PrintKit® Network Print Services

pkdoc 11-Sep-08 23:08, cover

PrintKit®

Version 2.3
September 2008

User Guide

Northlake Software
Portland, Oregon

© 1993 – 1997, 2008 by Northlake Software, Inc. All rights are reserved. No part of this document may be photocopied, reproduced, or translated to another language without the prior written consent of Northlake Software, Inc.

Northlake Software, Inc. disclaims any implied warranties with regard to this material or products described herein, including, but not limited to, the merchantability and fitness for a particular purpose. Northlake Software shall not be liable for errors contained herein or for incidental consequential damages in connection with the furnishing, performance, or use of this material. The information in this document is subject to change without notice and should not be construed as a commitment by Northlake Software. This document contains proprietary information which is protected by copyright.

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

PrintKit is a registered trademark of Northlake Software, Inc.

Printed in the United States of America.

Quick Start Installation

Chapter 1 **Printing with PrintKit**

1.2 Introduction
1.4 Printing from OpenVMS
The OpenVMS PRINT command
DECwindows FileView print dialog box
1.6 Capabilities of the PrintKit software
Document data types
Paper handling and document finishing
Page layup
1.7 How to use this manual
1.8 Related documentation

Chapter 2 **Using PrintKit**

2.2 Introduction
2.3 Selecting the print queue
2.3 Selecting the data type for the file
2.4 Printing multiple copies
2.5 Printing two-sided documents
2.6 Selecting paper stock and manual feed
DEFAULT_MEDIUM parameter
PAGE_MEDIA parameter
SHEET_SIZE parameter
INPUT_TRAY parameter
MEDIUM_SUBSTITUTION parameter
2.9 Selecting the output tray
2.9 Controlling page layout
Specifying page orientation
Scaling pages
Combining multiple pages per sheet
Complex imposition controls
2.12 Controlling document finishing
2.13 Selecting a document specification

2.13 Controlling page format with setup modules and forms
- Where setup modules are stored
- How setup modules are named
- Specifying setup modules

Chapter 3 **Managing the Print Job**

3.2 Introduction
3.2 Using fonts
3.2 Controlling flag page printing
3.3 Selecting files for printing
3.3 Scheduling print jobs
3.4 Monitoring the status of a print job

Chapter 4 **Configuring PrintKit**

4.2 Introduction
4.3 Determining the PrintKit version
4.3 Examining PrintKit queue configuration
4.4 PrintKit communications configuration
- Supported TCP/IP interfaces
- Assigning internet addresses
- Printer lpr/lpd queue names
- LAT node and port names

4.6 Execution queues
- Determining printer model and configuration
- Adding, modifying, and removing queues

4.8 Generic and logical queues
4.9 Document and subsidiary specifications
4.10 Document specifications
4.12 Medium specifications
4.13 Imposition specifications
4.14 Finishing process specifications
4.15 Sheet package specifications
4.17 Saving the PrintKit configuration
4.18 Managing print forms

Managing form definitions
Assigning forms to a queue
4.20 Managing device control libraries
Setup modules
Examining library contents
Adding new library modules
Adding new libraries
4.23 Removing the PrintKit software

Chapter 5 **Troubleshooting PrintKit**

5.2 Introduction
5.2 Files affected by the installation
5.2 System requirements
OpenVMS system version
Disk space
SYSGEN parameters
5.3 Commands for managing the print queue
5.4 Where to start troubleshooting
5.4 Enabling error and status messages
5.5 Printer status
5.5 Stopping execution queues
5.6 Verifying your configuration
Queue configuration
5.7 Verifying the network connection

Chapter 6 **Error and Status Messages**

6.2 Introduction
6.2 Operator messages
6.3 Notification messages
6.5 Queue status
6.6 Job status
6.7 Message log file
6.7 OpenVMS system messages

Appendix A **PRINT Command Reference**

A.2 Introduction
A.2 Command qualifiers
A.9 Parameter options
A.23 Page selection expressions

Appendix B **KITCP Command Reference**

B.2 Introduction
B.2 Version numbers
B.3 Commands

Appendix C **ANSI-PPL3 Initial State and Fonts**

C.2 Initial state settings
C.6 Font sets
Standard font sets

Appendix D **ANSI-PPL3 Page Format Controls**

D.2 Introduction
Coding control sequences
D.3 Page coordinate system
Setting the page orientation
Changing the margins
Changing column and line spacing
Changing the origin
D.7 Selecting fonts
Selecting an SGR parameter
Selecting font size
Assigning fonts to an SGR parameter

Appendix E **Forms Overlays**

E.2 Introduction

Forms overlay definition requirements

Creating forms overlays

Quick Start Installation

Here are the steps required to install and configure the PrintKit software. You perform the installation using the VMSLICENSE and VMSINSTAL commands. After installing the software, you use the KITCP command, supplied with PrintKit, to configure the software for the printers on your system. Once installation and configuration are complete, PrintKit queues are available to all users on the system.

If you are familiar with typical OpenVMS installation procedures, these instructions will give you a "quick start" with the PrintKit software. If you need additional information about installing or configuring the software, or you encounter problems, these instructions will direct you to additional information in later chapters.

Installing PrintKit

To install PrintKit you must be logged in to the SYSTEM account, or another account with privileges to create and modify files in the system directories. You can install the software on an active OpenVMS system without disturbing other operations.

The PrintKit VMSINSTAL kit verifies that your system configuration allows the software to operate successfully, or tells you how to make any required changes. Changing your system's SYSGEN parameters may require a reboot. If you prefer to check your configuration before performing the installation, refer to System requirements, page 5.2.

Step 1. register and load License PAK

Before installing PrintKit, use the VMSLICENSE command to register and load the license Product Authorization Key (PAK) supplied with your distribution kit.

If you are replacing an existing PAK in the course of upgrading your PrintKit software, it is safest to delete the old PAK before registering the new one. When you do this, be sure all PrintKit queues are stopped, so the PAK is not in use. Otherwise, you may end up with

[1]
installation
KITCP Administration Utility
configuration
installation
configuration
installation
SYSGEN parameters
License PAK

an incompletely deleted license that interferes with the newly registered license. Once the queues are stopped, use the following command to delete the license:

```
$ LICENSE UNLOAD PRINTKIT /PRODUCER=NORTHLAKE
$ LICENSE DELETE PRINTKIT /PRODUCER=NORTHLAKE
```

To load your new PrintKit license PAK, use the VMSLICENSE command:

```
$ @SYS$UPDATE:VMSLICENSE
```

When the license procedure prompts you, select option 1 and respond to the questions using the information from your PAK.

Step 2. install the PrintKit software

The VMSINSTAL command prompts you for the information needed to install PrintKit on your system. The installation displays progress messages, and upon completion it lists the files it has created or modified.

To start VMSINSTAL, enter the following command. If you are installing from CD-ROM media, make the appropriate substitution for the device CD when you enter the VMSINSTAL command. If you are installing a downloaded kit, specify the directory where the installation savesets are located instead of CD:[PRINTKIT023.KIT]. Specify OPTIONS N to review the release notes before performing the installation.

```
$ @SYS$UPDATE:VMSINSTAL PRINTKIT023 CD:[PRINTKIT023.KIT]
```

Here are some tips to remember during the installation:

> Enter a question mark (?) after any prompt to display help information.

> Most prompts display a default response, enclosed in brackets ([]), which you can select by pressing RETURN.

> You may cancel the installation at any time by pressing CTRL-Y. The system will do some cleanup, which may take some time. *Do not type* CTRL-Y *during the cleanup* – this may stop the cleanup before it has removed all files from the installation, leaving an unusable combination of new and old files.

VMSINSTAL asks you a series of questions as it installs the PrintKit software:

> Are you logged in to the SYSTEM account?

> Have you backed up your system disk?

[1]
Installation Verification Procedure (IVP)
installation
messages
release notes

> Have you registered and loaded your license PAKs?

> Do you want to purge old PrintKit files?

> Do you want to run the Installation Verification Procedure?

> Do you want to copy definitions from your existing database? (Asked only when upgrading from a pre-2.3 version of PrintKit.)

For more information on the VMSINSTAL utility, refer to the *OpenVMS System Manager's Manual: Essentials*. For PrintKit's system requirements and a list of the files created or modified by the installation, refer to Chapter 5, Troubleshooting PrintKit.

If you answered YES when asked if you wanted to run the Installation Verification Procedure (IVP), your installation will be verified immediately after the installation process is complete.

Step 3. review results of installation verification procedure

The Installation Verification Procedure (IVP) creates a test queue (directed to a null output, so it can be used without a printer connected), prints jobs to it, and then removes it. You should see messages announcing the startup of the test queue, and submit and completion messages for each test job.

You can run the IVP at any time using the PRINTKIT_IVP command:

```
$ @SYS$TEST:PRINTKIT_IVP
```

For help with any problems during the IVP, refer to Chapter 5, Troubleshooting PrintKit.

This completes the installation of the software. To configure PrintKit queues, proceed to the next section.

Configuring PrintKit queues

To configure PrintKit queues, you must be logged in to the SYSTEM account, or another account with privileges to perform operator functions. You can configure queues on an active OpenVMS system without disturbing other operations.

If you need more information on managing OpenVMS print queues, refer to the *OpenVMS System Manager's Manual: Essentials*.

[1]
Installation Verification Procedure (IVP)
installation
messages
managing queues

Step 1. get ready to configure

Make sure you have done the following before starting to configure a PrintKit queue:

Enable operator messages The PrintKit software reports configuration and printer communications problems using operator messages. These messages are an important source of information when troubleshooting problems, particularly when configuring a new queue.

Enter the following command to enable delivery of operator messages to your terminal:

```
$ REPLY/ENABLE=PRINTER
```

Define the KITCP command The KITCP administration utility, provided with the PrintKit software, lets you create, display, and modify PrintKit queues. Define the KITCP command as follows:

```
$ KITCP :== $KITCP
```

Step 2. collect configuration information

Your printer should be configured and connected to your network before you configure its PrintKit queue.

Once your printer is configured and connected, generate a configuration page on the printer. You will use it in creating the PrintKit queue. Consult your printer's User Manual for instructions on producing the configuration page.

Step 3. start KITCP

If you enter the KITCP command with no arguments, it responds with a KITCP> prompt. You can then enter the commands you need to configure PrintKit queues.

```
$ KITCP
KITCP>
```

You can also enter a command directly on the KITCP command line. When the command is finished, KITCP returns you directly to the system prompt.

```
$ KITCP ADD QUEUE /INTERACTIVE queue_name
```

KITCP accepts normal DCL command qualifiers, but it can also be told to prompt interactively for the information it needs. If you are configuring your PrintKit queues for the first time, you will probably find it much easier to use the /INTERACTIVE option when entering KITCP commands. Used interactively, KITCP will supply default values for most queue settings, and will prompt you for those that cannot be supplied from defaults, such as the printer's model and address.

[1]
KITCP Administration Utility
configuration
messages
operator messages
troubleshooting
configuration

Adding a queue interactively, you'll be shown the default setting for each prompt. You can enter a "?" at any prompt to get more information. If the answer is one of a list of choices, you can browse through the list using the up and down arrows.

Step 4. create print queue

Create a print queue for your printer using the KITCP's ADD QUEUE command, where queue_name is the name you have chosen for this print queue:

```
$ KITCP ADD QUEUE /INTERACTIVE queue_name
```

KITCP asks you a series of questions; the exact questions asked vary depending on the type of printer and interface being configured. Type a "?" at any prompt for more information, and use the up and down arrows to browse through available choices.

Step 5. check queue configuration

You can check the settings of the queue you have created using KITCP's SHOW QUEUE command:

```
$ KITCP SHOW QUEUE /FULL queue-name
```

The /FULL option displays all configuration settings, not just those you were prompted for by the /INTERACTIVE option.

Step 6. adjust queue settings

The KITCP ADD QUEUE /INTERACTIVE command prompts you for the settings that configure a queue for a particular printer; it initializes the rest with their default values. If you make a mistake when adding a new queue, or simply wish to make changes, use the KITCP MODIFY QUEUE command. Like ADD QUEUE, MODIFY QUEUE also has an /INTERACTIVE option.

The PrintKit software normally selects North American Letter size paper by default. If your locale uses A4 paper, you will need to configure this as the default for your queues:

```
$ KITCP MODIFY QUEUE queue_name /DEFAULT=SHEET_SIZE=A4
```

This should be enough to get you started with a typical configuration. However, there is more to KITCP. For details on configuring PrintKit, refer to Chapter 4, Configuring PrintKit; for descriptions of all the KITCP commands, refer to Appendix B, KITCP Command Reference.

[1]
KITCP Administration Utility
ADD QUEUE KITCP command
KITCP Administration Utility
SHOW QUEUE command
configuration
KITCP Administration Utility
ADD QUEUE KITCP command
MODIFY QUEUE KITCP command
configuration

Step 7. start the queue

To start the queue:

```
$ START/QUEUE queue-name
```

Step 8. verify queue operation

You can run the Installation Verification Procedure on the queue to verify its operation. The IVP prints test files to verify the correct operation of the queue:

```
$ @SYS$TEST:PRINTKIT_IVP queue-name
```

Step 9. make it automatic

The final step in configuring a new queue is to ensure that it starts correctly when your OpenVMS system is restarted. You can configure PrintKit queues as autostart queues, so they are started automatically by the ENABLE AUTOSTART command, or you can start them individually using START/QUEUE commands.

To start the queue explicitly during system startup, add a START/QUEUE command to your System Startup File, SYS$MANAGER:SYSTARTUP_VMS.COM.

To autostart the queue, use KITCP to set the autostart attribute for the queue.

```
$ KITCP MODIFY QUEUE queue-name /AUTOSTART_ON=node::device:
```

See the *OpenVMS System Manager's Manual: Essentials* for more information on setting up autostart queues.

Make sure your system startup procedure initializes any communications interfaces required by PrintKit queues before the queues are autostarted or started by individual START/QUEUE commands. The PrintKit software attempts to verify its printer connection when a queue is started, and it will fail if the communications interface is not yet initialized.

For a print queue using a LAT interface, make sure you add the LATCP CREATE PORT for the queue's LAT device to the system startup procedure.

This completes the queue configuration.

[1]
Installation Verification Procedure (IVP)
KITCP Administration Utility
AUTOMATIC data type
START/QUEUE command
autostart queue
configuration
LAT network protocol

Chapter 1
Printing with PrintKit

1.2 Introduction

1.4 Printing from OpenVMS

- The OpenVMS PRINT command
- DECwindows FileView print dialog box

1.6 Capabilities of the PrintKit software

- Document data types
- Paper handling and document finishing
- Page layup

1.7 How to use this manual

1.8 Related documentation

Introduction

The PrintKit Network Printing Services software extends OpenVMS printing to a diverse range of third-party printers. The PrintKit Production Printing Services software (Production PrintKit) adds features to meet the specific needs of high-volume, print-on-demand, and other production printing environments.

If you're used to OpenVMS printing and the DECprint architecture, PrintKit will look very familiar. It supplies the same user interface as the DECprint Supervisor software, including advanced layup controls. PrintKit also provides a full implementation of the ANSI-PPL3 (LN03) document data type, including its soft fonts, sixel graphics, and paper handling controls; it also implements the listing data type. Where it has been necessary to extend the user interface beyond that provided by DECprint, PrintKit relies on the ISO Document Printing Application Standard (ISO/IEC DP 10175), the standard from which the DECprint user interface is also derived.

PrintKit began as a product for HP LaserJet printers, Hewlett-Packard's market-leading work group and personal page printers. Laserjets have diverse paper handling features, PostScript or PCL5 data types (or both), PJL job controls, and sharable network interfaces. So, from the beginning, PrintKit's design anticipated a broad range of hardware configurations, emulation data types, printer job control, and communications interfaces.

To provide adaptability, the PrintKit software configures itself using PostScript Printer Description files, a standard developed by Adobe Systems. A PPD file describes the features of a particular printer and provides the printer-specific details required to control the printer. PrintKit ships with a base set of PPD files, and printer manufacturers routinely provide PPD files for new printers.

Network interfaces have become the preferred connection for work group printers, and are attractive for personal printers as well. However, as a practical matter, many existing printers used from OpenVMS rely on serial interfaces, either through direct connections or terminal servers. PrintKit supports both styles of communications, the old and the new.

PrintKit lets you use printers with TCP/IP Ethernet interfaces, such as HP LaserJets with JetDirect cards, straight out of the box. It is compatible with all the major layered OpenVMS TCP/IP network software. PrintKit can also perform the TCP/IP processing itself, using Northlake's own Printer Ethernet Protocol technology. Tailored for printer communications, PEP is simpler to configure and maintain than a general-purpose network interface – in fact, simpler than a LAT or serial link.

[1]
ANSI data type
LIST data type
PCL data type
data type
DECprint Supervisor
Job Control Program
LaserJet printers
LAT network protocol
page layout
paper handling
PostScript Printer Definition (PPD) files
symbiont
TCP/IP network protocol

PrintKit's LAT interface is compatible with built-in printer LAT network interface cards and with printers connected to LAT servers. It provides job control if bi-directional communications are present, and functions uni-directionally as well. It is tuned for the high-volume data transfers required for printing.

In summary, the PrintKit software links together the pieces of an open systems printing solution. It does a lot, but is designed to keep a low profile. It matches the full DECprint user interface, instead of inventing one of its own. Behind the scenes are a set of robust communications interfaces that perform quietly, but help you track down problems when they occur.

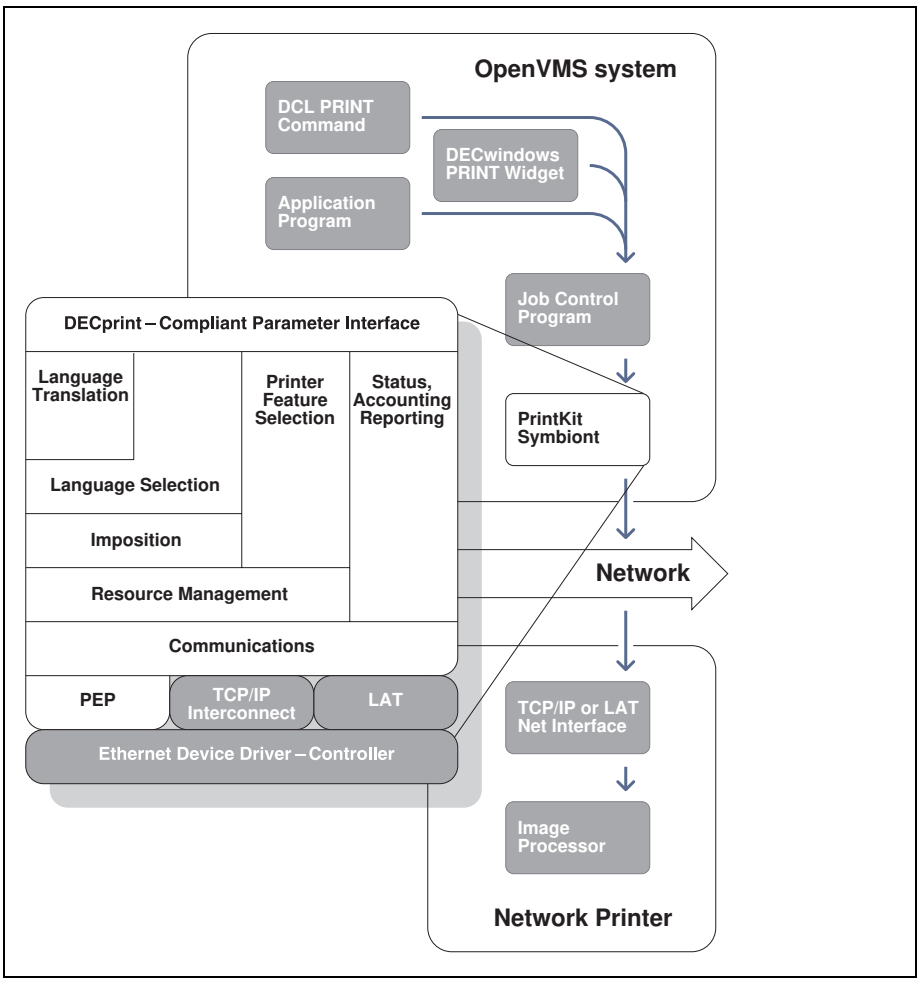


Figure 1-1, PrintKit Job Flow

Printing from OpenVMS

Typically, you print documents from OpenVMS using one of the following:

- The OpenVMS PRINT command at the DCL command line
- DECwindows FileView print dialog box

[1]
DECwindows FileView Print Dialog
PRINT command

The OpenVMS PRINT command

PrintKit is fully compatible with the OpenVMS PRINT command – PRINT options remain the same with the installation of the PrintKit software. Refer to the *OpenVMS DCL Dictionary* or the *OpenVMS User's Manual* for a detailed description of the OpenVMS PRINT command, including syntax rules and available options.

PrintKit uses the /PARAMETERS qualifier to provide an additional set of controls. Parameters are used in the same way by Digital's DECprint Supervisor software, and hence this interface is commonly called the DECprint parameter interface.

DECwindows FileView print dialog box

The DECwindows FileView print interface allows you to select print options from a dialog box (shown in Figure 1-2, DECwindows FileView Print Dialog, page 1.5).

This interface, however, assumes that you are using a standard Digital printer, and therefore displays options that may not be relevant to PrintKit or your printer.

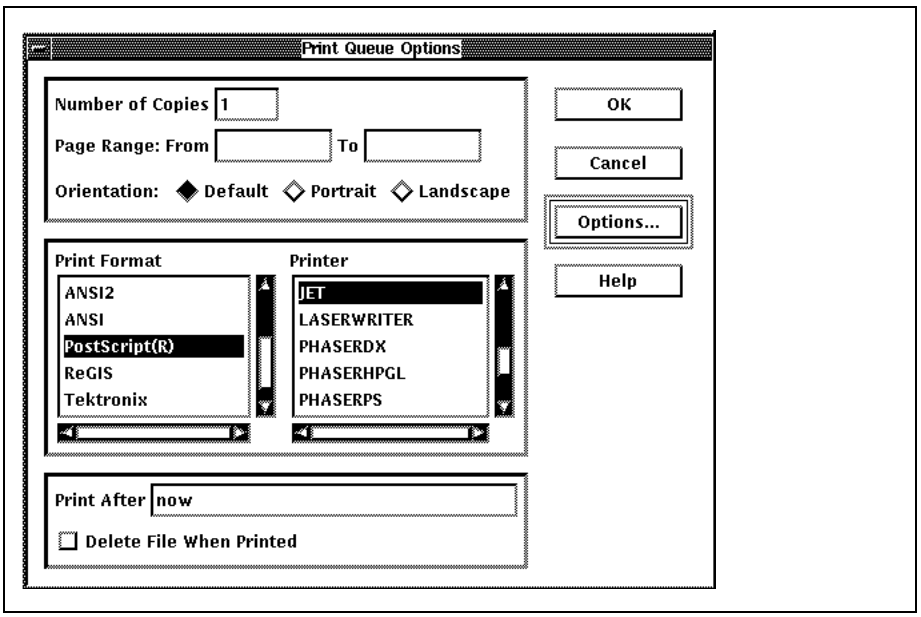


Figure 1-2, DECwindows FileView Print Dialog

For more information about printing from the DECwindows print dialog box, see the *OpenVMS DECwindows Motif User's Guide*.

