints record on current line.
- blank* Moves down 1 line and prints record.
- 0 Moves down 2 lines and prints record.
- Moves down 3 lines and prints record.

All other control character values are treated as blanks and result in the record printing on the next line. Channel values are not supported.

\*NONE User-supplied data is in its final form. No transform operation is done, and the data must contain the appropriate commands corresponding to the printer data stream type.

For the printer data stream type field of \*SCS and \*IPDS, data stream commands controlling the drawer, rotations, and fonts may be sent to the printer before the user-generated data. This puts the printer in the same state as it is when system default separators are printed. The commands controlling the drawer are based on the SEPDRAWER parameter of the Start Printer Writer (STRPRTWTR) command. The remaining controls are set to defaults or have values set from the spooled file attributes of the spooled file that prints next.

For a printer data stream type of \*AFPDS, the initial commands are not sent to the printer and the user must include all commands needed to define an AFPDS document.

**User-generated separator data.** The data that prints on the separator page. The format of this text is dependent on the transform option and the printer data stream type fields. When \*FCFC has been specified for the transform option, the data must be in the \*FCFC format. When \*NONE is specified for the transform option, the data (text and controls) must be in the format specified by the printer data stream type field.

## Separator Information Format

The following table shows the information passed from the printer writer to the exit program. This information is used to create a customized separator page. For more details about the fields in the following table, see “Field Descriptions.”

Offset		Type	Field
Dec	Hex		
0	0	CHAR(16)	Internal job identifier
16	10	CHAR(16)	Internal spooled file identifier
32	20	CHAR(10)	Job name
42	2A	CHAR(10)	User name
52	34	CHAR(6)	Job number
58	3A	CHAR(10)	Spooled file name
68	44	BINARY(4)	Spooled file number
72	48	CHAR(10)	Printer device name
82	52	CHAR(10)	Printer data stream type
92	5C	CHAR(10)	Type of separator
102	66	CHAR(8)	Job system name
110	6E	CHAR(7)	Spooled file create date
117	75	CHAR(1)	Reserved
118	76	CHAR(6)	Spooled file create time
124	7C	CHAR(50)	Reserved

## Field Descriptions

**Internal job identifier.** The internal identifier of the job that owns the spooled file that prints next. Only the i5/OS® APIs use this identifier. The identifier is not valid following an initial program load (IPL). If you attempt to use it after an IPL, an exception occurs.

**Internal spooled file identifier.** The internal identifier of the spooled file that prints next. This value is used as input to other programs to improve the performance of locating the spooled file on the system. Only the i5/OS APIs use this identifier. The identifier is not valid following an IPL. If you attempt to use it after an IPL, an exception occurs.

**Job name.** The name of the job that created the spooled file that prints next.

**Job number.** The number of the job that created the spooled file that prints next.

**Job system name.** The name of the system where the job that created the spooled file ran.

If the separator data length is less than 124 bytes, the exit program should assume \*ONLY.

**Printer device name.** The name of the printer device description for the writer that was started.

**Printer data stream type.** The type of data stream recognized by the printer writer. When the transform option is specified as \*NONE, this field must be used to determine the format of the user-generated separator data field.

The possible values are:

\*AFPDS           Advanced Function Presentation™ data stream  
\*IPDS            Intelligent Printer Data Stream™  
\*SCS             Systems Network Architecture (SNA) character stream

**Reserved.** An ignored field.

**Spoiled file create date.** The date the spoiled file was create on the system in the format CYYMMDD. See field Date file opened in API QUSRSPAL under field descriptions for more information on the date format.

If the separator data length is less than 124 bytes, the exit program should assume \*ONLY.

**Spoiled file create time.** The time the spoiled file was created on the system in the format HHMMSS. See field Time file opened in API QUSRSPAL under field descriptions for more information on the time format.

If the separator data length is less than 124 bytes, the exit program should assume \*ONLY.

**Spoiled file name.** The name of the spoiled file that prints next.

**Spoiled file number.** The spoiled file number of the spoiled file that prints next.

**Type of separator.** The type of separator you want printed.

The possible values are:

\*FILE            Separator printed before each copy of a spoiled file.  
\*JOB             Separator printed between spoiled files created by different jobs.

**User name.** The name of the user that created the spoiled file that prints next.

Exit program introduced: V2R3

Top | "Print APIs," on page 1 | APIs by category

---

## Image Print Transform Exit Program

Required Parameter Group:

1	Transform values	Input	Char(*)
2	Process values	I/O	Char(*)
3	Input data stream	Input	Char(*)
4	Output data stream	Output	Char(*)

Exit Point Name: QIBM\_QIMG\_TRANSFORMS

Exit Point Format Name: "XFRM0100 Format" on page 393

QSYSINC Member Name: eimgph

The Image Print Transform (IPT) exit point can be used to transform user ASCII data streams for the following reasons:

- Input data stream formats not supported directly by IPT.
- Output data stream formats not supported directly by IPT.

An exit program that takes the requested data stream format as input and produces the requested output data stream format must be registered to this exit point by using the Add Exit Program (QUSADDEP, QusAddExitProgram) API or the ADDEXITPGM command.

Image Print Transform (IPT) tries to determine the input data stream format and the output data stream format. Once the two data stream formats have been determined, the Retrieve Exit Information (QUSRTVEI, QusRetrieveExitInformation) API is called to determine if any exit programs have been registered that support the specified input and output data stream format combination. The following input and output data stream combinations are used in the exit program data field of the selection criteria for the API:

Input	Output
BMP	AFPDS
BMP	PCL
BMP	PS
GIF	AFPDS
GIF	PCL
GIF	PS
PCL	AFPDS
PCL	PS
PDF	AFPDS
PDF	PCL
PDF	PS
PS	AFPDS
PS	PCL
TIF	AFPDS
TIF	PCL
TIF	PS
UNKNOWN	UNKNOWN

**Note:** If the input or output data stream format is unrecognized, "UNKNOWN UNKNOWN" is used in the exit program data field of the selection criteria for the Retrieve Exit Information (QUSRTVEI, QusRetrieveExitInformation) API. IPT will query up to the first ten registered exit programs using the "UNKNOWN UNKNOWN" value in the data field in an attempt to find an exit program that can successfully transform the input data stream format to the requested output data stream format. If, after querying up to ten exit programs, an exit program is not found that can handle the transform, IPT returns to the caller, stating it cannot transform the image.

IPT queries the exit point to determine if one matches both the requested input and output data stream formats. If one or more is found, IPT then queries only the first exit program listed in the Retrieve Exit Information (QUSRTVEI, QusRetrieveExitInformation) API response to see if it can handle the requested transform, using a value of 20 (process file) in the process option field of the process values parameter. If the exit program can handle the transform, it is called again with the actual transform request. If there is no exit program registered with the specified input and output data stream formats (that is, something

other than "UNKNOWN UNKNOWN" in the data field area), or if there are no registered exit programs that can handle the requested transform, IPT then queries the first ten exit programs registered using "UNKNOWN UNKNOWN" in the data field area and handles it as described above. If there are no exit programs found that can handle the transform, IPT handles the request as normal, by either transforming it if it is able to or notifying the caller that the data cannot be converted.

See the Add Exit Program (QUSADDEP, QusAddExitProgram) API or the Add Exit Program (ADDEXITPGM) command for more information.

## Authorities and Locks

None.

## Required Parameters

### Transform values

INPUT; CHAR(\*)

Conversion information that is input to the exit program from Image Print Transform. For the format and description of this information, see "Transform Values to Exit Program."

### Process values

I/O; CHAR(\*)

Process information that is input to the exit program from Image Print Transform. In addition, there is process information passed back from the exit program. For the format and description of this information, see "Process Values to Exit Program" on page 398.

### Input Data Stream

INPUT; CHAR(\*)

The data stream to be transformed by the exit program. The input data stream is broken into manageable sized buffers and sent to the exit program one at a time.

### Output Data Stream

OUTPUT; CHAR(\*)

The data stream that has been transformed by the exit program. If the data stream is larger than the buffer sent, the exit program sends it back in a buffer at a time and IPT puts it back together.

## Transform Values to Exit Program

### XFRM0100 Format

The following table shows the format of the XFRM0100 structure for the transform values parameter. For more details about the fields in the following table, see "Field Descriptions" on page 394.

Offset		Type	Field
Dec	Hex		
0	0	CHAR(8)	Structure version
8	8	BINARY(4)	Structure length
12	C	BINARY(4)	Input file format
16	10	BINARY(4)	Output file format
20	14	BINARY(4)	Invert image
24	18	BINARY(4)	Input decompression value
28	1C	BINARY(4)	Output compression value
32	20	BINARY(4)	Output horizontal resolution

Offset		Type	Field
Dec	Hex		
36	24	BINARY(4)	Output vertical resolution
40	28	BINARY(4)	Convert color to gray
44	2C	BINARY(4)	Convert gray to black and white
48	30	BINARY(4)	Output page orientation
52	34	BINARY(4)	Horizontal justification from left border
56	38	BINARY(4)	Vertical justification from top border
60	3C	BINARY(4)	Bits per pixel
64	40	BINARY(4)	Output page size
68	44	PACKED(15,5)	Paper width
76	4C	PACKED(15,5)	Paper length
84	54	BINARY(4)	Left unprintable area
88	58	BINARY(4)	Right unprintable area
92	5C	BINARY(4)	Top unprintable area
96	60	BINARY(4)	Bottom unprintable area
100	64	CHAR(12)	Reserved
112	70	CHAR(10)	Spooled file job name
122	7A	CHAR(10)	Spooled file user name
132	84	CHAR(6)	Spooled file job number
138	8A	CHAR(10)	Spooled file name
148	94	BINARY(4)	Spooled file number
152	98	CHAR(8)	Job system name
160	A0	CHAR(7)	Spooled file create date
167	A7	CHAR(1)	Reserved
168	A8	CHAR(6)	Spooled file create time
174	AE	CHAR(2)	Reserved
176	B0	CHAR(10)	Writer message queue name
186	BA	CHAR(10)	Writer message queue library name
196	C4	CHAR(16)	Internal spooled file identifier
212	D4	CHAR(16)	Internal job identifier
228	E4	CHAR(28)	Reserved

## Field Descriptions

**Bits per pixel.** Number of bits per pixel to be used in the output image.

Possible values are:

- 1 Black and white output
- 8 Grayscale output
- 24 Color output

**Bottom unprintable area.** The number of pixels of unprintable area on the bottom border.

**Convert color to gray.** Whether to reduce the output data stream's pixel information content from color to gray. If this field is set to No, the pixel information content of the output data stream will be that of the input data stream. Setting this to Yes reduces the output pixel information content to gray scale if it would have been color and will otherwise have no effect.

0 No  
1 Yes

**Convert gray to black and white.** Whether to reduce the output data stream's pixel information content from gray to black and white. If this field is set to No, the pixel information content of the output data stream will be that of the input data stream. Setting this to Yes forces the output to black and white.

Possible values are:

0 No  
1 Yes

**Horizontal justification from left border.** Distance in pixels from left border to justify the image.

**Input decompression value.** Compression algorithm used on input data stream.

Possible values are:

1 None  
3 CCITT Group 3 Fax  
4 CCITT Group 4 Fax  
5 LZW  
6 JPEG  
7 Packbits  
8 IBM<sup>®</sup> MMR (modified-modified read)  
10 BMP-RLE

**Input file format.** Format of the input print or image data stream. Possible values are:

1 TIFF (Tag Image File Format)  
2 BMP (Bitmap)  
6 GIF (Graphics Interchange Format)  
20 AFPDS  
22 PCL  
23 PostScript  
25 PDF  
99 Unknown

**Internal job identifier.** The internal job identifier of the job that created the spooled file. This field will be set to blanks when the input stream is not a spooled file.

**Internal spooled file identifier.** The internal spooled file identifier of the spooled file being processed. This field will be set to blanks when the input stream is not a spooled file.

**Invert image.** If yes, black becomes white and white becomes black. This parameter has no effect for color output. It also has no effect for color images output as gray on a Printer Control Language (PCL) color printer.

