11 Programming Tips for Using PJL

Introduction

When using most printer languages, there is usually more than one way to perform a certain printing function. PJL is no exception. Properly formed PJL jobs provide the developer with the desired features while eliminating glitches. This chapter provides tips for creating efficient PJL jobs that eliminate potential problems.

Note Improperly coded print jobs may work fine in a single-computer/ single-printer environment running only one application, but cause problems in a multi-application or shared network environment.

The first part of this chapter categorizes PJL applications into three general types, and provides guidance and examples for each type. The remainder of the chapter provides hints for handling specific situations, including using status readback in a multi-user system.

PJL Services

This section segments applications into three types based on the services provided. Applications using PJL can be categorized as one of the following:

- **Print Data Generators**—Applications that generate print jobs, such as Lotus 1-2-3, Microsoft Word, and WordPerfect.
- **Printer Utilities**—Applications that control printer features, or request and provide printer status. These applications also can monitor the current state of the printer.
- **Spoolers**—Applications that control print jobs generated by other applications.

Many applications provide more than one of the services listed above. For example, a Spooler can be used to monitor the printer and provide control over printer features, and print jobs. If an application does provide more than one service, only use those PJL commands appropriate for the provided services.

If applications providing different printer services follow the guidelines described in this section, they can work together properly. This chapter lists the commands used by each application type, explains the functions of each command, and offers several examples illustrating different PJL functions.

Print Data Generators

Print Data Generators are applications used to create and print information. Examples include WordPerfect, Harvard Graphics, and Lotus 1-2-3.

Software applications of this type should work properly when the printer is under the direct control of the application, and also when there is a spooler between the application and the printer. This is accomplished by using the ENTER command to select the proper printer language before sending print data to the spooler or printer. The UEL command should precede and follow the print job.

Print Data Generators should only use PJL commands when a corresponding command does not exist in the printer language. This is recommended so that the application does not override feature settings and adversely affect future jobs. To change feature settings for the duration of the job, or until the next PJL reset condition, use the SET command. Do not use the PJL INITIALIZE command for Print Data Generators.

Commands Used by Print Data Generators

UEL	Use before and after every PJL job.
COMMENT	Use to add explanations within PJL code.
ENTER	To select the correct printer language for the print job.
SET	To select a desired feature when it is not possible using the desired printer language (such as PCL or PostScript).
RESET	Use at the end of the job if the SET command is used in the job.

Print Data Generator Examples

The following two examples demonstrate how applications categorized as “Print Data Generators” should use PJL. Both examples work well if sent directly to the printer or through a spooler.

Jobs That Select a Printer Language

This example selects a printer language using the ENTER command. It is the most common job format for Print Data Generators.

```
<ESC>%-12345X@PJL <CR><LF>
@PJL ENTER LANGUAGE = POSTSCRIPT <CR><LF>
%!PS-ADOBE ... PostScript code ... ^D
↩<ESC>%-12345X
```

Jobs That Set Printer Features

This example specifies two feature settings that cannot be selected using the specified printer language:

```
<ESC>%-12345X@PJL <CR><LF>
@PJL SET RET = LIGHT <CR><LF>
@PJL SET RESOLUTION = 600 <CR><LF>
@PJL ENTER LANGUAGE = PCL <CR><LF>
<ESC>E ... PCL print data ... <ESC>E
↩<ESC>%-12345X@PJL <CR><LF>
@PJL RESET <CR><LF>
<ESC>%-12345X
```

Printer Utilities

Printer utilities are those applications that perform one or more of the following functions:

- **Control printer resources**—These programs set printer features such as the print resolution and number of copies using the SET, RESET, DEFAULT, and INITIALIZE commands.
- **Request information from the printer**—Printer status and feature settings can be requested from the printer using the ECHO, INFO, INQUIRE, and DINQUIRE commands. Applications should only use PJL commands that request status if the application is in direct control of the printer and there is a bi-directional communication channel between the printer and the application.
- **Monitor the printer status**—Device status and timed status can inform the application of printer events such as printer open, paper out, or low toner. The USTATUS DEVICE and USTATUS TIMED commands are used to monitor printer status. Applications can continuously monitor the printer for changes in printer state. When the printer changes state, the application takes appropriate action, such as informing the user.

In general, the Printer Utilities described here are not used to print information, but enable the user to request the current printer feature settings and modify features as desired.

Commands Used by Printer Utilities

UEL	Use before and after every PJL job.
COMMENT	Use to add explanations within PJL code.
SET	To modify a printer feature until the next PJL reset condition.
RESET	To set the printer to its user default feature settings.
DEFAULT	To modify the user default value of a printer feature. (NOTE: In shared environments, this feature should be used by system administrators only.)
INITIALIZE	To set the printer to its factory default settings. (NOTE: In shared environments, this feature should be used by system administrators only.)

