





Reference Material

Study of DP-DS820/DP-DS820 (A) System Realization

(Single Printer)

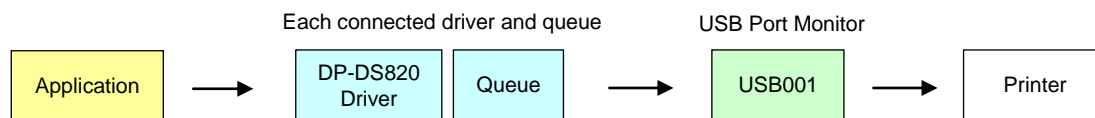
Mar. 18, 2016
Dai Nippon Printing Co., Ltd.

-  The copyrights for this document are the property of the holders of rights. Reproduction of any or all of the contents is prohibited.
-  The contents of this document are subject to change without prior notice.
-  Company and product names used in this document are trademarks or registered trademarks of their respective companies.
-  You cannot resale/use the DP-DS820 without destination number (A) in the U.S.

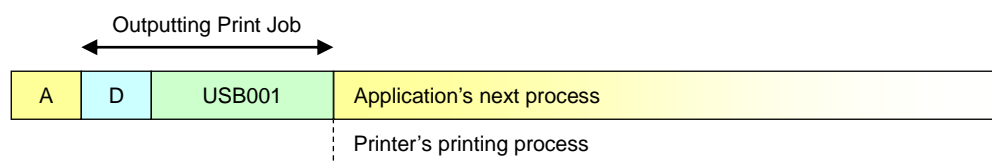
1. About the Data Flow on Windows




In the printer driver settings, there is a Spool On/Off setting. The data flow when using this setting is shown below.

1-1. Operation with Spool Off

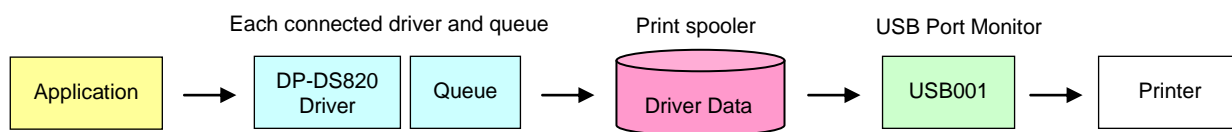


When the application outputs data to the driver, the driver can't release the print job until the data is output to the USB port. Therefore, the application waits until data output to the USB port is complete before performing the next process.

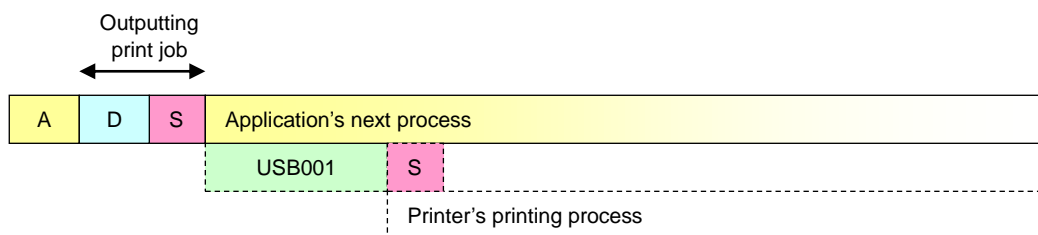


-  Application section: data resizing, status processing (API), sheet no. control, color management, color adjust, etc
-  Driver processing: color management, color adjust, printer command processing, etc.
-  USB data transferring, data send/receive management

1-2. Operation with Spool On



When the application outputs the data to the driver, the driver outputs the data to the print spooler and releases the print job. At this point, the application can begin the next process without waiting for the data to be output to the USB port. When the print spooler takes in the data, USB Port Monitor control is performed, and the data output to the port begins.