Possible values are:

- 0 No
- 1 Yes

**Job system name.** The name of the system where the job that created this spooled file ran. This field will be set to blanks when the input stream is not a spooled file.

**Left unprintable area.** The number of pixels of unprintable area on the left border.

**Output compression value.** The algorithm used to compress the output image or print data stream. Possible values are:

- 1 No compression
- 4 CCITT Group 4 Fax
- 7 PackBits
- 8 IBM MMR (modified-modified read)

**Output file format.** The format of the output data stream. Possible values are:

- 20 AFPDS (Advanced Function Presentation™ data stream)
- 22 PCL (Printer Control Language)
- 23 PostScript
- 99 Unknown

**Output page orientation.** How the output is oriented on paper. Possible values follow:

- 0 Portrait. For image input files, the top of each output page is placed on the narrowest dimension of the paper. For PostScript input data streams, the orientation specified in the input data stream is used.
- 90 Landscape. For image input files, the top of each output page is placed on the widest dimension of the paper. For PostScript input data streams, the output is rotated 90 degrees from the orientation specified in the input data stream.
- 180 Portrait, rotated 180 degrees. For PostScript input data streams, the output is rotated 180 degrees from the orientation specified in the input data stream.
- 270 Landscape, rotated 180 degrees. For PostScript input data streams, the output is rotated 270 degrees from the orientation specified in the input data stream.

**Output page size.** The paper size to be used for the output image or print data stream. See the table following the values for dimensions of each size value. Possible values are:

- 1 Executive
- 2 Letter
- 3 Legal
- 6 Ledger
- 13 A5
- 26 A4
- 27 A3
- 45 B5
- 46 B4
- 50 CONT80
- 51 CONT132
- 80 Monarch envelope
- 81 COMM10 envelope
- 89 COMM9 envelope
- 90 DL envelope

91 C5 envelope  
 100 B5 envelope

Name	Millimeters	Inches
<b>Paper:</b>		
Letter	215.9 x 279.4	8.5 x 11
Legal	215.9 x 335.6	8.5 x 14
Ledger	279.4 x 431.8	11 x 17
Executive	184.2 x 266.7	7.25 x 10.5
A3	297 x 420	11.7 x 16.54
A4	210 x 297	8.27 x 11.7
A5	148 x 210	5.83 x 8.27
B4	257 x 364	10.1 x 14.33
B5	182 x 257	7.17 x 10.1
CONT80	203.2 x 279.4	8 x 11
CONT132	335.3 x 279.4	13.2 x 11
<b>Envelope:</b>		
Monarch	98.4 x 190.5	3.875 x 7.5
Commercial 9	98.4 x 225.4	3.875 x 8.875
Commercial 10	104.8 x 241.3	4.125 x 9.5
DL	110 x 220	4.33 x 8.66
B5	176 x 250	6.93 x 9.84
C5	162 x 229	6.38 x 9.02

**Output horizontal resolution.** The number of horizontal pixels per inch.

**Output vertical resolution.** The number of vertical pixels per inch.

**Paper length.** Length of paper to use; units in inches.

**Paper width.** Width of paper to use; units in inches.

**Right unprintable area.** The number of pixels of unprintable area on the right border.

**Reserved.** The field is reserved.

**Spooled file create date.** The date the spooled file was created on the system in the CYYMMDD format. See field Date file opened in API QUSRSPLA under field descriptions for more information on the date format. This field will be set to blanks when the input stream is not a spooled file.

**Spooled file create time.** The time the spooled file was created on the system in the HHMMSS format. See field Time file opened in API QUSRSPLA under field descriptions for more information on the time format. This field will be set to blanks when the input stream is not a spooled file.

**Spooled file job name.** The name of the job that the file being processed was spooled under. This field will be set to blanks when the input stream is not a spooled file.

**Spooled file job number.** The job number that the file being processed was spooled under. This field will be set to blanks when the input stream is not a spooled file.

**Spooled file name.** The name of the spooled file being processed. This field will be set to blanks when the input stream is not a spooled file.

**Spooled file number.** The number of the spooled file being processed. This field will be set to zero when the input stream is not a spooled file.

**Spooled file user name.** The user profile that the file being processed was spooled under. This field will be set to blanks when the input stream is not a spooled file.

**Structure length.** The length of the structure. Currently, this is set to decimal 256.

**Structure version.** The format identifier of the transform parameter structure. Currently, this is set to "XPRM0100".

**Top unprintable area.** The number of pixels of unprintable area on the top border.

**Vertical justification from top border.** Distance in pixels from top border to justify the image.

**Writer message queue library name.** The name of the library in which the writer message queue resides. This field will be set to blanks when no writer message queue is associated with the file or buffer.

**Writer message queue name.** The name of the message queue to which the writer issues messages. This field will be set to \*NONE when no writer message queue is associated with the file or buffer.

## Process Values to Exit Program

The following table shows the structure for the process values parameter. For more details about the fields in the following table, see "Field Descriptions."

Offset		Type	Field
Dec	Hex		
0	0	CHAR(8)	Structure version
8	8	BINARY(4)	Structure length
12	C	BINARY(4)	Process option
16	10	BINARY(4)	Input data stream length
20	14	BINARY(4)	Output data stream length
24	18	BINARY(4)	Length of returned data
28	1C	CHAR(1)	Transform ability
29	1D	CHAR(3)	Reserved
32	20	BINARY(4)	Finished transform

## Field Descriptions

**Finished transform.** (OUTPUT) Signals when transform of the input data stream has completed. Possible values are:

0 No  
1 Yes

**Input data stream length.** (INPUT) Length of data being sent to transform. The maximum value is 16700000. If the total is larger than this maximum, it will be sent a buffer at a time.

**Length of returned data.** (OUTPUT) Length of transformed data returned by the exit program. If this field is larger than the output buffer length, it must be returned as separate buffers in the correct order. This is accomplished by setting the finished transform field to No for all output data streams returned except the last one, which must have the finished transform field set to Yes. If a terminating error occurs in the process of transforming, the exit program may set the length to a negative value, and the process will be halted.

**Output data stream length.** (OUTPUT) Length of buffer for transformed output data stream.

**Process option.** (INPUT) The action for the exit program to take. The first option is process (20). Possible values are:

- 20 (*Process file*) A data stream is available to be transformed. The transform and process parameter structures and other information is passed. The exit program decides whether to transform the data and sets the transform ability field in the process parameter structure and returns.
- 30 (*Transform data*) The exit program is called if the transform ability field is set to 1 (can transform). The data to be transformed is passed in the input data stream parameter. The transformed data is returned in the output data stream parameter. The exit program is called repetitively with process option 30 to retrieve multiple output data streams until the exit program indicates it is done with this input data stream by setting the finished transform to Yes. This process is repeated until all the data required in the input data stream has been passed to the exit program. At that time, the exit program is called with a Process option 40.
- 40 (*End file*) This option informs the exit program that it has received all the data from the input data stream. No data is passed on this call. The exit program returns additional transformed data through the output data stream buffer until the exit program indicates it is complete by setting the finished transform to Yes.
- 50 (*Terminate*) The exit program will be notified that processing on this data stream has stopped and the exit program should terminate. Option 50 may be used after any other process option.

**Reserved.** Space reserved for future use.

**Structure length.** (INPUT) The length of the structure. Currently, this is set to decimal 36.

**Structure version.** (INPUT) The format identifier of the transform parameter structure. Currently, this is set to "IPRM0100".

**Transform ability.** (OUTPUT) Used by the exit program, when running during process option 20, to notify Image Print Transform whether the exit program can transform the data stream.

Possible values are:

- 0 Unable to transform
- 1 Able to transform

Exit program introduced: V5R1

Top | "Print APIs," on page 1 | APIs by category

---

## IPDS to PDF Transform Interface Exit Program

Required Parameter Group:

1	PDF tranform information	Input	Char(*)
---	--------------------------	-------	---------

Default Public Authority: \*USE

PSF/400 uses the IPDS™ to PDF Transform Interface exit program to submit an IPDS to PDF transform when a PDF transform is requested to the QIBM\_QPQ\_Transform exit point and the PDFGEN parameter on the PSF configuration object has a value other than \*NONE.

If the IPDS to PDF Transform exit program encounters an error in starting the transform (TCP/IP is not started or any other start-up error), the transform returns to the program that called it and writes an error message to the PDF transform job log. If no transform is registered to the exit point, PSF/400 will time out when it tries to communicate with the PDF transform and will log its own error.

## Authorities and Locks

There are no locks for this exit program.

Subdirectories and files created by the PDF transform are owned by the spooled file owner and have \*EXCLUDE public authority. The transform has \*USE authority to the QIPS library.

## Parameters

### PDF Transform Information

INPUT; CHAR(\*)

PDF transform information that is input to the IPDS to PDF Transform Interface exit program from the PSF/400 print writer.

### Length of PDF transform information to the IPDS to PDF Transform Interface

INPUT; CHAR(4)

The length (in zoned decimal) of the PDF transform information specified in the PDF transform input information to the IPDS to PDF Transform Interface exit program.

## PDF Transform Input Information

The following table shows the structure for the PDF transform input parameter.

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	Device type
10	A	CHAR(10)	Paper size for drawer 1
20	14	CHAR(10)	Paper size for drawer 2
30	1E	CHAR(256)	File directory
286	11E	CHAR(5)	TCP/IP port
291	123	CHAR(10)	Print writer's message queue
301	12D	CHAR(10)	Print writer's message queue library
311	137	CHAR(8)	Printer font ID
312	138	CHAR(1)	Inline font indicator
313	139	CHAR(1)	Group boundary operation

The example below is provided to show the interaction between the PSF/400 and PDF transform exit program.

1. Printer writer is started by user.

This ties a PSF configuration object to a printer device. The configuration object must have a PDFGEN value other than \*NONE and the other appropriate PDF parameters appropriately defined based on the the PDFGEN value chosen.

2. PSF/400 does a submit job to get the PDF transform started.

The PDF Transform exit program uses the input information to initialize itself. PSF/400 saves the job number from the submit job that is associated with the transform.

3. The writer selects the first eligible spooled file to process.

The PDF transform communicates with PSF/400 as a virtual printer device, but instead of printing the IPDS, the transform converts it to PDF and stores it either in a user specified path (when PDFGEN is \*STMF) or in a temporary path location chosen by PSF/400.

When PSF/400 receives an indication from the virtual printer device that all pages have been stacked, it knows that the PDF transform is finished with that spooled file.

4. PSF/400 mails or spools the data.

At this time, if PDFGEN had a value of \*MAIL or \*SPLF, PSF/400 will complete the request by either electronically mailing the PDF file or spooling it to the appropriate output queue. When either function is complete, PSF/400 deletes the temporary path name.

5. Printer writer is ended by user.

PSF/400 does its normal clean up and ends the job that is running the PDF transform.

## Field Descriptions

**Device type.** The device type that the transform should emulate. It is specified in the PDFDEVTYPE parameter of the PSF configuration object.

**File directory.** The first two components of the file path names specified in the PDFDIR parameter of the PSF configuration object and assigned to the PDF files created by the PDF transform. The rest of the path name is assigned by the PDF transform.

**Group boundary operation.** The value of PDFMULT parameter in the PSF configuration object. Valid values are:

- 1 Separate the output into multiple PDF files at the DGB commands.
- 2 Place an index tag at the DGB Location.

**Inline font indicator** Specifies whether fonts should be placed inline with the PDF output. Valid values are:

- 0 File should not contain the fonts inline.
- 2 File should contain the fonts inline.

**Paper size for drawer 1.** The paper size specified in the PDFPPRDWR1 value of the PSF configuration object that the transform should report to PSF/400.

**Paper size for drawer 2.** The paper size specified in the PDFPPRDWR2 value of the PSF configuration object that the transform should report to PSF/400.

**Printer font ID.** The printer default font ID to use if a font is not specified in the IPDS data. It is composed of the graphic character set ID, code page global ID, font character set global ID, and character width.