Printer utilities that are equipped for bi-directional communications can request printer status using the following commands:

INQUIRE	To request the current setting for a particular feature.
DINQUIRE	To request the default setting for a particular feature.
INFO	To find the printer model, printer configuration, memory available, status information, page count, or printer variables.
ECHO	To synchronize requested status information.
USTATUS DEVICE	To be informed when the printer status changes.
USTATUS TIMED	To receive unsolicited printer status at periodic intervals.

Printer Utility Examples

The following three examples demonstrate the kind of PJL jobs used by Printer Utilities:

- Jobs that request information and perform printer setup
- Jobs that request information, set features, and print existing files
- Jobs that monitor printing status

Jobs that Request Information and Perform Print Setup

The following example demonstrates a job setup utility that first asks for printer information, then sets default features to a desired state. This application functions like a remote control panel. This type of utility enables users to select PJL features before printing from another application that cannot select PJL features. In this example, the DEFAULT command is used instead of the SET command, so that changes apply to all future print jobs.

Note Since this application uses the DEFAULT command, it changes the User Default Environment and control panel settings. In multi-user situations, it should be used only by printer administrators to avoid unexpected results by other users.

```
(Reading Status Information)
<ESC>%-12345X@PJL <CR><LF>
@PJL COMMENT Requesting features <CR><LF>
@PJL ECHO 15:18:25.3 07-25-92 <CR><LF>
[. . application discards any printer status readback
information received before the ECHO response . .]
@PJL DINQUIRE RET <CR><LF>
@PJL DINQUIRE RESOLUTION <CR><LF>
[. . application receives status readback from the
DINQUIRE commands showing that RET = OFF and
RESOLUTION = 300.  . . .]
<ESC>%-12345X
```

```
(Changing the Default Values)
<ESC>%-12345X@PJL <CR><LF>
@PJL COMMENT Set desired values <CR><LF>
@PJL DEFAULT RET = DARK <CR><LF>
@PJL DEFAULT RESOLUTION = 600 <CR><LF>
@PJL COMMENT Reset PJL to ensure default
↪values take effect <CR><LF>
@PJL RESET <CR><LF>
<ESC>%-12345X
```

Jobs That Request Information, Set Features and Print Existing Files

The following example requests information using the INQUIRE command, sets printer features using the SET command, and then sends an existing application print file to the printer. To restore the default values, the RESET command is issued before the job closes.

```
<ESC>%-12345X@PJL <CR><LF>
@PJL COMMENT Requesting features <CR><LF>
@PJL ECHO Job #53 11-17-92 08:52:03.7 <CR><LF>
[. . application discards any printer status readback
information received before the ECHO response . .]
@PJL INQUIRE RET <CR><LF>
@PJL INQUIRE PAGEPROTECT <CR><LF>
@PJL INQUIRE RESOLUTION <CR><LF>
<ESC>%-12345X
[. . application receives status readback from the
INQUIRE commands showing that RET = OFF, PAGEPROTECT
= LETTER, and RESOLUTION = 300, which are not the
desired settings. The application then sends SET
commands to modify the settings to the desired value
. .]
<ESC>%-12345X@PJL <CR><LF>
@PJL COMMENT Set desired values <CR><LF>
@PJL SET RET = MEDIUM <CR><LF>
@PJL SET PAGEPROTECT = OFF <CR><LF>
@PJL SET RESOLUTION = 600 <CR><LF>
@PJL ENTER LANGUAGE = PCL <CR><LF>
<ESC>E . . . PCL file . . .<ESC>E
↪<ESC>%-12345X@PJL <CR><LF>
@PJL COMMENT Reset PJL to restore <CR><LF>
@PJL COMMENT control panel defaults <CR><LF>
@PJL RESET <CR><LF>
<ESC>%-12345X
```