[1]
/PARAMETERS qualifier
DECprint Supervisor installation
PRINT command
DECwindows FileView Print Dialog

Capabilities of the PrintKit software

The PrintKit software implements all the standard OpenVMS printing functions. Also, it provides additional functions to let you exploit the capabilities of current page-printing technology. These include support for multiple document data types, paper handling, page layup operations, and document finishing controls. Production PrintKit, moreover, supports more extensive paper handling controls.

Document data types

PrintKit can process the following data types for printing:

Digital ANSI-Compliant Printing Protocol Level 3 (Often referred to as LN03 emulation, after the Digital printer for which it was first implemented.) PrintKit translates ANSI-PPL3 to PostScript for printing.

HP-GL/2 Transferred to the printer for interpretation and printing.

Listing Text file in a simple listing format. The text is assumed to be encoded using the ISO Latin 1 character set. The formatting paginates the text, adds page headings, line numbers and a graybar overlay, and wraps lines longer than the page width. PrintKit translates listings to PostScript for printing.

PCL5 Transferred to the printer for interpretation and printing.

PostScript Both ordinary PostScript and Encapsulated PostScript documents are transferred to the printer for interpretation and printing. Encapsulated PostScript processing makes adjustments to ensure that all pages print.

Normally, PrintKit automatically selects the document data type based on document content. However, you can also directly control the document data type.

Paper handling and document finishing

PrintKit provides the following paper handling and document finishing features:

Paper Selection Select paper stock by attributes (size, color, weight, and type) or input tray. Production PrintKit provides additional paper selection controls, including multiple selections within a document, and the ability to substitute paper stocks for input tray selections encoded in the document being printed.

Overlays Display a forms overlay on a paper stock, such as graybars or replacements for specialized pre-printed stocks.

Duplexing Print one-sided or two-sided, normal orientation or "tumbled."

Partial Document Printing Print a selected part of a document.

[1]
data type
page layout
paper handling
ANSI data type
LIST data type
PCL data type
POSTSCRIPT data type
data type
Encapsulated PostScript (EPSF)
overlay
INPUT_TRAY parameter
duplex printing
overlay
paper handling

Output Tray Selection Direct the document to a particular printer output tray.

Finishing Staple, bind, fold, and punch the printed document, on printers which have these finishing capabilities.

Page layup

PrintKit provides the following page layup controls:

Page Scaling Adjust the size of the printed image to fit a smaller paper size.

Margins Realign the printed image or allow space for binding.

Multi-Up Printing Combine multiple pages onto a single page.

Page layup controls are implemented using PostScript, and hence are available only for the PostScript, ANSI-PPL3 and listing data types.

How to use this manual

This manual describes how to install and use the PrintKit software. It serves for both the standard and Production PrintKit versions of the software. Production PrintKit includes all the features of the base PrintKit software, so most of the manual applies to both products. Features specific to Production PrintKit are noted in the text that describes them.

Quick Start Installation, at the beginning of the manual, describes the basic PrintKit software installation and configuration procedures. You'll probably only need to glance at that chapter. In addition to installing the software, the Install procedures check to make sure your system is configured to allow PrintKit to operate correctly. To help you configure the software, checklists are provided for the most common printer configurations.

Chapter 1, Printing with PrintKit, the chapter you're reading now, provides an overview of the PrintKit software.

Chapter 2, Using PrintKit, describes how to use the most common PrintKit features to print your documents. For example, this chapter describes how to specify multiple copies, how to specify page size and orientation, and how to select the input and output trays.

Chapter 3, Managing the Print Job, describes how to manage your print job, including how to schedule and monitor print jobs, use fonts, and recover from printer and system errors.

[1]
ANSI data type
LIST data type
binding orientation
data type
page layout
page scaling
KITCP Administration Utility
ANSI data type
configuration
copies
error messages
font set
installation
messages
release notes
troubleshooting
version, software

Chapter 4, Configuring PrintKit, describes how to configure the PrintKit software. It goes into more detail than Quick Start Installation, and shows how to use the KITCP administration utility to examine and modify your PrintKit configuration.

Chapter 5, Troubleshooting PrintKit, is a troubleshooting guide, and Chapter 6, Error and Status Messages, contains information on error messages.

Appendix A, PRINT Command Reference, defines all the print qualifiers and parameters you can use with PrintKit.

Appendix B, KITCP Command Reference, defines all the commands you can use with KITCP.

Appendix C, ANSI-PPL3 Initial State and Fonts, specifies the state settings in effect for the ANSI-PPL3 translator at the beginning of a document, and describes the font sets supplied for ANSI-PPL3 documents.

Appendix D, ANSI-PPL3 Page Format Controls, provides an introduction to ANSI-PPL3 page format controls, and describes control sequences commonly used to modify page orientation, margins, column and line spacing, and the font selected for printing.

Appendix E, Forms Overlays, describes PrintKit forms overlays and how to create them.

Related documentation

The following manuals from the OpenVMS Documentation Set provide additional information on topics discussed in this manual:

> *OpenVMS DCL Dictionary*, AA-PV5KA-TK & AA-PV5LA-TK (two volumes). Provides detailed reference information for all DCL commands.

> *OpenVMS System Manager's Manual: Essentials*, AA-PV5MA-TK. Describes procedures needed to set up print queues and manage their daily operation.

> *OpenVMS System Management Utilities Reference Manual*, AA-PV5PA-TK & AA-PV5PA-TK (two volumes). Provides detailed reference information on system management utilities.

These documents are the standard references for the document data types used with the PrintKit software:

[1]
ANSI data type
PCL data type
data type
page layout
related documentation

> Adobe Systems Incorporated, *PostScript Language Reference Manual*, Second Edition, Addison Wesley, Reading, Massachusetts, 1990. The definitive PostScript language specification. Also includes Version 3.0 of the Document Structuring Conventions Specification, and Version 3.0 of the Encapsulated PostScript Files Specification.

> *Digital ANSI-Compliant Printing Protocol Level 3 Programming Reference Manual*, AA-PBWGA-TE, Digital Equipment Corporation, Maynard, Massachusetts, 1990. The definitive ANSI-PPL3 language specification. Also provides character set layout charts.

> *PCL 5 Printer Language Technical Reference Manual*, Hewlett-Packard, 1992. The PCL5 language specification.

Chapter 2
Using PrintKit

2.2 Introduction
2.3 Selecting the print queue
2.3 Selecting the data type for the file
2.4 Printing multiple copies
2.5 Printing two-sided documents
2.6 Selecting paper stock and manual feed
- DEFAULT_MEDIUM parameter
- PAGE_MEDIA parameter
- SHEET_SIZE parameter
- INPUT_TRAY parameter
- MEDIUM_SUBSTITUTION parameter

2.9 Selecting the output tray
2.9 Controlling page layout
- Specifying page orientation
- Scaling pages
- Combining multiple pages per sheet
- Complex imposition controls

2.12 Controlling document finishing
2.13 Selecting a document specification
2.13 Controlling page format with setup modules and forms
- Where setup modules are stored
- How setup modules are named
- Specifying setup modules

Introduction

This chapter describes how you use PrintKit to perform your printing tasks. For a complete discussion of the command interface that PrintKit uses, see Appendix A, PRINT Command Reference.

The PRINT command allows you to print files. It does this by creating a *print job* out of the files and specifications you provide and then sends the job for printing. In the background, the OpenVMS system routes and schedules the job for printing. The result is the set of printed documents described by the job.

When you use the PRINT command, you complete the specification of the print job by using command *qualifiers*. When you use the PrintKit software, an additional set of controls are also available to you. They allow you to specify additional job characteristics not anticipated by the standard PRINT qualifiers.

You specify these added controls using the PRINT /PARAMETER qualifier. The /PARAMETER qualifier is used in the same way by Digital's DECprint Supervisor software, and is commonly called the *DECprint parameter interface*. The controls themselves are called *parameters*, to distinguish them from qualifiers, which are the controls provided by the standard PRINT command. The parameters provided by the PrintKit software include the parameters provided by DECprint Supervisor, so you can use the same job specifications for PrintKit as you do for DECprint Supervisor.

Parameter Handling for Multi-File Jobs

Some PRINT qualifiers apply to the entire print job, and others you can specify separately for each file in the job. The /PARAMETER qualifier, in particular, always applies to the entire job. Therefore, parameters also always apply to the entire job, and cannot be specified separately for individual files.

Features Depend on Printer Capabilities

Some PrintKit controls require particular printer features or capabilities. Be sure your printer is configured to support the controls you are trying to use.

[1]
DECprint Supervisor
PRINT command

Selecting the print queue

The /QUEUE qualifier allows you to print files on the specified queue. This qualifier applies to the entire print job. For example, to print a file to the PRINTKIT_PS queue:

```
$ PRINT/QUEUE=PRINTKIT_PS
```

In a typical PrintKit configuration, the PRINTKIT_PS queue might be a logical queue that selects PostScript as the default data type for jobs and directs them to the PRINTKIT execution queue.

The default print queue is SYS$PRINT, the system print queue.

Selecting the data type for the file

Documents printed by PrintKit are encoded in a variety of data types – ANSI-PPL3, HP-GL/2, listing, PCL, and PostScript. The processing appropriate for a particular document depends on its data type. Usually, PrintKit can select the correct data type automatically by examining the contents of the file being printed.

In some cases, you may need to specify the data type to get the desired results. The ANSI-PPL3 and PCL data types, in particular, can be difficult to distinguish, because they use similar character sets and control sequences. Also, the ANSI-PPL3 and listing data types are both appropriate choices for printing a text file. For example, to send a PCL file directly to the printer with no translation, use the following:

```
$ PRINT/PARAM="DATA_TYPE=PCL"
```

If the DATA_TYPE option is not used, the system uses the default data type associated with the print queue to which the job is submitted. Your site may have logical queues that are configured for different default data types. The PrintKit execution queue also has a default data type (usually AUTOMATIC) that applies to jobs not assigned a data type explicitly or through a logical queue default.

The data types are as follows:

ANSI (default) Data is in ANSI-PPL3 (LN03) format. It is translated to the PostScript language for printing. PostScript must be available on the printer.

ASCII, LINE, TEXT Data is in ASCII format. It is translated to the PostScript language for printing, or, if PostScript is not available on the printer, to PCL instead.

[1]
/QUEUE qualifier
configuration
data type
execution queue
logical queue
ANSI data type
ASCII data type
AUTOMATIC data type
DATA_TYPE parameter
HPGL data type
LIST data type
PCL data type
POSTSCRIPT data type
data type
Encapsulated PostScript (EPSF)
execution queue
logical queue

AUTOMATIC The beginning of the data file is examined to determine the data type. If the data type cannot be determined or the data consists only of text, the TEXT data type will be assumed.

EPSF Data is in Encapsulated PostScript format. The file is assumed to contain PostScript describing a single page. This is useful for printing those EPSF files that do not normally print by themselves, because it forces a page to print.

HPGL Data is in Hewlett-Packard Graphics Language. It is sent directly to the printer. HPGL must be available on the printer.

LIST Data is text encoded using the ISO Latin 1 character set (which includes the ASCII character set). It is formatted in a simple listing format and translated to the PostScript language for printing. PostScript must be available on the printer.

PCL Data is in Hewlett-Packard Printer Control Language. It is sent directly to the printer. PCL must be available on the printer.

POSTSCRIPT, PS Data is in PostScript language. It is sent directly to the printer. PostScript must be available on the printer.

Printing multiple copies

You can specify the number of copies of your job to print. Depending on the qualifier or parameter you choose, the printed output may be uncollated or collated. Uncollated means that the printer prints all copies of the first sheet, all copies of second sheet, and so on. Collated means that the printer assembles the output so that all sheets of the first copy prints as a unit, followed by the second copy, and so on.

If you want uncollated copies, use the SHEET_COUNT parameter. For example to specify five copies, use the following:

```
$ PRINT/PARAM="SHEET_COUNT=5"
```

This is the quickest way to print multiple copies.

If you need collated output, use the /COPIES or /JOB_COUNT PRINT qualifier.

[1]
/COPIES qualifier
/FLAG qualifier
/JOB_COUNT qualifier
/TRAILER qualifier
SHEET_COUNT parameter
copies
flag page
multiple file jobs
trailer page

Multiple File Jobs and Multiple Copies

Every file in a multiple file job begins on a new sheet, regardless of number-up, orientation, or any other factor. The same principle applies for multiple copies – new copies begin on a new sheet.

If you would like a flag page or trailer page between files within a multiple file job, use the /FLAG or /TRAILER PRINT qualifier.

Printing two-sided documents

Some printers allow you a choice of printing on one side or both sides of a sheet and select whether pages are arranged for binding along the long edge or the short edge.

Although simplex normally implies one-sided printing, and duplex implies two-sided printing, one-sided duplex and two-sided simplex are exceptions.

For illustrations of the page arrangements produced by SIDES settings, and the interaction between NUMBER_UP, PAGE_ORIENTATION, and SIDES settings, see Figure 2-1, Page Layouts, page 2.11.

The sides settings are as follows:

1, ONE, ONE_SIDED_SIMPLEX Print on one side, with long-edge binding.

2, TWO, TWO_SIDED_DUPLEX Print on both sides, with long-edge binding. In landscape mode, the printing orientation allows for the long-edge binding (providing the "flip-up" ability).

TUMBLE, TWO_SIDED_TUMBLE Print on both sides, with short-edge binding.

ONE_SIDED_DUPLEX Print on one side. This results in a duplex-formatted copy that can be taken to a copying facility to make two-sided copies. With PrintKit, this format is identical to ONE_SIDED_SIMPLEX.

ONE_SIDED_TUMBLE Print on one side, rotating alternating pages 180 degrees. This option results in a tumble-formatted copy that can be taken to a copying facility to make two-sided copies.

TWO_SIDED_SIMPLEX Print on both sides in simplex format. With PrintKit, the result is identical to TWO_SIDED_DUPLEX. TWO_SIDED_SIMPLEX can be used to keep a two-sided master copy, in order to save space and paper.

[1]
NUMBER_UP parameter
PAGE_ORIENTATION parameter
POSTSCRIPT data type
SIDES parameter
binding orientation
copies
duplex printing
landscape orientation
portrait orientation
simplex printing
tumble duplex printing

For example to print a PostScript job in tumble mode, using portrait format:

```
$ PRINT/PARAM="SIDES=TUMBLE,DATA=POST"
```

Selecting paper stock and manual feed

Different print jobs may require different paper stock. For example, some print jobs require three-hole white paper while others may require transparencies.

You can select the stock to use for printing by using the DEFAULT_MEDIUM, PAGE_MEDIA, SHEET_SIZE, or INPUT_TRAY parameter. INPUT_TRAY may also be used to specify a manual feed operation. If your printer has only one input tray or if you're using a special media, you most likely will use manual feed to supply the stock to the printer.

It is usually better practice to select stock using media specifications, rather than input trays. This makes it easier to move your print job from one printer to another, and it helps avoid errors when another user changes the tray configuration on the printer. ANSI-PPL3 documents are often coded with input tray selection commands, and the MEDIUM_SUBSTITUTION parameter lets you redefine these tray selections as media selections.

Input Tray and Sheet Size Conflicts

If you use both the INPUT_TRAY and media selection (DEFAULT_MEDIUM, PAGE_MEDIA, or SHEET_SIZE) parameters, and the tray you specify does not contain a compatible medium, the effect depends on how your printer is configured. Some printers will cancel the job and report an error message; others will report that the required medium is not present and suspend printing until the problem is corrected.

DEFAULT_MEDIUM parameter

Specifying the DEFAULT_MEDIUM parameter selects the default paper stock for a job using the name of a predefined *medium specification*. For example, if GRAYBAR is the name of a medium specification for a stock with a graybar overlay, you would refer to it as follows:

```
$ PRINT /PARAM="DEFAULT_MEDIUM=GRAYBAR"
```

PAGE_MEDIA parameter

Specifying the PAGE_MEDIA parameter selects the paper stock for particular pages using the name of a predefined medium specification. By specifying multiple PAGE_MEDIA parameters, or a combination of DEFAULT_MEDIUM and PAGE_MEDIA parameters, you can print a document on several paper stocks.

[1]
ANSI data type
DEFAULT_MEDIUM parameter
INPUT_TRAY parameter
MEDIUM_SUBSTITUTION parameter
SHEET_SIZE parameter
configuration
error messages
messages
DEFAULT_MEDIUM parameter
medium specification
overlay
ANSI data type
DEFAULT_MEDIUM parameter
medium specification

This example selects a separate LETTER_BLUE medium for the first and last sheets of a duplexed document.

```
$ PRINT /PARAM=("SIDES=2",-
   "PAGE_MEDIA=(LETTER_BLUE,(1,2),(LAST-1,LAST))")
```

When specifying multiple page ranges for a PAGE_MEDIA parameter, parenthesize each range as shown to distinguish it from single pages.

In this example, pages of an ANSI-PPL3 document are printed on a SHEET_A medium, except for selected pages (those with "Continued" in line 1, columns 1-9), which are printed on SHEET_B medium.

```
$ PRINT /PARAM=("DEFAULT_MEDIUM=SHEET_A",-
   "PAGE_MEDIA=(SHEET_B,(FIELD(1,1,9)=""Continued""))"
```

This has the same effect, but with the roles of SHEET_A and SHEET_B media exchanged in the selection expressions:

```
$ PRINT /PARAM=("DEFAULT_MEDIUM=SHEET_B",-
   "PAGE_MEDIA=(SHEET_A,(FIELD(1,9,1)<>""Continued""))")
```

Selecting pages by content, rather than position, works only for ANSI-PPL3 documents.

For more information on the expressions used to select pages, see Table A-3, PRINT Page Selection Expressions, page A.24.

The PAGE_MEDIA parameter is provided only with the Production PrintKit software.

SHEET_SIZE parameter

Specifying the SHEET_SIZE allows the system to select the appropriate input tray based on the desired sheet size. This is the preferred method of selecting stock.

The available sheet sizes depend on the capabilities of your printer. Some common sizes, and their dimensions, are:

A4	8.3 X 11.7in (210 X 297mm)
LEDGER (or B)	11 X 17in (279 X 432mm)
LEGAL	8.5 X 14in (216 X 356mm)
LETTER (or A)	8.5 X 11in (216 X 279mm)

[1]
INPUT_TRAY parameter
SHEET_SIZE parameter

INPUT_TRAY parameter

Some printers have more than one input tray. Typically, if your printer supports multiple trays, each tray is used for a different type of media. For example, the top tray may contain 8 1/2 x 11 white paper, and the bottom tray may contain transparencies.

For example, to print from the bottom input tray:

```
$ PRINT/PARAM="INPUT_TRAY=BOTTOM"
```

The standard input tray names recognized by PrintKit are as follows:

TOP The top input tray (default).

MIDDLE The middle input tray (same as BOTTOM if there is no middle tray).

BOTTOM The bottom input tray (same as TOP if there is only one tray).

LCIT The large capacity input tray (same as BOTTOM if there is no large capacity tray).

CASSETTE The cassette input tray.

MULTIPURPOSE The multipurpose input tray.

ENVELOPE_FEEDER The optional input tray designed to accommodate envelopes.

MANUAL_FEED The manual paper feed.

NOMANUAL_FEED The current default input tray.

Input tray names vary among printers; your printer probably does not provide all these trays, and it may have different ones of its own.

MEDIUM_SUBSTITUTION parameter

If you are printing an ANSI-PPL3 document coded for a particular configuration of input trays, you can replace its tray selections with media selections using the MEDIUM_SUBSTITUTION parameter.

For instance, if your document is configured to draw plain paper from tray 1 and separator paper from tray 3, you can substitute LETTER_WHITE and LETTER_BLUE media selections by specifying:

```
$PRINT/PARAM=MEDIUM_SUBSTITUTION=-
   "((TRAY_1,LETTER_WHITE),(TRAY_3,LETTER_BLUE))"
```

The MEDIUM_SUBSTITUTION parameter is provided only with the Production PrintKit software.

[1]
INPUT_TRAY parameter
ANSI data type
MEDIUM_SUBSTITUTION parameter
configuration

Selecting the output tray

Some printers have more than one output tray. For those printers, you can specify the output tray for the print job.

The output trays are as follows:

TOP, UPPER (default unless print queue specifies otherwise) Send printed output to the upper paper output tray.

FACE_UP Produce output face up, uncollated. This option is useful if you are printing on heavier paper because it sends the paper through fewer bends while being transported through the printer.

LOWER Send printed output to the lower paper output tray. If there is no lower output tray, use the default tray.

For example, to specify the upper output tray:

```
$ PRINT/PARAM=OUTPUT_TRAY=UPPER
```

The default is TOP.

Controlling page layout

In understanding how to control the appearance of documents, it is useful to remember that the process PrintKit uses to assemble pages into final printed form is organized into two stages. The first stage is the formatting of the *page image* as described by the document source. The second stage is *imposition* (also called *page layup*), the arrangement of the formatted page images into final form.

The appearance of the page image is, for the most part, coded in the document itself, or in setup modules that PrintKit incorporates into the document as it is printed. For ANSI-PPL3 and PCL data types, the print form associated with the document may also affect the format.

The arrangement of formatted page images into final form is controlled by PrintKit parameters. There are PrintKit parameters to specify the orientation of page images, scale page images, arrange margins, and combine multiple page images into a single output page. These imposition controls are described fully in Appendix A, PRINT Command Reference.

[1]
ANSI data type
PCL data type
data type
page layout
setup modules

The topics that follow describe the group of PrintKit parameters that control page layup and show some of the adjustments you can make to your document's appearance using these parameters. Formatting controls are discussed in a separate section that outlines how to use setup modules and print form definitions.

Specifying page orientation

PrintKit allows you to specify the orientation of printed output on the page. For example to print landscape:

```
$ PRINT/PARAM="PAGE_ORIENTATION=LANDSCAPE"
```

A landscape page orientation changes the page format of an ANSI-PPL3, listing or PCL document so the primary axis of text is parallel to the long axis of the page, and it changes the default combination of font, character spacing and line spacing for text to one that is smaller and more compact. The PAGE_ORIENTATION parameter does not affect the format or orientation of pages in PostScript documents – they are formatted prior to processing by PrintKit and are assumed to already have appropriate orientation.

For PostScript, ANSI-PPL3 and listing documents, the page orientation affects how page images are arranged when layup specifications are used to combine multiple pages. For illustrations of the page arrangements produced by PAGE_ORIENTATION settings, and the interaction between NUMBER_UP, PAGE_ORIENTATION, and SIDES settings, see Figure 2-1, Page Layouts, page 2.11.

The orientations are as follows:

PORTRAIT The text runs parallel to the short edge of the page.

LANDSCAPE The text runs parallel to the long edge of the page.