**Print writer's message queue.** The name of the PSF writer's message queue.

**Print writer's message queue library.** The library in which the PSF writer's message queue can be found.

**TCP/IP port.** The TCP/IP port of the transform's virtual printer.

Exit program introduced: V5R1

Top | "Print APIs," on page 1 | APIs by category

---

## Print Driver Exit Program

Required Parameter Group:

1	Process option	Input	Binary(4)
2	Option specific input information to print driver	Input	Char(*)
3	Length of option specific input information	Input	Binary(4)
4	Option specific output information from print driver	Output	Char(*)
5	Length of option specific output information	Input	Binary(4)

QSYSINC Member Name: ESPDRVXT

This Print Driver exit program defines how a user-defined print driver exit program must be written in order to be used with the i5/OS<sup>®</sup> open writer program. When a user has enabled a print driver exit for a particular device, a writer started to that device opens the writer as its main processing program

This open writer program handles the scheduling of spooled files to print, and relies on the Print Driver exit program to complete all device specific tasks.

A typical scenario is:

- Printer writer is started by an operator.
- The writer calls the print driver with the initialize (10) process option.

The print driver initializes its environment.

- The writer selects the first eligible file to process.

The writer calls the print driver with the process file (20) process option. The name of the spooled file is passed in the input information. After the file has been processed by the print driver, control returns to the writer.

- The writer waits for x seconds for another file to become available.

The number of seconds (x) to wait is returned by the print driver in the idle time field of the output information from the print driver parameter. If a file is not available, the writer calls the print driver with the idle (30) process option.

## Authorities and Locks

None.

## Required Parameter Group

### Process option

INPUT; BINARY(4)

The action for the Print Driver exit program to take. This information is filled in by the writer. The first option should be initialize (10). This is done when the writer is started.

Possible values are:

- |                            |  |
|----------------------------|--|
| 10 ( <i>Initialize</i> )   | The print driver sets any initial information with this option.  |
| 20 ( <i>Process file</i> ) | A spooled file is available to print. The spooled file name and other information is passed in the option specific input information to driver exit program. |

- 21 (*Reprocess file*) The print driver was interrupted while being called by the process file option (20). The print driver determines the appropriate action by using the Extract Writer Status (QSPEXTWI) API. The print driver can be interrupted only if the allow interrupts field was specified as \*YES (in the Option specific output information parameter) and a spooled file was deleted, held with OPTION(\*IMMED), or restarted (CHGSPLFA RESTART()).
- 30 (*Idle*) The writer calls the print driver after x seconds to inform the print driver that there is not a ready file on the output queue. The print driver can take the opportunity to end existing connections. The idle timer (x) value is filled in by the print driver when the writer calls the print driver with the initialize (10) process option.
- 40 (*Writer held*) The process has been held. The print driver does cleanup to ensure the proper start of the writer when it is released.
- 50 (*Terminate*) The writer is ending. The print driver does cleanup.

### Option specific input information to print driver

INPUT; CHAR(\*)

Information which is input to the Print Driver exit program from the open writer program. For the format and description of this information, see "Option Specific Input Information to Print Driver."

### Length of option specific input information

INPUT; BINARY(4)

The length of the option specific input information specified in the option specific input information to print driver parameter.

### Option specific output information from print driver

OUTPUT; CHAR(\*)

Information which is output from the Print Driver exit program to the open writer program. For the format and description of this information, see "Option Specific Output Information from Print Driver" on page 407.

### Length of option specific output information

INPUT; BINARY(4)

The length of the option specific output information to return in the option specific output information from the print driver parameter.

## Option Specific Input Information to Print Driver

The following table shows the structure for the option specific input information parameter. For more details about the fields in the following table, see "Field Descriptions" on page 404.

### Structure of the Option Specific Input Information Parameter

Offset		Type	Field
Dec	Hex		
0	0	CHAR(16)	Writer handle
16	10	CHAR(10)	Writer name
26	1A	CHAR(10)	Printer device name
36	24	CHAR(10)	Writer output queue name
46	2E	CHAR(10)	Writer output queue library name
56	38	CHAR(10)	Writer message queue name
66	42	CHAR(10)	Writer message queue library name
76	4C	CHAR(10)	Align file
86	56	CHAR(10)	Spooled file handle

Offset		Type	Field
Dec	Hex		
96	60	CHAR(16)	Internal job identifier
112	70	CHAR(16)	Internal spooled file identifier
128	80	CHAR(26)	Qualified job name
154	9A	CHAR(10)	Spooled file name
164	A4	BINARY(4)	Spooled file number
168	A8	BINARY(4)	Starting page number
172	AC	BINARY(4)	Drawer for separator pages
176	B0	BINARY(4)	Number of job separators
180	B4	BINARY(4)	Number of file separators
184	B8	BINARY(4)	Termination type
188	BC	CHAR(10)	Current writer form type
198	C6	CHAR(8)	Job system name
206	CE	CHAR(7)	Spooled file create date
213	D5	CHAR(1)	Reserved
214	D6	CHAR(6)	Spooled file create time
220	DC	CHAR(23)	Reserved

## Field Descriptions

**Align file.** Specifies how to control the forms alignment.

Possible values are:

- \*WTR            The writer keeps track of the output that is printed and issues a forms alignment message whenever it determines that forms may need to be aligned.
- \*FILE           The forms alignment message is issued for every file that has \*YES specified for the align page prompt.
- \*FIRST          The forms alignment message is issued only for the first file printed.
- \*SKIP           The forms alignment message is skipped for the next file.

**Current writer form type.** The form type the writer is currently printing.

**Drawer for separator pages.** The paper drawer to be selected when printing the job and file separators.

**Internal job identifier.** The internal job identifier of the job that created the spooled file.

**Internal spooled file identifier.** The internal spooled file identifier of the spooled file being processed.

**Job system name.** The name of the system where the job that created the spooled file ran.

**Number of file separators.** The number of the file separators to be printed.

**Number of job separators.** The number of the job separators to be printed.

**Printer device name.** The name of the printer device.

**Qualified job name.** The qualified job name for the job that created the file.

**Reserved.** Reserved to maintain compatibility with the writer transform exit and the Host Print Transform (QWPZHPTR) API.

**Spooled file create date.** The date the spooled file was created on the system in the CYYMMDD format. See field Date file opened in API QUSRSPLA under field descriptions for more information on the date format.

**Spooled file create time.** The time the spooled file was created on the system in the HHMMSS format. See field Time file opened in API QUSRSPLA under field descriptions for more information on the time format.

**Spooled file handle.** The handle for the spooled file to be printed.

**Spooled file name.** The name of the spooled file being processed.

**Spooled file number.** The number of the spooled file being processed.

**Starting page number.** The number of the page to start printing.

**Termination type.** The type of termination for the writer.

Possible values are:

- 1 (*normal*)            The writer ended normally.
- 2 (*immediate*)        The writer ended immediately.
- 3 (*abnormal*)         The writer ended abnormally.

**Writer handle.** The handle to the writer job.

**Writer name.** The name of the writer.

**Writer message queue library name.** The name of the library in which the writer message queue resides.

**Writer message queue name.** The name of the message queue to which the writer issues messages.

**Writer output queue library name.** The name of the library in which the output queue resides.

**Writer output queue name.** The name of the output queue from which the writer is processing spooled files.

Fields Defined for the Option Specific Input Information Parameter (page 405) presents, in the same order, all the fields previously described in the Structure of the Option Specific Input Information Parameter (page 403). Each entry indicates if a field is defined for a particular process option as defined by the process option parameter. Possible values are Y for yes, and N for no.

**Fields Defined for the Option Specific Input Information Parameter**

Field	Initialize (10)	Process file (20)	Re-process (21)	Idle (30)	Writer held (40)	Terminate (50)	Description
Writer handle	Y	Y	Y	Y	Y	Y	For use with new writer APIs.
Writer name	Y	Y	Y	Y	Y	Y	Name specified on STRPRTWTR command.

Field	Initialize (10)	Process file (20)	Re-process (21)	Idle (30)	Writer held (40)	Terminate (50)	Description
Printer device name	Y	Y	Y	Y	Y	Y	The name of the printer the writer is started to.
Writer output queue name	Y	Y	Y	Y	Y	Y	The name of the output queue that the writer is processing.
Writer output library name	Y	Y	Y	Y	Y	Y	The library in which the writer output queue resides.
Writer message queue name	Y	Y	Y	Y	Y	Y	Name of the message queue for the started writer.
Writer message queue library name	Y	Y	Y	Y	Y	Y	Name of the library in which the message queue resides.
Align file	Y	Y	Y	Y	Y	Y	Value from STRPRTWTR command.
Spooled file handle	N	Y	Y	N	N	N	For use with new writer APIs.
Internal job identifier	N	Y	Y	N	N	N	For use with existing spool APIs.
Internal spooled file identifier	N	Y	Y	N	N	N	For use with existing Spool APIs.
Qualified job name	N	Y	Y	N	N	N	Job owning spooled file
Spooled file name	N	Y	Y	N	N	N	Name of spooled file being processed
Spooled file number	N	Y	Y	N	N	N	Number of spooled file being processed
Starting page number	N	Y	Y	N	N	N	Page at which to start printing
Drawer for separator pages	N	Y	Y	N	N	N	Paper drawer for separator pages
Number of job separators	N	Y	Y	N	N	N	Number of job separators to print
Number of file separators	N	Y	Y	N	N	N	Number of file separators to print
Termination type	N	N	N	N	N	Y	This is how the writer has been ended (normal, immediate, or abnormal for unrecoverable error).
Current writer form type	N	Y	Y	N	N	N	Form type writer is currently processing.

Field	Initialize (10)	Process file (20)	Re-process (21)	Idle (30)	Writer held (40)	Terminate (50)	Description
Job system name	N	Y	Y	N	N	N	Name of the system the job resides on.
Spooled file create date	N	Y	Y	N	N	N	Date the spooled file was created.
Reserved	N	N	N	N	N	N	Reserved for compatibility.
Spooled file create time	N	Y	Y	N	N	N	Time the spooled file was created.

## Option Specific Output Information from Print Driver

The following table shows the structure for the option specific output information parameter. For more details about the fields in the following table, see "Field Descriptions."

### Structure of the Option Specific Output Information Parameter

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Error code
4	4	BINARY(4)	Initial status
8	8	BINARY(4)	Idle timer
12	C	CHAR(1)	Allow interrupt
13	D	CHAR(3)	Reserved
16	10	CHAR(10)	LAN driver name

## Field Descriptions

**Allow interrupt.** Whether to allow the writer to interrupt the print driver. This determines whether Hold spooled file (HLDSPLF) with option \*IMMED, Delete spooled file (DLTSPLF), or Change spooled file attributes (CHGSPLFA) changing the Restart printing (RESTART) parameter causes the print driver to be interrupted before it returns to the writer normally. If the print driver program allows interrupts, it is necessary that the print driver program is written such that it is able to resume processing following an interruption.

Possible values are:

- 0 Do not allow interrupts.
- 1 Allow interrupts.

**Error code.** The error code returned by the print driver after each call.

Possible values are:

- 0 No error occurred.
- 1 Writer ending normally
- 2 Writer ending immediately

10 Processing of the file was interrupted (Note: This error code should be used when the driver program has detected a Hold Writer (HLDWTR) command with option \*PAGEEND or \*CNTRLD, and wishes to continue processing the spooled file after the writer is released.)

**Idle timer.** The number of seconds to allow between spooled files. A value of 0 indicates that no calls are made to the print driver with the Idle (30) process option.

**Initial status.** The initial status for the spooled file selected by the writer.

Possible values are:

- 1 (*pending*)            The spooled file is being converted.
- 2 (*writing*)           The spooled file is being selected by the writer.
- 3 (*sending*)           The spooled file is being sent to a remote system

**LAN driver name.** The name of the LAN driver to process the spooled files.

**Reserved.** An ignored field.

Fields Defined for the Option Specific Output Information Parameter (page 408) presents, in the same order, all the fields previously described in the Structure of the Option Specific Output Information Parameter (page 407). Each entry indicates if a field is used with a particular process option as defined by the Process Option parameter. Possible values are Y for yes, and N for no.