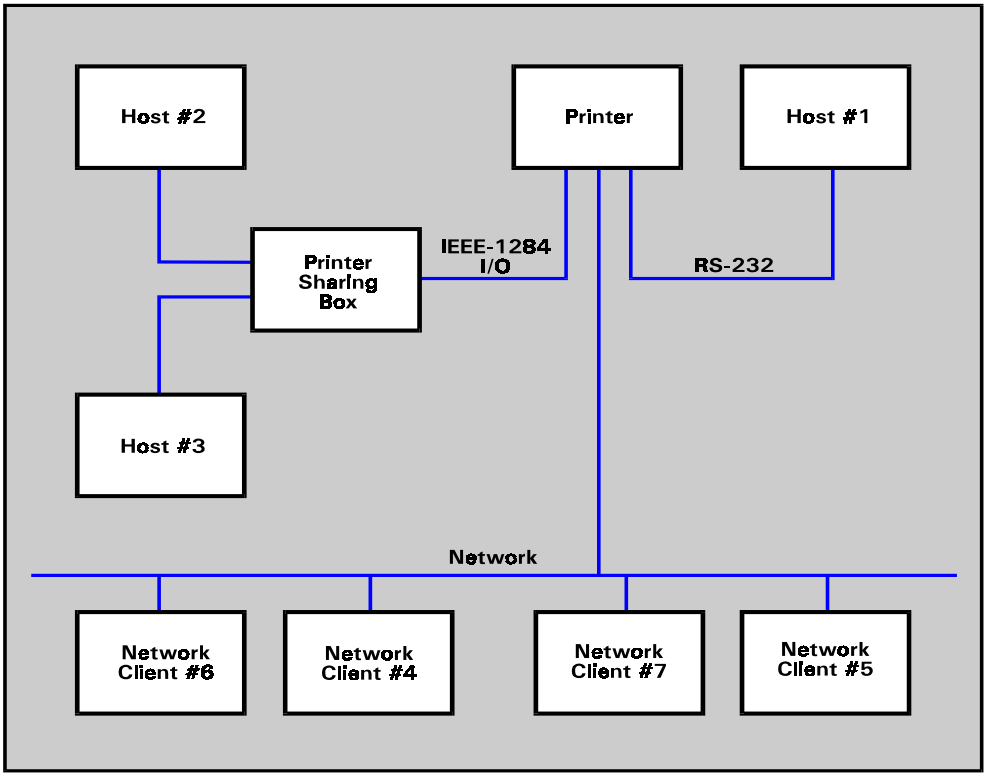

Figure 11-1 Sample Printing System Components

Jobs That Monitor Printing Status

The following example enables unsolicited printer status using the USTATUS DEVICE command. After unsolicited device status is enabled, the printer automatically sends a status response whenever the printer's status changes.

```
<ESC>%-12345X@PJL <CR><LF>
@PJL ECHO Acme #17 8-28-92 5:39:02.9 <CR><LF>
@PJL COMMENT Turn on unsolicited device
↪status <CR><LF>
@PJL USTATUS DEVICE = ON <CR><LF>
<ESC>%-12345X
```

Spoolers

Spoolers are applications that control and/or monitor print jobs generated by other applications.

Spooling applications should use only PJL job-related commands when controlling the flow of print jobs to the printer. Encapsulate each print job with the JOB command at the beginning and the EOJ command at the end (the UEL command is always the first and last command in all PJL jobs).

Spoolers perform one or more of the following operations:

- **Confirm job completion**—To notify the spooler when the job is completely processed, use the USTATUS JOB = ON command. When this command is used, the EOJ command at the end of the job triggers the printer to send unsolicited job status information when the last page of the job is printed and in the output tray.
- **Cancel jobs** (only HP LaserJet 5/5M, 5Si/5SiMx/5Si Mopier, and other printers that support job cancel)—The spooler can cancel the print job using the PML printer language or a user can cancel the job using the control panel. If unsolicited job status is enabled, the printer notifies the application that the job has indeed been canceled.
- **Monitor job progress**—If you want the spooler to monitor the page-by-page progress of a job, enable the page status with the USTATUS PAGE=ON command. As each page lands in the output tray, the printer sends an unsolicited page status message to the spooler.
- **Interact with the user**—To flash a message on the printer's display and take the printer off-line, use the OPMSG or STMSG commands. These commands are used to notify the user of a required action before printing can continue. For example, you can display “LOAD BLUE PAPER” and then wait until the user presses the Continue or On Line key. To display a message without taking the printer off-line, use the RDYMSG command.
- **Select a Printer Language or Print a Banner Page**— If the spooler is responsible for printer language selection or printing banner pages, use the ENTER LANGUAGE command. (Performance is not adversely affected if both the application file and the spooler send the ENTER LANGUAGE command.)

- **Print selected pages of jobs**—Use the JOB command with the START and END options to specify which pages you want to print.

Commands Used by Spoolers

UEL	Use before and after every PJL job.
COMMENT	Use to add explanations within PJL code.
DEFAULT	To modify the default value of a printer feature. (NOTE: In shared environments, this feature should be used by system administrators only.)
ENTER	Use if responsible for printer language selection or banner printing.
JOB/EOJ	Use to monitor printing status, name jobs, print portions of a job, mark job boundaries.
OPMSG	To display a customized message and take the printer off-line until the operator presses the On Line, Continue, or Reset key.
RDYMSG[1]	To replace the control panel “00 READY” message with a customized message.

[1] For the LaserJet 4000 and newer printers, use the JOB DISPLAY = “display text” method instead of (or as well as) RDMSG. Refer to the note on page 8-2.