Scaling pages

The NUMBER_UP option can be used in combination with the PAGE_SIZE and SHEET_SIZE options to scale page sizes to fit within sheets. For instance, a document formatted for LEDGER page size can be scaled to fit a LETTER sheet size with the following:

```
$ PRINT/PARAM=(NUMBER_UP=1,-
      PAGE_SIZE=LEDGER,SHEET_SIZE=LETTER)
```

Scaling also reduces the image to allow for sheet margins, if any.

[1]
ANSI data type
LIST data type
NUMBER_UP parameter
PAGE_ORIENTATION parameter
PCL data type
POSTSCRIPT data type
SIDES parameter
landscape orientation
page layout
NUMBER_UP parameter
PAGE_SIZE parameter
SHEET_SIZE parameter
page scaling

Combining multiple pages per sheet

The NUMBER_UP option sets the number of input document pages printed on a physical sheet. For example, if you set NUMBER_UP=4, four pages from your original document will be combined on each page of the printed output. Specifying a non-zero NUMBER_UP also enables borders and sheet margins.

The positioning specification affects the orientation and placement of these pages on the printed sheet. For instance, you could specify landscape orientation for the print job, and the printed pages would be in landscape format. A portrait specification would place pages in the same number of page spots, but rotated to the portrait format and placed appropriately within the page.

For illustrations of the page arrangements produced by NUMBER_UP settings, and the interaction between NUMBER_UP, PAGE_ORIENTATION, and SIDES settings, see Figure 2-1, Page Layouts, page 2.11.

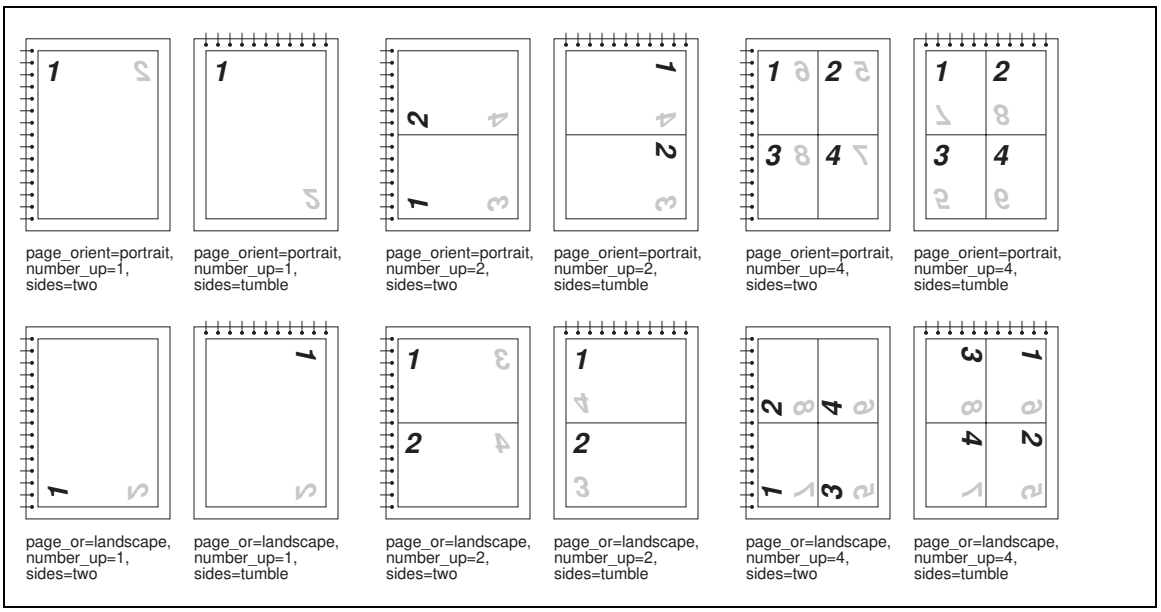


Figure 2-1, Page Layouts

[1]
BORDERS parameter
NUMBER_UP parameter
PAGE_ORIENTATION parameter
SIDES parameter
landscape orientation
portrait orientation

Complex imposition controls

More complex page layup can be specified using the LAYUP_SPECIFICATION parameter. Layup specifications provide a very flexible method of positioning page images on the output sheet; there are layup options to position multiple page images on a grid, and to adjust the margins surrounding the page images.

You can specify layup options directly with the LAYUP_SPECIFICATION parameter, or you can create an imposition specification containing the layup, and refer to it by name. Use a predefined imposition specification when you expect to use a particular layup repeatedly, or if it is complex enough that you risk making errors entering it on the command line. For instance, here a specification has been defined to shift margins to allow for binding:

```
$ PRINT /PARAM="SIDES=TWO,LAYUP=BINDING_SHIFT"
```

You create imposition specifications with the KITCP utility. Use KITCP to list the imposition specifications defined at your site:

```
$ KITCP SHOW IMPOSITION
```

For more information on working with imposition specifications, see Imposition specifications, page 4.13.

If the layup specification referenced above were specified directly, it would look like:

```
$ PRINT /PARAM=("SIDES=TWO",LAYUP=-
    "(MARGINS=0,0,36,-36;ALTERNATE=LEFT;NOBORDER)")
```

This specifies offsetting half-inch (36-point) left and right margins that are swapped on alternate sides of sheets in a duplexed document. Specifying offsetting margins, rather than a single margin, keeps the page image at full size; if only one margin were specified, PrintKit would scale the page image to fit the reduced space.

Controlling document finishing

Finishing is the process of assembling a completed document from the printed output. PrintKit lets you control this process by specifying a predefined finishing specification to be applied to your document after it is printed. For instance, here is a specification that staples the document in its upper-left corner:

```
$ PRINT /PARAM="FINISHING=STAPLE_TOP_LEFT"
```


[1]
KITCP Administration Utility
binding orientation
imposition specification
page layout
KITCP Administration Utility
FINISHING_INCLUDES_DOCUMENT parameter

Separator pages, by default, are not included in the finishing operations. You can override the default with the FINISHING_INCLUDES_DOCUMENT parameter. (This may not work if PrintKit is configured to use built-in printer separator pages.)

PrintKit is supplied with a standard set of finishing specifications, and more can be created with the KITCP utility. Use KITCP to list the finishing specifications defined at your site:

```
$ KITCP SHOW FINISHING
```

For more information on working with finishing specifications, see Finishing process specifications, page 4.14.

Selecting a document specification

When you have a document that requires a complicated combination of parameters, or you have a set of parameters that you use repeatedly, you can save them as a named *document specification*, using the KITCP command.

Once the parameters are saved as a document specification, you refer to them using the INITIAL_VALUE_DOCUMENT parameter. You can supersede parameters in the document specification by specifying them directly as PRINT parameters. For example, to print a job using a predefined PAMPHLET specification, use the following:

```
$ PRINT /PARAM="INITIAL_VALUE_DOCUMENT=PAMPHLET"
```

PrintKit is supplied with a standard set of document specifications, and more can be created with the KITCP utility. Use KITCP to list the document specifications defined at your site:

```
$ KITCP SHOW DOCUMENT
```

For more information on working with document specifications, see Document and subsidiary specifications, page 4.9.

Controlling page format with setup modules and forms

In addition to the controls PrintKit provides for document appearance through the use of parameter settings, most document data types allow an additional level of control from within the document. For example, the ANSI-PPL3 and PCL data types provide control

[1] setup modules
KITCP Administration Utility
INITIAL_VALUE_DOCUMENT parameter
document specification
ANSI data type
PCL data type
data type

sequences, and PostScript, as a general page description language, provides great flexibility in controls.

Setup modules let you manage these document controls separately from your documents, and supply them to the document during printing. Although you could always place the controls in the document itself, setup modules make it easier to specify and reuse common sets of controls. Setup modules also allow you to hide the details of potentially complex, device-specific material.

For ANSI-PPL3 and PCL data types, the print form associated with the job also affects the page format. The form dimensions, margins, and line wrap settings are all reflected in the initial settings for processing the document.

Interaction between ANSI-PPL3 setup modules and form settings

By default, form dimensions and margins for ANSI-PPL3 documents are implemented by changing the initial ANSI-PPL3 page format bounds for the document. Setup modules that reset or otherwise modify the page format will undo the form settings. The NOTAB parameter causes form settings to be implemented using space and linefeed characters instead. It can be useful in controlling the effects of setup modules on the form settings.

Where setup modules are stored

Setup modules are stored in *Device Control Libraries* associated with each print queue. When you print a document with setup modules, PrintKit collects the modules from the libraries and prepends them to the document before transferring it to the printer.

Each device control library associated with a queue contains setup modules for a particular data type, so PrintKit can be sure the setup modules it uses match the document data type. Also, PrintKit uses Adobe PostScript Printer Description (PPD) files, stored in the device control library as well, to provide printer-specific configuration data. Much of the information in the PPD file is available through command parameters, but entries that have no parameter counterpart can be accessed as setup modules. The places PrintKit checks for a setup module are as follows:

> modules in each library with a data type matching the document data type;

> modules in each library with a data type matching the translated data type, if the document data type is one that PrintKit translates before printing (ANSI-PPL3 to PostScript, for instance);

> entries in the printer-specific PPD file being used for the queue, if the document data type or translated data type is PostScript.

[1] setup modules

ANSI data type
POSTSCRIPT data type
configuration
data type
device control library
PostScript Printer Definition (PPD) files

How setup modules are named

You refer to setup modules by name. For ordinary setup modules, this is simply the name of the module as it appears in the library. For modules taken from PPD file entries, use the entry name with some minor adjustments. The names of PPD entries that can be used as setup modules have the form *MainKey OptionKey – you specify mainkey-optionkey as the setup module name. That is, drop the leading asterisk (*), replace the space with a hyphen (-), and ignore upper and lower case distinctions.

Specifying setup modules

There are two ways to specify setup modules for a print job, each with its own advantages:

/SETUP qualifier The PRINT /SETUP qualifier allows you to specify setup modules on the command line. It lets you specify modules directly, and in arbitrary combinations as required. However, the names are not checked by the PRINT command. If you make a mistake, it will not be detected until your job is assembled for printing, and your job will fail to print.

Forms Setup modules OpenVMS print form definitions can include setup modules, which you specify by selecting the form with the PRINT /FORM qualifier. In this case, the setup modules are bundled with the other form attributes, and each form selects only one combination of modules. However, form names are checked by the PRINT command, and, since the form is predefined, there is less chance of error in its specification of setup modules.

For instance, to print an ANSI-PPL3 document that requires condensed pitch settings for a compact listing format, you might have a setup module called COMPACT_SETTINGS. Once installed in a device control library for the queue, you can reference the module directly:

```
$ PRINT/SETUP=COMPACT_SETTINGS
```

Or, if you define a form COMPACT that designates the module, you can reference it indirectly:

```
$ PRINT/FORM=COMPACT
```

Similarly, if you have a special error handler for debugging PostScript documents stored as module EHANDLER, you can load it as you need it with:

```
$ PRINT/SETUP=EHANDLER
```

The person at your site responsible for configuring the PrintKit software should be able to provide the names of the setup modules and form definitions available for your use.

[1]
device control library
PostScript Printer Definition (PPD) files
setup modules
/FORM qualifier
/SETUP qualifier
ANSI data type
LIST data type
device control library
managing queues
PRINT command
setup modules

(Instructions for managing device control libraries are provided in Managing device control libraries, page 4.20.)

Chapter 3
Managing the Print Job

3.2 Introduction
3.2 Using fonts
3.2 Controlling flag page printing
3.3 Selecting files for printing
3.3 Scheduling print jobs
3.4 Monitoring the status of a print job

Introduction

The previous chapter described how to print your document. This chapter contains more advanced information on managing your print job. Depending on your specific network configuration, some of these operations may need to be performed by your System Administrator.

The major topics discussed in this chapter are:

- Using Fonts
- Controlling Flag Page Printing
- Selecting Files for Printing
- Scheduling Print Jobs
- Monitoring the Status of a Print Job
- Recovering from Printer and System Errors

Using fonts

Some print jobs may use fonts that are not permanently resident in the printer. In these cases, you'll need to use the FONTS_USED option to download the font to the printer for use by the print job. The fonts to be downloaded must be stored in the Device Control Libraries associated with the printer execution queue.

For example, to load Triumvirate and Times font sets for 8, 10, and 12 point sizes, enter the following:

```
$ PRINT/PARAM="FONTS_USED=(TRIUM8-10-12,TIMES8-10-12)"
```

Controlling flag page printing

PrintKit allows you to control whether to print a flag page, a burst page, or a trailer page. The flag and burst pages precede the file; the trailer page follows the file.

You can specify information to appear on flag pages by using the /NOTE command qualifier.

See the /BURST, /FLAG, /TRAILER, and /NOTE command qualifier descriptions for more information.

[1]
configuration
flag page
managing queues
FONTS_USED parameter
device control library
execution queue
font set
/BURST qualifier
/FLAG qualifier
/NOTE qualifier
/TRAILER qualifier
flag page
trailer page

Selecting files for printing

PrintKit provides numerous ways to select a file for printing. You can:

> Select only those files having the specified owner UIC. Use the /BY_OWNER command qualifier.

> Select files interactively. The PRINT command will prompt you with the name of each file. Use the /CONFIRM command qualifier.

> Exclude the specified files from printing. File specifications may specify directory, file name, type, and absolute version number. You may use wildcards; device names and relative version number, however, are not allowed. Use the /EXCLUDE command qualifier.

> Select only files dated before the specified time. You may specify backup, creation, expiration, or modification time for the selection. By default, the creation time is used. Use the /BEFORE command qualifier.

> Select only files dated since the specified time. You may specify backup, creation, expiration, or modification time for the selection. By default, the creation time is used. Use the /SINCE command qualifier.

See also the /BACKUP, /CREATION, /EXPIRED, and /MODIFIED command qualifiers.

Scheduling print jobs

Print jobs may be scheduled according to various criteria. For example, you may want to print jobs by a specific user, or hold printing of some jobs.

The following qualifiers allow you to schedule print jobs:

> /AFTER delays printing the files until after the specified time. If the time is always past, there is no delay in queueing the files for printing.

> /HOLD controls whether the files are immediately available for printing. This qualifier keeps files from being taken off the print until you use SET ENTRY/RELEASE command to release them. By default, files are available for printing once they are placed in the queue.

> /PRIORITY sets the priority of the print job. The priority is a number in the range 0 to 255, with 0 being the lowest priority. By default, the SYSGEN parameter DEFQUEPRI is used as the priority. You must have the OPER or ALTPRI privilege to raise the priority above the value of SYSGEN parameter MAXQUEPRI.

[1]
/BACKUP qualifier
/BEFORE qualifier
/BY_OWNER qualifier
/CONFIRM qualifier
/EXCLUDE qualifier
/EXPIRED qualifier
/MODIFIED qualifier
/SINCE qualifier
PRINT command
/AFTER qualifier
/CHARACTERISTICS qualifier
/FORM qualifier
/HOLD qualifier
/OPERATOR qualifier
/PRIORITY qualifier
/USER qualifier
messages
SYSGEN parameters

> /FORM specifies the form required for printing the files. Printing will be delayed until a compatible form (one that specifies the same paper stock) is set for the queue. Print forms are defined at your site. In conjunction with print characteristics, forms inform the system operator of steps that must be taken to configure a printer before a particular job can be printed.

> /CHARACTERISTICS specifies the characteristics required for printing the files. Printing will be delayed until all specified characteristics are set for the queue. Print characteristics are defined at your site. In conjunction with print forms, print characteristics inform the system operator of steps that must be taken to configure a printer before a particular job can be printed.

> /OPERATOR sends the specified message to operators when the print job begins.

> /USER prints the job on the behalf of the specified user. This qualifier requires CMKRNL privilege and R (READ) access to the user authorization file.

Monitoring the status of a print job

You can monitor and manage your print jobs using several DCL commands. You have privileges to display the queues, and change, display, or delete your own jobs. You can do this through the following commands:

SET ENTRY Change the current status or attributes of a job.

SHOW ENTRY Display the current status or attributes of a job.

SHOW QUEUE Display information about queues and the jobs that are currently in queues.

DELETE/ENTRY Remove a print job from the queue.

These commands are documented in the *OpenVMS DCL Dictionary* and the *OpenVMS User's Manual*.

Use the /NOTIFY qualifier to receive notification when the print job is completed or aborted. By default, you will not be notified.

By using the MESSAGES parameter, you can save error messages and data returned from a PostScript print job. You can specify to write them to a log file or print them on a log page following the print job. This parameter affects only the PostScript data type and is useful only for printers with bidirectional interfaces.

[1]
/NOTIFY qualifier
MESSAGES parameter
POSTSCRIPT data type
data type
error messages
messages
notification messages
status of print job

Chapter 4
Configuring PrintKit

4.2 Introduction
4.3 Determining the PrintKit version
4.3 Examining PrintKit queue configuration
4.4 PrintKit communications configuration
- Supported TCP/IP interfaces
- Assigning internet addresses
- Printer lpr/lpd queue names
- LAT node and port names

4.6 Execution queues
- Determining printer model and configuration
- Adding, modifying, and removing queues

4.8 Generic and logical queues
4.9 Document and subsidiary specifications
4.10 Document specifications
4.12 Medium specifications
4.13 Imposition specifications
4.14 Finishing process specifications
4.15 Sheet package specifications
4.17 Saving the PrintKit configuration
4.18 Managing print forms
- Managing form definitions
- Assigning forms to a queue

4.20 Managing device control libraries
- Setup modules
- Examining library contents
- Adding new library modules
- Adding new libraries

4.23 Removing the PrintKit software

Introduction

This chapter describes how to configure the PrintKit software for correct operations with your printers, and how to update that configuration as your printing needs change. It also discusses how the configuration information is organized.

The PrintKit software operates as a symbiont, which means it is controlled by the OpenVMS Job Control Program. Job Control provides the PrintKit software with configuration information taken from a common system queue management database. However, PrintKit provides many capabilities that go beyond those managed by Job Control. For these, the PrintKit symbiont refers to a separate PrintKit configuration database.

Briefly, the PrintKit configuration database contains entries for the following entities:

Print queues An entry for each print queue managed by PrintKit supplements the common system queue attributes;

Printer models An entry for each printer model supported by PrintKit provides the information needed to tailor PrintKit's handling of the printer and to provide access to printer features;

Document, sheet package and subsidiary specifications Document specifications, and the subsidiary specifications that they reference (medium, imposition, and finishing process), provide the information to direct PrintKit in printing documents. These predefined specifications provide an alternative to parameters specified at the time a document is printed, and they allow more complex specifications. Sheet package specifications (which also reference medium specifications) direct media selection for separator pages. They are assigned as attributes of PrintKit execution queues.

The KITCP administration utility lets you configure and maintain the PrintKit software. You use it to manage PrintKit queues, printer models, and document and sheet package specifications. For a complete discussion of the command interface KITCP uses, see Appendix B, KITCP Command Reference.

Before you use KITCP, you need to define the KITCP command, as follows:

```
$ KITCP :== $KITCP
```


[1]
KITCP Administration Utility
INITIALIZE/QUEUE command
configuration
document specification
execution queue
Job Control Program
medium specification
sheet package specification
symbiont

KITCP and INITIALIZE/QUEUE Conflicts

The KITCP command incorporates the capabilities of the INITIALIZE/QUEUE command, and for PrintKit queues, you should use KITCP instead of INITIALIZE/QUEUE. Similarly, you should not use the SET QUEUE command to modify static queue attributes that are also managed by KITCP. Using INITIALIZE/QUEUE or SET QUEUE this way can leave the configuration information in the system queue management database inconsistent with that in the PrintKit configuration database. If KITCP detects such inconsistencies, its normal action is to reset the system configuration to match PrintKit's.

Determining the PrintKit version

To determine the installed version of the PrintKit software, use the KITCP SHOW VERSION command. For example:

```
$ KITCP SHOW VERSION
PrintKit Version 2.3
```

Examining PrintKit queue configuration

For a brief display of all the queues managed by PrintKit, use the KITCP SHOW QUEUE command. For example:

```
$ KITCP SHOW QUEUE
Printer queue PRINTKIT, idle, on LES::TCPIP,
   mounted form DEFAULT
<PrintKit Execution Queue>

Generic queue PRINTKIT_PS, idle, mounted form DEFAULT
<PrintKit PostScript Queue>
...
```

The output displays the same queue information as the DCL SHOW QUEUE command: the type of queue (printer or generic), the name of the queue, the queue status (see Queue status, page 6.5), the node and device on which it is located, and its current mounted form.

To examine the configuration for a particular queue, specify the name of the queue. To display more detailed information, use the /FULL qualifier. For example:

```
$ KITCP SHOW QUEUE/FULL PRINTKIT
```


[1]
SHOW VERSION KITCP command
version, software
SHOW QUEUE KITCP command
SHOW QUEUE command
configuration
queue status
troubleshooting

```
Printer queue PRINTKIT, idle, on LES::TCPIP,
   mounted form DEFAULT
<PrintKit Execution Queue>
/BASE_PRIORITY=4 /DEFAULT=(FEED,FORM=DEFAULT,SHEET_SIZE=LETTER)
/OWNER=[1,4] /PROCESSOR=PRINTKIT
/PROTECTION=(S:M,O:D,G:R,W:S) /REQUEST_TO=(PRINTER)
/RETAIN=ERROR /SCHEDULE=NOSIZE

/LIBRARY=(PRINTKIT023,PRINTKIT_ANSI/DATA_TYPE=ANSI,
   PRINTKIT_PCL/DATA_TYPE=PCL,PRINTKIT_PS/DATA_TYPE=POSTSCRIPT)
/PRINTER=(MODEL="ACME LaserPrinter Model II")
/COMMUNICATIONS=(ADDRESS=192.0.1.200)

Process name PrintKit_80, Process ID 21c1f22a
```

The use of the KITCP SHOW QUEUE command for troubleshooting problems with a PrintKit queue is discussed in more detail in Chapter 5, Troubleshooting PrintKit.

PrintKit communications configuration

PrintKit provides support for TCP/IP socket, TCP/IP lpr/lpd, TCP/IP TRANSPORT1/STATUS1, TCP/IP DQP, LAT, and serial communications protocols. Each has its own set of options.

You may find it useful to refer to the following notes when configuring the communications settings for a PrintKit queue.

Supported TCP/IP interfaces

The PrintKit software is compatible with all the commonly-used OpenVMS TCP/IP communications services. It also has its own built-in TCP/IP services, PEP, that it uses when no separate services are present.

When you configure a queue using a TCP/IP interface, you specify a network device name in the /ON qualifier for the queue. Normally, you specify TCPIP as the network device name, which causes PrintKit to select the appropriate device, based on your system configuration. (It first looks for a TCP/IP services device, and if it finds none, looks for an Ethernet device for use by PEP.)

