Tru64 UNIX Best Practice

Implementing Forms Management with Advanced Printing Software

January 2002

Product Version:	**Advanced Printing Software Version 1.1**
Operating System and Version:	**Tru64 UNIX Version 4.0F or higher.**

This Best Practice describes how to configure your print system to efficiently print and manage print jobs that require different types of print formats such as preprinted forms, envelopes, company letterheads, inventory lists, bills of material, invoices, transparencies, and fanfold paper.

Compaq Computer Corporation
Houston, Texas

Contents

Implementing Forms Management with Advanced Printing Software

Implementing Forms Management with Advanced Printing Software

This Best Practice describes how to configure your print system to efficiently print and manage print jobs that require different types of print formats such as preprinted forms, envelopes, company letterheads, inventory lists, bills of material, invoices, transparencies, and fanfold paper.

See the Tru64 UNIX Best Practices Web page for more information about Best Practices documentation:

http://tru64unix.compaq.com/faqs/publications/best_practices/

Is This Best Practice Right for You?

Not all Best Practices apply to all configurations, so you must be sure that it is appropriate for your system and circumstances. To use this Best Practice, you must meet the requirements described in the following table:

Requirement	Description
Operating System	Tru64 UNIX™ Version 4.0F or higher
Printing Software	Advanced Printing Software Version 1.1

Before You Begin

Before you apply the Best Practice for *Implementing Forms Management with Advanced Printing Software*, you must understand the methodology and perform some preliminary tasks.

1. Identify the print areas you serve with Advanced Printing Software. Print areas typically consist of similar printers, in one location, serving the same types of task; for example, printing envelopes and letters on company stationery. Focus on one print area at a time.
2. Determine the different types of paper forms and media used in each print area.

3. Create a set of logical printers that have default job attributes, such as the kind of paper form required, setup modules that select a font or font size, or specific input trays or output bins.
4. Create physical printer objects that represent your printer devices and their capabilities.
5. Associate all the area's printers with a queue.

   By declaring which physical printers are ready to print on a particular form or media, the Advanced Printing Software spooler ensures that jobs are printed on the correct form.
6. Create the `initial-value-job` object and `initial-value-document` object and assign attribute values to them. These are the default attributes.
7. Associate the initial-value objects with one or more logical printers.

Note

Using the `pdprintadmin` GUI simplifies this process but it limits you to one set of initial-value objects per logical printer. It does not share initial-value objects across multiple logical printers. You can use the `pdprintadmin` GUI to set up the default values for one printer and use a CLI command to share the default values with one or more printers.

When the printers and queues are set up with the default attributes, you can submit print requests to a logical printer destination that implicitly defines a set of job defaults. You will not have to specify the attributes with each print request. In System V printing, this is referred to as printing to a printer `class`. A physical printer can accept jobs submitted to any logical printer on the queue, but it will only be given jobs where the requirements are met by the physical printer's supported and ready attributes.

Applying the Best Practice

Before you implement forms management with Advanced Printing Software, be sure to read and understand the methodology explained in *Before You Begin*.

This Best Practice uses an example to illustrate the tasks for implementing forms management with Advanced Printing Software.

The example defines three logical printers: `listing_A` which prints listings, `invoice_A` which prints invoices, and `letter_A` which prints letterhead. Two identical printers, `A1_pp` and `A2_pp`, accept jobs from the logical printers through queue `A_q`. In the example, the spooler is named `my_spl` and the supervisor is named `my_sup`. The print area to which the printers belong is print area A, thus `invoice_A` and so on and `A1_pp` and so on. To set up this configuration:

1. Create the logical printers:

```
# pdcreate -c printer my_spl:listing_A
# pdcreate -c printer my_spl:invoice_A
# pdcreate -c printer my_spl:letter_A
```

2. Create the physical printer objects:

```
# pdcreate -c p -x printer-model='"FormPrint 9000"' my_sup:A1_pp
# pdcreate -c p -x printer-model='"FormPrint 9000"' my_sup:A2_pp
```

   The example does not use PAF files. However, if your printer is a supported text printer, you should use the appropriate PAF file. Replace the printer-model definition with the `-X` option to specify the PAF file. The example uses a hyphothetical printer model, `FormPrint 9000`.