Spoolers equipped for bi-directional communications can monitor printer status using the following commands:

ECHO	To synchronize status readback information. The application should send an ECHO command before any other status readback commands; all status responses before the returned ECHO response should be discarded to ensure proper synchronization.
STMSG	To display a customized message and take the printer off-line until the operator presses the On Line, Continue, Job Cancel, or Reset key. Returns the name of the key that the operator pressed.
USTATUSOFF	To turn off all unsolicited status before specifying the desired USTATUS.
USTATUS PAGE	To receive notification as each page lands in the output tray.
USTATUS JOB	To monitor job progress. Sends status message when job has completed.

Spooler Examples

The following examples show jobs used by spooling applications:

- Jobs that monitor job progress and confirm job completion
- Jobs that notify the user to load specific forms or paper
- Jobs that control printer resources and print selected pages
- Jobs that print a banner page followed by an application-generated job (application specifies printer language)
- Jobs that print a banner page followed by an application-generated job (application does not specify printer language)

In each example, the indented lines indicate commands belonging to a print file created by another application.

Jobs That Monitor Job Progress and Confirm Job Completion

This example enables the spooler to receive notification as each page is finished printing, and when the entire job has finished. The USTATUS JOB = ON command enables unsolicited job status and the USTATUS PAGE = ON command enables page status.

```
<ESC>%-12345X@PJL <CR><LF>
@PJL COMMENT Monitoring Job Progress <CR><LF>
@PJL COMMENT and Completion <CR><LF>
@PJL ECHO 12:07:54.5 07-26-92 <CR><LF>
[. application discards printer status readback data
received before ECHO response .]
@PJL USTATUS JOB = ON <CR><LF>
@PJL USTATUS PAGE = ON <CR><LF>
@PJL JOB NAME = "Monitoring Job" <CR><LF>
  <ESC>%-12345X@PJL <CR><LF>
  @PJL ENTER LANGUAGE = PCL <CR><LF>
  <ESC>E... PCL print data ...<ESC>E
  ↩<ESC>%-12345X
↩<ESC>%-12345X@PJL <CR><LF>
@PJL EOJ NAME = "End of Monitor Job" <CR><LF>
@PJL USTATUS JOB = OFF <CR><LF>
@PJL USTATUS PAGE = OFF <CR><LF>
<ESC>%-12345X
```

When the printer processes the JOB command, the following unsolicited job status message is sent:

```
@PJL USTATUS JOB
START
NAME="Monitoring Job"
<FF>
```

When USTATUS PAGE = ON, the first unsolicited page status message is sent after page 1 lands in the output tray:

```
@PJL USTATUS PAGE
1
<FF>
```

Additional page status messages are sent after each succeeding page lands in the output tray. After the job is completely printed, the host receives unsolicited page and job status as follows (assuming a 38-page job):

```
@PJL USTATUS PAGE
38
<FF>

@PJL USTATUS JOB
END
NAME="End of Monitor Job"
PAGES=38
<FF>
```

The unsolicited *page* status indicates that the 38th page printed. The *job* status indicates that the EOJ command was received, and that 38 pages were printed. (The page count is incremented by one for each simplex page formatted, and by two for each duplex page formatted.)

Jobs That Notify the User to Load Specific Forms or Paper

This example displays a message to “LOAD FORM 1040” and takes the printer off-line. When the form is loaded and the On Line or Continue key is pressed, the form is printed.

```
<ESC>%-12345X@PJL <CR><LF>
@PJL JOB NAME = "Job #65432" <CR><LF>
@PJL OPMSG DISPLAY = "LOAD FORM 1040" <CR><LF>
  <ESC>%-12345X@PJL <CR><LF>
  @PJL ENTER LANGUAGE = PCL <CR><LF>
  <ESC>E ... PCL print data ... <ESC>E
  ↩<ESC>%-12345X
↩<ESC>%-12345X@PJL <CR><LF>
@PJL EOJ NAME = "Job #65432" <CR><LF>
<ESC>%-12345X
```

Jobs That Control Printer Resources and Print Selected Pages

The following job prints 1 copy of pages 5 through 8 of the print job. The START command tells the printer to process pages in non-printing mode until the specified page (in this case, page 5). The END command specifies the last page to print. If the END command is not included, the job prints from page 5 through the end of the document.