- A** Application section: data resizing, status processing (API), sheet no. control, color management, color adjust, etc
- D** Driver processing: color management, color adjust, printer command processing, etc.
- S** File writing to and deletion from the print spooler
- USB** USB data transferring, data send/receive management

[About status check]

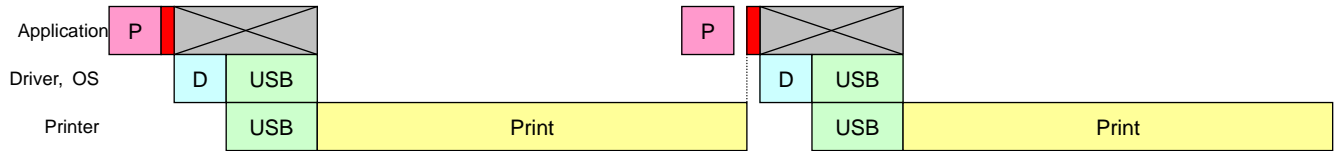
With Spool ON, the application can perform the next process when the data output to the spooler is complete, but until the data output to the port is complete, status confirmation for the printer cannot be done.

2. Relation between printer buffer operation (single/double) and Spool ON/OFF

The relationship between printer buffer operation (single/double) and Spool ON/OFF is shown below.

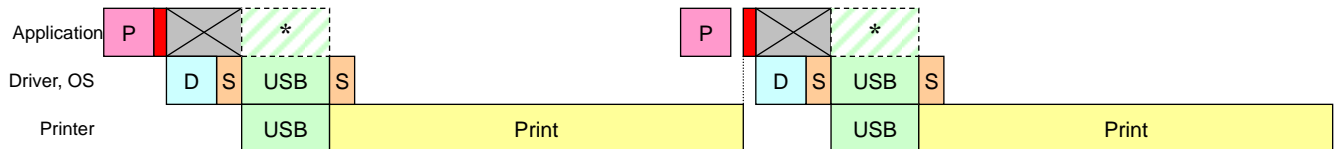
(1) Single buffer, Spool OFF

Because it is Spool OFF, it waits for the data to be sent. Moreover, it waits for printing to be complete before sending the next data.



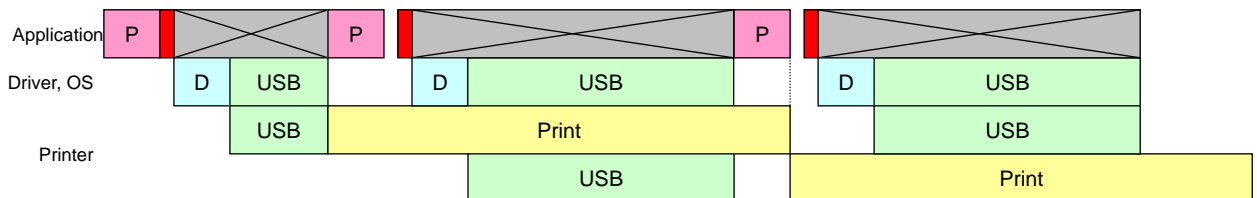
(2) Single buffer, Spool ON

By setting Spool ON, the application is freed from the print processing earlier, but with the spool processing time, the total processing time is slightly longer than with Spool OFF.



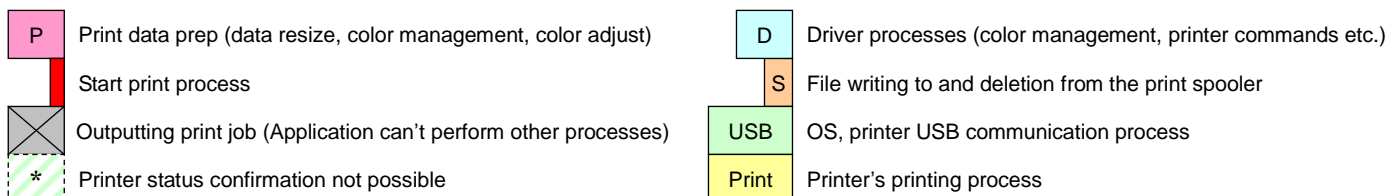
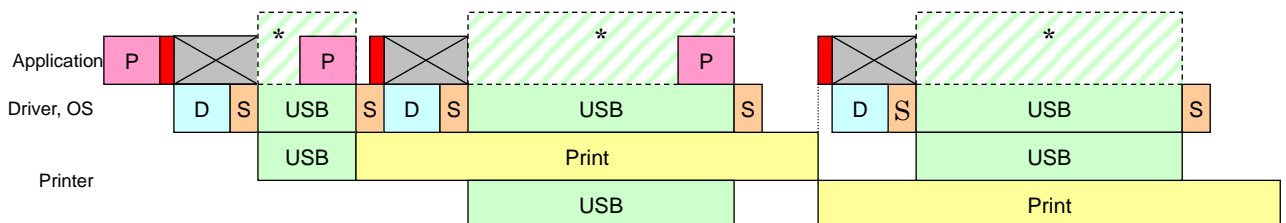
(3) Double buffer, Spool OFF