[1]
configuration
LAT network protocol
TCP/IP network protocol
AUTOMATIC data type
configuration
TCP/IP network protocol

You can also specify a specific network device name with the /ON qualifier, but you should do this *only* if PrintKit's automatic TCPIP device selection does not work with your configuration. It may be necessary if your system has multiple network interface devices, or if the network interface is not one that PrintKit recognizes. PrintKit looks for a network device in the following order:

BG0:	Digital TCP/IP Services for OpenVMS
INET0:	Wollongong PathWays, Cisco MultiNet, Process Software TCPware, others
FUA0:	Network Research FUSION
ERA0:, ESA0:, ETA0:, EXA0:, EZA0:, XEA0:, XQA0:	Ethernet (used by PEP)
E%A0:, X%A0:	More Ethernet (used by PEP)

Assigning internet addresses

To connect to a printer using TCP/IP communications protocols, the PrintKit software needs the printer's Internet address (also called an IP address, for Internet Protocol). An internet address may be specified as a symbolic node name or a numeric address. Use of node names requires standard OpenVMS TCP/IP Services or a third-party service with a compatible implementation of name translation. A numeric address is expressed as four decimal values, separated by periods (d.d.d.d). If you are using the built-in PEP TCP/IP services, you will also need to supply the Internet address for OpenVMS system on which the PrintKit software is installed.

If you are already using TCP/IP communications on your network, your printer will need an Internet address that is compatible with your TCP/IP configuration. Ask your Network Administrator to supply the address.

Network compatibility is not a problem if you are using PEP on the host where you are installing PrintKit, and you are not otherwise using TCP/IP on your network. If you are uncertain of what Internet address to specify for the printer, 192.0.0.1 is a good choice. In Internet terminology, this is a Class C network address – the first three numbers (192.0.0) are the network number, and the fourth number (1) identifies the node on the network. You can then use 192.0.0.2 as the address of the system where you are installing the PrintKit software. If you have additional printers or host systems, assign them other node numbers, keeping the same network number (192.0.0.3, 192.0.0.4, and so forth).

[1]
configuration
TCP/IP network protocol

Printer lpr/lpd queue names

Printer network interfaces are preconfigured with lpr/lpd queue names. The name should be displayed on the printer configuration page; if no queue name is indicated, check for a service name or similar nomenclature, or refer to the user manual for the printer.

If you have trouble locating the queue name in the user manual, look for a description of the UNIX printcap entry required by the interface. The printcap rp (remote printer) entry is the lpr/lpd queue name.

Be sure you specify the queue name exactly as it appears on the configuration page, matching upper and lower case in particular.

LAT node and port names

LAT printer servers are preconfigured with LAT node and port names. A typical LAT node name consists of a prefix of three or four letters identifying the manufacturer or model, followed by the last six digits (in hexadecimal) of the server's Ethernet address.

If you have configured a service name for your printer's LAT interface, you may use it in place of the port name.

Execution queues

Execution queues process jobs for printing. Each queue has an associated *symbiont*, the software component that performs the processing of jobs and manages the connection with the printer. For PrintKit, this is the PrintKit symbiont. Each printer accessed from a system must have its own unique execution queue. A printer's execution queue is normally shared across nodes in a VMScluster.

Execution queues are complemented by generic and logical queues, which peform no processing, but are used for routing and scheduling of jobs. These queue types are discussed in the following section Generic and logical queues, page 4.8.

To create a new PrintKit execution queue, first collect configuration information for the printer to be controlled by the queue and the communications interface used to access it. Then verify that the printer model is one supported by PrintKit, and determine the model name used by PrintKit. With this information in hand, use the KITCP ADD QUEUE command to create the queue.

Instructions for a typical configuration are provided in Quick Start Installation. The following instructions cover the general case, and include tips on where to look for further information.

[1]
configuration
TCP/IP network protocol
LAT network protocol
ADD QUEUE KITCP command
configuration
execution queue
logical queue
symbiont

Determining printer model and configuration

Use the KITCP SHOW MODEL command to determine the model name PrintKit uses for your printer. If there appears to be no model definition for your printer, check the Northlake web site, www.nls.com, for up-to-date information on supported printers and printer model supplements.

PrintKit's selectable communications attributes are summarized in the definition for the KITCP ADD QUEUE command in Table B-1, KITCP Commands, page B.4; these attributes are specified by the /COMMUNICATIONS qualifer.

For further discussion of the communications options, refer to the preceding PrintKit communications configuration, page 4.4.

PrintKit's selectable printer attributes are summarized in the definition for the KITCP ADD QUEUE command in Table B-1, KITCP Commands, page B.4; these attributes are specified by the /PRINTER qualifer.

Model definitions provide default settings for communications and printer attributes. These match the standard configuration for the printer, so you normally don't need to specify settings explicitly. (Use the KITCP SHOW MODEL/FULL command to view these settings.) Typically, you need to specify only the printer model and network address.

Adding, modifying, and removing queues

Once you have collected all the configuration information for your printer, use the KITCP ADD QUEUE command to create a PrintKit execution queue for it. For example:

```
$ KITCP ADD QUEUE PRINTKIT /ON=TCPIP -
    /PRINTER=(MODEL="ACME LaserPrinter Model II") -
    /COMMUNICATIONS=(ADDRESS=192.0.1.2)
```

If you have an existing queue that you want to configure to use the PrintKit software, or if you are updating queues from an earlier version of PrintKit that did not use a separate configuration database, use the /REPLACE qualifier with the KITCP ADD QUEUE command. It causes KITCP to use the existing queue specification to provide defaults for the PrintKit queue definition.

If you are adding PrintKit queues for several printers with similar configurations, the KITCP COPY QUEUE command can simplify the task. It functions like the ADD QUEUE command, and accepts the same qualifiers. You specify only those attributes that are different from the "copied" queue. The new queue takes on the attributes of the copied queue, except those you specify explicitly. For example:

```
$ KITCP COPY QUEUE PRINTKIT PRINTKIT2 -
```


[1]
ADD QUEUE KITCP command
SHOW MODEL KITCP command
configuration
KITCP Administration Utility
ADD QUEUE KITCP command
COPY QUEUE KITCP command
MODIFY QUEUE KITCP command
REMOVE QUEUE KITCP command
configuration
execution queue

```
/COMMUNICATIONS=(ADDRESS=192.0.1.2)
```

If you make a mistake in creating the queue, use the KITCP MODIFY QUEUE command to make changes. You may also find it convenient to create the queue in stages, checking your work as you go along – first create the queue, then modify it to complete the configuration. You can also use the MODIFY QUEUE command later, to change the queue configuration.

A queue must be stopped to be modified. If you try to modify a queue that is not stopped, KITCP will wait for any active job to complete printing, stop the queue, make the modifications, then restart the queue. You can also manually stop the queue while making changes. For example:

```
$ STOP/QUEUE/NEXT PRINTKIT
$ KITCP MODIFY QUEUE PRINTKIT -
    /SCHEDULE=NOSIZE
$ START/QUEUE PRINTKIT
```

Be sure to specify the /NEXT qualifier when you stop the queue; without it, the queue only pauses.

Use the KITCP REMOVE QUEUE command to remove a PrintKit queue. For example:

```
$ KITCP REMOVE QUEUE PRINTKIT
```

Generic and logical queues

Generic queues are used with the PrintKit software for two purposes. Their traditional use is to allow jobs to be directed to more than one queue for printing, without users or applications software having to be aware of which queue is the correct one to use at any particular time. The generic queue remains unchanged, but its associated execution queues may change according to load, printer availability, or changes in cluster configuration.

With PrintKit, generic queues are also used to supply default parameter settings for print jobs. In this case, there may be multiple generic queues associated with a single fixed execution queue, each supplying a different set of job defaults. This can provide a reliable method of supplying commonly-used combinations of parameter settings. It also provides a way of specifying parameters for jobs created by applications that do not allow them to be specified directly.

[1]
ADD QUEUE KITCP command
COPY QUEUE KITCP command
MODIFY QUEUE KITCP command
POSTSCRIPT data type
REMOVE QUEUE KITCP command
SHOW QUEUE KITCP command
autostart queue
configuration
data type
document specification
execution queue
generic queue
logical queue

Logical queues are similar to generic queues, except that they direct jobs to a single execution queue for printing. With PrintKit, they can be used in the same manner as generic queues to supply default parameter settings.

Use the KITCP ADD QUEUE command to create PrintKit generic or logical queues, much the same as for execution queues, except that you specify the /GENERIC or /LOGICAL qualifier. Typically, the only other qualifier you need specify is /DEFAULT, to provide a default DOCUMENT for the queue. However, the /CLOSE, /DESCRIPTION, /OPEN, /OWNER, /PROTECTION, /RETAIN, and /SCHEDULE qualifiers are also allowed.

For example, assuming a DEFAULT_PS has been defined as a document specification that selects the PostScript data type, the following creates a generic queue for printing PostScript jobs:

```
$ KITCP ADD QUEUE PRINTKIT_PS -
    /DEFAULT=(DOCUMENT=DEFAULT_PS) -
    /GENERIC=(PRINTKIT)
```

Generic queues are also used in a VMScluster environment to maintain availability of printing services should a node leave the cluster. (Autostart queues can perform a similar function.) Here is a generic queue that routes jobs to execution queues on two nodes:

```
$ KITCP ADD QUEUE PRINTKIT -
    /GENERIC=(BERT$PRINTKIT,ERNIE$PRINTKIT)
```

One execution queue would remain stopped, unless the other became unavailable.

The KITCP SHOW QUEUE, MODIFY QUEUE, COPY QUEUE, and REMOVE QUEUE commands work with generic and logical queues, in the same manner as with execution queues, as was described in the preceding section.

Document and subsidiary specifications

A *document* specification is the collection of information used by PrintKit to prepare a file for printing. A document specification may in turn refer to subsidiary specifications (*medium*, *imposition*, and *finishing process*). Medium specifications describe the paper or other media on which the document is printed. Imposition specifications describe adjustments made to the placement and scale of the document page images as they are assembled onto printed sheets. Finishing process specifications describe the steps involved in assembling the printed sheets into the final document.

[1] medium specification

KITCP Administration Utility
LAYUP_DEFINITION parameter
configuration
document specification
finishing process specification
imposition specification

The attributes that make up a document specification may be supplied individually, as PRINT parameters. A collection of attributes may also be stored in the PrintKit configuration database as a named document specification. The specification can then be used in printing a document by referring to it by name, rather than specifying attributes individually.

Subsidiary specifications (medium, imposition, and finishing process) are likewise stored in the PrintKit configuration database, and are referred to by name. The attributes comprising an imposition specification may also be specified individually as a LAYUP_DEFINITION parameter, rather than being defined in the database. Other subsidiary specifications are only referenced as named entities.

The following sections describe how to manage specifications in the PrintKit configuration database. They show examples of KITCP commands to display, add, modify, and remove each type of specification.

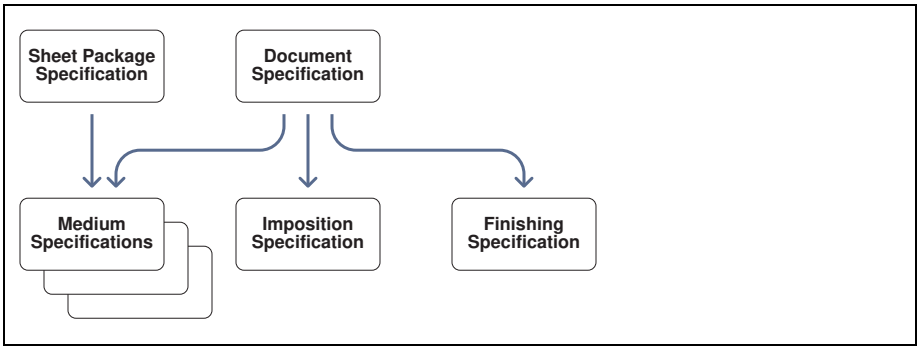


Figure 4-1, Document Specification Organization

Document specifications

A document specification supplies the same attributes as can be specified with PRINT parameters. Because of this, the parameter descriptions in Appendix A, PRINT Command Reference, and the parameter examples in Chapter 2, Using PrintKit, also apply to document specifications. You simply substitute the KITCP qualifier syntax for the parameter syntax – the meaning is the same.

[1]
KITCP Administration Utility
document specification

Adding, modifying, and removing document specifications

Use the KITCP ADD DOCUMENT command to add a new document specification to the PrintKit database. For example:

```
$ KITCP ADD DOCUMENT LISTING -
    /DESCR="Program listing"-
    /DEFAULT_MEDIUM=LETTER_WHITE -
    /LAYUP=LISTING_MARGINS /TAB
```

If you already have a document specification similar to one you want to create, you can copy it to create the new specification. The KITCP COPY DOCUMENT functions like the ADD DOCUMENT command, and accepts the same qualifiers. You specify only those attributes that are different from the "copied" specification. The new specification takes on the attributes of the copied specification, except those you specify explicitly. For example:

```
$ KITCP COPY DOCUMENT LISTING DUPLEX_LISTING -
    /DESCR="Duplex Program Listing"-
    /LAYUP=DUPLEX_LISTING_MARGINS /SIDES=DUPLEX
```

Use the KITCP MODIFY DOCUMENT command to modify an existing PrintKit document specification. For example:

```
$ KITCP MODIFY DOCUMENT LISTING -
    /DEFAULT_MEDIUM=LETTER_GRAYBAR
```

Use the KITCP REMOVE DOCUMENT command to remove a document specification from the PrintKit configuration database. For example:

```
$ KITCP REMOVE DOCUMENT LISTING
```

Showing document specifications

Use KITCP SHOW DOCUMENT to display document specifications. By default, it displays all documents. You may specify a document name, or a partial name using wildcards, to restrict the display to selected documents. The /FULL qualifier displays all the document attributes; without it, you get a more compact summary display. For example:

```
$ KITCP SHOW DOCUMENT/FULL DEFAULT*
Document object DEFAULT, version 1.0
  <Default Document Specification>
  /DATA_TYPE=AUTOMATIC=ANSI /TAB

Document object DEFAULT_PS
```


[1]
ADD DOCUMENT KITCP command
COPY DOCUMENT KITCP command
MODIFY DOCUMENT KITCP command
REMOVE DOCUMENT KITCP command
configuration
document specification
SHOW DOCUMENT KITCP command
document specification

```
  <Default PostScript Document Specification>
  /DATA_TYPE=POSTSCRIPT
  /DEFAULT_MEDIUM=LETTER_WHITE

Document object DEFAULT_PCL
  <Default PCL Document Specification>
  /DATA_TYPE=PCL /DEFAULT_MEDIUM=LETTER_WHITE

$ KITCP SHOW DOCUMENT/FULL *PAMPHLET
Document object A4_PAMPHLET
  <Saddle-stitched A4 pamphlet>
  /DEFAULT_MEDIUM=A3_WHITE
  /FINISHING=SADDLE /LAYUP=SIGNATURE
  /PG_MEDIA_SELECT=((A3_BLUE,"(1,2),(LAST-1,LAST)"))
  /PG_SIZE=A4 /SIDES=DUPLEX

Document object PAMPHLET
  <Saddle-stitched letter pamphlet>
  /DEFAULT_MEDIUM=LEDGER_WHITE
  /FINISHING=SADDLE /LAYUP=SIGNATURE
  /PG_MEDIA_SELECT=((LEDGER_BLUE,"(1,2),(LAST-1,LAST)"))
  /PG_SIZE=LETTER /SIDES=DUPLEX
```

Medium specifications

A medium specification designates a particular print medium, identified by a set of attributes: its type, size, color, and weight. The medium is typically some type of paper, but printers may support other media, too, such as transparency stock. PrintKit uses medium specifications to control the media the printer uses for printing a document.

Printers vary in which attributes they use for media selection. The media attributes provided by PrintKit match the PostScript Level 2 medium model, but particular printers may recognize only the size attribute, or a combination of size and type.

A medium specification may also designate forms overlays to be displayed on the front and back sides of the medium. For more information on using forms overlays, see Appendix E, Forms Overlays.

[1]
POSTSCRIPT data type
medium specification

Adding, modifying, and removing medium specifications

Use the KITCP ADD MEDIUM command to add a new medium specification. For example:

```
$ KITCP ADD MEDIUM A3_BLUE -
    /DESCRIPTION="A3 Blue Stock"/SIZE=A3 -
    /COLOR=BLUE /WEIGHT=80-GSM /TYPE=PLAIN
```

If you already have a medium specification similar to one you want to create, you can copy it to create the new specification. The KITCP COPY MEDIUM functions like the ADD MEDIUM command, and accepts the same qualifiers. But, in addition, you specify the name of an existing medium specification that is "copied" to create the new specification. That is, unless you explicitly specify an attribute for the new specification, its value is taken from the specification being copied. For example:

```
$ KITCP COPY MEDIUM A3_BLUE A3_RED -
    /COLOR=RED
```

Use the KITCP MODIFY MEDIUM command to modify an existing PrintKit medium specification. For example:

```
$ KITCP MODIFY MEDIUM A3_BLUE -
    /WEIGHT=80#-COVER
```

Use the KITCP REMOVE MEDIUM command to remove a PrintKit medium specification. For example:

```
$ KITCP REMOVE MEDIUM A3_BLUE
```

Showing medium specifications

Use the KITCP SHOW MEDIUM/FULL command to show medium attributes. For example:

```
$ KITCP SHOW MEDIUM/FULL A3_BLUE
Medium object A3_BLUE
  <A3 Blue Stock>
  /COLOR=BLUE /SIZE=A3 /TYPE=PLAIN /WEIGHT=80-GSM
```

Imposition specifications

Each document specification may refer to an imposition or number-up specification. Imposition attributes determine how many page images will appear on each printed page, how the images will be oriented, separated, and arranged.

[1]
ADD MEDIUM KITCP command
COPY MEDIUM KITCP command
MODIFY MEDIUM KITCP command
REMOVE MEDIUM KITCP command
medium specification
medium specification
document specification
imposition specification

Adding, modifying, and removing imposition specifications

Use the KITCP ADD IMPOSITION command to add a new imposition specification. For example:

```
$ KITCP ADD IMPOSITION DIAGONAL -
    /MARGINS=(72,72,108,72) /ALTERNATE -
    /GRID=(2,2) /PG_ORDER=LEFTDOWN -
    /FIRSTPAGE=2 /PG_PER_SHEET=2
```

Use the KITCP COPY IMPOSITION command to copy an existing imposition specification to a new specification. For example:

```
$ KITCP COPY IMPOSITION DIAGONAL -
    REVERSE_DIAGONAL /PG_ORDER=RIGHTDOWN
```

Use the KITCP MODIFY IMPOSITION command to modify an existing imposition specification. For example:

```
$ KITCP MODIFY IMPOSITION DIAGONAL -
    /MARGINS=(36,36,72,36) -
```

Use the KITCP REMOVE IMPOSITION command to remove an imposition. For example:

```
$ KITCP REMOVE IMPOSITION DIAGONAL
```

Showing imposition specifications

Use the KITCP SHOW IMPOSITION/FULL command to show imposition attributes. For example:

```
$ KITCP SHOW IMPOSITION/FULL DIAGONAL
Imposition object DIAGONAL
  <PrintKit Imposition Object>
  /ALTERNATE /FIRST_PAGE=2 /GRID=(2,2)
  /MARGINS=(72,72,108,72)
  /PG_ORDER=LEFTDOWN /PG_PER_SHEET=2
```

Finishing process specifications

Each document specification may refer to a finishing specification. Finishing attributes determine how a document will be folded, stapled and bound. Every finishing specification defined in the PrintKit database contains a list of finishing processes.

[1]
ADD IMPOSITION KITCP command
COPY IMPOSITION KITCP command
MODIFY IMPOSITION KITCP command
REMOVE IMPOSITION KITCP command
imposition specification
imposition specification
document specification
finishing process specification

Adding, modifying, and removing finishing process specifications

Use the KITCP ADD FINISHING command to add a new finishing specification. For example:

```
$ KITCP ADD FINISHING PAMPHLET -
    /DESCRIPTION="Pamphlet finishing (folded, stapled)"-
    /SPECIFICATION=(FOLDING=SADDLE,-
        STITCHING=SADDLE)
```

Use the KITCP COPY FINISHING command to copy an existing finishing specification to a new specification file. For example:

```
$ KITCP COPY FINISHING PAMPHLET PAMPHLET2
```

Use the KITCP MODIFY FINISHING command to modify an existing finishing specification. For example:

```
$ KITCP MODIFY FINISHING PAMPHLET -
    /SPECIFICATION=(FOLDING=SADDLE)
```

Use the KITCP REMOVE FINISHING command to remove a finishing specification. For example:

```
$ KITCP REMOVE FINISHING PAMPHLET
```

Showing finishing process specifications

Use the KITCP SHOW FINISHING/FULL command to show finishing process attributes. For example:

```
$ KITCP SHOW FINISHING PAMPHLET
Finishing object PAMPHLET
  <Pamphlet finishing (folded, stapled)>
  /SPECIFICATION=(FOLDING=SADDLE,STITCHING=SADDLE)
```

Sheet package specifications

A sheet package specification selects the media used for printing separator pages (job flag, burst and trailer pages, and file flag, burst, trailer and log pages) by assigning medium specifications to separator page types.

[1]
ADD FINISHING KITCP command
COPY FINISHING KITCP command
MODIFY FINISHING KITCP command
REMOVE FINISHING KITCP command
finishing process specification
finishing process specification
medium specification
sheet package specification

Adding, modifying, and removing sheet package specifications

Use the KITCP ADD SHEET_PACKAGE command to add a new sheet package specification to the PrintKit database. For example:

```
$ KITCP ADD SHEET_PACKAGE FLAG_COLOURED -
    /DESCR="Coloured Flag Pages Sheet Package"-
    /SHEETS=(FLAG=(MEDIUM=LETTER_COLOURED))
```

If you already have a sheet package specification similar to one you want to create, you can copy it to create the new specification. The KITCP COPY SHEET_PACKAGE functions like the ADD SHEET_PACKAGE command, and accepts the same qualifiers. You specify only those attributes that are different from the "copied" specification. The new specification takes on the attributes of the copied specification, except those you specify explicitly. For example:

```
$ KITCP COPY SHEET_PACKAGE FLAG_COLOURED ALL_COLOURED -
    /SHEETS=(DEFAULT=(MEDIUM=LETTER_COLOURED))
```

Use the KITCP MODIFY SHEET_PACKAGE command to modify an existing PrintKit sheet package specification. For example:

```
$ KITCP MODIFY SHEET_PACKAGE FLAG_COLOURED -
    /SHEETS=(FLAG=(MEDIUM=LETTER_COLOURED), -
        DEFAULT=(MEDIUM=LETTER_WHITE))
```

Use the KITCP REMOVE SHEET_PACKAGE command to remove a sheet package specification from the PrintKit configuration database. For example:

```
$ KITCP REMOVE SHEET_PACKAGE LISTING
```

Showing sheet package specifications

Use KITCP SHOW SHEET_PACKAGE to display sheet package specifications. By default, it displays all sheet packages. You may specify a sheet package name, or a partial name using wildcards, to restrict the display to selected sheet packages. The /FULL qualifier displays all the sheet package attributes; without it, you get a more compact summary display. For example:

```
$ KITCP SHOW SHEET_PACKAGE DEFAULT*
Sheet package object DEFAULT, version 1.0
  <Default sheet package>
$ KITCP SHOW SHEET_PACKAGE/FULL *COLOURED
Sheet package object ALL_COLOURED
```


[1]
ADD SHEET_PACKAGE KITCP command
COPY SHEET_PACKAGE KITCP command
MODIFY SHEET_PACKAGE KITCP command
REMOVE SHEET_PACKAGE KITCP command
SHOW SHEET_PACKAGE KITCP command
configuration
sheet package specification
sheet package specification

```
  <Coloured Pages Sheet Package Definition>
  /SHEETS=(DEFAULT=(MEDIUM=LETTER_COLOURED))

Sheet package object FLAG_COLOURED
  <Coloured Flag Pages Sheet Package Definition>
  /SHEETS=(FLAG=(MEDIUM=LETTER_COLOURED))
```

Saving the PrintKit configuration

In addition to displaying the information in the PrintKit configuration database, the KITCP SHOW commands can generate a set of KITCP commands that will recreate the information in the database. This allows you to make a human-readable record of the configuration information that can also be used to recreate the database. It can also be handy for moving configuration information from one system to another.