If you need to print more than one copy of each page, send the job multiple times using the START and END options to get collated copies. Using the “@PJL SET COPIES = *number of copies*” command to print multiple copies of a selected page causes unexpected results because the printer counts physical pages, not multi-copy collated pages.

```
<ESC>%-12345X@PJL <CR><LF>
@PJL COMMENT Printing selected pages <CR><LF>
@PJL JOB START = 5 END = 8 <CR><LF>
  <ESC>%-12345X@PJL <CR><LF>
  @PJL ENTER LANGUAGE = PCL <CR><LF>
  <ESC>E . . . PCL print job . . .<ESC>E
  ↩<ESC>%-12345X
↩<ESC>%-12345X@PJL <CR><LF>
@PJL EOJ <CR><LF>
<ESC>%-12345X
```

Spooler Job That Prints a Banner Page Followed by an Application-Generated Job (Application specifies printer language)

The following example demonstrates a spooling application that prints a banner page, and then prints a PostScript job. The application that produced the print file specifies the printer language.

```
<ESC>%-12345X@PJL <CR><LF>
@PJL COMMENT Spooling Job with Banner <CR><LF>
@PJL JOB NAME = "Spooler #1" <CR><LF>
@PJL ENTER LANGUAGE = PCL <CR><LF>
<ESC>E . . . Banner Page Data . . . <ESC>E
↩<ESC>%-12345X
  ↩<ESC>%-12345X@PJL <CR><LF>
  @PJL ENTER LANGUAGE = POSTSCRIPT <CR><LF>
  %!PS-ADOBE . . PostScript print data . . ^D
  ↩<ESC>%-12345X
↩<ESC>%-12345X@PJL <CR><LF>
@PJL EOJ <CR><LF>
<ESC>%-12345X
```

Spooler Job That Prints a Banner Page Followed by an Application-Generated Job (Application Does Not Specify Printer Language)

The following example demonstrates a spooling application that prints a banner page. The application was not created with a PJL-ready driver, so it does not begin and end with UEL commands, or specify the printer language—the printer implicitly switches to the correct printer language.

```
<ESC>%-12345X@PJL <CR><LF>
@PJL JOB NAME = "Spooler #2" <CR><LF>
@PJL ENTER LANGUAGE = PCL <CR><LF>
<ESC>E . . Banner Page Data. .<ESC>E
↩<ESC>%-12345X
   . . . PCL or PostScript print data . . .
↩<ESC>%-12345X@PJL <CR><LF>
@PJL EOJ <CR><LF>
<ESC>%-12345X
```

Requesting Printer Status

When querying the printer for status, the response is not immediate. Wait a fixed amount of time for a response and then time out. If the printer is still working on a previous print job, it may take a while before a response is received.

Your application should be able to discard unexpected status, such as unsolicited status, and discard unrecognizable lines. Lines within the PJL status response begin with a specific keyword, as described in the command description in Chapter 7, and end with the <CR><LF> control codes. Future printers may support new keywords in the PJL status response. Your application should ignore those lines which it does not understand.

Using Status Readback in a Multi-User System

The printing system consists of all components involved in the process of turning an application document into a printed sheet of paper. Common printing system components include the host computer, applications on the host computer, the operating system used by the host computer, the cable connecting the host computer to the printer, and the printer. Other printing system components can include printer sharing boxes, network servers, spoolers running on network servers, and I/O cards installed in the printer. For printer status readback to be useful, all components must be bi-directional.

Some operating system environments, like Microsoft's Windows, provide the components that interact with the printer so Windows applications generally do not need to support printer status readback.

The components that may generate printer queries include the host application, the printer sharing box, the network spooler, and the I/O card installed in the printer. Printer status readback can allow many printing system components to function more effectively, not just the host application. For example, a network I/O card may inject a PJL JOB command at the beginning of each job and a PJL EOJ command at the end of each job. Using unsolicited PJL job status, the I/O card could sent a network packet to an application on the client that supplied the job, notifying the user the job had been printed. Printer status readback allows many printer system components to solve user's needs.

Example Printing System

Figure 11-1 contains an example printing system. The print system contains one I/O-switching LaserJet printer which is connected to seven computers. Host computer #1 is connected via the serial I/O interface. Host computers #2 and #3 are connected through an external printer sharing box that also buffers print job data. The printer sharing box is connected to the printer through the IEEE-1284-compatible I/O interface. Network servers #4 and #5 spool print jobs from network clients #6 and #7.

Although all the network clients and spoolers are connected to the same network cable as the printer, only the two network servers are capable of communicating directly with the printer. In this example, the network clients spool print jobs to a network server; they can not interact directly with the printer.

Given the printing system as shown in Figure 11-1, not all of the printing system components can query the printer and expect a response back.

Printer Status Readback and Printer Sharing

Host computer #1 is directly connected to the printer via the bi-directional serial I/O link. If the software on the host computer supports bi-directional printer communication, applications on host computer #1 can expect the printer to respond to queries. However, at the time an application on host computer #1 requests information, the printer may be processing a print job from a different I/O interface, so the response to host computer #1's query may be delayed, as described in the “Printer Status Readback and Printer I/O Switching” section in this chapter.