**Fields Defined for the Option Specific Output Information Parameter**

Field	Initialize (10)	Process file (20)	Re-process (21)	Idle (30)	Writer held (40)	Terminate (50)	Description
Error code	Y	Y	Y	Y	Y	N	This indicates if the driver exit program has encountered an error. A value 0 indicates no error. A value of 1 causes the writer to wait for all current print requests to complete, and then call the print driver with an normal termination. A value of 2 causes the writer to immediately call the print driver with an immediate termination type. A value of 10 causes the writer to call the print driver with a reprocess file (21) process option for the currently active file.

Field	Initialize (10)	Process file (20)	Re-process (21)	Idle (30)	Writer held (40)	Terminate (50)	Description
Initial status	Y	N	N	N	N	N	Specifies the initial status of spooled files that are selected for processing by the writer.
Idle timer	Y	Y	Y	Y	N	N	Specifies the time to wait in seconds before calling the print driver with the Idle (30) process option. A value of 0 indicates that no calls are made with the Idle process option.
Allow interrupt	Y	N	N	N	N	N	Allows the writer to interrupt the print driver during a call from the writer to the print driver.
LAN driver name	Y	N	N	N	N	N	Specifies the name of the local area network (LAN) and the name of the printer driver it is started to.

Exit program introduced: V3R7

Top | "Print APIs," on page 1 | APIs by category

---

## PSF/400 Mapping Exit Program

Required Parameter Group:

1	Mapping exit input information	Input	Char(*)
2	Length of input information	Input	BINARY(4)
3	Mapping exit output information	Output	Char(*)
4	Length of output information buffer	Input	BINARY(4)
5	Length of output information available	Output	BINARY(4)

Default Public Authority: \*USE

Threadsafe: Yes

The PSF/400 mapping exit program allows a customer to specify a routing tag instead of a mail address in the USRDFNDDTA parameter of the printer file, using indices and Infoprint Server's CRTAFPDTA command, or the DDS STRPAGGRP parameter. The PSF/400 mapping exit program also allows a customer or vendor to provide a program to map a routing tag (such as a customer number) to one or more distribution actions, and configure the details of the distribution. The PDF file can be stored as a streamfile in the Integrated File System, respooled as AFP™ or PDF on an output queue, e-mailed, or any combination of those distributions. In addition, the mapping exit program allows configuration of those distributions, including the ability to specify the name of the PDF streamfile, the ability to specify the subject, encryption settings and additional attachments of the e-mail, or the output queue to spool the PDF.

This exit program is called for each successfully generated PDF file whenever a mapping exit program is specified on the PDFMAPPGM parameter of the PSFCFG, and the PDFGEN parameter does not have a value of \*NONE. Once PSF/400 has received notification from the virtual printer that all pages have been stacked, it calls the exit program for each PDF output file to resolve the routing tag information to the proper distribution action.

## Authorities and Locks

None.

## Required Parameter Group

### Mapping exit input information

INPUT; CHAR(\*)

The mapping exit program information that is input to the mapping exit program from the PSF/400 print writer. For the format and description of the information, see “Mapping Exit Program Input Information.”

### Length of input information

INPUT; BINARY(4)

The length of the input information specified in the mapping exit input information to the mapping exit program parameter.

### Mapping exit output information

OUTPUT; CHAR(\*)

The information that is output from the mapping exit program to the PSF/400 print writer. For the format and description of this information, see “Mapping Exit Program Output Information” on page 412.

### Length of output information buffer

INPUT; BINARY(4)

The size, in bytes, of the mapping exit program output information buffer.

### Length of output information available

OUTPUT; BINARY(4)

The length of the mapping exit program output information available. If this length is less than or equal to the length of the routing tag output information, all available information has been returned. If this length is greater than the length of the mapping program output information, only data up to the length provided is returned in the mapping exit program output information buffer.

## Mapping Exit Program Input Information

The following table shows the structure for the input information that is passed to the PSF/400 Mapping exit program.

### Printer File Parameter Table

Offset		Type	Field
Dec	Hex		
0	0	CHAR(26)	Qualified job name
26	1A	CHAR(10)	Spooled file name
36	24	BINARY(4)	Spooled file number
40	28	CHAR(250)	Routing tag
290	122	CHAR(340)	Path and name of PDF file

Offset		Type	Field
Dec	Hex		
630	276	CHAR(1)	Mail server type
631	277	CHAR(1)	Reserved
632	278	BINARY(4)	Path name CCSID
636	27C	CHAR(10)	Mail sender
646	286	CHAR(10)	User Data
656	290	CHAR(8)	Job system name
664	298	CHAR(8)	Creation time stamp
672	2A0	CHAR(10)	Output queue on which the spooled file is located
682	2AA	CHAR(10)	Output queue library
692	2B4	CHAR(10)	PDF mapping object name
702	2BE	CHAR(10)	PDF mapping object library
712	C8	CHAR(10)	Formtype

## Field Descriptions

**Creation time stamp.** The date and time the spooled file was created.

**Formtype.** The value from the Formtype parameter on the print file.

**Job system name.** The name of the system on which the spooled file was created.

**Mail sender.** The sender of the electronic mail. The value is specified on the printer file USRDFNDTA parameter or on the PSF configuration object PDFSENDER parameter. If no mail sender is identified on USRDFNDTA, the value from the PDFSENDER parameter of the PSF configuration object is used.

**Mail server type.** The mail server in use. Valid values are:

- 1 The SNDDST (Send Distribution) command is used to send e-mail messages.
- 2 The SMTP protocol is being used to send e-mail messages.

**Routing tag.** The mail address or routing tag specified on the printer file USRDFNDTA parameter or the DDS STRPAGGRP keyword. This is set to blanks ('40'X) if no value was provided by either parameter. It is possible that the mapping program could return mail addresses based on job, file, and USRDTA information. If both provide tag or address information, the DDS STRPAGGRP keyword takes precedence.

**Output queue on which the spooled file is located.** The output queue to which the writer was started.

**Output queue library.** The library that contains the output queue to which writer was started.

**Path and name of PDF file.** The path and name of the PDF file.

**Path name CCSID.** The CCSID of the path and name of the PDF file.

**PDF mapping object library.** Name of the PDF mapping object library, if one was specified on the PSFCFG object. This is set to blanks (X'40') if no value was provided on the PDFMAPBOBJ parameter of the PSFCFG object.

**PDF mapping object name.** Name of the PDF mapping object, if one was specified on the PSFCFG object. This is set to blanks (X'40') if no value was provided on the PDFMAPBOBJ parameter of the PSFCFG object.

**Qualified job name.** The qualified job name of the job that created the spool file that was converted to PDF. The data is comprised of the following information:

CHAR(10)      Job name  
 CHAR(10)      User name  
 CHAR(6)        Job number

**Reserved.** An unused field.

**Spooled file name.** The name of the spooled file that was converted to PDF.

**Spooled file number.** The unique number of the spooled file that was converted to PDF.

**User data.** The value in the printer file USRDTA parameter. It contains blanks if USRDTA contains blanks.

## Mapping Exit Program Output Information

The following table shows the structure for the output information from the Mapping exit program.

Offset		Type	Field
Dec	Hex		
0	0	CHAR(1)	Disposition of PDF e-mail
1	1	CHAR(1)	More processing
2	2	CHAR(2)	Reserved - initialized to null
4	4	BINARY(4)	Length of message text (0 to 255 bytes)
8	8	BINARY(4)	Length of mail address data (0-16,000,000)
12	C	CHAR(255)	Message text data
267	10B	CHAR(1)	Reserved - initialized to null
268	10C	BINARY(4)	Offset to extension area
272	110	BINARY(4)	CCSID of message text data and subject
276	114	CHAR(1)	Disposition of PDF streamfile
277	115	CHAR(1)	Disposition of PDF spoolfile
278	116	CHAR(1)	Disposition of PDF error
279	117	CHAR(1)	Disposition of AFPDS spoolfile
280	118	BINARY(4)	Offset to message text data
284	122	CHAR(3)	Reserved - initialized to null
287	11F	CHAR(*)	E-mail addresses

## Field Descriptions

**CCSID of message text data and subject.** The CCSID associated with the message text data and subject, if specified.

0                    A value from 1 to 65533.  
 ccsid-value        A value from 1 to 65533.

This field is used only when an SMTP mail server is specified in your PSF configuration object and must be set to zero otherwise.

This CCSID will be used to convert the data to Unicode for inclusion within the e-mail. The receiver of the e-mail may have to have the receiving e-mail program configured for the appropriate character set to be able to view the e-mail correctly.

**Disposition of AFPDS spoolfile.** Specifies whether the PDF should be spooled. If file is to be spooled as AFPDS, an output queue has to be defined on the PSFCFG or in the “Spooled File AFPDS Distribution” on page 421 in the Output Queue Name field.

'0' Do not spool AFPDS on an output queue.  
'1' Spool AFPDS on an output queue

**Disposition of PDF e-mail.** Specifies whether the PDF should be e-mailed.

'0' Do not e-mail  
'1' E-mail to specified addresses

**Disposition of PDF error.** This field can be used to signal to PSF/400 that an error was detected in the Exit Program. If this flag is set, PSF/400 will e-mail the PDF output to the e-mail address on the PSFCFG PDFADMIN, if one is available.

'0' No errors mapping input.  
'1' Errors were encountered.

**Disposition of PDF spoolfile.** Specifies whether the PDF should be spooled. If file is to be spooled as PDF, an output queue has to be defined on the PSFCFG or in the “Spooled File PDF Distribution” on page 420 in the Output Queue Name field

'0' Do not spool PDF file on an outq  
'1' Spool PDF file on an outq

**Disposition of PDF streamfile.** Specifies whether the PDF should be stored as a stream file in the Integrated File System. If file is to be stored as a stream file, user will have to have the path name defined in the PSFCFG or specified in the PDF file name field of the “Extension Area Format” on page 414.

'0' Do not store PDF as a stream file.  
'1' Store PDF as a stream file.

**E-mail addresses.** The e-mail addresses that should be used for sending the PDF file. They must be in the following format: ('e-mail address' 'e-mail address'.) Each address must be delimited by single quotes.

**Length of e-mail address data.** The length of the e-mail address data to be used in sending the PDF file. The maximum length allowed is 16 MB, since this is the largest value that can be used allocate a user space. If the PDF file should not be e-mailed, the length of the e-mail address data should be set to 0.

**Length of message text data.** The length of the message text to be used in sending the PDF file. This must be a value ranging from 0 to 255. If no message text is to be used in sending the PDF file, the length of the message text data should be set to 0.

**Message text data.** The message text data to be used in sending the PDF file. This value should be blanks if the text length is 0.

**More processing.** Whether the exit program should be called again because it has more processing to do for the same spooled file.

'0'(X'F0')

Do not call exit program again. Since this field occupies the offset of a field that was reserved in V5R1, a value of X'00' will be treated the same as 0 (X'F0') for compatability.

'1'(X'F1')

Call exit program again. The input information presented to the exit program is not modified. If at any point an error occurs while processing an e-mail:

- The exit program is not called again, irrespective of the setting for this field.
- The spooled file is held.

**Offset to extension area.** Offset from the beginning of this structure. The value specified for this entry must be a 4-byte multiple. A value of 0 indicates that the extension area is not being used. See "Extension Area Format" for additional information.

**Offset to message text data.** Offset from the beginning of this structure. The value must be a 4-byte multiple Use this field if you specify more than 255 bytes of message text data. If you are using the Message text data field, specify a value of 0 for Offset to message text data.

**Reserved - initialized to null.** A reserved field.

**Reserved (set to X'00').** A reserved field.

## Extension Area Format

This format is specified only when needed. If your application has no need for it, specify a value of 0 for Offset to extension area in the e-mail exit program output information.