The /DCL qualifier causes the SHOW commands to write their output as KITCP commands, the /OUTPUT qualifier redirects the output to a file, and the /APPEND qualifier causes the /OUTPUT qualifier to append to an existing output file. The following commands will save all the PrintKit configuration database entries in a file:

```
$ KITCP SHOW MODEL/DCL -
    /OUTPUT=PRINTKIT_RESTORE.COM
$ KITCP SHOW MEDIUM/DCL -
    /OUTPUT=PRINTKIT_RESTORE.COM /APPEND
$ KITCP SHOW IMPOSITION/DCL -
    /OUTPUT=PRINTKIT_RESTORE.COM /APPEND
$ KITCP SHOW FINISHING/DCL -
    /OUTPUT=PRINTKIT_RESTORE.COM /APPEND
$ KITCP SHOW DOCUMENT/DCL -
    /OUTPUT=PRINTKIT_RESTORE.COM /APPEND
$ KITCP SHOW SHEET_PACKAGE/DCL -
    /OUTPUT=PRINTKIT_RESTORE.COM /APPEND
$ KITCP SHOW QUEUE/DCL -
    /OUTPUT=PRINTKIT_RESTORE.COM /APPEND
```

In order to use the output from the KITCP SHOW MODEL command, you will need copies of the PPD files referenced by the model definitions.

[1]
KITCP Administration Utility
SHOW MODEL KITCP command
configuration
PostScript Printer Definition (PPD) files

Managing print forms

Each OpenVMS print job has an associated print form, specified explicitly or supplied from the default setting for the queue to which the job is submitted for printing. Each queue, in turn, has a current mounted form, in addition to its default form.

A form definition specifies a paper stock, formatting attributes, and setup modules. The paper stock assists in determining when jobs are scheduled for printing. Both the formatting attributes and setup modules affect the printed appearance of the job.

Jobs are allowed to print on a queue only if their form and the queue's mounted form specify the same paper stock. Jobs with a form that does not match the mounted form's stock are not processed. They remain in a pending state in the queue until the queue's mounted form is changed to a compatible form. (If you display such a job with the SHOW QUEUE or SHOW ENTRY command, its status will be displayed as "pending," with "stock mismatch" as the reason.) When a job prints, its form automatically becomes the current mounted form for the queue on which it is printing. This will not change the stock of the queue's current form, so it has no effect on job scheduling, but it may change other form attributes.

A form's print stock, despite its name, has no effect on the physical medium used for printing the job. It is only used for scheduling. The name simply suggests its intended use – to allow an operator to configure the printer, typically by loading required media, before allowing a job to print.

Both uses for forms, scheduling and formatting control, are relevant to PrintKit jobs. Formatting attributes apply to ANSI-PPL3 and PCL jobs, and setup modules can be used with all data types. For more complete information on using print forms, refer to the *OpenVMS System Manager's Manual: Essentials*.

To use a VMS form, you must define the form, make any needed changes to the default form, assign a default form for each execution queue, and inform users of the available forms and the queues with which they should be used.

Managing form definitions

Use the SHOW QUEUE/FORM/FULL command to display form definitions. Specify the form name, or omit it to display all forms.

```
$ SHOW QUEUE/FORM/FULL DEFAULT
Form name   Number   Description
_____       ___      _______
DEFAULT     0        System-defined default
```


[1]
ANSI data type
PCL data type
SHOW ENTRY command
SHOW QUEUE KITCP command
data type
execution queue
managing queues
setup modules
managing queues
setup modules

```
/LENGTH=66 /MARGIN=(BOTTOM=6) /STOCK=DEFAULT
/TRUNCATE /WIDTH=132
```

To create a new form, enter the DEFINE/FORM command:

```
$ DEFINE/FORM form-name form-number [/qualifiers]
```

Briefly, the DEFINE/FORM qualifiers relevant for PrintKit are:

/DESCRIPTION="string" A text description of the form.

/LENGTH=number /WIDTH=number The length and width of the form, in units of lines and columns, respectively. Defaults are 66 and 132, respectively.

/MARGIN=(margin=number,...) Settings for BOTTOM, LEFT, RIGHT, and TOP margins. Top and bottom margins are in units of lines, left and right are in columns. Default is 6 lines for bottom margin, 0 for others.

/SETUP=(module,...) /PAGE_SETUP=(option,...) Device control modules to be incorporated into the document when it is printed. Setup modules are placed at the beginning of the document, page setup modules at the beginning of each page.

/STOCK=name Paper stock. Default is the name of the form.

/TRUNCATE /WRAP Treatment of lines exceeding the form's line length (its width minus left and right margins). Truncate causes extra characters to be discarded, wrap places them on the next line.

The DEFINE/FORM command is also used to modify existing form definitions. If the form-name and form-number match an existing form, the command qualifiers modify the current definition.

The following example creates a form without margins, with a paper stock that matches the DEFAULT form:

```
$ DEFINE/FORM NOMARGINS 100 -
    /DESCR="DEFAULT stock, no margins"
    /MARGINS=(TOP=0,BOT=0,LEF=0,RIG=0)
    /STOCK=DEFAULT
```

Use the DELETE/FORM command to delete a form definition. For example, to delete the form MEMO, you would enter:

```
$ DELETE /FORM MEMO
```

For a more detailed description of the DEFINE/FORM command, refer to the *OpenVMS DCL Dictionary*.

Assigning forms to a queue

To display the current mounted form and the default form for a queue, use the SHOW QUEUE/FULL command. For example:

```
$ SHOW QUEUE/FULL JEAN_PRINT
```

Set the current mounted form for a queue by specifying the /FORM_MOUNTED qualifier with the KITCP MODIFY QUEUE command:

```
$ KITCP MODIFY QUEUE/FORM_MOUNTED=form_name
```

Set the default form for an execution queue by specifying the /DEFAULT qualifier with the KITCP MODIFY QUEUE command:

```
$ KITCP MODIFY QUEUE/DEFAULT=FORM=form-name
```

The INITIALIZE/QUEUE, START/QUEUE, and SET QUEUE commands can also be used to change the current and default forms for a queue. However, these changes are not visible to KITCP. If KITCP is subsequently used to modify any queue settings, the forms settings will be reset to the last values set using KITCP.

Managing device control libraries

Device Control Libraries, associated with each PrintKit execution queue, contain standard control and configuration modules required by the PrintKit software, as well as any setup modules defined for use at your site. There may be several libraries associated with each execution queue. PrintKit uses this capability to segregate its standard modules from site-defined modules, and to differentiate setup modules according to their data types (PostScript or ANSI-PPL3, for instance).

Device Control Libraries are always located in SYS$LIBRARY, and each may have an associated data type. The first library associated with a PrintKit queue, normally PRINTKIT023.TLB, contains the standard modules required by PrintKit. It has no specific

[1]
INITIALIZE/QUEUE command
MODIFY QUEUE KITCP command
START/QUEUE command
execution queue
ANSI data type
SHOW QUEUE command
PRINTKIT023.TLB Device Control Library
configuration
data type
device control library
execution queue
managing queues
setup modules

data type, and its contents are managed directly by PrintKit. You should *not* place setup modules in this library. Additional data-typed libraries follow the first. They are intended as storage for setup modules.

You can determine the libraries for a PrintKit queue by using the KITCP SHOW QUEUE/FULL command to display the /LIBRARY attribute for the queue. (The DCL SHOW QUEUE command will *not* display the full library list for the queue.)

Setup modules

When you print a document using setup modules, PrintKit collects the modules from the Device Control Libraries associated with the print queue and prepends them to the document before transferring it to the printer. The places PrintKit checks for a setup module are as follows:

> modules in each library with a data type matching the document data type;

> if the document data type is one that PrintKit translates before printing (ANSI-PPL3 to PostScript, for instance), modules in each library with a data type matching the translated data type;

> if the document data type or translated data type is PostScript, entries in the printer-specific PPD file being used for the queue.

Examining library contents

You can use the OpenVMS Librarian to list the contents of a Device Control Library, using the LIBRARIAN/LIST command.

For example, to list the contents of the PRINTKIT_ANSI library you would use the following command:

```
$ LIBRARIAN/LIST SYS$LIBRARY:PRINTKIT_ANSI.TLB
Directory of TEXT library SYS$LIBRARY:PRINTKIT_ANSI.TLB;1
Creation date: 24-NOV-1993 11:52   Creator: VAX-11 Librarian
Revision date: 7-JUL-1995 15:41    Library format: 3.0
Number of modules: 2               Max. key length: 39
Other entries: 0                   Allocated index blocks: 11
Recoverable deleted blocks: 0      Total index blocks used: 1
Max. Number history records: 20    Library history records: 2

PPL$CCOURIR_FONTS
```


[1]
ANSI data type
data type
device control library
PostScript Printer Definition (PPD) files
setup modules
ANSI data type
device control library
setup modules

```
PPL$CLUCSMR_FONTS
```

If you want to examine the contents of a library module, you will need to extract the module from the library, using the LIBRARIAN/EXTRACT command.

For example, to examine the ANSI-PPL3 setup module COMPACT_SETTINGS in the PRINTKIT_ANSI library, you would use the following commands:

```
$ LIBRARIAN SYS$LIBRARY:PRINTKIT_ANSI.TLB -
    /EXTRACT=COMPACT_SETTINGS -
    /OUTPUT=COMPACT_SETTINGS.TXT
```

Be sure to specify the /OUTPUT qualifier – by default, the Librarian uses the name of the library, not the module, as the name of the file created for the extracted module.

Adding new library modules

To add a setup module to the PrintKit Device Control Libraries, follow these steps:

> Choose the library in which you will load the module, making sure it has the required data type if you are adding a setup module.

> Stop any active PrintKit execution queues that use the library (you cannot update a library that is in use – the librarian will hang, waiting to gain access to the library). On a lightly used system, you may be able to make changes while queues are idle, but on a more active system you will need to stop all PrintKit queues. Use the KITCP STOP QUEUE/ALL command to stop all PrintKit queues, or use the DCL STOP/QUEUE command to stop individual queues.

> Add the module to the library, using the LIBRARIAN/REPLACE command.

> Restart any queues you stopped.

For example, to add an ANSI-PPL3 setup module COMPACT_SETTINGS, stored in the file COMPACT_SETTINGS.TXT, to the PRINTKIT_ANSI library, used by the PRINTKIT queue, you would use the following commands:

```
$ STOP/NEXT PRINTKIT
$ LIBRARIAN/REPLACE SYS$LIBRARY:PRINTKIT_ANSI.TLB -
    COMPACT_SETTINGS.TXT
$ START/QUEUE PRINTKIT
```


[1] setup modules

KITCP Administration Utility
ANSI data type
STOP/QUEUE command
data type
device control library
execution queue

Adding new libraries

PrintKit's ability to use multiple Device Control Libraries can help you organize setup modules by grouping them into separate libraries. This also allows you to selectively share setup modules between printers.

Use the KITCP MODIFY QUEUE command to add a new library to a PrintKit queue. For instance, to add a new LOCAL_PS library, containing PostScript setup modules:

```
$ KITCP MODIFY QUEUE PRINTKIT -
    /LIBRARY=(PRINTKIT023,-
        PRINTKIT_ANSI/DATA_TYPE=ANSI,-
        PRINTKIT_PCL/DATA_TYPE=PCL,-
        PRINTKIT_PS/DATA_TYPE=POSTSCRIPT,-
        LOCAL_PS/DATA_TYPE=POSTSCRIPT)
```

The /LIBRARY qualifier replaces the previous list of libraries, so be sure to specify all existing libraries when you add a new one.

Removing the PrintKit software

A deinstallation command procedure, SYS$MANAGER:PRINTKIT_DEINSTALL.COM, is provided with the PrintKit software. It removes all the files created when PrintKit was installed and deletes all PrintKit queues from the system.

To remove PrintKit, you execute the command file:

```
$ @SYS$MANAGER:PRINTKIT_DEINSTALL
```

The last file it removes is the deinstall command file itself.

[1]
MODIFY QUEUE KITCP command
device control library
setup modules
removing the PrintKit software

Chapter 5
Troubleshooting PrintKit

5.2 Introduction
5.2 Files affected by the installation
5.2 System requirements
- OpenVMS system version
- Disk space
- SYSGEN parameters

5.3 Commands for managing the print queue
5.4 Where to start troubleshooting
5.4 Enabling error and status messages
5.5 Printer status
5.5 Stopping execution queues
5.6 Verifying your configuration
- Queue configuration

5.7 Verifying the network connection

Introduction

This chapter provides supplementary information you may find useful in troubleshooting problems as you install the PrintKit software, and, later on, during normal use.

Files affected by the installation

The PrintKit installation creates or modifies these files:

SYS$SYSROOT:[SYSHLP.EXAMPLES.PRINTKIT]PRINTKIT_TEST.*
SYS$SYSROOT:[SYSHLP.EXAMPLES.PRINTKIT]*.PS
SYS$HELP:HELPLIB.HLB
  Modified by adding PrintKit help module.
SYS$HELP:KITCP.HLB
SYS$HELP:KITCP_INTERACTIVE.HLB
SYS$LIBRARY:PRINTKIT023.TLB
SYS$LIBRARY:PRINTKIT_ANSI.TLB,PRINTKIT_PCL.TLB,PRINTKIT_PS.TLB
  Empty libraries, for user-defined setup modules.
QMAN$MASTER:PRINTKIT_CONFIG_023.DAT
  (Placed in SYS$SYSTEM: if QMAN$MASTER: undefined)
SYS$MANAGER:PRINTKIT_DEINSTALL.COM
SYS$SYSTEM:KITCP.EXE
SYS$SYSTEM:PRINTKIT.EXE
SYS$TEST:PRINTKIT_IVP.COM

During the installation, you can elect to keep previous versions, or you can let the installation purge files, leaving only the new versions.

System requirements

The following system version and resources are required for successful operation of the PrintKit software.

[1]
troubleshooting
device control library
installation
setup modules
version, software

OpenVMS system version

All OpenVMS production versions are supported on Alpha and Integrity Servers. On VAX systems, you must be using OpenVMS V5.4 or higher. You can check the version with the SHOW SYSTEM command.

Disk space

You must have at least 3000 blocks of free space on the system disk. PrintKit requires a large amount of disk space for temporary files used during the installation. However, after installation it occupies less than 1500 blocks.

SYSGEN parameters

You must set the following SYSGEN parameters:

VIRTUALPAGECNT 15000 (minimum)
MAXBUF 8192 (minimum)

Use the following DCL commands to check that you set the SYSGEN parameters properly:

```
$ WRITE SYS$OUTPUT F$GETSYI("VIRTUALPAGECNT")
$ WRITE SYS$OUTPUT F$GETSYI("MAXBUF")
```

Use the AUTOGEN utility to increase these parameters if necessary. For more information on running AUTOGEN, see the *OpenVMS System Manager's Manual: Tuning, Monitoring, and Complex Systems* and the *OpenVMS System Management Utilities Reference Manual*.

Commands for managing the print queue

The following OpenVMS commands are available to you for managing the PrintKit device queues:

KITCP ADD QUEUE Create new PrintKit queues. Use it in place of the INITIALIZE/QUEUE command. For information on its use and information on related KITCP commands, see Execution queues, page 4.6, and Generic and logical queues, page 4.8.

KITCP SHOW QUEUE Display queue characteristics, configuration, and current state. For PrintKit queues, it provides more complete information than the SHOW QUEUE command. Specify /FULL for a complete listing.

START/QUEUE Start a queue that is paused or stopped.

STOP/NEXT Stop a print queue when the current job, if any, completes. The queue's PrintKit process will exit.

[1]
version, software
installation
AUTOGEN program
SYSGEN parameters
KITCP Administration Utility
INITIALIZE/QUEUE command
SHOW QUEUE command
configuration
managing queues

STOP/QUEUE Pause a queue. If the queue is processing a job, data transmission will stop, but the network link will remain open. Since access to the printer by other hosts is blocked, and the printer may time-out, you should not leave a job paused except for short periods of time. If the queue is not processing a job when this command is issued, it will simply prevent the queue from processing new jobs until the queue is resumed.

STOP/RESET Stop a print queue immediately, aborting any active print job. The queue's PrintKit process will exit.

Refer to the *OpenVMS DCL Dictionary* and the *OpenVMS System Manager's Manual: Essentials* for a full description of these commands.

Where to start troubleshooting

Depending on the nature of the problem you are having in printing with the PrintKit software, there are several techniques that can be used to identify and correct the problem.

Before you start to troubleshoot a problem, always be sure that you are receiving all error and status messages.

If you have a new installation or some or all of your queues have never performed correctly, check the status of the printer, then reset all queues, verify your configuration, and check your network connection.

If your queues have been working correctly, but printing is stalled, hung, has crashed, or you are getting communication errors, check the status of the printer, reset the problem queue or queues, and verify your network connection.

Enabling error and status messages

To troubleshoot any problems with PrintKit, be sure to have a terminal with operator messages enabled. Use the REPLY/ENABLE or REPLY/ENABLE=PRINTER commands to enable messages. You can use the REPLY/STATUS command to determine if messages are already enabled.

You can also examine the operator log file for messages already sent.

Jobs submitted for printing, especially any test jobs, should be submitted with the /NOTIFY qualifier. You may want to define a symbol PRINT as PRINT/NOTIFY in your LOGIN.COM file to make user notification the default.

[1]
configuration
installation
messages
status messages
troubleshooting
/NOTIFY qualifier
messages
notification messages
operator messages
status messages

For more information on these messages, see Chapter 6, Error and Status Messages.

Printer status

Physically check the printer to be sure that no error messages are being displayed, that the input trays contain paper, that it is not waiting for a manual feed, that its output tray is not full, and that it is on-line.

Even if there are no obvious problems with the printer, restarting a printer can correct many problems, and is worth trying before undertaking more extensive troubleshooting.

If the printer is hung or you need to change its configuration, reset all print queues that talk to the printer and then reset the queues before you reset or power-cycle the printer.

Stopping execution queues

To stop execution queues and close any network connection, queues must be stopped with the STOP/NEXT or STOP/RESET commands.

STOP/NEXT performs an orderly shutdown of a queue; it will wait for the current job, if any, to complete before stopping. STOP/RESET performs an immediate shutdown, aborting any current job. (STOP/QUEUE does not stop the queue at all – it causes the queue to pause.)

Use STOP/RESET if the queue is hung. When you stop the queue this way, you should check that the PrintKit process for the queue has also stopped. In certain error situations, the process may still be active, even though the queue appears to be stopped. When this happens, you will not be able to restart the queue until you manually stop the process.

There are several ways to determine whether the PrintKit process for a queue is active. For queues connected to a LAT or terminal device, you can specify SHOW DEVICE/FULL for the device to see if it has a non-zero Owner Process ID. You can also specify KITCP SHOW QUEUE/FULL for the queue – the last line of the display lists the process name and id for the queue's most recently active process. Do this from the node on which the queue is located – the information is not visible from other nodes. If neither of these two techniques identify an active process, and you still suspect that one is present, you can specify SHOW SYSTEM and look for process names beginning with PRINTKIT. In this case, it can be helpful to stop all the PrintKit queues on the system, rather than trying to identify

[1]
configuration
error messages
messages
STOP/NEXT command
STOP/QUEUE command
STOP/RESET command
execution queue
LAT network protocol
messages
operator messages
troubleshooting

which process is matched with which queue. If you find a process still active for a stopped queue, use STOP/ID to remove it.

If you have a particular job which fails, reset the queue with STOP/RESET, and then set the job to hold with SET ENTRY/HOLD so that a different job prints during the next attempt.

If PrintKit crashes (you receive a QMAN-E-SYMDEL, unexpected symbiont process termination operator message), it may create a PRINTKIT.DMP file in the SYS$SYSTEM: directory. This dump file can provide support personnel with valuable information in diagnosing problems with the software. It also consumes several thousand blocks of disk space. If you are having problems with the PrintKit software, you should check whether dump files have been created, and remove any that are not needed for troubleshooting.

Verifying your configuration

To verify that your configuration is set up properly for your network connection, compare configuration information for consistency in your PrintKit startup command file, printer configuration pages, and current definitions of logical names and queues on your system.

Queue configuration

You can examine the current configuration of a PrintKit queue with the KITCP SHOW QUEUE/FULL command. For example:

```
$ KITCP SHOW QUEUE PRINTKIT/FULL
Printer queue PRINTKIT, idle, on DON::ESA0:, mounted form DEFAULT
   <PrintKit Automatic Queue>
   /BASE_PRIORITY=4
   /DEFAULT=(FEED,FORM=DEFAULT,SHEET_SIZE=LETTER)
   /OWNER=[SYSTEM] /PROCESSOR=PRINTKIT
   /PROTECTION=(S:E,O:D,G:R,W:W) /RETAIN=ERROR
   /SCHEDULE=(NOSIZE)

/LIBRARY=(PRINTKIT,PRINTKIT_ANSI/DATA_TYPE=ANSI,
   PRINTKIT_PCL/DATA_TYPE=PCL,
   PRINTKIT_PS/DATA_TYPE=POSTSCRIPT)
/PRINTER=(MODEL="ACME LaserPrinter Model II")
/COMMUNICATIONS=(ADDRESS=192.0.1.209)
```


[1]
configuration
logical name
startup command file
PRINTKIT023.TLB Device Control Library
configuration
device control library
execution queue
queue status
setup modules
symbiont

```
Process name PrintKit_257, Process ID 404002a9
```

Verify the queue status. An inactive queue should be stopped, paused, or idle. A queue will be busy or stop pending for most of the time it is processing a print job, although it will remain starting while it waits for the printer to finish another job. Otherwise, it is an indication of problems if a queue is aborting, stalled, or starting for an extended period of time. For hints on interpreting the queue status, see Queue status, page 6.5.

Verify that the /ON qualifier specifies the correct node and device for the queue. Make sure the processor is PRINTKIT and the list of device control libraries includes PRINTKIT023.TLB, plus any libraries containing setup modules you have created for use with the printer.

Verify that the /COMMUNICATIONS and /PRINTER qualifiers match the configuration of your printer and its communications interface. A *current* test page from your printer will help you verify that all options are correctly specified, and that nothing has changed in your printer configuration.

Logical and generic queues should be idle and assigned to valid PrintKit execution queues.

The process name and process id identify the symbiont for an active queue, or the last active symbiont for a queue that is stopped. This information is visible only from the node on which the queue is located. You can use it to check the state of the symbiont, or to verify that the symbiont has exited properly when a queue is stopped.