Host computers #2 and #3 are connected to the printer via an external printer sharing box that buffers data. If the printer sharing box was not designed to support bi-directional communication with the printer, then applications on host computers #2 and #3 will not be able to receive printer status readback information. However, since the printer sharing box is connected to the printer via the Bi-Tronics I/O interface, which does support bi-directional communication, it is possible for printer sharing boxes to be designed to support printer status readback. In fact, the printer sharing box itself may be able to inject a “print job” to query the printer for specific information.

External printer sharing boxes can be designed to support sending data from the printer to the host computer. However, the printer sharing box has to determine which host computer should receive the

printer's response. The simplest solution for printer sharing boxes is to forward all data received from the printer to all attached host computers, and let the applications on each host computer ignore any impertinent data, as described in the “Handling Unexpected Printer Responses” section in this chapter.

The printer in this example is also attached directly to a network via a network interface card installed into the printer's MIO interface. Network servers #4 and #5 spool print jobs from network clients #6 and #7. In this example, the network clients can not interact directly with the network-attached printer. The network spoolers can inject a “print job” to query the printer and can expect to get responses back. However, for this example, the network operating system does not provide a mechanism which allows applications running on a network client to query the printer and get a response.

The printer's I/O card, the printer sharing box, and the spoolers running on the network servers can query the printer and expect to receive a response since those components are connected directly to the printer (i.e., no other printing system component except for the communication cable is between the printer and those components).

In summary, an application running on host computer #1 can receive printer responses if the operating system on the host computer supports bi-directional communication over the serial I/O interface. The application may have to wait for the printer's response if the printer is processing a job from either the IEEE-1284-compatible I/O interface or the MIO interface.

An application running on host computer #2 or #3 may receive printer responses if the printer sharing box supports bi-directional printer communication. However, since the printer sharing box does not have enough information to route the response to the correct host computer, the printer sharing box can forward all printer responses to all attached host computers and let the application on the host computer ignore any unneeded printer responses.

Applications on network clients #6 and #7 cannot receive any printer responses due to the underlying network operating system. However, the printer sharing box, the I/O card installed in the printer, and spoolers running on network servers #4 and #5 can all query the printer and expect a printer response (although the response will not be immediate if the printer is processing a print job from a different I/O interface).

Printer Status Readback Usage

Applications use printer status readback for two primary purposes (although there are other uses): printer driver configuration and monitoring device status. Since printer configuration does not change often (i.e., the amount of installed printer memory does not change frequently) using printer status readback to assist in initial configuration (or when the printer's configuration changes) is straightforward. If the application determines printer status readback is unavailable (discussed in the “Determining If Printer Status Readback is Available” section), the application can ask the user for the information instead of querying the printer.

The other primary use for printer status readback is to allow applications to monitor the state of the printer. In particular, applications can inform the user when the printer requires intervention (e.g. fill paper trays). If the application enables unsolicited PJL device status, the printer will inform the application whenever the printer's status changes. Since the printer informs the application using unsolicited status, even when the printer is processing a print job from a different I/O interface, the application gets timely printer status information. If printer status readback is unavailable, the application will not be able to provide the user with current printer state information.

Another use for printer status readback is to determine what volatile resources (e.g. soft fonts) are available in the printer right before a print job is sent to the printer. If an application optimizes the printer job using printer status readback (i.e., does not send soft fonts if the printer already has the font), then if printer status readback is unavailable, the application should assume worst case (i.e., the font is not in the printer) when generating print jobs.

In summary, the application should be designed to take advantage of printer status readback, if available, but still function if it is unavailable.

Determining If Printer Status Readback is Available

If the application sends a PJL ECHO command (preceded by the Universal Exit Language command, of course) to the printer and receives the expected response, then the application can assume bi-directional communication with the printer is available and supported solicited status queries will be answered in a timely fashion.

However, under some conditions the application will send, or attempt to send, a PJL ECHO command and not receive a timely response because of one of two reasons: the printer may not be accepting data (due to the printer processing a print job from a different I/O interface or the printer being off-line), or the printing system does not support printer status readback to the application. If the application does not receive a response within three seconds, then the application can assume printer status readback is currently unavailable.

In some printing environments, the application may be able to distinguish between the host operating system not being able to send the query, and the query being sent without receiving a response in a reasonable amount of time. Application developers should not assume that because a query was sent and a response was not received in a timely fashion, that the printer system does not support bi-directional communication. Printer sharing devices may buffer the query and the response may be available, however the data buffered before the query may take more than three seconds to process, delaying the response by more than three seconds.

The application should notify the user that printer status readback is currently unavailable and query the user to determine how to proceed. The common choices are: 1) printer busy, continue retrying until printer status readback is available, 2) printer busy on a long printer job, temporarily do not use printer status readback, or 3) printer status readback unavailable due to host-printer configuration, do not use printer status readback. After obtaining the user's response, the application can continue as appropriate.

Printer Status Readback and Printer I/O Switching

I/O switching occurs when a printer contains more than one I/O interface and accepts data from any interface, based on a first-come first-served basis. Some PJL printers do not support I/O switching.