This format must begin on a 4-byte multiple. All offsets are computed from the beginning of the mapping exit program output information structure.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Length of extension area format
4	4	BINARY(4)	Offset to subject
8	8	BINARY(4)	Length of subject
12	C	BINARY(4)	Offset to ReplyTo e-mail address
16	10	BINARY(4)	Length of ReplyTo e-mail address
20	14	BINARY(4)	Offset to CC e-mail addresses
24	18	BINARY(4)	Length of CC e-mail addresses
28	1C	BINARY(4)	Offset to BCC e-mail addresses
32	20	BINARY(4)	Length of BCC e-mail addresses
36	24	BINARY(4)	Offset to list of path names for body of e-mail
40	28	BINARY(4)	Offset to path name for directory for files
44	2C	BINARY(4)	Length of path name for directory for files
48	30	BINARY(4)	Offset to list of path names of attachments

Offset		Type	Field
Dec	Hex		
52	34	BINARY(4)	Offset to PDF file name for storing as a file
56	38	BINARY(4)	Length of PDF file name for storing as a file
60	3C	BINARY(4)	Offset to file name for PDF attachment for e-mail
64	40	BINARY(4)	Length of file name for PDF attachment for e-mail
68	44	BINARY(4)	Offset to PDF file public authority
72	48	BINARY(4)	Length of PDF file public authority
76	4C	BINARY(4)	Offset to spooled file PDF distribution
80	50	BINARY(4)	Length of spooled file PDF distribution
84	54	BINARY(4)	Offset to spooled file AFP distribution
88	58	BINARY(4)	Length of spooled file AFP distribution
92	5C	BINARY(4)	Offset to PDF encryption information
96	60	BINARY(4)	Length of PDF encryption information
100	64	CHAR(10)	Mail sender

## Field Descriptions

**Length of BCC e-mail addresses.** The length of the BCC e-mail address data to be used in sending the PDF file. This field is used only when an SMTP mail server is specified in your PSF configuration object; otherwise, it must be set to zero.

**Length of CC e-mail addresses.** The length of the CC e-mail address data to be used in sending the PDF file. This field is used only when an SMTP mail server is specified in your PSF configuration object; otherwise, it must be set to zero.

**Length of file name for PDF attachment for e-mail** Length of the file name used to rename the PDF attachment for e-mail distribution. This field is only used when Disposition for PDF e-mail is set to '1'; otherwise, it must be set to zero.

**Length of extension area format.** The length of the extension area format. Specify one of the following values:

- 52 This length is used by the oldest mapping programs. Fields after Offset to list of path names of attachments will not be recognized.
- 100 This length is used by some mapping programs. Fields after Length of PDF encryption information will not be recognized.
- 110 This length allows full use of the Extension Area Format. All fields will be recognized.

**Length of path name for directory for files.** The length of the path name containing the directory for which files may be stored. This field is used only when an SMTP mail server is specified in your PSF configuration object; otherwise, it must be set to zero.

**Length of PDF encryption information .** The length of the PDF file encryption information. This field is used only when PDF encryption information is specified; otherwise, it must be set to zero.

**Length of PDF file name for storing as a streamfile.** The length of the PDF file name used when Disposition of PDF streamfile is set to 1. This field is used only when Disposition of PDF streamfile is set to '1'; otherwise, it must be set to zero.

**Length of PDF file public authority** The length of public authority to the PDF file. This field is used only when the PDF file is to be stored as a stream file and the PDF Streamfile Disposition is equal to '1'; otherwise, it must be set to zero.

**Length of spooled file AFP distribution** The length of the AFP spooled file distribution information. This field is used only when a additional spooled file information is specified; otherwise, it must be set to zero.

**Length of spooled file PDF distribution** The length of the PDF spooled file distribution information. This field is used only when additional spooled file information is specified; otherwise, it must be set to zero.

**Length of ReplyTo e-mail address.** The length of the ReplyTo e-mail address. This field is used only when an SMTP mail server is specified in your PSF configuration object; otherwise, it must be set to zero.

**Length of subject** The length of the character string to be used as the subject text. Maximum length is 255. Data longer than the maximum length will be truncated.

**Offset to BCC e-mail addresses.** Offset to BCC e-mail addresses. This area contains the BCC e-mail addresses that should be used for sending the PDF file. These addresses must be in the following form: ('e-mail address' 'e-mail address'...). Each e-mail address must be delimited by single quotes.

A value of 0 indicates that BCC e-mail addresses are not specified. This field is used only when an SMTP mail server is specified in your PSF configuration object; otherwise, it must be set to zero.

**Mail sender.** The name to be used for the sender of the electronic mail. Pad the field with blanks ('40'X), if necessary.

*Blanks ('40'X)* The mail sender specified on the PSF configuration object is used.  
*mail-sender-name* The name of the mail sender. This name must be a valid user profile enrolled in the System Distribution Directory. If your PSF configuration object specifies PDFMAILSVR(mail-server-name) or PDFMAILSVR(\*LOCAL), the entry in the System Distribution Directory must have an SMTP user ID and SMTP domain specified.

**Offset to CC e-mail addresses.** Offset to BCC e-mail addresses. This area contains the CC e-mail addresses that should be used for sending the PDF file. These addresses must be in the following form: ('e-mail address' 'e-mail address'...). Each e-mail address must be delimited by single quotes.

A value of 0 indicates that CC e-mail addresses are not specified. This field is used only when an SMTP mail server is specified in your PSF configuration object; otherwise, it must be set to zero.

**Offset to file name for PDF attachment for e-mail** The offset to the file name that will be used for the PDF output that is attached to the e-mail when Disposition of PDF e-mail is equal to '1'. If Disposition of PDF e-mail is set to one, and this field is set to '0', PSF will determine a default file name to use. If Disposition of PDF e-mail is set to 0, this field must be set to zero as well.

**Offset to list of path names of attachments.** The offset to the list of path names listing the attachments. One or more attachments may be specified. The PDF file you are sending must not be listed here; otherwise, it will be attached again. A value of 0 indicates that no attachments are specified.

If you have specified a directory to be used for files (see Offset to path name for directory for files), then the path name specified for this field will be appended to the value specified for the directory. If you have not specified a directory to be used for files, the path name specified for each attachment must be fully resolved. These files must be in the Integrated File System. Please note the following:

- PSF can only access file systems that are threadsafe; therefore, you must specify a file system that is threadsafe. The following file systems are not threadsafe:
  - QFileSvr.400
  - Network File System (NFS)
  - QDLS
- All files must be encoded in a CCSID that matches the CCSID/CodePage of the target system.

See “Stream File Format” on page 419 for information about the format to use when specifying this information. This field is used only when an SMTP mail server is specified in your PSF configuration object; otherwise, it must be set to zero.

**Offset to list of path names for body of e-mail.** Offset to list of path names listing the files to be used for the body of the e-mail. One or more files may be specified. The PDF file you are sending must not be listed here or it will be attached again. A value of 0 indicates that a stream file is not specified.

The order in which you list these files determines the order in which they are placed in the body (message portion) of the e-mail. Files are placed in the body in the order in which they are specified. If you have specified Message text data, this is placed in the body of the e-mail first. These files must be in the Integrated File System. Please note the following:

- PSF can only access file systems that are threadsafe; therefore, you must specify a file system that is threadsafe. The following file systems are not threadsafe:
  - QFileSvr.400
  - Network File System (NFS)
  - QDLS
- All files must be encoded in a CCSID that matches the CCSID/CodePage of the target system.

See “Stream File Format” on page 419 for information about the format to use when specifying this information.

The CCSID of the file is used if a code page conversion is required.

If a file is not text, it will be placed as an attachment to the e-mail. For example, if you specify a sound file to be part of the body of the e-mail, it will actually be an attachment.

This field is used only when an SMTP mail server is specified in your PSF configuration object; otherwise, it must be set to zero.

**Offset to path name for directory for files** The offset to a path name containing the directory for which files may be stored. This path name must be fully resolved. If you use this field, then path names of specified attachments and body files will be appended to the path name specified at the offset for this field.

A value of 0 indicates that a path name is not specified. If a value of 0 is specified, you must specify the complete path name for all listed files used for the body of the e-mail and as attachments to the e-mail.

This directory is specified by using a value of 1 in the use-specified directory field in the Base format. This format is used when specifying files to be used as attachments or as part of the body of the e-mail. If you specify 0 for the Use specified directory field, then that particular file must have its full path specified.

This field is used only when an SMTP mail server is specified in your PSF configuration object; otherwise, it must be set to zero.

**Offset PDF encryption information.** The offset to the PDF file encryption information. See “PDF Encryption Structure” on page 422 for information about the format to use when specifying this information. A value of 0 indicates that no encryption information is specified.

**Offset to PDF file name for storing as a file.** The offset to the file name the PDF will be stored as in the Integrated File System. Requires that the Disposition of PDF streamfile be set to 1. If Disposition of PDF streamfile is set to 1, a value of 0 in this field indicates that PSF/400 will determine a default file name to be used.

**Offset to PDF file public authority.** The offset to the public authorities specified for the PDF stream file.

#### *Valid Authority Values*

**\*ALL** PSF specifies \*RWX for the Public data authority and \*ALL for the Public object authorities

#### **\*EXCLUDE**

The users do not have any of the data authorities to the objects.

**\*RWX** The users are given \*RWX authority to perform all operations on the object except those limited to the owner or controlled by object existence, object management, object alter, and object reference authority. The user can change the object and perform basic functions on the object. \*RWX authority provides object operational authority and all the data authorities.

**\*RX** The users are given \*RX authority to perform basic operations on the object, such as run a program or display the contents of a file. The user is prevented from changing the object. \*RX authority provides object operational authority and read and execute authorities.

**\*RW** The users are given \*RW authority to view the contents of an object and change the contents of an object. \*RW authority provides object operational authority and data read, add, update, and delete authorities.

**\*WX** The users are given \*WX authority to change the contents of an object and run a program or search a library or directory. \*WX authority provides object operational authority and data add, update, delete, and execute authorities.

**\*R** The users are given \*R authority to view the contents of an object. \*R authority provides object operational authority and data read authority.

**\*W** The users are given \*W authority to change the contents of an object. \*W authority provides object operational authority and data add, update, and delete authorities.

**\*X** The users are given \*X authority to run a program or search a library or directory. \*X authority provides object operational authority and data execute authority.

**Offset to spooled file AFP distribution.** The offset to the AFP spooled file information. See “Spooled File AFPDS Distribution” on page 421 for information about the format to use when specifying this information. A value of 0 indicates that no spooled file information is specified.

**Offset to spooled file PDF distribution.** The offset to the PDF spooled file information. See “PDF Encryption Structure” on page 422 for information about the format to use when specifying this information. A value of 0 indicates that no spooled file information is specified.

**Offset to ReplyTo e-mail address.** Offset to ReplyTo e-mail address. Use this address if you wish to have replies to your e-mail be sent to an e-mail address other than the one specified for the FROM e-mail address.

The address must be in the following form: ('e-mail address').

A value of 0 indicates that ReplyTo e-mail addresses are not specified.

This field is used only when an SMTP mail server is specified in your PSF configuration object and must be set to zero otherwise.

**Offset to subject** The offset to the character string to be used as the subject text. A value of 0 indicates that a subject is not specified here. If you do not specify a subject, then the default subject is obtained from message PQT4133 in message file QPQMSGF.

If you have specified an SMTP mail server, the CCSID of message text data and subject will be used to convert the data to Unicode for inclusion within the e-mail. The receiver of the e-mail may have to have the receiving e-mail program configured for the appropriate character set to be able to view the e-mail correctly.

## Integrated File System Authorizations

*Authority to directories in the path where the stream file resides*  
\*X

*Public authority to stream files*  
\*R

## Stream File Format

Use this format only when an SMTP mail server is specified in your PSF configuration object, and you are specifying the use of stream files for either the body of the e-mail or additional attachments.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Total length for stream file information
4	4	BINARY(4)	Number of stream files specified
8	8	CHAR(*)	Stream file information

## Field Descriptions

**Number of stream files specified.** The number of stream files specified within this format.

**Stream file information.** The initial offset at which information for each stream file is positioned.

See “Individual Stream File Information Format” for information about the format to use when specifying this information. Use one instance of the base format for each stream file. Each instance of this format must be contiguous to the previous one.

**Total length for stream file information.** The total length for all data used with this structure. This field is included within the length. The value specified for this entry must be a 4-byte multiple.