Verifying the network connection

Check the cabling between your host and the printer. Verify the operation of all hubs, transceivers, bridges, etc. Transceiver cables often make a poor connection at the printer end.

Chapter 6
Error and Status Messages

6.2 Introduction
6.2 Operator messages
6.3 Notification messages
6.5 Queue status
6.6 Job status
6.7 Message log file
6.7 OpenVMS system messages

[1]
messages
status messages

Introduction

Error and status messages are provided in a number of ways:

Operator messages These include general messages that do not pertain to a specific print job, such as execution queue startup, or that relate to the communication link or the status of the device, such as queue stall.

User notification messages These are messages that are broadcast to a specific user's terminal and describe the status of the print job being processed. These messages are only broadcast to the user if the /NOTIFY qualifier was specified when the job was submitted. Operator messages that occur during the processing of a user's job are also broadcast to the user if /NOTIFY was specified.

Queue status The SHOW QUEUE command can be used to determine the current queue status, which may be busy, idle, paused, stalled, starting, stop pending, or stopped.

Job status The SHOW ENTRY or SHOW QUEUE commands can be used to determine the current print job status, which may be aborting, holding, pending, printing, retained, stalled, starting, or suspended. If a job fails, up to three OpenVMS system messages describing the error are shown.

OpenVMS system messages Operator, user notification, queue status, and job status messages all use OpenVMS system messages to describe in detail error conditions. Additional information on these messages can be found in the *OpenVMS System Messages and Recovery Procedures Reference Manual*.

Operator messages

You receive Operator messages sent by the Operator Communication facility (OPCOM) at terminals designated to receive messages with the REPLY/ENABLE command. These messages are also logged in the system operator message file. Operator messages generated by print symbionts, such as PrintKit, are normally sent to terminals who have enabled receipt of messages of the PRINTER class. In addition, PrintKit can be configured to send its messages to any class or classes of operators using the REQUEST_TO parameter as a default for an execution queue.

The following operator messages are generated by PrintKit. With the exception of queue startup, stall, and resume messages, all operator messages include one or more OpenVMS

[1]
/NOTIFY qualifier
SHOW ENTRY command
SHOW QUEUE KITCP command
SHOW QUEUE command
broadcast messages
execution queue
job status
messages
notification messages
operator messages
queue status
retained job
status messages
REQUEST_TO parameter
STOP/QUEUE command
configuration
device control library
execution queue
messages
OPCOM (Operator Communications Facility)
operator messages
version, software
version, specification

system messages to describe the error condition in detail. All operator messages begin with:

PrintKit Queue *queue-name* on *device-name*

Connect failure Initial attempts to make a network connection with the device have failed. Detail is provided with OpenVMS system messages. Connection attempts will be retried at five second intervals until the connection succeeds or the queue is manually reset.

Job (entry nnn) startup failure A print job failed to start due to a bad configuration parameter or a failure of the network link configuration. Detail is provided with OpenVMS system messages.

Resumed Data transmission has resumed after being stalled.

Stalled A network connection has been made to the device, but data transmission has stalled for a period of time which may indicate that the printer is out of paper, or is waiting for a manual feed, etc. The reason for the stall is not known by PrintKit. Data has not been lost.

Startup The start queue operation has succeeded and the execution queue is ready to accept jobs for processing. This message includes the PrintKit version number, the license token, and a copyright message.

Startup failure The network device or device control library are not available or some other fatal condition prevents the execution queue from startup. Detail is provided with OpenVMS system messages.

Startup has stalled Attempts to connect to the device to process a print job are waiting for another user of the device.

Task resume failure An execution queue which has been paused with the STOP/QUEUE command is unable to resume. Detail is provided with OpenVMS system messages.

Notification messages

You may receive user notification messages broadcast at your terminal to describe the status of your print jobs. To receive these messages you must submit your print jobs specifying the /NOTIFY command qualifier, be logged in to the same cluster as the PrintKit execution queue at the time it processes your job, and have broadcast messages enabled for your terminal. (Broadcast is controlled by the SET TERMINAL/BROADCAST and SET

[1]
/NOTIFY qualifier
DELETE/ENTRY command
STOP/ABORT command
STOP/QUEUE command
STOP/RESET command

broadcast messages
execution queue
message log file
messages
notification messages
retained job
setup modules

BROADCAST commands.) The following user notification mesages are broadcast by PrintKit. Except where noted, the messages include one or more OpenVMS system messages to describe the error condition in detail. OpenVMS system messages are described later in this chapter. User notification messages all begin with:

```
PrintKit Job name (queue name, entry number)
```

Job aborted A print job was aborted by a STOP/ABORT, STOP/RESET or DELETE/ENTRY command. By default, the job will be retained in the queue showing the abort status.

Job completed Indicates successful completion of a print job.

Job connect failed A print job, which is starting, has failed to form a network connection to the printer. Detail is provided with one or more OpenVMS system messages, which also indicate whether attempts to connect will continue. Attempts to connect to a printer that is busy with a job from another host will not normally produce this message – the print job simply remains in the starting state and the connect request is queued. In other cases, such as when the printer is not on-line, is not accessible, or an excessive number of hosts are trying to connect so the connect queue overflows, a connect failure will be reported. Failure can also occur if the connection made by another host does not terminate normally. Occasionally, manual intervention is required to allow the connect to succeed, such as putting the printer on-line, verifying the network connection, resetting the printer, or in extreme cases, power-cycling the printer.

Job failed A print job has completed with a failure status, after successfully starting. Detail is provided with one or more OpenVMS system messages. Print jobs may fail if the input files or setup modules cannot be read, or the network link to the printer is lost. By default, the job will also be retained in the queue showing the failure status, although the user notification message may show more detail than the retained queue entry.

Job paused A print job has been paused with the STOP/QUEUE command. The network link remains open, blocking other hosts from accessing the printer. If the job remains paused for an excessive period of time, the printer will time-out and the job will not successfully resume.

Job resumed A print job that was paused has been resumed.

Job resume failure The attempt to resume a print job that was paused has failed. Detail is provided with one or more OpenVMS system messages.

Job startup failed A print job has failed its initial startup which includes parsing the print command parameters, creating a message log file, opening the input files, or failing to queue the initial network connection request. Detail is provided with one or more OpenVMS system messages. By default, the job will also be retained in the queue showing the failure status, although the user notification message may show more detail than the retained queue entry.

Queue status

Use the SHOW QUEUE command to determine the current queue status, which may be busy, idle, paused, stalled, starting, stop pending, or stopped. It also shows job status for any jobs owned by you. See the *OpenVMS DCL Dictionary* or the *OpenVMS User's Manual* for additional information. For example:

```
$ SHOW QUEUE PRINTKIT
Printer queue PRINTKIT, idle, on DON::_ESA0:
  mounted form DEFAULT
  <PrintKit Automatic Queue>
```

In this example, queue PRINTKIT is a "Printer queue" (execution queue), which is idle. It executes on node DON, and communicates using Ethernet device ESA0 to a printer with an Internet interface. The user who issued the command has no jobs waiting in the queue. The queue may contain jobs belonging to other users – the /ALL qualifier can be used to display these jobs.

The queue may be in one of the following states:

busy A print job is being processed. The job entry status will show additional information about the state of the job.

idle The print queue is running, and there is no print job to process. There is one PrintKit symbiont process for each running execution queue.

paused The print queue has been paused with the STOP/QUEUE command. If there is no current job, no new ones will be started. If a current job is running, data transmission to the device will pause until the queue is resumed or the printer times out. If the printer times out, the print job will fail to resume. Since the network connection to the printer is held open, the printer is made unavailable to other users on the network; therefore you should not leave queues in a paused state for very long. The PrintKit symbiont process remains running for an execution queue that is paused.

[1]
SHOW QUEUE command
STOP/NEXT command
STOP/QUEUE command
STOP/RESET command
configuration
execution queue
job status
License PAK
messages
operator messages
queue status
symbiont

stalled No data has been accepted by the printer for over 60 seconds for a print job being processed by an execution queue. This may indicate that the printer is jammed, out of paper, or waiting for a manual feed. The reason for the stall is unknown to the system since the printer cannot transmit status information when data is stalled.

starting The execution queue is starting. It should take only a few seconds for the queue to start. The configuration information, default parameters, and license PAK are validated, then PrintKit announces the completion of queue startup with an operator message.

stop pending The queue is processing a job and has been asked to stop with the STOP/NEXT command. The PrintKit symbiont process will exit when the current job completes. A STOP/RESET will force the queue to stop immediately.

stopped The queue is not running and has been stopped with a STOP/NEXT or STOP/RESET command. There should be no PrintKit symbiont process running for the queue.

SHOW QUEUE/FULL includes additional configuration information about the queue, and SHOW QUEUE/ALL includes job status for all jobs in the queues, depending on their user name and whether you have GROUP or WORLD privileges.

Job status

The SHOW ENTRY or SHOW QUEUE commands can be used to determine the current print job status, which may any one of the following:

aborting The system is attempting to abort processing of the job. The queue should remain in this state for only a very short time.

holding The print job was submitted with the /HOLD qualifier, and will not be processed until the job is released with the SET ENTRY/RELEASE command.

pending The print job is waiting to begin processing. It may be waiting for an execution queue to be started, to complete processing another job, or for other jobs who were submitted earlier or are of higher priority.

printing The print job is being processed. Also, check the queue status to see if processing has stalled.

retained The print job has completed. By default, PrintKit execution queues are configured to retain jobs only when they complete with an error status. The entry will display up to three OpenVMS system messages to describe the error. Unfortunately, the message text

[1]
/HOLD qualifier
/NOTIFY qualifier
SHOW ENTRY command
SHOW QUEUE KITCP command
error messages
execution queue
job status
messages
queue status
retained job

for all errors are not known to the system, and error message arguments are not retained. To view the complete messages when a print job completes, specify the /NOTIFY qualifier when submitting your jobs.

stalled Printing of the job has stalled. Either the printer has reported a problem that has caused it to stop printing the job, or transfer of data to the printer has stalled.

starting The job has been scheduled for execution, but PrintKit has not yet acknowledged that it is processing the job. PrintKit attempts to establish communications with the printer before reports that the job is executing. When a job stays in the starting state, it typically means the printer is not responding, or it is refusing connection requests from PrintKit.

suspended Printing of the job is suspended because the queue on which it is executing is paused.

Message log file

If the user specifies the MESSAGES=KEEP print parameter, any messages sent to PrintKit by the printer as it processes the print job are written to a file in the user's default login directory. The file is given a name of PRINTKIT_JOB_*entry-number*.LOG. There is no limit to the number of messages written to this file.

In addition to the job messages, the file includes job description information, in a similar form as on file flag pages. This information will be written to the file even if no messages are recorded.

OpenVMS system messages

Operator, user notification, queue status, and job status messages all use OpenVMS system messages to describe error conditions. Messages used by PrintKit queues and the KITCP command are described below. In addition, messages produced by other OpenVMS facilities used by PrintKit may not be listed here, for example, RMS facility messages encountered while reading a print job input file, LIBRARY facility messages encountered while reading setup modules from a setup module library, CLI facility messages encountered while processing job parameters or command input. For information on messages not listed here, or for additional information, see the *OpenVMS System Messages and Recovery Procedures Reference Manual*.

[1]
flag page
message log file
messages
KITCP Administration Utility
/NOTIFY qualifier
device control library
job status
message log file
messages
notification messages
operator messages
queue status
retained job
setup modules
status messages

Some of these messages contain variable text which is shown when the message is displayed in an operator message, user notification message, or in a message log. This text is lost from the messages generated from the completion status of operations such as print jobs or command execution; a coded marker (such as !AS or !UL) is displayed in its place. You may see these incomplete messages for job entries retained on error and when starting or stopping PrintKit queues. Use the /NOTIFY qualifier to view the full completion status of PRINT commands. Enable operator messages (REPLY/ENABLE=PRINTER) to view the full completion status of queue operations.

KIT-F-ACCVIO, access violation, reason mask=00, virtual address=*address*, PC=*address*, PS=*status*

PrintKit or KITCP has attempted an invalid memory access. This may be caused by a bug in the software, and may also be caused by insufficient virtual memory. If the error is reported for a PrintKit queue, check that you have configured adequate virtual memory for the PrintKit symbiont. If the error was caused by a software bug, you may need to delete the job that was printing at the time of the error before restarting the queue. Please submit a problem report.

KIT-E-ATTRERR, error processing *name* attribute: *explanation*

PrintKit encountered a problem processing the named attribute. This error can occur during queue startup, as PrintKit processes configuration attributes, and it can also occur during printing, as PrintKit processes job parameters. The explanation provides information as to the source of the problem:

invalid attribute syntax The value specified for the attribute does not conform to the syntax required for the attribute.

undefined attribute value The value specified for the attribute is not recognized.

unsupported attribute type The attribute is recognized, but is not supported by the version of the PrintKit software in use.

unsupported attribute value The value specified for the attribute is recognized, but is not supported by the printer.

mandatory attribute omitted A required attribute was not specified.

[1]
KITCP Administration Utility
ACCVIO message
symbiont
ATTRERR message
configuration

KIT-W-BADLAYDEF, syntax error in layup definition file *name*, line *number*

KITCP encountered an error interpreting the named layup definition file, on the indicated line. Inspect the file and correct the error.

KIT-F-BADLOGIC, internal logic error detected

An internal PrintKit sanity check has failed. A secondary KIT-I-TEXT message will describe the error. The software will not be able to continue operation. Stop the execution queue using STOP/RESET if it is not already stopped, then restart it. Please submit a problem report.

KIT-W-BOOTPERR, error occurred attempting to perform bootp services

PrintKit is configured to perform BOOTP services to provide the printer with the IP address it needs for TCP/IP communications. The BOOTP processing has failed, and PrintKit has not supplied addressing for the printer.

SYSTEM-F-BUGCHECK, internal consistency failure

An internal PrintKit sanity check has failed. The software will not be able to continue operation. Stop the execution queue using STOP/RESET if it is not already stopped, then restart it. Please submit a problem report.

KIT-W-CFGQUEUPD, queue updated in configuration database

KITCP detected an inconsistency between a queue's definition in the system job queue file and its definition the PrintKit configuration database. KITCP has adjusted the definition in the PrintKit configuration database to match the system job queue file. No further action is required.

PSM-E-CLOSEIN, error closing *file-name* as input

PrintKit encountered an error closing the indicated input file or setup module library. A secondary error message further describes the error. This error should not occur under normal circumstances.

PSM-E-CLOSEOUT, error closing *file-name* as output

PrintKit encountered an error disconnecting or deallocating the indicated output device. A secondary error message further describes the error.

[1]
KITCP Administration Utility
BADLAYDEF message
page layout
BADLOGIC message
STOP/RESET command
execution queue
BOOTPERR message
TCP/IP network protocol
BUGCHECK message
STOP/RESET command
execution queue
KITCP Administration Utility
CFGQUEUPD message
configuration
CLOSEIN message
device control library
error messages
setup modules
CLOSEOUT message
error messages

KIT-I-CONATMPT, continuing attempts to connect to *device-name*

Attempts to establish communications with the printer on the indicated device are continuing after a failure. A secondary error message further describes the error.

KIT-W-CONFAIL, trial connection to printer not established

PrintKit attempts to make a trial connection to the printer when the queue is started. This message indicates the trial connection failed. This can indicate that the PrintKit queue is incorrectly configured and is looking for the printer at an incorrect address. It can also simply indicate that the printer is unavailable, because it is turned off, is busy with another job, or there is a problem in the communications path.

PSM-W-ENDABORT, *data-type* compilation aborted

Processing of your print job has been aborted by the printer. This error commonly occurs when there are errors in a PostScript document. It may also result from printer hardware errors, or a printer configuration that does not allow the job to be printed. Check the printer and any output for further information. Also, specifying the MESSAGES=KEEP parameter with your print job will produce a log file containing any PostScript errors reported by the printer.

LICENSE-F-EXLICENSE, licensed product has exceeded current license limits
SYSTEM-F-EXLICENSE, licensed product has exceeded current license limits

Your PrintKit license PAK is for a limited number of printers, and starting this execution queue would exceed that limit. Verify that you registered and loaded all your license PAKs, in accordance with the terms and conditions of your license, on nodes that will run PrintKit execution queues. You may need additional license units to support all of your printers. Contact your PrintKit distributor for assistance.

This error can also occur if the PrintKit symbiont fails to terminate properly when a queue is stopped. See Stopping execution queues, page 5.5, for more information on detecting and correcting this condition.

PSM-F-FUNNOTSUP, function not supported

The PrintKit symbiont has received an unrecognized request from the job control program. This error should not occur under normal circumstances.

[1]
CONATMPT message
error messages
CONFAIL message
ENDABORT message
POSTSCRIPT data type
configuration
EXLICENSE message
execution queue
License PAK
symbiont
FUNNOTSUP message
Job Control Program
symbiont

KIT-E-INCCONFIGVER, incompatible configuration database version

The format of the PrintKit configuration database is not compatible with this version of the software. The database file, PRINTKIT_CONFIG_023.DAT, is located in the same directory as QMAN$MASTER.DAT (normally SYS$SYSTEM:). This error can occur if the database file has inadvertantly been replaced with a file from a different version of PrintKit.

SYSTEM-F-INSFMEM, insufficient dynamic memory

An attempt to allocate dynamic memory for internal buffers or data structures has failed. Check that you have configured adequate virtual and paging memory for the PrintKit symbiont. The software may not be able to continue successful operation. Stop the execution queue using STOP/RESET if it is not already stopped, then restart it.

LICENSE-F-INVLICENSE, licensed product is not authorized for this operating mode

SYSTEM-F-INVLICENSE, licensed product is not authorized for this operating mode

Your PrintKit license PAK is not valid for your configuration, has been tampered with, or you are using an incompatable version of the License Management Facility. Verify that you registered and loaded all your license PAKs, in accordance with the terms and conditions of your license, on nodes that will run PrintKit execution queues. Contact your PrintKit distributor for assistance.

LICENSE-F-LICENSE_LEVEL, license level does not match product version

SYSTEM-F-LICENSE_LEVEL, license level does not match product version

Your PrintKit license PAK is for an earlier version of the software than you have installed on your system. Verify that you registered and loaded all your license PAKs, in accordance with the terms and conditions of your license, on nodes that will run PrintKit execution queues. You may need additional license units to support all of your printers. Contact your PrintKit distributor for assistance.

PSM-E-MODNOTFND, library module *module-name* not found in record *number*

A setup module was not found in the execution queue setup module library. The record number argument is not used. This message is followed by another message further describing the error. Setup modules may be explicitly requested when the print job is submitted or implicitly by a print form. PrintKit also loads setup modules for its own use. Verify that the print request was submitted with the correct setup module name, that the form used is properly defined, that the execution queue is configured for the correct library, and that the library contains the modules being requested.

[1]
INCCONFIGVER message
configuration
INSFMEM message
STOP/RESET command
execution queue
symbiont
INVLICENSE message
configuration
execution queue
License PAK
LICENSE_LEVEL message
execution queue
License PAK
version, software
MODNOTFND message
device control library
execution queue
setup modules

KIT-E-NOCONFIG, no configuration database specified

You have attempted to apply a KITCP command to the PrintKit configuration database, but there is no database currently specified. Use the SET DATABASE command, then retry the command.

LICENSE-F-NOLICENSE, operation requires software license
SYSTEM-F-NOLICENSE, operation requires software license

PrintKit was unable to find a valid license on the node the execution queue runs on. Verify that you registered and loaded all your license PAKs, in accordance with the terms and conditions of your license, on nodes which will run PrintKit execution queues. Contact your PrintKit distributor for assistance.

KIT-W-NOMATCHOBJ, no matching objects

A KITCP RELINK command did not find any references matching the object name you specified. Check that you specified the object name as intended.

KIT-E-NOREFOBJ, referenced object is not defined

A name you have specified in a KITCP command qualifier does not refer to a defined object. You can use the SHOW command to display defined objects of different types.

KIT-E-NOSUCHDEV, no such device available

No suitable network device was found for a queue configured for printing on the symbolic TCPIP device. Network device selection is described in Supported TCP/IP interfaces, page 4.4. This message should only appear as a secondary message.

KIT-E-NOSUCHOBJ, no such object

The object you have specified in a KITCP COPY, MODIFY, SHOW, or REMOVE command is not defined. You can use the SHOW command to display defined objects of different types.

KIT-E-NOWILD, no wildcard permitted

You have specified a name containing wildcard characters in a KITCP COPY command in a context where wildcards are not allowed. Wildcards are allowed only when copying from one database to another using the /DATABASE qualifier. In this case, if the old name contains wildcards, the new name must be omitted or be specified as a single asterisk wildcard (*).

[1]
KITCP Administration Utility
NOCONFIG message
SET DATABASE KITCP command
configuration
NOLICENSE message
execution queue
KITCP Administration Utility
NOMATCHOBJ message
KITCP Administration Utility
NOREFOBJ message
NOSUCHDEV message
messages
KITCP Administration Utility
NOSUCHOBJ message
KITCP Administration Utility
NOWILD message

KIT-E-OBJEXISTS, object already exists

The object you have specified in a KITCP ADD or COPY command already exists. Either specify the /REPLACE qualifier or use the MODIFY command instead.

KIT-E-OBJNOTADDED, object *name* not added

KITCP was unable to perform an ADD command for the named object. A secondary error message further describes the error.

KIT-E-OBJNOTCOPIED, object *name* not copied

KITCP was unable to perform a COPY command for the named object. A secondary error message further describes the error.

KIT-E-OBJNOTMODIFIED, object *name* not modified

KITCP was unable to perform a MODIFY command for the named object. A secondary error message further describes the error.

KIT-E-OBJNOTREMOVED, object *name* not removed

KITCP was unable to perform a REMOVE command for the named object. A secondary error message further describes the error.

KIT-E-OBJNOTSHOWN, object *name* not shown

KITCP was unable to perform a SHOW command for the named object. A secondary error message further describes the error.

KIT-E-OPENIN, error opening *file-name* as as input

KITCP was unable to open the named configuration database file. A secondary error message further describes the error. Check that the file exists, and that you have privileges to access it.

PSM-E-OPENIN, error opening *file-name* as as input

PrintKit encountered an error opening the indicated input file or setup module library. A secondary error message further describes the error. Verify that the files submitted for printing were not purged or deleted before the print job began processing, and that the setup module library exists and is accessible to the symbiont.

[1]
KITCP Administration Utility
OBJEXISTS message
KITCP Administration Utility
OBJNOTADDED message
error messages
messages
KITCP Administration Utility
OBJNOTCOPIED message
error messages
messages
KITCP Administration Utility
OBJNOTMODIFIED message
error messages
messages
KITCP Administration Utility
OBJNOTREMOVED message
error messages
messages
KITCP Administration Utility
OBJNOTSHOWN message
error messages
messages
KITCP Administration Utility
OPENIN message
configuration
error messages
messages
OPEN
device co
error mes
setup mo
symbiont

KIT-E-OPENOUT, error opening *file-name* as output

KITCP was unable to open the named file for output from a SHOW command. A secondary error message further describes the error.