3. Set the device address for each physical printer. For example, if the printers support raw socket connections on port 9100:

```
# pdset -x printer-address=bingo.mycomp.com:9100 A1_pp
# pdset -x pr-address=bango.mycomp.com:9100 A2_pp
```

4. Set up one queue:

```
# pdcreate -c q my_spl:A_q
```

5. Associate the printers with the queue:

```
# pdset -c p -x associated-queue=A_q listing_A invoice_A letter_A
# pdset -c p -x assoc-qu=A_q A1_pp A2_pp
```

6. View the logical printers and physical printers, attributes of the physical printers, and the queues defined on the spooler to verify what you have done.

   a. List the logical printers on the spooler and all physical printers on the supervisor:

```
# pdls -c p my_spl: my_sup:
```

   b. List specific attributes of the physical printers:

```
# pdls -c p -r 'pr-name pr-state enabled pr-model pr-addr' \
  my_sup:
```

c. List all of the attributes of the physical printers:

```
# pdls -c p -r all -s line A1_pp A2_pp
```

d. List the queues defined on the spooler:

```
# pdls -c q my_spl:
```

7. Enable the logical printers and queue to allow them to accept print jobs:

```
# pdenable -c queue A_q
# pdenable listing_A invoice_A letter_A
```

8. Enable the physical printers that have the correct media loaded and disable the physical printers that do not. For example:

```
# pddisable A2_pp
# pdenable A1_pp
```

9. Try printing a job using the printer, `listing_A`:

```
# pdpr -p listing_A /etc/motd
# pdq -p listing_A
```

You can also pipe `stdin` to `pdpr`:

```
# cat /etc/motd | pdpr -p listing_A
```

Setting Default Attributes

You can associate default attributes with a logical printer so that each print job uses those defaults. In the following example, each logical printer has a default medium defined. Print jobs submitted to a logical printer are automatically tagged to require the default medium. The medium specifies the kind of paper, the size of paper, or the form. The ISO-DPA standard defines over seventy standard media types; you can also define your own. Refer to Appendix A in the *Advanced Printing Software System Administration and Operation Guide*.

Most useful attributes are placed in the `initial-value-document` object rather than the `initial-value-job` object. It is suggested that you set them up in pairs. The naming conventions used in the example are the ones that the `pdprintadmin` GUI accepts. That is, a printer's initial value objects are name `<printername>_IVJ_DEFAULT` and `<printername>_IVD_DEFAULT`.

The steps in the example that follows set up the initial-value objects using the `pdcreate` command and associates the initial-value objects with the logical printers using the `pdset` command. Once the initial-value objects are set up and associated with the logical printers, the `pdset` command sets up the specific defaults, the media-supported attribute on all printer

objects, and the media-ready attribute on the physical printers. Lastly, the printers are enabled.

To set the default attributes:

1. Set up the initial-value objects:

```
# pdcreate -c i-v-j my_spl:listing_A_IVJ_DEFAULT
# pdcreate -c i-v-d my_spl:listing_A_IVD_DEFAULT
# pdcreate -c i-v-j my_spl:invoice_A_IVJ_DEFAULT
# pdcreate -c i-v-d my_spl:invoice_A_IVD_DEFAULT
# pdcreate -c i-v-j my_spl:letter_A_IVJ_DEFAULT
# pdcreate -c i-v-d my_spl:letter_A_IVD_DEFAULT
```

2. Associate the initial-value objects with the logical printers:

```
# pdset -c printer -x printer-i-v-j=listing_A_IVJ_DEFAULT listing_A
# pdset -c printer -x pr-i-v-d=listing_A_IVD_DEFAULT listing_A
# pdset -c printer -x pr-i-v-j=invoice_A_IVJ_DEFAULT invoice_A
# pdset -c printer -x pr-i-v-d=invoice_A_IVD_DEFAULT invoice_A
# pdset -c printer -x pr-i-v-j=letter_A_IVJ_DEFAULT letter_A
# pdset -c printer -x pr-i-v-d=letter_A_IVD_DEFAULT letter_A
```