## Individual Stream File Information Format

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Total length of individual stream file format
4	4	BINARY(4)	Length of individual stream file format
8	8	BINARY(4)	Offset to path name

Offset		Type	Field
Dec	Hex		
12	C	BINARY(4)	Length of path name
16	10	CHAR(1)	Use-specified directory
17	11	CHAR(3)	Reserved (set to X'00')

## Field Descriptions

**Length of individual stream file format.** The length of the structure used for the individual stream file format. This value must be 20.

**Length of path name.** The length of the path name that specifies the stream file.

**Offset to path name.** The offset to the path name that specifies the stream file. The offset is from the beginning of the base format in which this value is specified.

**Use-specified directory.** Whether the directory specified for stream files used as either attachments or in the body of the e-mail is to be prefixed to the name of the stream file. For example, if you have specified '/SpecialApp/customerOne/' for a directory of files in the extension area format, this value will be prefixed to stream file names when specified using this field. Valid values are:

- 0 Do not prefix directory to this path name.
- 1 Prefix directory to this path name.

**Total length of individual stream file format.** The total contiguous length used with this format, including this field. This value is used to determine the location of the next stream file information format. The value specified for this entry must be a 4-byte multiple.

## Spooled File PDF Distribution

Use this format when a PDF is to be spooled as a PDF Spoolfile.

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	Name of output queue to spool PDF
10	A	CHAR(10)	Output queue library
20	14	CHAR(10)	Spooled File Name
30	1E	CHAR(10)	User Data
40	28	CHAR(255)	User Defined Data
295	127	CHAR(10)	Formtype

## Field Descriptions

**Formtype** The Form type to be used for the PDF spooled file.

- \*SPLF Specifies that the form type of the original spoolfile will be used.
- Formtype Specify the form type defined data.

**Name of output queue to spool PDF.** The name of the output queue that the PDF will be spooled on.

The following special values are supported for the library name:

*\*PSFCFG* Use value specified in the PSF Configuration object.  
*Output-queue* Specify the output queue name.

**Output queue library** The library in which the output queue to spool PDF exists.

*Output-queue-library* Specify the output queue library name.

**Spooled File Name** The name used for the new spooled file.

The following special values are supported for the spooled file name:

*\*SPLF* Use the original spooled file name for the new spooled file.

**User Data** The user data for the PDF spooled file. The possible values are:

*\*SPLF* Specifies that the user data of the original spoolfile will be used.  
*User-data* Specify the user data.

**User Defined Data** The user defined data for the PDF spooled file. The possible values are:

*\*SPLF* Specifies that the user defined data of the original spoolfile will be used.  
*User-data* Specify the user defined data.

## Spooled File AFPDS Distribution

Use this format when a Spoolfile segment is to be spooled as a AFP Spoolfile.

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	Name of output queue to spool AFPDS
10	A	CHAR(10)	Output queue library
20	14	CHAR(10)	Spooled File Name
30	1E	CHAR(10)	User Data
40	28	CHAR(255)	User Defined Data
295	127	CHAR(10)	Formtype

## Field Descriptions

**Formtype** The Form type to be used for the ADPDS spooled file.

*\*SPLF* Specifies that the form type of the original spoolfile will be used.  
*Formtype* Specify the form type defined data.

**Name of output queue to spool AFPDS** The name of the output queue that the AFPDS will be spooled on.

The following special values are supported for the library name:

\*PSFCFG            Use value specified in the PSF Configuration object.  
*Output-queue*        Specify the output queue name.

**Output queue library** The library in which the output queue to spool AFPDS exists.

*Output-queue-library*        Specify the output queue library name.

**Spooled File Name** The name used for the new spooled file.

The following special values are supported for the spooled file name:

\*SPLF                Use the original spooled file name for the new spooled file.

**User Data** The user data for the AFPDS spooled file. The possible values are:

\*SPLF                Specifies that the user data of the original spoolfile will be used.  
*User-data*            Specify the user data.

**User Defined Data** The user defined data for the AFPDS spooled file. The possible values are:

\*SPLF                Specifies that the user defined data of the original spoolfile will be used.  
*User-defined-data*    Specify the user defined data.

## PDF Encryption Structure

Use this format when a PDF is to be encrypted.

Offset		Type	Field
Dec	Hex		
0	0	CHAR(32)	PDF Master Password
32	20	CHAR(32)	PDF User Password
64	40	CHAR(1)	PDF Printing
65	41	CHAR(1)	PDF Document Changing
66	42	CHAR(1)	PDF Copy Document
67	43	CHAR(1)	PDF Encryption Level
68	44	CHAR(1)	PDF Content Access Enablement
69	45	CHAR(1)	PDF Changing Comments
70	46	CHAR(1)	PDF Document Assembly

## Field Descriptions

**PDF Content Access Enablement.** Specify the PDF viewer's security settings for content access to the PDF document.(Only configurable with 128-bit encryption.For 40-bit, set to '0'.)

'0'        Content Access not enabled.  
'1'        Content Access enabled

**PDF Changing Comments.** Specify the PDF viewer's security settings for adding or modifying text annotations in the PDF document.

- '0' Comments not allowed
- '1' Comments allowed

**PDF Copy.** Specify the PDF viewer's security settings for copying from the PDF document.

- '0' Copy not allowed
- '1' Copy allowed

**PDF Document Assembly.** Specify the PDF viewer's security settings for document assembly from the PDF document.*(Only configurable with 128-bit encryption. For 40-bit, set to '0'.)*

- '0' Assembly not allowed
- '1' Assembly allowed

**PDF Document Changing.** Specify the PDF viewer's security settings for changing the PDF document.

- '0' Changing not allowed
- '1' Changing allowed

**PDF Encryption Level.** Specify the encryption level the PDF document using PDF Viewer security settings.

- '1' 40-bit
- '2' 128-bit

**PDF Master Password.** A string of up to 32 alphanumeric letters and/or numbers. Acceptable characters are A-Z, a-z and 0-9. X'00' will indicate that there is no password on this document.

\*NONE No master password specified.

**PDF Printing.** Specify the PDF viewer's security settings for printing the PDF document

- '0' Not Allowed
- '1' Allowed
- '2' Low Resolution Printing.*(Only configurable with 128-bit encryption.)*

**PDF User Password.** A string of up to 32 alphanumeric letters and/or numbers. Acceptable characters are A-Z, a-z and 0-9. X'00' will indicate that there is no password on this document.

\*NONE No user password specified.

Exit program introduced: V5R1

Top | "Print APIs," on page 1 | APIs by category

---

## Writer Transform Exit Program

Required Parameter Group:

1	Process option	Input	Binary(4)
2	Option specific input information to transform exit	Input	Char(*)
3	Length of option specific input information to transform exit	Input	Binary(4)
4	Spoiled file data buffer	Input	Char(*)
5	Length of spoiled file data buffer	Input	Binary(4)
6	Option specific output information from transform exit	Output	Char(*)
7	Length of option specific output information from transform exit	Input	Binary(4)
8	Length of option specific output information available from transform exit	Output	Binary(4)
9	Transformed data buffer	Output	Char(*)
10	Length of transformed data buffer	Input	Binary(4)
11	Length of transformed data available	Output	Binary(4)

QSYSINC Member Name: ESPTRNXT

The Writer Transform exit program describes the interface between a user-defined transform exit program (transform exit) and an i5/OS printer writer (writer).

**Note:** This exit interface can also be used for remote writers and on the Send TCP/IP Spooled File (SNDTCPSPLF or LPR) command.

A transform exit program can be enabled by an operator when a writer is started to an output queue. The writer handles the scheduling of spooled files to print, and relies on the transform exit program to complete transforming of the data necessary to print the file. The data is processed by the writer as DEVTYPE(\*USERASCII).

For informational purposes, the example below is provided to show the interaction between the printer writer and the transform exit program.

1. Printer writer is started by an operator.

This ties a specific output queue to a specific printer. In this case, the printer is capable of accepting the Printer Control Language (PCL) data stream.
2. The writer calls the transform exit program with the initialize (10) process option.

The transform exit program creates its work spaces and returns to the writer a successful return code.
3. The writer selects the first eligible file to process.

The writer calls the transform exit program with the process file (20) process option. The name and identifier of the spooled file are passed in the input information.  
The transform exit program determines if it will transform the file, returns to the writer a successful return code and an indication whether it will transform the file.
4. The writer calls the transform exit program with the transform data process option (30).

The writer calls the transform exit program, passing the data to transform. The transform exit program transforms the data and passes it back to the writer. This step happens repeatedly, until the entire spooled file has been passed to the transform exit program.
5. The writer calls the transform exit program with the end file process option (40).

The transform exit program returns to the writer any remaining data to be sent to the printer.
6. The writer selects the next eligible spooled file from the output queue.

Steps starting with the process file option are repeated.
7. The operator ends the printer writer.

The writer calls the transform exit program with the terminate process option (50). The transform exit program cleans up any work spaces that it has created and returns to the writer.

The transform program should be capable of handling the case where the program could lose control while processing a spooled file.

**Note:** This could happen when the spooled file being processed is held using the Hold Spooled File (HLDSPLF) command, deleted using the Delete Spooled File (DLTSPLF) command, or a restart page specified using the Change Spooled File Attributes (CHGSPLA) command.

When this happens, the transform program will be called with the end file (40) process option. The transform program is expected to be able to continue processing normally.

## Authorities and Locks

None.

## Required Parameter Group

### Process option

INPUT; BINARY(4)

The action for the transform exit program to take. This information is filled in by the writer. The first option should be initialize (10). This is done when the writer is started.

Possible values are:

- |                              |  |
|------------------------------|--|
| 10 ( <i>Initialize</i> )     | The transform exit program sets any initial information with this option.  |
| 20 ( <i>Process file</i> )   | A spooled file is available to print. The spooled file name and other information are passed in the Option specific input information parameter to the transform exit program. The transform exit program decides whether to transform the data and sets the Transform file field in the output information structure. The transform exit may pass data to the writer. This data is treated as open time commands and is sent to the printer before the spooled file is printed. |
| 30 ( <i>Transform data</i> ) | The writer calls the transform exit program after reading one or more buffers of data from the spooled file. The data is in the Spooled file data buffer. This process is repeated for a spooled file until all data has been processed. Alignment data could also be returned.  |
| 40 ( <i>End file</i> )       | The writer calls the transform exit to inform the transform exit that all data for the spooled file has been passed. No data is passed on this call. The transform exit may pass additional data to the writer as an append to the end of the printed data.  |
| 50 ( <i>Terminate</i> )      | This option is used even if the transform exit did not transform the spooled file.<br>The writer is ending. The transform exit will do cleanup.  |

### Option specific input information to transform exit

INPUT; CHAR(\*)

Information which is input to the transform exit program from the writer. For the format and description of this information, see "Option Specific Input Information to Transform Exit Program" on page 427.

### Length of option specific input information to transform exit

INPUT; BINARY(4)

The length of the option specific input information specified in the Option specific input information to transform exit program parameter.

### Spooled file data buffer

INPUT; CHAR(\*)

The spooled file data to be transformed by the transform exit program.

The space is created and destroyed by the writer.

**Length of spooled file data buffer**

INPUT; BINARY(4)

The length of the spooled file data specified in the Spooled file data buffer parameter.

**Option specific output information from transform exit**

OUTPUT; CHAR(\*)

Information which is output from the transform exit program to the writer. For the format and description of this information, see "Option Specific Output Information From Transform Exit" on page 431.

**Length of option specific output information from transform exit**

INPUT; BINARY(4)

The size, in bytes, of the option specific output information buffer.

**Length of option specific output information available from transform exit**

OUTPUT; BINARY(4)

The length of the option specific output information available. If this length is less than or equal to the length of the option specific output information, all available information has been returned. If this length is greater than the length of the option specific output information, only data up to the length provided is returned in the option specific output information buffer.

**Transformed data buffer**

OUTPUT; CHAR(\*)

Spooled file data which has been transformed by the transform exit program. The data in this buffer will be sent by the writer to the printer. It will not be validity checked or syntax checked by the writer. If the print data is not a valid data stream for the device, errors will be returned from the device. The data is processed by the writer as DEVTYPE(\*USERASCII).