I/O switching adds complexity to printing solutions taking advantage of printer status readback. If the printer is currently processing a print job from a different I/O interface, applications querying the printer will have to wait for the printer to respond to the query, (or proceed without using printer status readback). The amount of time until the printer provides the response is dependent on the length of the current print job and the length of any other print jobs the printer may process before processing the application's query.

When processing a job, I/O switching paces off all I/O interfaces except for the one which is providing the current print job. However, it is possible for an application to get a few bytes of data into the inactive I/O interface before the printer paces off that I/O interface. If the application times out while trying to send the initial UEL command after one or more bytes of the command are sent, but before the entire UEL command is sent, then the application can proceed using one of the following choices: 1) continue attempting to send the data to the printer, 2) remember how much of the initial UEL command has been sent to the printer and attempt to send the rest of the command at a later time, or 3) stop trying to send the command, making sure the next command the application sends to the printer is the UEL command. This is done so the parser inside the printer properly recognizes all commands following the previously sent partial UEL command. The printer will properly recognize the UEL command, even if the
command appears after a partial UEL command.

For example, assume the initial query is:

```
<ESC>%-12345X@PJL ECHO 08/27/92 09:53:46.5
↩033288925
```

If the printer only accepts "<ESC>%-12", and if the application later sends:

```
<ESC>%-12345X@PJL ECHO 08/27/92 09:57:46.5
↩6202323802 <LF>
```

The printer accepts the query and responds with:

```
@PJL ECHO 08/27/92 09:57:46.5 6202323802<LF>
<FF>
```

One of the conditions I/O-switching printers use to determine when to switch to another I/O interface is the idle time of the I/O interface supplying the current print job. Idle time is defined as the time elapsed since the I/O interface received the last data byte. If the application queries the printer and waits for a response, it is possible for I/O interface idle time to exceed the set timeout value (the TIMEOUT environment variable). When the printer detects the current I/O interface timed out, the printer allows I/O switching and treats any data received over the I/O interface as unrelated to the data previously received over that interface.

To keep the I/O interface from timing out, the application can either send data to the printer more frequently than the timeout setting or can cause the printer to use a different setting by sending a PJL JOB command. When I/O-switching printers receive the PJL JOB command, the printer uses an I/O timeout value equal to ten times the TIMEOUT environment variable setting or five minutes, whichever is greater. (If the application sends the PJL JOB command, the application should send a PJL EOJ command at the end of the job.)

Old Printer Status Readback Responses

Applications need to be designed to handle receiving printer status readback responses that were generated because of an action performed by a different application. For example, a word processing application may send a query to the printer, but because the user turned off the host computer, the application never got a chance to retrieve the response from the printer before the host computer was turned off. When another application, or even the same application at a later time, queries the printer, the first response is the response to the word processing application's query.

The currently executing application should accept old printer status readback responses and discard them until the expected response is received. Applications should include unique information as part of the ECHO command, such as the current time and date and a random number, so that the application can resynchronize with printer status readback responses.

There is also a chance the PJL ECHO response will never by sent by the printer, due to overflowing of the printer's response data buffer as discussed in the following section.

Response Data Buffer Overflow

The printer contains a limited amount of memory. It is possible for an application to fill all available printer memory with printer status readback responses that have not yet been sent to the host computer, simply by sending many queries and never accepting the responses from the printer. Also, if an application enables unsolicited status and never accepts responses, the printer's response data buffer can overflow. The printer limits the amount of memory allotted to hold responses, and when that memory is full (and the host has not recently accepted any responses), the printer discards all future responses until the host accepts the currently queued responses in the printer.

When the printer discards responses, the printer inserts the PJL unsolicited “printer to host data buffer overflow” message:

```
CE
CODE=10010
DISPLAY="00 READY"
```

Note Refer to the “HP LaserJet 4 and 5 Family Comments” section in Appendix A under “Printer Status Readback” for the exact “printer to host buffer overflow” message sent by HP LaserJet 4 and 5 printer families.

The PJL unsolicited “printer to host data buffer overflow” message is always inserted when the printer discards one or more responses, even if unsolicited status is turned off.

I/O-switching printers set aside memory for each I/O interface. A response data buffer overflow on one I/O interface does not mean another I/O interface will also experience the overflow condition. To keep the printer's response data buffer from overflowing, applications should not send more than five queries without accepting the printer's response to those queries.

The application should always send a PJL ECHO command to resynchronize with the printer status readback responses (i.e., get rid of any old responses). However, if the printer's response data buffer is full and the printer is discarding all future responses, then the host will receive a PJL unsolicited “printer to host data buffer overflow” message, but not the PJL ECHO response. Applications must be designed to handle situations where the printer's response data buffer overflows and the printer does not provide the expected response, but instead provides a PJL unsolicited “printer to host data buffer overflow” message. The application should resend all unanswered queries whenever the application receives a PJL unsolicited “printer to host data buffer overflow” message.