3. Set up specific defaults:

```
# pdset -c i-v-d -x default-medium=fanfold my_spl:listing_A_IVD_DEFAULT
# pdset -c i-v-d -x default-medium=invoice my_spl:invoice_A_IVD_DEFAULT
# pdset -c i-v-d -x default-medium=letterhead my_spl:letter_A_IVD_DEFAULT
# pdset -c i-v-d -x new-line-option=lf my_spl:listing_A_IVD_DEFAULT \
 my_spl:invoice_A_IVD_DEFAULT my_spl:letter_A_IVD_DEFAULT
```

4. Set up the media-supported attribute on all printer objects:

```
# pdset -c p -x media-supported="fanfold invoice letterhead" listing_A \
 invoice_A letter_A A1_pp A2_pp
```

5. Set the media-ready attribute on the physical printers (printer `A1_pp` has fanfold paper loaded; printer `A2_pp` has invoice forms loaded):

```
# pdset -c p -x media-ready=fanfold A1_pp
# pdset -c p -x media-ready=invoice A2_pp
```

6. Enable the printers.

If you submit a mix of jobs (invoices, letterhead, fanfold, listings, and so on), the jobs should flow to the physical printer that has the appropriate form loaded. If the appropriate form is not loaded, the job is held in the queue. For example, jobs submitted to logical printer, `letter_A` will be held in the queue when the physical printers do not have letterhead paper loaded.

When the printer operator changes forms on a printer, the operator must disable the physical printer to prevent jobs from being printed on it, change the forms, change the media-ready attribute of that physical printer to reflect the change, and reenable the physical printer.

The pddisable and pdenable commands can be performed by a designated printer operator. They do not need to be performed by an administrator.

To change the media of printer A1_pp from fanfold to letterhead:

1. Disable the physical printer A1_pp:

```
# pddisable A1_pp
```

2. Change the forms in printer A1_pp to letterhead.
3. Set media-ready=letterhead on the physical printer and enable it:

```
# pdset -x media-ready=letterhead A1_pp
# pdenable A1_pp
```

Jobs requiring the letterhead media will begin to print on physical printer A1_pp. Jobs submitted to logical printer invoice_A continue to flow to printer A2_pp, because it has invoice media loaded. Fanfold jobs submitted to logical printer listing_A will wait in the queue until an operator sets media-ready=fanfold on one of the physical printers.

Creating Printer Setup Modules

Sometimes a form requires a printer to print text in a specific font size or printer mode. A printer setup module is a file that the supervisor prepends to the document stream when printing. The command sequences in the module select the printing mode or format that you want. Refer to the printer's programming documentation to find the escape sequences required and add them to a printer setup module.

To create a printer setup module:

1. Create the printer setup file in a user directory.
2. Create a new directory and copy the printer setup file into it. For example, /usr/local/pd/setup.
3. Set the cfg-prologue-path attribute on the supervisor to indicate where the setup file is stored:

```
# pdset -c server -x cfg-prologue-path=/usr/local/pd/setup my_sup
```

Consider the following example. You have created the following setup module files to cause your PCL-compatible printer to print in various font sizes:

Setup Module	Lines per Inch	Characters per Inch
6x8.pcl	6	8
6x10.pcl	6	10
6x12.pcl	6	12
6x12_hq.pcl	6	12
8x12.pcl	8	12

1. Create the PCL sequence:

   The escape sequence files must be binary files with no trailing carriage return or line feed, if one is not wanted. Consult your printer's technical or programming documentation, supplied by the printer vendor, for specific control sequences that apply to your printer.

   To remove newline characters, use the `tr` command:

   ```
   mv 6x8.pcl 6x8.tmp
   tr -d '\n' <6x8.tmp >6x8.pcl
   ```

   Put the setup sequences you need into individual setup files. Store them in your setup directory.

2. Add the corresponding `printer-setup-module` attributes to the `initial-value-document` objects to print the listings with the 6x10 font, invoices with the 6x12 high quality font, and the letterhead with the standard 6x12 font:

   ```
   # pdset -c i-v-d -x printer-setup-module=6x10.pcl \
   my_spl:listing_A_IVD_DEFAULT
   # pdset -c i-v-d -x printer-setup-module=6x12_hq.pcl \
   my_spl:invoice_A_IVD_DEFAULT
   # pdset -c i-v-d -x printer-setup-module=6x12.pcl \
   my_spl:letterhead_A_IVD_DEFAULT
   ```