Alignment data, if requested, is also returned in the transformed data buffer. The lengths and offsets of the alignment information is returned in the option specific output information. For the format and description of this information, see "Option Specific Output Information From Transform Exit" on page 431.

The space is created and destroyed by the writer.

**Length of transformed data buffer**

INPUT; BINARY(4)

The size, in bytes, of the transformed data buffer.

**Length of transformed data available**

OUTPUT; BINARY(4)

The length of the transformed data that is available to be returned in the transformed data buffer. If this length is less than or equal to the length of the transform data buffer, all available information has been returned. If this length is greater than the length of the transformed data buffer, only data up to the length provided was returned in the transformed data buffer.

Parameters Used by the Printer Writer or Transform Exit Program (page 426) presents, in the same order, all the required parameters previously described. Each entry indicates if a parameter is to be used by the writer or transform exit program. Possible values are Y and N.

**Parameters Used by the Printer Writer or Transform Exit Program**

Parameter	Initialize (10)	Process file (20)	Transform data (30)	End file (40)	Terminate (50)	Description
Process option	10	20	30	40	50	

Parameter	Initialize (10)	Process file (20)	Transform data (30)	End file (40)	Terminate (50)	Description
Option specific input information to transform exit	Y	Y	Y	Y	Y	See description of this parameter.
Length of option specific input information	Y	Y	Y	Y	Y	Length of option specific input information
Spoiled file data buffer	N	N	Y	N	N	Contains data only on spoiled data to transform option.
Length of spoiled file data buffer	N	N	Y	N	N	The length of spoiled data buffer.
Option specific output information from transform exit	Y	Y	Y	Y	Y	See description of this parameter.
Length of option specific output information	Y	Y	Y	Y	Y	Length of option specific output information
Length of option specific output information available	Y	Y	Y	Y	Y	The number of bytes of option specific output information that is available. If this is greater than the length of the buffer, only information that can fit in the option specific output information buffer will be returned.
Transformed data buffer	N	Y	Y	Y	N	This buffer can contain data to send to the device before the transformed data (Start of file option) or after the transformed data (End of file option), as well as transformed data (Spoiled data to transform option).
Length of transformed data buffer	N	Y	Y	Y	N	The size, in bytes, of the transformed data buffer.
Length of transformed data available	N	Y	Y	Y	N	The number of bytes of transformed data that is available.

## Option Specific Input Information to Transform Exit Program

The following table shows the structure for the option specific information input parameter. For more details about the fields in the following table, see "Field Descriptions" on page 428.

Offset		Type	Field
Dec	Hex		
0	0	CHAR(16)	Writer handle
16	10	CHAR(10)	Writer name
26	1A	CHAR(10)	Printer device name
36	24	CHAR(10)	Writer output queue name

Offset		Type	Field
Dec	Hex		
46	2E	CHAR(10)	Writer output queue library name
56	38	CHAR(10)	Writer message queue name
66	42	CHAR(10)	Writer message queue library name
76	4C	CHAR(10)	Reserved
86	56	CHAR(10)	Spooled file handle
96	60	CHAR(16)	Internal job identifier
112	70	CHAR(16)	Internal spooled file identifier
128	80	CHAR(26)	Qualified job name
154	9A	CHAR(10)	Spooled file name
164	A4	BINARY(4)	Spooled file number
168	A8	CHAR(12)	Reserved
180	B4	BINARY(4)	End file type
184	B8	BINARY(4)	Termination type
188	BC	CHAR(10)	Current writer form type
198	C6	CHAR(1)	Return alignment data
199	C7	CHAR(5)	Reserved
204	CC	BINARY(4)	Number of complete pages
208	D0	CHAR(10)	Workstation customizing object name
218	DA	CHAR(10)	Workstation customizing object library
228	E4	CHAR(15)	Manufacturer type and model
243	F3	CHAR(31)	Reserved
274	112	CHAR(8)	Job system name
282	11A	CHAR(7)	Spooled file create date
289	121	CHAR(1)	Reserved
290	122	CHAR(6)	Spooled file create time

## Field Descriptions

**Current writer form type.** The form type the writer is currently printing.

This field is set to blanks when the transform exit program is called from the Send TCP/IP Spooled File (SNDTCPSPLF or LPR) command.

**End file type.** The type of ending for the file.

Possible values are:

- 1 (*normal*)            Processing of the file ends normally.
- 2 (*immediate*)        Processing of the file ends immediately.
- 3 (*page end*)         Processing of the file ends at the end of the current page.

**Internal job identifier.** The internal job identifier of the job that created the spooled file.

**Internal spooled file identifier.** The internal spooled file identifier of the spooled file being processed.

**Job system name.** The name of the system where the job that created the spooled file ran.

**Manufacturer type and model.** The manufacturer, type and model for a printer using transform support.

**Number of complete pages.** The number of complete pages that are contained in the data to be transformed.

**Printer device name.** The name of the printer device.

**Qualified job name.** The qualified job name of the job that created the spooled file.

**Reserved.** This field is reserved.

**Return alignment data.** Whether the transform should return data to the caller which can be used for alignment purposes.

Possible values are:

- 0 Do not return alignment information.
- 1 Return alignment information.

**Spooled file create date.** The date the spooled file was created on the system in the CYYMMDD format. See field Date file opened in API QUSRSPLA under field descriptions for more information on the date format.

**Spooled file create time.** The time the spooled file was created on the system in the HHMMSS format. See field Time file opened in API QUSRSPLA under field descriptions for more information on the time format.

**Spooled file handle.** The handle for the spooled file to be printed.

This field is set to blanks when the transform exit program is called from the Send TCP/IP Spooled File (SNDTCPSPLF or LPR) command.

**Spooled file name.** The name of the spooled file being processed.

**Spooled file number.** The number of the spooled file being processed.

**Termination type.** The type of termination by the writer:

- 1 (*normal*) The writer ended normally.
- 2 (*immediate*) The writer ended immediately.
- 3 (*abnormal*) The writer ended abnormally.

**Workstation customizing object library.** The library containing the workstation customizing object.

**Workstation customizing object name (WSCST).** The name of an object that consists of a table of attributes used to customize a given ASCII device.

**Writer handle.** The handle to the writer job.

This field is set to blanks when the transform exit program is called from the Send TCP/IP Spooled File (SNDTCPSPLF or LPR) command.

**Writer message queue library name.** The name of the library in which the writer message queue resides.

This field is set to blanks when the transform exit program is called from the Send TCP/IP Spooled File (SNDTCPSPLF or LPR) command.

**Writer message queue name.** The name of the message queue to which the writer issues messages.

This field is set to blanks when the transform exit program is called from the Send TCP/IP Spooled File (SNDTCPSPLF or LPR) command.

**Writer name.** The name of the writer.

This field is set to blanks when the transform exit program is called from the Send TCP/IP Spooled File (SNDTCPSPLF or LPR) command.

**Writer output queue library name.** The name of the library in which the output queue resides.

**Writer output queue name.** The name of the output queue from which the writer is processing spooled files.

Fields Used by the Option Specific Input Information Parameter (page 430) presents, in the same order, all the fields previously described in the option specific input information table. Each entry indicates if a field is defined for a particular option. Possible values are Y and N.

#### Fields Used by the Option Specific Input Information Parameter

Field	Initialize (10)	Process file (20)	Transform data (30)	End file (40)	Terminate (50)	Description
Writer handle	Y	Y	Y	Y	Y	For use with writer APIs.
Writer name	Y	Y	Y	Y	Y	Name specified on STRPRTWTR command.
Printer device name	Y	Y	Y	Y	Y	Device name of the printer.
Writer output queue name	Y	Y	Y	Y	Y	The name of the output queue that the writer is processing.
Writer output queue library name	Y	Y	Y	Y	Y	The library in which the writer output queue resides.
Writer message queue name	Y	Y	Y	Y	Y	Name of the message queue for the started writer.
Writer message queue library name	Y	Y	Y	Y	Y	Name of the library in which the message queue resides.
Reserved	N	N	N	N	N	Reserved to preserve consistency with the print driver exit.
Spooled file handle	N	Y	Y	Y	N	For use with future APIs.
Internal job identifier	N	Y	Y	Y	N	For use with existing Spool APIs.
Internal spooled file identifier	N	Y	Y	Y	N	For use with existing Spool APIs.
Qualified job name	N	Y	Y	Y	N	Job owning spooled file
Spooled file name	N	Y	Y	Y	N	Name of spooled file being processed

Field	Initialize (10)	Process file (20)	Transform data (30)	End file (40)	Terminate (50)	Description
Spooled file number	N	Y	Y	Y	N	Number of spooled file being processed.
Reserved	N	N	N	N	N	Reserved to preserve consistency with the print driver exit.
End file type	N	N	N	Y	N	Type of end for spooled file.
Termination type	N	N	N	N	Y	The type of termination by the writer.
Current writer form type	N	Y	Y	Y	N	Form type writer is currently processing.
Return alignment information	N	Y	Y	N	N	Returns alignment information along with transformed data.
Reserved	N	N	N	N	N	Reserved to preserve consistency with the print driver exit.
Number of complete pages	N	N	Y	N	N	Specifies the number of complete pages in the data to be transformed.
Workstation customizing object name	N	Y	N	N	N	The name of an object that consists of a table of attributes used to customize a given ASCII device.
Workstation customizing object library	N	Y	N	N	N	The library containing the workstation customization object.
Manufacturer type and model	N	Y	N	N	N	The manufacturer, type and model for a printer using transform support.
Reserved	N	N	N	N	N	Reserved to maintain compatibility with the Host Print Transform (QWPZHPTR) API.
Job system name	N	Y	Y	Y	N	Name of system job is located.
Spooled file create date	N	Y	Y	Y	N	Date the spooled file was created.
Reserved	N	N	N	N	N	Reserved to maintain compatibility with the Host Print Transform (QWPZHPTR) API.
Spooled file create time	N	Y	Y	Y	N	Time the spooled file was created.

## Option Specific Output Information From Transform Exit

The following table shows the structure for the option specific information output parameter. This structure is used to pass information from the transform exit program to the writer. For more details about the fields in the following table, see “Field Descriptions” on page 432.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Return code
4	4	CHAR(1)	Transform file
5	5	CHAR(1)	Pass input data
6	6	CHAR(1)	Send single copy
7	7	CHAR(1)	Send open time commands
8	8	CHAR(1)	Done transforming file
9	9	CHAR(3)	Reserved
12	C	BINARY(4)	Offset to vertical positioning commands
16	10	BINARY(4)	Length of vertical positioning commands
20	14	BINARY(4)	Offset to print data
24	18	BINARY(4)	Length of print data
28	1C	BINARY(4)	Offset to carriage return commands
32	20	BINARY(4)	Length of carriage return commands
36	24	BINARY(4)	Offset to form feed commands
40	28	BINARY(4)	Length of form feed commands
44	2C	CHAR(*)	Carriage return commands
		CHAR(*)	Form feed commands

## Field Descriptions

**Carriage return commands.** The carriage return commands contain the data stream commands which can be used to do end of line processing for alignment purposes.

**Done transforming file.** Indicates the transform exit program requires no further data to process the current file. This field is ignored when the pass input data field is set to 0.

Possible values are:

- 0 Continue calling the transform exit program with the transform data (30) option.
- 1 The next call to the transform exit program will be with the end file (40) option.

**Form feed commands.** The form feed commands contain the data stream commands which can be used to do end of page processing for alignment purposes.

**Length of carriage return commands.** The length, in bytes, of the carriage return commands. The carriage return commands are returned in the output specific information.

**Length of form feed commands.** The length, in bytes, of the form feed commands. The form feed commands are returned in the output specific information.

**Length of print data.** The length, in bytes, of the print data for the first line of data on the page. This includes any horizontal positioning which is done in the first line.

**Length of vertical positioning commands.** The length, in bytes, of the vertical positioning commands that are returned in the transformed data buffer.