Handling Unexpected Printer Responses

Printer sharing devices, like the external printer sharing box or the MIO card inserted into the printer as shown in Figure 11-1, do not have sufficient information to allow the printer sharing device to properly route printer status readback responses to the host computer (the one running the application that queried the printer). In particular, unsolicited PJL status is difficult for the printer sharing

device to properly route since the unsolicited status response may be received by the printer sharing box hours after sending the data from the host computer to the printer that enabled unsolicited status.

Manufacturers of printer sharing devices can choose from the following options to solve this printer-to-host data routing problem.

1 A printer sharing device that is a uni-directional device and does not support printer-to-host data transfer. Applications need to be designed to work with uni-directional printer sharing devices.

2 A printer sharing device that examines the data received from the printer and only forwards solicited responses to the host computer.

A printer sharing device can track which of its I/O interfaces is currently sending data to the printer, and thus know which I/O interface to use when sending solicited printer responses. When the printer sharing device sends data to the printer from a different I/O interface, the printer sharing device must know to the exact byte boundary what printer response data should be sent to the previous I/O interface and what data should be sent to the current I/O interface.

Before the printer sharing device sends the printer data from a different I/O interface than the source of the current print job, the printer sharing device can inject a PJL ECHO command. All solicited printer responses received before the PJL ECHO response should be sent to the previously active I/O interface. All solicited printer responses received after the PJL ECHO response should be sent to the currently active I/O interface. The printer sharing device should consume the PJL ECHO response which was a result of the PJL ECHO command injected by the printer sharing device.

Applications that use unsolicited PJL device, job, page, or timed status need to work properly with printer sharing devices that do not support unsolicited printer-to-host responses.

3 A printer sharing device that can route solicited responses as described in 2 above, plus send all unsolicited responses to all attached host computers. Applications must be designed to properly ignore unexpected printer status readback responses. Ideas on how to design an application to ignore unexpected printer status readback responses are described in the next section.

Application Design Ideas

The previous pages have described various printing system topologies and how components in those topologies can operate so each printer system component can receive printer status readback responses. This section discusses application design ideas that handle the complexities involved with using printer status readback information.

The obvious design approach of querying the printer and waiting for an answer to that query is sure to lead to failure. A busy printer, old printer responses, overflowed printer response buffer, and a uni-directional printing environment all cause the obvious design approach to be inadequate. A better design approach is to treat printer responses as events that cause the application to take some action. Solicited and unsolicited printer status readback queries/responses are discussed separately.

Whenever the application sends a solicited query to the printer, the application should add the query to the “outstanding solicited query” list. Also, if the application can not proceed until the response is received, then a watchdog timer should be set so that some event will be guaranteed to occur in case the application does not receive a timely response. Whenever a solicited response is received, the response should be compared to the list of outstanding solicited queries. If there is no match, then the solicited response should be discarded. If there is a match, then the query should be removed from the outstanding solicited query list, any associated watchdog timer should be disabled, and the application should process the response. If the application was blocked waiting for the response, then the application should become unblocked.

If the application does not need an unsolicited status response, then the application can discard the response.

If the application ever receives the unsolicited PJL “printer to host data buffer overflow” message, then the application should resend all queries in the “outstanding solicited query” list. When the application resends the outstanding solicited queries, it is possible that the application will receive two responses (one generated because of the initial query and one generated because of the resent query). If the application is using an outstanding solicited query list, then when the first response is received, the query will be removed from the list. When the second response is received, the second response will be ignored because there is no matching query in the outstanding solicited query list.

Given the above structure for sending queries and processing responses, applications should adhere to the following guidelines:

1 If the application sends data to the printer in blocks, followed by long pauses between blocks (i.e., greater than five seconds), then it should cause the printer to use a larger I/O interface idle time value by using the PJL JOB command as described in the “Printer Status Readback and Printer I/O Switching” section.

2 Check status readback availability as described in the “Determining if Printer Status Readback is Available” section.

3 Synchronize with the printer-to-host data stream as described in the “Old Printer Status Readback Responses” section.

4 Do not have more than five outstanding queries before accepting printer status readback responses. This prevents response data loss as described in the “Response Data Buffer Overflow” section.

5 Before quitting, turn off all unsolicited status enabled by the application. Also, accept all responses from the printer that were generated because of actions performed by the application.

6 Do not be dependent on the timing characteristics of a single printer model. For example, the response time of the HP LaserJet 4 printer's built-in IEEE-1284-compatible I/O interface will be different than the timing characteristics of a IEEE-1284- compatible interface in an I/O card installed into the printer.