Using a double buffer, while the first image is being printed, data for the second can be sent, so processing time is that much shorter. However, with Spool OFF, the application cannot perform any other processes while the print job is being sent to the printer.



(4) Double buffer, Spool ON

Double buffer processing allows data to be sent while printing, and with Spool ON, the application can perform other processes (like preparing the next print data), except for printer status confirmation, even while the print job is being sent to the printer.



3. System Composition

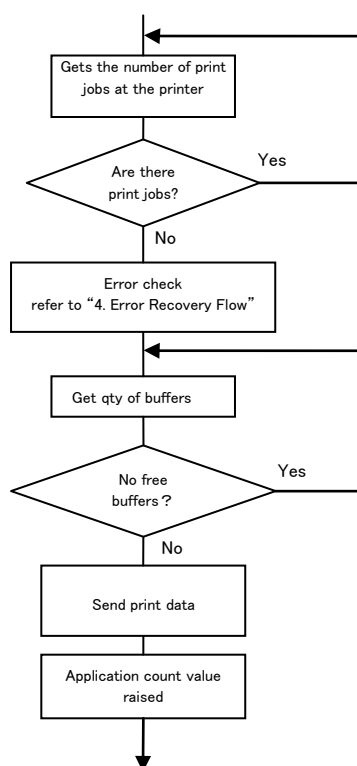
- Connected printers : 1 Unit
- Printer operating mode : Double buffer operation
- Spool setting : Spool ON



[Sample Flow]

In the overall process, when a print request is received, it checks the printer to see if data can be sent, and then it sends the data. Details of the process are listed below.

- ① When a print request is received, it checks to see if there is a print job at the printer (WindowsAPI EnumJobs). If there is a job, it repeats the check process.
- ② It performs an error check. If there is an error, it performs the error recovery process. (for details, refer to “2. Error Recovery Flow”)
- ③ If there is no error, it gets the number of free buffers at the printer (status API GetFreeBuffer).
- ④ If there are no free buffers, it repeats the check for free buffers.
- ⑤ If there are 1 or more free buffers, it sends the print data to the printer.
- ⑥ After the print data is sent, the application’s print count value is raised.



4. Error Recovery Flow

When an error occurs, it's necessary to find the image that was being printed at that time to find out the right image(s) to be recovered, based on the printer's counter (GetCounterL in the Status API).

- (1) Checks for print jobs.
- (2) If there are no print jobs, gets the status of the printer and checks for any errors.
- (3) If there is any error found, gets counter value of the printer.
- (4) Compares the value of the printer's counter and the counter of the application side.
- (5) If the difference of the two counter values is 2, recovers the two previous images. (In a multi-printer system, sends the image to be reprinted to another printer that is ready. If it is a single printer, sends the image to be reprinted after the printer error is solved.) In the case when the difference is 1, recovers the previous image.

[Note]

- In the case where a print job is stacked or is not released for a long time, it's possible there is a communication error, and it may be necessary to go into the error recovery process.
- If there is no print job and communication is suspended for some reason, a failure code will be returned approx. 1 minute after sending the DP-DS820 printer status API to get the printer status. (During this time, the application is idling, waiting for the status to be returned) So, when confirming the status, it's recommended to check the time from the start of the status check to the end. If the time in between is too long, it's necessary to go into the error recovery process.