**Offset to carriage return commands.** Offset to the carriage return commands in the output specific information. The carriage return commands contain the data stream commands which can be used to do end of line processing for alignment purposes.

**Offset to form feed commands.** Offset to the form feed commands in the output specific information. The form feed commands contain data stream commands which can be used to do end of page processing for alignment purposes.

**Offset to print data.** Offset to the first line of the print data for the page in the transformed data buffer. This first line can be printed for alignment purposes.

**Offset to vertical positioning commands.** Offset to the vertical positioning commands in the transformed data buffer.

**Pass input data.** Whether the writer passes the input data to the transform exit program.

Possible values are:

- 0 The writer will pass the input data to the transform exit program.
- 1 The writer will not pass the input data to the transform exit program.

**Return code.** The return code returned by the transform exit after each call.

Possible values are:

- 0 No error occurred.
- non zero* An error occurred.

When the return code indicates that an error has occurred, the following is performed by the writer:

- Initialize error  
The next call to the transform exit program is the terminate process option.
- Start of file error  
The next call to the transform exit program is the end file process option.
- Spooled file data transform error  
The next call to the transform exit program is the end file process option.
- End of file error  
The next call to the transform exit program is the terminate process
- Terminate  
There are no additional calls to the transform exit program for this printer writer.

**Send open time commands.** Indicates if the open time commands returned by the transform exit program on the process file (20) call are sent before the spooled file prints.

This value is ignored when transform file is set to 0 or 1. In this case, the data returned on the process file (20) call will always be sent before the spooled file prints.

- 0 The printer writer decides if the open time commands should be sent
- 1 Send the open time commands before this spooled file prints. The commands vary with the spooled file attributes.
- 2 Do not send any open time commands.

**Send single copy.** Indicates if the transform exit program is called with process options 20, 30, and 40 for each copy requested.

- 0 The transform exit program is called for each copy requested to print.
- 1 The transform exit program is called with the 20, 30, and 40 process option sequence only once for each spooled file.

**Transform file.** Whether the spooled file data will be transformed.

- 0 The spooled file data cannot be transformed.
- 1 The spooled file data will be transformed.
- 2 The spooled file data is in its final form.

Fields Used by the Option Specific Output Information Parameter (page 434) presents, in the same order, all the fields in previously described in the option specific output information table. Each entry indicates if a field is used with a particular process option. Possible values are Y and N.

**Fields Used by the Option Specific Output Information Parameter**

Field	Initialize (10)	Process file (20)	Transform data (30)	End file (40)	Terminate (50)	Description
Return code	Y	Y	Y	Y	Y	Indicates if the transform exit program was successful in completing the specified option.
Transform file	N	Y	N	N	N	Allows the transform exit program to end the processing of a file, with either the Process File option.
Pass input data	N	Y	N	N	N	Allows the transform exit program to specify whether the writer will provide the input data or the transform exit program will read the data directly from the file.
Send single copy	N	Y	N	N	N	Allows the transform exit program to convert a file once, even if there are multiple copies requested. In the case of multiple copies requested, the transform exit program inserts the proper commands to print multiple copies.
Send open time commands	N	Y	N	N	N	Allows the transform exit program to specify if the system-generated initial (open time) commands are sent to the printer before the file prints.
Done transforming file	N	N	Y	N	N	For the spooled file currently being processed, it allows the transform exit program to stop the writer from calling the transform exit program with any additional process option 30 calls.

Field	Initialize (10)	Process file (20)	Transform data (30)	End file (40)	Terminate (50)	Description
Reserved	N	N	N	N	N	Reserved to preserve consistency with the print driver exit.
Offset to vertical positioning commands	N	N	Y	N	N	The offset to the vertical positioning commands in the transformed data buffer. When alignment information has been requested, vertical positional commands are also returned in the transformed data buffer.
Length of vertical positioning commands	N	N	Y	N	N	The length of the vertical positioning commands that are returned the transformed data buffer. This can be used in alignment processing.
Offset to print data	N	N	Y	N	N	The offset to the print data for the first line of the page. This can be used for alignment processing.
Length of print data	N	N	Y	N	N	The length of print data for the first line of data for a page. This can be used for alignment processing.
Offset of carriage return commands	N	N	Y	N	N	Offset to the carriage return commands in the output specific information. The carriage return commands are data stream commands that can be used to do end of line processing for alignment purposes.
Length of carriage return commands	N	N	Y	N	N	The length of the carriage return commands.
Offset of form feed commands	N	N	Y	N	N	Offset to the form feed commands in the output specific information. The form feed commands are data stream commands that can be used to do end of page processing for alignment purposes.
Length of form feed commands	N	N	Y	N	N	The length of the form feed commands.
Carriage return commands	N	N	Y	N	N	The carriage return commands are data stream commands that can be used to do end of line processing for alignment purposes.
Form feed commands	N	N	Y	N	N	The form feed commands are data stream commands that can be used to do end of page processing for alignment purposes.

Exit program introduced: V3R7

---

## UIM Spooled File User-defined List Action Exit Program

Required Parameter Group:

1	Exit point name	Input	Char(20)
2	Format	Input	Char(8)
3	User-defined action	Input	Char(1)
4	Spooled file identity	Input	Char(*)
5	Length of spooled file identity	Input	Binary(4)

Exit Point Name: QIBM\_QSP\_SPLF\_LSTACT

Exit Point Format Name: "LASP0100 Format" on page 437

QSYSINC Member Name: espxpts

The UIM Spooled File User-defined List Action exit point can be used to define additional list actions on the following green screen interfaces:

- Work with Printer Output (WRKSPLF ASTLVL(\*BASIC))
- Work with Spooled Files (WRKSPLF ASTLVL(\*INTERMED))
- Work with Job Spooled Files (WRKJOB OPTION(\*SPLF))
- Work with Output Queue (WRKOUTQ *output queue name*)
- Work with Spooled File Status (D P or WRKSPLF DSPFMT(\*S36FMT))

An exit program that takes the user-defined list action and spooled file identity as input and performs the desired action must be registered to this exit point by using the Add Exit Program (QUSADDEP, QusAddExitProgram) API or the ADDEXITPGM command.

### Authorities and Locks

None.

### Required Parameters

#### Exit point name

INPUT; CHAR(20)

The name of the registered exit point. The value of this parameter will always be QIBM\_QSP\_SPLF\_LSTACT.

#### Format

INPUT; CHAR(8)

The name of the format describing the spooled file identity information passed to the exit program. The value of this parameter will always be LASP0100.

#### User-defined action

INPUT; CHAR(1)

The user-defined action specified by the user on the UIM list panel.

#### Spooled file identity

INPUT; CHAR(\*)

The identity of the spooled file on which the user-defined action is to be performed. The identity of the spooled file consists of the qualified job name, spooled file name, spooled file number, job system name, create date and time, output queue name and output queue library of the spooled file.

## Spooled file identity length

INPUT; BINARY(4)

The length of the spooled file identity information in number of bytes.

## LASP0100 Format

The following table shows the format of the LASP0100 structure for the Spooled file identity parameter. For more details about the fields in the following table, see “Field Descriptions.”

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	Spooled file job name
10	A	CHAR(10)	Spooled file user name
20	14	CHAR(6)	Spooled file job number
26	1A	CHAR(10)	Spooled file name
36	24	BINARY(4)	Spooled file number
40	28	CHAR(8)	Job system name
48	30	CHAR(7)	Spooled file create date
55	37	CHAR(6)	Spooled file create time
61	3D	CHAR(10)	Output queue name
71	47	CHAR(10)	Output queue library name

## Field Descriptions

**Job system name.** The name of the system where the job that created this spooled file ran.

**Output queue name.** The name of the output queue where the spooled file is located.

**Output queue library name.** The name of the library that contains the output queue.

**Spooled file create date.** The date the spooled file was created on the system in the CYYMMDD format. See field Date file opened in API QUSRSPPL under field descriptions for more information on the date format.

**Spooled file create time.** The time the spooled file was created on the system in the HHMMSS format. See field Time file opened in API QUSRSPPL under field descriptions for more information on the time format.

**Spooled file job name.** The name of the job that owns the spooled file.

**Spooled file job number.** The job number of the job that owns the spooled file.

**Spooled file name.** The name of the spooled file being processed.

**Spooled file number.** The number of the spooled file being processed.

**Spooled file user name.** The user that owns the spooled file.

◀ Exit program introduced: V6R1

---

## Concepts

These are the concepts for this category.

---

### Spool and Print Tools in Library QUSRTOOL

There are tools in QUSRTOOL that use some of the spooled file and print APIs. To show how to use these APIs, source code for these tools is provided in the examples.

These tools use most of the spooled file and print APIs discussed in this part. Additionally, some CL commands are used in the tools.

The tools are:

- Save and restore spooled files (SAVRSTSPLF) tool

This tool uses the following spooled file APIs:

- QSPCLOSP
- QSPCRTSP
- QSPGETSP
- QUSLSPL
- QUSRSPLA
- QSPOPNSP
- QSPPUTSP

It also makes use of the following save and restore APIs:

- QSRSAVO
- QSRRSTO

- Transform AFP™ to ASCII data stream (TRNAFP) tool

This tool uses the QWPZTAFP print API.

- Customized separator page exit program (QSPBLSEP) example

This example of the customized print separator exit program builds program QSPBLSEP. QSPBLSEP can be used to generate data for separator pages.

- Page range support exit program (TSPRWPR) example

This example of the Writer Transform exit program provides page range support for remote writers and CL commands Send TCP/IP Spooled File (SNDTCPSPLF) and LPR.

- LPR Print driver exit program (TSPLPRD) example

This example of the Print Driver exit program provides the function of sending spooled files from an output queue to an ASCII printer device which is attached to an LPD server.

A description of the SAVRSTSPLF tool can be found in member TSRINFO. A description of TRNAFP can be found in member TWPINFO. A description of QSPBLSEP can be found in member TBSINFO. A description of TSPRWPR can be found in member TSPRWPRI. A description of TSPLPRD can be found in member TSPLPRDI. All members are associated with file QATTINFO, library QUSRTOOL. File QATTINFO also contains information about how to use QUSRTOOL.

Top | “Print APIs,” on page 1 | APIs by category

---

## Appendix. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation  
Licensing  
2-31 Roppongi 3-chome, Minato-ku  
Tokyo 106-0032, Japan

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:** INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation  
Software Interoperability Coordinator, Department YBWA  
3605 Highway 52 N  
Rochester, MN 55901  
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, IBM License Agreement for Machine Code, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

#### COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. \_enter the year or years\_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

---

## Programming interface information

This API descriptions publication documents intended Programming Interfaces that allow the customer to write programs to obtain the services of IBM i5/OS.

---

## Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

Advanced 36  
Advanced Function Presentation  
Advanced Peer-to-Peer Networking  
AFP  
AIX  
AnyNet  
AS/400  
BCOCA  
C/400  
COBOL/400  
Common User Access  
CUA  
DB2  
DB2 Universal Database  
Distributed Relational Database Architecture  
Domino  
DPI  
DRDA  
Enterprise Storage Server  
eServer  
FlashCopy  
GDDM  
i5/OS  
IBM  
IBM (logo)  
InfoColor  
Infoprint  
Integrated Language Environment  
Intelligent Printer Data Stream  
IPDS  
Lotus  
Lotus Notes  
MO:DCA  
MVS  
Net.Data  
NetServer  
Notes  
OfficeVision  
Operating System/2  
Operating System/400  
OS/2  
OS/400  
PartnerWorld  
POWER5+  
PowerPC  
Print Services Facility  
PrintManager  
PROFS  
RISC System/6000  
RPG/400  
RS/6000

SAA  
SecureWay  
SOM  
System i  
System i5  
System Object Model  
System/36  
System/38  
System/390  
TotalStorage  
VisualAge  
WebSphere  
xSeries  
z/OS

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

---

## Terms and conditions

Permissions for the use of these publications is granted subject to the following terms and conditions.

**Personal Use:** You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative works of these publications, or any portion thereof, without the express consent of IBM.

**Commercial Use:** You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER

EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.







Printed in USA