3. Verify the setup by displaying the attributes and values of the print system objects:

   ```
   # pdls -c i-v-d -r 'i-v-d-ident def-med pri-setup-mod' my_spl:

   initial-value-document-identifier default-medium printer-setup-module
   --------------------------------- -------------- --------------------
   listing_A_IVD_DEFAULT             fanfold        6x10.pcl
   invoice_A_IVD_DEFAULT             invoice        6x12_hq.pcl
   letter_A_IVD_DEFAULT              letterhead     6x12.pcl

   # ls -l /usr/local/pd/setup

   total 1
   -rw-r--r-- 1 root system 14 Jun 8 15:53 6x10.pcl
   -rw-r--r-- 1 root system 14 Jun 8 15:53 6x12.pcl
   -rw-r--r-- 1 root system 14 Jun 8 15:53 6x12_hq.pc1
   ```

```
# pdls -c server -r cfg-prolog-path my_sup

cfg-prologue-path
-----------------
/usr/local/pd/setup
```

When you print to the printers, `invoice_A`, `listing_A`, or `letter_A`, the jobs will be printed on the appropriate printer in the desired type fonts because the printers receive the setup escape sequences prior to the document data.

If you need to further modify the document format, consider using a filter program or script to set up a `filter-definition` on the supervisor and use the `translation-filter` or `modification-filter` attribute in either the `initial-value-document` object or with your print request. Refer to the *Advanced Printing Software System Administration and Operation Guide* for information about data filtering.

Text Line Termination

If you are printing a UNIX text file where the lines are terminated by newline characters, the printer might print the lines of text in stair-step fashion, off the page. Either the print software has to insert carriage return characters or, at the front panel of the printer, set the printer to interpret newlines as CR/LF.

Advanced Printing Software Version 1.1 has a document attribute, `new-line-option` that allows you to specify how newlines are to be treated. With each print request, include `new-line-option=lf`, or set it in the `initial-value-document` object.

Using Legacy Print Commands

Some applications contain a hard-coded reference to the `lp` or `lpr` UNIX commands, and they do not provide a way to submit jobs using the `pdpr` command.

To use the `lp` or `lpr` commands:

1. Set up the Inbound Gateway client daemon with the `/usr/pd/scripts/inbound_gw_config.sh` shell script. If your system is running Tru64Cluster Server software, run the daemon on all cluster members.
2. Set up a `printcap` entry for each logical printer.

   Use either `printconfig` or `lprsetup`, or edit the `printcap` file.

3. Configure the printers as remote and set the remote machine (rm) characteristic to `@dpa`.

   For example:

```
# Printcap entries for printers invoice_A, listing_A and letter_A

lp0|invoice_A:\
    :lp=:\
    :rm=@dpa:\
    :rp=invoice_A:\
    :sd=/var/spool/printer/invoice_A:\
    :mx#0:
lp1|listing_A:\
    :lp=:\
    :rm=@dpa:\
    :rp=listing_A:\
    :sd=/var/spool/printer/listing_A:\
    :mx#0:
lp2|letter_A:\
    :lp=:\
    :rm=@dpa:\
    :rp=letter_A:\
    :sd=/var/spool/printer/letter_A:\
    :mx#0:
```

4. Create spool directories that match the `printcap` entries' sd characteristics and give them daemon ownership:

```
# mkdir /var/spool/printer/invoice_A
# mkdir /var/spool/printer/listing_A
# mkdir /var/spool/printer/letter_A
# chown daemon:daemon /var/spool/printer/invoice_A
# chown daemon:daemon /var/spool/printer/listing_A
# chown daemon:daemon /var/spool/printer/letter_A
```

5. Start and enable the queues:

```
# lpc start invoice_A listing_A letter_A
# lpc enable invoice_A listing_A letter_A
```

The print queue definitions in the `printcap` file send the print jobs to logical printer targets. The Advanced Printing Software spooler schedules the jobs to print on the appropriate physical device. When the `printcap` entries and spool directories are in place and the inbound gateway daemon is running, users and applications can use the `lp` or `lpr` command to print invoices or other form-specific data. For example:

```
# lpr -P invoice_A invoices.txt
```

or

```
# lp -d invoice_A invoices.txt
```

Comments and Questions

We value your comments and questions on the information in this document. Please mail your comments to us at this address:

best_practices@zk3.dec.com

Legal Notice