PSM-E-OPENOUT, error opening *device-name* as output

PrintKit encountered an error allocating or connecting the indicated output device. A secondary error message further describes the error. Verify that the execution queue is configured for the correct device and that the device has been created and initialized. This error may occur because an execution queue was started without first successfully completing the PrintKit startup command file. It may also occur when another process already has a channel assigned to the device.

PSM-F-PARSEFAIL, error parsing '*parameter*'

The indicated print parameter, specified when the print job was submitted, contains invalid syntax. Verify the syntax of all print parameters specified for the job with the SHOW ENTRY/FULL command.

KIT-F-PPDFMTERR, PostScript Printer Description file format error encountered at line *number*

A syntax error was encountered in a PostScript Printer Description file used to configure PrintKit, on the indicated line.

SYSTEM-F-PROTOCOL, network protocol error

This error is reported when the device on which a queue is configured for printing is not compatible with the communications protocol for which the queue is configured. It also may result from errors generated by the protocol used for communications with the printer. This message should only appear as a secondary message.

KIT-E-QUALMISS, required qualifier *name* is missing

A KITCP command is missing the specified qualifier, which is required in order to perform the command.

KIT-E-QUENOTSTART, queue *name* not started

KITCP was unable to start the named queue. A secondary error message further describes the error.

[1]
KITCP Administration Utility
OPENOUT message
error messages
messages
OPENOUT message
error messages
execution queue
startup command file
PARSEFAIL message
POSTSCRIPT data type
PPDFMTERR message
PROTOCOL message
messages
KITCP Administration Utility
QUALMISS message
KITCP Administration Utility
QUENOTSTART message
error messages
messages

KIT-E-QUENOTSTOP, queue *name* not stopped

KITCP was unable to stop the named queue. A secondary error message further describes the error.

KIT-S-QUESTART, queue *name* started

KITCP started the named queue.

KIT-S-QUESTOP, queue *name* stopped

KITCP stopped the named queue.

PSM-E-READERR, error reading *file-name*

PrintKit encountered an error reading from the indicated input file, setup module library, or device. A secondary error message further describes the error. This error should not occur under normal circumstances.

KIT-I-RESTARTINGQUE, restarting queue *name*

KITCP is restarting the named queue after performing modifications. No action is required.

KIT-I-STOPPINGQUE, stopping queue *name*

KITCP is stopping the named queue in order to perform modifications. The queue will stop as soon as it becomes idle. Normally, no action is required; however, you will need to manually stop a queue that is processing a stalled print job. (The KITCP STOP QUEUE command will perform an immediate shutdown of the queue, aborting any current job.)

QMAN-E-SYMDEL, unexpected symbiont process termination

An execution queue has crashed and is now stopped. This message appears as part of a operator message and is normally preceeded by other messages which describe the error. Please submit a problem report.

PSM-F-SYNTAX, error parsing *parameter*

An unrecoverable error has occurred in parsing the indicated print parameter, and the execution queue is stopping. The queue must be restarted. Verify the syntax of all print parameters specified for the job with the SHOW ENTRY/FULL command before resubmitting the print job. Please submit a problem report.

[1]
KITCP Administration Utility
QUENOTSTOP message
error messages
messages
KITCP Administration Utility
QUESTART message
KITCP Administration Utility
QUESTOP message
READERR message
device control library
error messages
setup modules
KITCP Administration Utility
RESTARTINGQUE message
KITCP Administration Utility
STOP QUEUE KITCP command
STOPPINGQUE message
SYMDEL message
execution queue
operator messages
symbiont
SYNTAX message
execution queue

KIT-W-SYSQUECRE, queue added to system job queue file

A queue defined in the PrintKit configuration database had no corresponding definition in the system job queue file. KITCP has added the queue to the system job queue file. No further action is required.

KIT-F-TABLEOVER, data table *name* overflowed, maximum size is *value*

The indicated PrintKit data structure has overflowed. The software will not be able to continue operation. Stop the execution queue using STOP/RESET if it is not already stopped, then restart it. Please submit a problem report.

SYSTEM-E-UNSUPPORTED, unsupported operation or function

The printer does not have the emulator necessary to print this job. This message is only used as a secondary error message with PSM-E-BADVALUE.

KIT-E-USESETDATA, use SET DATABASE

This message only appears as a secondary message. You have attempted to apply a KITCP command to the PrintKit configuration database, but there is no database currently specified.

PSM-E-WRITEERR, error writing *device-name*

PrintKit encountered an error when writing data to the indicated device, which is a device used for communications with the printer or for spooled output. A secondary error message further describes the error. If this error occurs for a queue that has been working correctly, it typically indicates a problem with the printer or a network disruption.
For a new queue, it usuall indicates that incorrect communications settings have been configured for the printer.

KIT-W-ZEROAREA, margins result in no usable sheet area

The margins in a layup specification, when combined with the selected sheet size, result in an image area that is zero or negative.

[1]
KITCP Administration Utility
SYSQUECRE message
configuration
STOP/RESET command
TABLEOVER message
execution queue
UNSUPPORTED message
error messages
KITCP Administration Utility
USESETDATA message
configuration
messages
WRITEERR message
error messages
messages
ZEROAREA message
page layout

Appendix A
PRINT Command Reference

A.2 Introduction
A.2 Command qualifiers
A.9 Parameter options
A.23 Page selection expressions

[1]
PRINT command

Introduction

The OpenVMS PRINT command prepares a *print job* from a list of files to be printed. You can also specify *qualifiers* to control the appearance of the printed files and the overall handling and arrangement of the print job.

```
$ PRINT input-file,...
```

You invoke the PRINT command using normal command conventions. One or more input file specifications indicate the files to be included in the print job; wildcards are allowed, and the default file type is .LIS.

In addition to the controls provided by the standard PRINT qualifiers, the PrintKit software uses the /PARAMETERS command qualifier to provide an additional level of control over your print job. These options are compatible with the DECprint interface. This compatibility allows you to use the same job specifications for PrintKit and DECprint.

Command qualifiers

Qualifiers may follow individual input file specifications or the PRINT command itself. Qualifiers that apply to the job as a whole have the same meaning regardless of their position, but qualifiers that apply to files apply to all files when specified following the PRINT command or to the individual file with which they are associated.

Command and input file qualifiers are summarized in Table A-1, PRINT Qualifiers, page A.3. The description of each qualifier explains its syntax and function, indicates its default setting, and lists related qualifiers.

[1]
/PARAMETERS qualifier
DECprint Supervisor
PRINT command
PRINT command

/AFTER=time	Delay printing the files until after the specified time. If the time is already past, there is no delay. The time may be absolute (dd-mmm-yyyy hh:mm:ss.cc) or a delta time (ddd hh:mm:ss.cc). For time format details, see the *OpenVMS User's Manual*. See also /HOLD.
/BACKUP /NOBACKUP	Use files' most recent backup times to select which are printed in processing a /BEFORE or /AFTER qualifier.
/BEFORE=time /NOBEFORE	Select for printing only files dated before the specified time. Backup, creation (default), expiration, or modification time may be used, as indicated by /BACKUP, /CREATED, /EXPIRED, or /MODIFIED. The time may be absolute (dd-mmm-yyyy hh:mm:ss.cc) or a delta time (ddd hh:mm:ss.cc). For time format details, see the *OpenVMS User's Manual*.
/BURST=[ALL ONE] /NOBURST	Print a burst page preceding the file.
/BY_OWNER=uic /NOBY_OWNER	Select for printing only those files having the specified owner UIC.
/CHARACTERISTICS=(name,...)	Specify characteristics required for printing the files. Printing will be delayed until all the specified characteristics are set for the queue. See also /FORMS.

Table A-1, PRINT Qualifiers

/CONFIRM /NOCONFIRM	Select files for printing interactively. The PRINT command will prompt with the name of each file. The following responses are allowed: YES, TRUE, 1 Print the file. NO, FALSE, 0, return Do not print the file. QUIT, CTRL/Z Stop processing. ALL Stop prompting, continue processing.
/COPIES=number	Set the number of copies to print. See also /JOB_COUNT.
/CREATED /NOCREATED	Use files' creation times to select which are printed in processing a /BEFORE or /AFTER qualifier.
/DELETE /NODELETE	Delete file after printing.
/DEVICE=name	See /QUEUE.
/EXCLUDE=(file-spec,...) /NOEXCLUDE	Exclude the specified files from printing. File specifications may specify directory, file name, type, and absolute version numbers, and may use wildcards. Device names and relative version numbers are not allowed.
/EXPIRED /NOEXPIRED	Use files' expiration times to select which are printed in processing a /BEFORE or /AFTER qualifier.
/FEED /NOFEED	Break file across pages, advancing to a new page on reaching the bottom margin, as specified by the forms definition for the job. Affects only the ANSI data type. Default setting is a print queue attribute.

Table A-1, PRINT Qualifiers (Continued)

/FLAG /NOFLAG	Print a flag page for the file or job. Default setting is a print queue attribute.
/FORM=type	Specify the form type required for printing the files. Printing will be delayed until a compatible form type is set for the queue. See also /CHARACTERISTICS.
/HEADER /NOHEADER	Display a header line containing the page number and the name and revision date of the file being printed at the top of each page. Affects only ANSI and PostScript data types. Default setting is a print queue attribute.
/HOLD /NOHOLD	Control whether the files are immediately available for printing. The SET ENTRY/RELEASE command makes a job on hold available for printing.
/IDENTIFY /NOIDENTIFY	Control whether the PRINT command displays an informational message giving the job number and queue name in which the files are queued for printing.
/JOB_COUNT=number	Specify the number of copies to be made of the entire print job. See also /COPIES.
/LOWERCASE /NOLOWERCASE	Indicate files must be printed on a device that can print lower case characters.

Table A-1, PRINT Qualifiers (Continued)

/MODIFIED /NOMODIFIED	Use files' most recent modification times to select which are printed in processing a /BEFORE or /AFTER qualifier.
/NAME=job-name	Provide a name for the print job; it is displayed by the SHOW QUEUE command and appears on the job flag page. The name is from 1 to 39 characters long. If it contains characters other than alphanumerics, underscores, and dollar signs, enclose it in quotes (""). Default is the name of the first file in the job.
/NOTE=string	Print informational string on flag pages. If it contains characters other than alphanumerics, underscores, and dollar signs, enclose it in quotes ("").
/NOTIFY /NONOTIFY	Notify the user when the print job is completed or aborted.
/OPERATOR=string	Send the specified message to operators when the print job begins. The message string may be up to 255 characters long. If it contains characters other than alphanumerics, underscores, and dollar signs, enclose it in quotes ("").
/PAGES=[last-page (first-page,last-page)]	Print only selected pages. Has the same effect as a PAGE_LIMIT parameter specifying a range of pages (first-page,last-page).

Table A-1, PRINT Qualifiers (Continued)

/PARAMETERS=(parameter,...)	Specify additional job parameters. If a parameter contains characters other than alphanumerics, underscores, and dollar signs, enclose it in quotes (""). PrintKit interprets parameters according to DECprint syntax. See Table A-2, PRINT/PARAMETER Options, page A.11.
/PASSALL /NOPASSALL	Suppress formatting of the file being printed. Ignored by PrintKit.
/PRIORITY=number	Set the priority of the print job. The priority is a number in the range 0 to 255, with 0 the lowest priority. You must have the OPER or ALTPRI privilege to raise the priority above the value set by the SYSGEN parameter MAXQUEPRI. Default is set by the SYSGEN parameter DEFQUEPRI.
/QUEUE=name /DEVICE=name	Print the files on the specified queue. /QUEUE and /DEVICE are equivalent, but /QUEUE is preferred because /DEVICE is reserved for use by Digital. Default queue is SYS$PRINT.
/REMOTE	Print on the queue SYS$PRINT on the remote node indicated by the file specifications for the job.
/RESTART /NORESTART	Allow the job to be restarted after a queue failure or a STOP/QUEUE/REQUEUE command.

Table A-1, PRINT Qualifiers (Continued)

/SETUP=(name,...)	Specify setup modules to be sent to the printer at the beginning of the print job.
/SINCE=time	Select for printing only files dated since the specified time. Backup, creation (default), expiration, or modification time may be used, as indicated by /BACKUP, /CREATED, /EXPIRED, or /MODIFIED. The time may be absolute (dd-mmm-yyyy hh:mm:ss.cc) or a delta time (ddd hh:mm:ss.cc). For time format details, see the *OpenVMS User's Manual*.
/SPACE /NOSPACE	Control whether lines are double spaced or printed normally. Ignored by PrintKit.
/TRAILER=[ALL ONE] /NOTRAILER	Print a trailing flag page following the file.
/USER=username	Print the job on behalf of the specified user. You must have CMKRNL privilege and R (READ) access to the user authorization file to use this qualifier.

Table A-1, PRINT Qualifiers (Continued)

Parameter options

Parameter options are checked for correctness when your print job is printed, not when you issue the PRINT command. This is because they are meaningful only to the PrintKit software that does the printing, and are not interpreted by the PRINT command.

Parameter errors will cause your job to abort, but since this takes place after the PRINT command has completed, you will not receive any direct error indication. You can specify the PRINT /NOTIFY qualifier to receive notification of errors. Also, SHOW QUEUE and SHOW ENTRY will show the status of failed jobs if the queue has been configured to "retain on error."

Parameter options appear almost the same as ordinary DCL command qualifier options. But, because they are not interpreted directly by the PRINT command, there are some important differences. Observe the following rules:

> You can abbreviate option keywords, so long as the abbreviation is unambiguous.

```
/PARAMETERS="MESSAGES=(KEEP,PRINT)"
/PARAMETERS="ME=(K,P)"
```

> The parameter options must be collected into no more than 7 parameter strings, each containing up to 256 characters. To specify multiple options as a single parameter string, separate them with commas and enclose them in double quotations marks (""). To specify more than one parameter string, separate them with commas and enclose them in parentheses.

```
/PARAMETERS="DATA=ASCII,MESSAGES=(KEEP,PRINT)"
/PARAMETERS=(DATA=ASCII,"MESSAGES=(KEEP,PRINT)")
```

Using a single quoted parameter, as shown in the first example, is the recommended way of specifying parameter options. Using multiple parameter strings, as shown in the second example, is necessary only when options exceed 256 characters in total length.

> If the value specified for a parameter option contains any delimiters, such as commas or parentheses, enclose either the value or the entire parameter string in double quotation marks.

```
/PARAMETERS="MESSAGES=(KEEP,PRINT)"
/PARAMETERS=MESSAGES="(KEEP,PRINT)"
```

> If there are duplicate options, the last is used.

[1]
/NOTIFY qualifier
SHOW ENTRY command
SHOW QUEUE KITCP command
notification messages
PRINT command
retained job

The PrintKit parameter options are summarized in Table A-2, PRINT/PARAMETER Options. The description of each option explains its syntax and function, and indicates its default setting, where appropriate.

DATA_TYPE=	ANSI (or ANSI2) ASCII (or LINE, TEXT) AUTOMATIC [=default] EPSF HPGL LIST PCL (or PCL5, PCL5C) POSTSCRIPT (or PS) TRANSPARENT	Select data type. Translates data into printable form if necessary. Data types are as follows: ANSI ANSI-PPL3 (LN03) format. It is translated to PostScript for printing. PostScript must be available on the printer. AUTOMATIC [=default] (default) The beginning of the data file is examined to determine the data type. If the data type cannot be determined, the specified default data type is used. If no default is specified, ANSI-PPL3 or TEXT is assumed, depending on whether PostScript is available on the printer. ASCII ASCII format. It is translated to PostScript for printing, or, if PostScript is not available on the printer, to PCL. EPSF Encapsulated PostScript format. The file is assumed to describe a single page. The EPSF processing forces the page to print, which is useful for EPSF files that do not themselves contain the commands to print the page. HPGL Hewlett-Packard Graphics Language. LIST Simple listing format. It is translated to PostScript for printing. PostScript must be available on the printer. POSTSCRIPT PostScript language. PCL Hewlett-Packard Printer Control Language. TRANSPARENT The data file is transferred to the printer without interpretation. PostScript, PCL, and HPGL data types are sent to the printer without translation, so these data types are available only if supported by the printer.

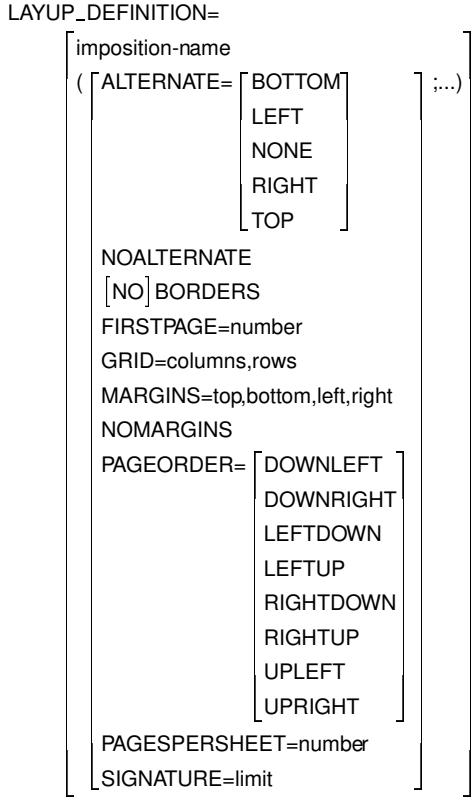
Table A-2, PRINT/PARAMETER Options

DEFAULT_MEDIUM=medium-name	Select default medium, used for pages for which no other medium is specified, either by PAGE_MEDIA_SELECT or by the data file itself. The medium-name refers to a medium specification defined in the PrintKit configuration database. The default medium also supplies the sheet size for the printed document. If SHEET_SIZE and DEFAULT_MEDIUM are both specified, the sheet size supplied by the default medium takes precedence.
FINISHING=finishing-name	Specify finishing process. The finishing-name refers to a finishing specification defined in the PrintKit configuration database.
FINISHING_INCLUDES_DOCUMENT NOFINISHING_INCLUDES_DOCUMENT	Include separator pages in document finishing operations.
FONTS_USED=(font-module,...)	Specify required ANSI-PPL3 soft fonts or PostScript fonts. The fonts must be resident on the printer, or they must be present in the device control libraries associated with the PrintKit execution queue, from which they will be downloaded before printing.
INITIAL_VALUE_DOCUMENT=document-name	Specify default document parameter settings, used for parameters which are not specified explicitly. The document-name refers to a document specification defined in the PrintKit configuration database.

Table A-2, PRINT/PARAMETER Options (Continued)

INPUT_TRAY=	BOTTOM CASSETTE ENVELOPE_FEEDER LCIT [NO] MANUAL_FEED MIDDLE MULTIPURPOSE TOP	Select an input tray. The listed input trays are standard settings. The LCIT and MIDDLE settings are treated the same as BOTTOM if their respective trays are not present; BOTTOM in turn is treated the same as TOP if its tray is not present. Additional trays may be present on certain printers. PrintKit determines a printer's input tray configuration from the printer model specified for the PrintKit execution queue – the information is provided by the PPD file designated by the printer's model definition in the PrintKit configuration database.

Table A-2, PRINT/PARAMETER Options (Continued)



Define page layup settings, either by reference to an imposition specification, or by specifying layup settings directly.

The imposition-name refers to a imposition specification defined in the PrintKit configuration database.

The layup settings are as follows:

ALTERNATE Swap the specified pair of margins on alternate sides of sheets in a duplexed document.

BORDERS Draw borders to indicate edges of page images.

FIRSTPAGE First grid position to be used on each page.

GRID Subdivide the page into rows and columns for multi-up printing. Within the layout grid, PAGEORDER specifies the order of positions, and FIRSTPAGE and PAGESPERSHEET specify which positions are occupied by page images.

MARGINS Sheet margins, specified in units of printer's points (1/72 inch). If less than four numbers are specified, the last is used for all remaining margins. In particular, specifying one number sets all margins. Default is 36 points for all margins.

PAGEORDER Order of positions within the page layout grid.

PAGESPERSHEET Number of grid positions occupied by page images.

SIGNATURE=limit Combine pages into signatures, limit specifies the maximum number of sheets in a signature.

Table A-2, PRINT/PARAMETER Options (Continued)

	Note the distinct syntax for LAYUP options: GRID and MARGINS values are not parenthesized; multiple options are separated by semicolons (;). Page layup is implemented using PostScript commands, so it applies only to PostScript, ANSI-PPL3 and listing jobs.
MEDIUM_SUBSTITUTION= ((medium-name | input-tray ,medium-name),...)	Replace media selections in file with alternate media. Replacements may be specified for selection by input tray or by medium specification. The medium-name refers to a medium specification defined in the PrintKit configuration database. The input-tray names corresponding to ANSI-PPL3 DECASFC tray positions are TRAY_1, TRAY_2, etc. Also for ANSI-PPL3 documents, a medium size may be specified as the medium-name to be substituted. Medium substitution is available only with Production PrintKit. The MEDIUM_SUBSTITUTION parameter currently has no effect for data types other than ANSI-PPL3.
MESSAGES=(KEEP | NOKEEP | NOTIFY | NONOTIFY | PRINT | NOPRINT ,...) NOMESSAGES	Select disposition of job-generated PostScript messages. KEEP saves messages to job log file. NOTIFY sends them to your terminal. PRINT displays them on a job trailer page. Default is NOMESSAGES.

Table A-2, PRINT/PARAMETER Options (Continued)

NUMBER_UP=number	Combine multiple page images onto the printed sheets. Multiple images up to 100 are allowed. Settings 1 or greater cause the page to be arranged with LAYUP_DEFINITION settings, which by default are BORDERS, MARGINS, and a GRID chosen to allow the specified number of images. Default is NUMBER_UP=0; one image, no scaling, margins or borders. Multi-up printing is implemented using PostScript commands, so it applies only to PostScript, ANSI-PPL3 and listing jobs.
OUTPUT_TRAY= [FACE_UP / LOWER / TOP (or UPPER)]	Select an output tray. Default is TOP.
PAGE_LIMIT=(page-selection)	Print selected pages of the document. Page selection expressions are described in Table A-3, PRINT Page Selection Expressions, page A.24. A page selection that specifies a single page position is treated as an upper limit, with an implicit lower limit of one; a pair of positions is treated as lower and upper limits. Page selection is implemented using PostScript commands, so it applies only to PostScript, ANSI-PPL3 and listing jobs.

Table A-2, PRINT/PARAMETER Options (Continued)

PAGE_MEDIA_SELECT= ((medium-name,page-selection),...)	Select medium for pages indicated by page-selection expressions. Each medium-name refers to a medium specification defined in the PrintKit configuration database. Page selection expressions are described in Table A-3, PRINT Page Selection Expressions, page A.24. Page media selection is available only with Production PrintKit. Page media selection is implemented using PostScript commands, so it applies only to PostScript, ANSI-PPL3 and listing jobs.
PAGE_ORIENTATION=[LANDSCAPE PORTRAIT]	Specify the orientation of the primary reading axis of pages. For the ANSI-PPL3 and PCL data types, this sets the appropriate page format; for PostScript, it indicates the primary axis of the already formatted page.

Table A-2, PRINT/PARAMETER Options (Continued)

PAGE_SIZE=paper-size	Select size of page images being printed. If the page and sheet sizes differ, or pages are printed multi-up, page images may be reduced in size to fit on the output sheet. otherwise, PAGE_SIZE must match SHEET_SIZE. Default is to match SHEET_SIZE. Scaling printing is implemented using PostScript commands, so it applies only to PostScript, ANSI-PPL3 and listing jobs. Standard paper-size settings are: ISO A sizes A0 841 mm by 1189 mm A1 594 mm by 841 mm A2 420 mm by 594 mm A3 297 mm by 420 mm A4 210 mm by 297 mm A5 148 mm by 210 mm A6 105 mm by 148 mm A7 74 mm by 105 mm A8 52 mm by 74 mm A9 37 mm by 52 mm A10 26 mm by 37 mm

Table A-2, PRINT/PARAMETER Options (Continued)

ISO B sizes	
B0	1000 mm by 1414 mm
B1	707 mm by 1000 mm
B2	500 mm by 707 mm
B3	353 mm by 500 mm
B4	250 mm by 353 mm
B5	176 mm by 250 mm
B6	125 mm by 176 mm
B7	88 mm by 125 mm
B8	62 mm by 88 mm
B9	44 mm by 62 mm
B10	31 mm by 44 mm
ISO envelope sizes	
C0_ENVELOPE	917 mm by 1297 mm
C1_ENVELOPE	648 mm by 917 mm
C2_ENVELOPE	458 mm by 648 mm
C3_ENVELOPE	324 mm by 458 mm
C4_ENVELOPE	229 mm by 324 mm
C5_ENVELOPE	162 mm by 229 mm
C6_ENVELOPE	114 mm by 162 mm
C7_ENVELOPE	81 mm by 114 mm
C8_ENVELOPE	57 mm by 81 mm
DL_ENVELOPE	110 mm by 220 mm

Table A-2, PRINT/PARAMETER Options (Continued)

JIS B sizes	
JIS_B0	1030 mm by 1456 mm
JIS_B1	728 mm by 1030 mm
JIS_B2	515 mm by 728 mm
JIS_B3	364 mm by 515 mm
JIS_B4	257 mm by 364 mm
JIS_B5	182 mm by 257 mm
JIS_B6	128 mm by 182 mm
JIS_B7	91 mm by 128 mm
JIS_B8	64 mm by 91 mm
JIS_B9	45 mm by 64 mm
JIS_B10	32 mm by 45 mm
North American sizes	
LETTER	8.5 in by 11 in
LEGAL	8.5 in by 14 in
EXECUTIVE	7.25 in by 10.5 in
FOLIO	8.5 in by 13 in
INVOICE	5.5 in by 8.5 in
LEDGER	11 in by 17 in
QUARTO	8.5 in by 10.83 in
North American envelope sizes	
10X13_ENVELOPE	10 in by 13 in
9X12_ENVELOPE	9 in by 12 in
COMM10_ENVELOPE	4.125 in by 9.5 in
7X9_ENVELOPE	7 in by 9 in
9X11_ENVELOPE	9 in by 11 in
10X14_ENVELOPE	10 in by 14 in
COMM9_ENVELOPE	3.875 in by 8.875 in
6X9_ENVELOPE	6 in by 9 in
10X15_ENVELOPE	10 in by 15 in
MONARCH_ENVELOPE	3.87 in by 7.5 in

Table A-2, PRINT/PARAMETER Options (Continued)

	Engineering drawing sizes A 8.5 in by 11 in B 11 in by 17 in C 17 in by 22 in D 22 in by 34 in E 34 in by 44 in Availability of paper sizes for printing without scaling depends on printer configuration. Additional sizes may also be available, depending on printer model.
ROUTE="string"	Routing information to display on burst, flag, and trailer pages.
SHEET_COUNT=number	Number of times to print each sheet. Sheet copies are not collated; that is, all copies of the first sheet print, then all copies of the second, and so forth.
SHEET_SIZE=paper-size	Select sheet size of printed document. See PAGE_SIZE for description of paper-size settings, and interaction between PAGE_SIZE and SHEET_SIZE. Default is to match PAGE_SIZE. The document sheet size may also be provided by the DEFAULT_MEDIUM parameter, which takes precedence over SHEET_SIZE if both are present.

Table A-2, PRINT/PARAMETER Options (Continued)

SIDES= [ONE_SIDED_SIMPLEX (or 1, ONE) ONE_SIDED_DUPLEX ONE_SIDED_TUMBLE TWO_SIDED_DUPLEX (or 2, TWO) TWO_SIDED_TUMBLE (or TUMBLE) TWO_SIDED_SIMPLEX]	Control arrangement of document onto sheet sides. Settings are: ONE_SIDED_SIMPLEX One side, long-edge binding. TWO_SIDED_DUPLEX Both sides, long-edge binding. This is normal binding for portrait orientation, "flip up" for landscape orientation. TWO_SIDED_TUMBLE Both sides, short-edge binding. This is normal binding for landscape orientation, "flip up" for portrait orientation. ONE_SIDED_DUPLEX One side, duplex format. This results in a duplex-formatted copy that can be taken to a copying facility to make two-sided copies. With PrintKit, this format is identical to ONE_SIDED_SIMPLEX. ONE_SIDED_TUMBLE One side, tumble duplex format. This results in a tumble-formatted copy that can be taken to a copying facility to make two-sided copies. With PrintKit, this format is identical to ONE_SIDED_SIMPLEX. TWO_SIDED_SIMPLEX Both sides, simplex format. With PrintKit, the result is identical to TWO_SIDED_DUPLEX.
TAB NOTAB	Control whether tab characters are processed according to ANSI-PPL3 interpretation or are expanded into space characters before ANSI-PPL3 processing occurs. Also controls whether form margins are processed by modifying the ANSI-PPL3 page format bounds or by inserting space and linefeed characters. TAB specifies ANSI-PPL3 interpretation, NOTAB specifies prior expansion. Applies only to ANSI-PPL3 files.

Table A-2, PRINT/PARAMETER Options (Continued)

Page selection expressions

The PAGE_LIMIT and PAGE_MEDIA_SELECT parameters share a common syntax for selecting pages from the document. Selection expressions designate pages in one of two ways: position or content.

Positional expressions select pages by their position relative to either the beginning or end of the document. They can select ranges of pages, or pages at intervals from the document. Positional expressions apply only to PostScript, ANSI-PPL3, and listing documents.

In order to select pages relative to the end of the document, PrintKit must be able to determine the total number of pages in the document. To allow this, PostScript documents must include a %%Pages Document Structuring Comment at the beginning of the file (if the comment is missing or appears at the end of the file, end-relative selection is disabled, and, if the comment is incorrect, the wrong pages will be selected). PrintKit counts pages for ANSI-PPL3 and listing documents, but, for efficiency's sake, it starts sending pages to the printer once it has collected 128 pages, so selection relative to the end of the document works only for the last 128 pages.

Content expressions select pages according to the printed matter on the page; expressions allow data fields to be checked for certain values, or compared with other fields on the same page or adjacent pages. Content expressions apply only to ANSI-PPL3 documents.

[1]
ANSI data type
LIST data type
PAGE_LIMIT parameter
PAGE_MEDIA_SELECT parameter
POSTSCRIPT data type
page selection expression

page-selection	A page selection expression selects pages from a file either based on their position in the file, or by testing data within the page itself.
page-position	Select single page specified by page-position expression.
first-page-position,last-page-position	Select pages starting with page specified by first-page expression, up to and including page specified by last-page expression.
first-page-position,next-page-position,...	Select pages at intervals from within page range from the first-page-position up to the last page. Pages selected are first-page-position, next-page-position, and succeeding pages at equal intervals.
first-page-position,next-page-position,...,last-page-position	Select pages at intervals from within page range specified by first-page-position and last-page-position expressions. Pages selected are first-page-position, next-page-position, and succeeding pages at equal intervals.
test-expression	Select pages specified by test-expression.
page-position	A page position selects a page at a particular position in the file, starting from either the first or last page in the file.
number	Page position relative to the first page in the file. First page is 1.
LAST-number	Page position relative to the last page in the file. Last page is LAST, next to last is LAST-1.

Table A-3, PRINT Page Selection Expressions

test-expression	A test expression selects pages that contain data matching the criteria specified by the expression.
(test-expression)	Parentheses may be used for grouping subexpressions.
test-expression AND test-expression	Logical AND of subexpressions; matches if both subexpressions match.
test-expression OR test-expression	Logical OR of subexpressions; matches if either subexpression matches.
NOT test-expression	Logical NOT of subexpression; matches if subexpressions do not match.
test-expression = test-expression	Comparison of subexpressions; matches if subexpressions are equal.
test-expression <> test-expression	Comparison of subexpressions; matches if subexpressions are not equal.
test-expression < test-expression	Comparison of subexpressions; matches if first subexpression is less than second.
test-expression <= test-expression	Comparison of subexpressions; matches if first subexpression is less than or equal to second.
test-expression >= test-expression	Comparison of subexpressions; matches if first subexpression is greater than or equal to second.
test-expression > test-expression	Comparison of subexpressions; matches if first subexpression is greater than second.
FIELD(line,first-column,last-column)	Page data field, has as its value the text from the specified line and column range on the current page. Any portion of the field that extends beyond the limits of the page text or is otherwise not represented by printable characters is filled with spaces.
PREV.FIELD(line,first-column,last-column)	Like FIELD, but selects text from the previous page.
NEXT.FIELD(line,first-column,last-column)	Like FIELD, but selects text from the next page.

Table A-3, PRINT Page Selection Expressions (Continued)

"string"	Literal string, has as its value the string within the quotes. The quote character itself (") may be represented by a pair of quotes ("").

Table A-3, PRINT Page Selection Expressions (Continued)

Appendix B
KITCP Command Reference

B.2 Introduction
B.2 Version numbers
B.3 Commands

Introduction

The KITCP administration utility configures and manages the PrintKit software. It lets you add and remove the specifications required by PrintKit, and display and modify the attributes of existing specifications. These specifications include print queues, printer models, and document specifications and their subsidiaries (medium, imposition, and finishing process).

Before you invoke KITCP, define the KITCP command as follows:

```
$ KITCP :== $KITCP
```

You can invoke individual KITCP commands directly:

```
$  KITCP command parameter ...
```

You can also invoke KITCP for multiple commands:

```
$ KITCP
KITCP> command parameter ...
KITCP> ...
```

In either case, the KITCP command uses normal OpenVMS conventions for commands and qualifiers.

Version numbers

Printer models, and document specifications and their subsidiaries may be assigned a version number to aid in documenting and tracking revisions to the specification.

A version number consists of one or more numeric values, separated by periods (.), and, optionally, a final numeric value, separated by a hyphen (-). The period-separated values comprise the primary version number, and the hyphen-separated value is the revision number.

```
number . ... [-number]
```

The primary version number remains unchanged unless set explicitly by a KITCP add, modify or copy command. If a specification has a version, its revision number is incremented automatically by KITCP whenever the specification is modified. KITCP will add a revision number to the version when the specification is modified, if one is not already present.

[1]
KITCP Administration Utility
document specification
KITCP Administration Utility
document specification
version, specification

Copying a specification does not copy its version. If the copy requires a version number, you must specify it explicitly.

Commands

Commands are summarized in Table B-1, KITCP Commands, page B.4. The description of each command explains its syntax and function.

ADD DOCUMENT document-name	Create a new document specification. The document-name may be up to 31 characters long. It consists of letters (upper and lower case are equivalent), digits, dollar sign ($), and underscore (_), and must include at least one letter. Qualifiers specified with the ADD DOCUMENT command correspond directly to PrintKit parameters. The parameter descriptions in Table A-2, PRINT/PARAMETER Options, p. A.11, provide more detailed descriptions than the summaries provided below.
/DATA_TYPE= [ANSI (or ANSI2) ASCII (or LINE, TEXT) EPSF HPGL LIST PCL (or PCL5, PCL5C) POSTSCRIPT (or PS) TRANSPARENT AUTOMATIC [=data-type]]	Select data type. If you specify automatic data type selection, PrintKit will examine the file being printed to determine its data type, using the specified data-type (which must match one of the other data type selectors) if the file's data type is indeterminate. Corresponds to DATA_TYPE parameter.
/DEFAULT_MEDIUM=medium-name	Select default medium, used for pages for which no other medium is specified, either by PAGE_MEDIA_SELECT or by the data file itself. The medium-name refers to a medium specification. Corresponds to DEFAULT_MEDIUM parameter; also provides functionality of SHEET_SIZE parameter.
/DESCRIPTION=text	Provide a text description of the document specification.
/FINISHING=finishing-name	Specify finishing process. The finishing-name refers to a finishing specification. Corresponds to FINISHING parameter.

Table B-1, KITCP Commands

[1]
ADD DOCUMENT KITCP command
ANSI data type
ASCII data type
AUTOMATIC data type
DATA_TYPE parameter
DEFAULT_MEDIUM parameter
FINISHING_INCLUDES_DOCUMENT_SETUP parameter
FINISHING parameter
FONTS_USED parameter
HPGL data type
INPUT_TRAY parameter
LAYUP_DEFINITION parameter
LIST data type
MEDIUM_SUBSTITUTION parameter
MESSAGES parameter
NUMBER_UP parameter
OUTPUT_TRAY parameter
PAGE_LIMIT parameter
PAGE_MEDIA_SELECT parameter
PAGE_ORIENTATION parameter
PAGE_SIZE parameter
PCL data type
POSTSCRIPT data type
SHEET_COUNT parameter
SHEE
SIDES
TAB pa
copies
data type
documen
impositio

/FIN_INCL_DOC_SHEET /NOFIN_INCL_DOC_SHEET	Include separator pages in document finishing operations. Corresponds to FINISHING_INCLUDES_DOCUMENT parameter.
/FONTS_USED=(font-name,...)	Specify required ANSI-PPL3 soft fonts or PostScript fonts. Corresponds to FONTS_USED parameter.
/FULL	Used with /INTERACTIVE to prompt for all attribute settings, not just the most common ones.
/INPUT_TRAY_SELECT=input-tray-name	Select an input tray. Corresponds to INPUT_TRAY parameter.
/INTERACTIVE	Prompt interactively for document attribute settings, instead of taking their values from command qualifiers. By default, prompts for most common attributes and supplies default settings for others; use /FULL to prompt for all.
/LAYUP_DEFINITION=imposition-name	Define page layup settings. The imposition-name refers to an imposition specification. Corresponds to LAYUP_DEFINITION parameter.
/MEDIUM_SUBSTITUTION= ((medium-name | input-tray-name ,medium-name),...)	Replace media selections in file with alternate media. Replacements may be specified for selection by input tray or by medium specification. The medium-name refers to a medium specification. Corresponds to MEDIUM_SUBSTITUTION parameter. Medium substitution is available only with Production PrintKit.

Table B-1, KITCP Commands (Continued)

/MESSAGES=(KEEP | NOKEEP | NOTIFY | NONOTIFY | PRINT | NOPRINT ,...) /NOMESSAGES	Select disposition of job-generated PostScript messages. KEEP Saves messages to a job log file. NOTIFY Sends messages to the terminal. TRAILER Displays messages on a job trailer page. Corresponds to MESSAGES parameter.
/NUMBER_UP=number	Combine multiple page images onto the printed sheets. Corresponds to NUMBER_UP parameter.
/OUTPUT_TRAY=tray-name	Select an output tray. Corresponds to OUTPUT_TRAY parameter.
/PG_LIMIT=(page-selection,...)	Print selected pages of the document. Corresponds to PAGE_LIMIT parameter. Page selection expressions are described Table A-3, PRINT Page Selection Expressions, page A.24.
/PG_MEDIA_SELECT= ((medium-name,page-selection,...),...)	Select medium for pages indicated by page-selection expressions. Each medium-name refers to a medium specification. Corresponds to PAGE_MEDIA_SELECT parameter. Page selection expressions are described in Table A-3, PRINT Page Selection Expressions, page A.24. Page media selection is available only with Production PrintKit.
/PG_ORIENTATION= LANDSCAPE | PORTRAIT	Specify the orientation of the primary reading axis of pages. Corresponds to PAGE_ORIENTATION parameter.

Table B-1, KITCP Commands (Continued)

/PG_SIZE=page-size	Select size of page images being printed. Corresponds to PAGE_SIZE parameter.
/REPLACE	Allow the document specification to replace an existing specification.
/SHEET_COUNT=number	Number of times to print each sheet. Sheet copies are not collated; that is, all copies of the first sheet print, then all copies of the second, and so forth. Corresponds to SHEET_COUNT parameter.
/SIDES= [ONE_SIDED_SIMPLEX (or 1, ONE) ONE_SIDED_DUPLEX ONE_SIDED_TUMBLE TWO_SIDED_DUPLEX (or 2, TWO) TWO_SIDED_TUMBLE (or TUMBLE) TWO_SIDED_SIMPLEX]	Control arrangement of document onto sheet sides. Corresponds to SIDES parameter.
/TAB /NOTAB	Control whether tab characters are processed according to ANSI-PPL3 interpretation or are expanded into space characters before ANSI-PPL3 processing occurs. Also controls whether form margins are processed by modifying the ANSI-PPL3 page format bounds or by inserting space and linefeed characters. /TAB specifies ANSI-PPL3 interpretation, /NOTAB specifies prior expansion. Applies only to ANSI-PPL3 files. Corresponds to TAB parameter.
/VERSION=version-number /NOVERSION	Assigns a version to the document specification. Version numbers are described in Version numbers, page B.2.

Table B-1, KITCP Commands (Continued)

[1]
ADD FINISHING KITCP command
binding orientation
execution queue
finishing process specification
PostScript Printer Definition (PPD) files

ADD FINISHING finishing-name	Create a new finishing process specification. The finishing-name may be up to 31 characters long. It consists of letters (upper and lower case are equivalent), digits, dollar sign ($), and underscore (_), and must include at least one letter.
/DESCRIPTION=text	Provide a text description of the finishing specification.
/FULL	Used with /INTERACTIVE to prompt for all attribute settings, not just the most common ones.
/INTERACTIVE	Prompt interactively for finishing attribute settings, instead of taking their values from command qualifiers. By default, prompts for most common attributes and supplies default settings for others; use /FULL to prompt for all.
/REPLACE	Allow the finishing specification to replace an existing specification.
/SPECIFICATION= ([BINDING | FOLDING | PUNCHING | STITCHING] =finishing-name,...)	References to the processes which are to be applied during finishing. BINDING specifies a binding finishing process; FOLDING specifies a folding finishing process; PUNCHING specifies a hole punching finishing process; STITCHING specifies a stitching finishing process (such as staples, brads, or sewn stitches). PrintKit determines a printer's finishing capabilities from the printer model specified for the PrintKit execution queue – the information is provided by the PPD file designated by the printer's model definition in the PrintKit configuration database.
/VERSION=version-number /NOVERSION	Assigns a version to the finishing specification. Version numbers are described in Version numbers, page B.2.

Table B-1, KITCP Commands (Continued)

[1]
ADD IMPOSITION KITCP command
ALTERNATE layup option
BORDERS parameter
FIRSTPAGE layup option
GRID layup option
LAYUP_DEFINITION parameter
MARGINS layup option
PAGEORDER layup option
PAGESPERSHEET layup option
imposition specification
page layout

ADD IMPOSITION imposition-name	Create a new imposition (number up) specification. The imposition-name may be up to 31 characters long. It consists of letters (upper and lower case are equivalent), digits, dollar sign ($), and underscore (_), and must include at least one letter. Qualifiers specified with the ADD IMPOSITION command correspond directly to PrintKit LAYUP parameter options. The LAYUP parameter description in Table A-2, PRINT/PARAMETER Options, p. A.11, provides more detailed descriptions than the summaries provided below.
/ALTERNATE [BOTTOM | LEFT | NONE | RIGHT | TOP] /NOALTERNATE	Swap the specified pair of margins on alternate sides of sheets in a duplexed document. Corresponds to ALTERNATE layup parameter.
/BORDERS [ALL | NONE] /NOBORDERS	Draw borders to indicate edges of page images. Corresponds to BORDERS layup parameter.
/DESCRIPTION=text	Provide a text description of the imposition specification.
/FIRSTPAGE=number	First grid position to be used on each page. Corresponds to FIRSTPAGE layup parameter.
/FULL	Used with /INTERACTIVE to prompt for all attribute settings, not just the most common ones.
/GRID=(columns,rows)	Subdivide the page into rows and columns for multi-up printing. Corresponds to GRID layup parameter.

Table B-1, KITCP Commands (Continued)

/INTERACTIVE	Prompt interactively for imposition attribute settings, instead of taking their values from command qualifiers. By default, prompts for most common attributes and supplies default settings for others; use /FULL to prompt for all.
/LAYUP_DEFINITION=filename	Specifies a layup definition file used as the source of settings for the imposition specification. The layup definition file has essentially the same syntax as a PrintKit LAYUP_DEFINITION parameter, except layup settings are placed on separate lines instead of being separated by semicolons. An exclamation point (!) introduces a comment: the exclamation point and any following text on the line are ignored. The LAYUP parameter description in Table A-2, PRINT/PARAMETER Options, p. A.11, provides more detailed descriptions of the layup settings and their syntax.
/MARGINS=(top,bottom,left,right)	Sheet margins, specified in units of printer's points (1/72 inch). If less than four numbers are specified, the last is used for all remaining margins. In particular, specifying one number sets all margins. Default is 36 points for all margins. Corresponds to MARGINS layup parameter.
/PG_ORDER= [DOWNLEFT | DOWNRIGHT | LEFTDOWN | LEFTUP | RIGHTDOWN | RIGHTUP | UPLEFT | UPRIGHT]	Order of positions within the page layout grid. Corresponds to PAGEORDER layup parameter.

Table B-1, KITCP Commands (Continued)

/PG_PER_SHEET=number	Number of grid positions occupied by page images. Corresponds to PAGESPERSHEET layup parameter.
/REPLACE	Allow the imposition specification to replace an existing specification.
/VERSION=version-number /NOVERSION	Assigns a version to the imposition specification. Version numbers are described in Version numbers, page B.2.
ADD MEDIUM medium-name	Create a new medium specification. The medium-name may be up to 31 characters long. It consists of letters (upper and lower case are equivalent), digits, dollar sign ($), and underscore (_), and must include at least one letter.
/COLOR=text	The medium color.
/DESCRIPTION=text	Provide a text description of the medium specification.
/FULL	Used with /INTERACTIVE to prompt for all attribute settings, not just the most common ones.
/INTERACTIVE	Prompt interactively for medium attribute settings, instead of taking their values from command qualifiers. By default, prompts for most common attributes and supplies default settings for others; use /FULL to prompt for all.
/OVERLAYS=(overlay-name,...)	Designate forms overlays to be printed on the medium. The first appears on the front side, and a second, if specified, appears on the back. The definitions for the overlays are provided by PostScript Form resources, stored in the device control libraries associated with the PrintKit execution queue. See Appendix E, Forms Overlays for information on creating overlay definitions.

Table B-1, KITCP Commands (Continued)

[1]
ADD MEDIUM KITCP command
POSTSCRIPT data type
device control library
execution queue
medium specification
overlay

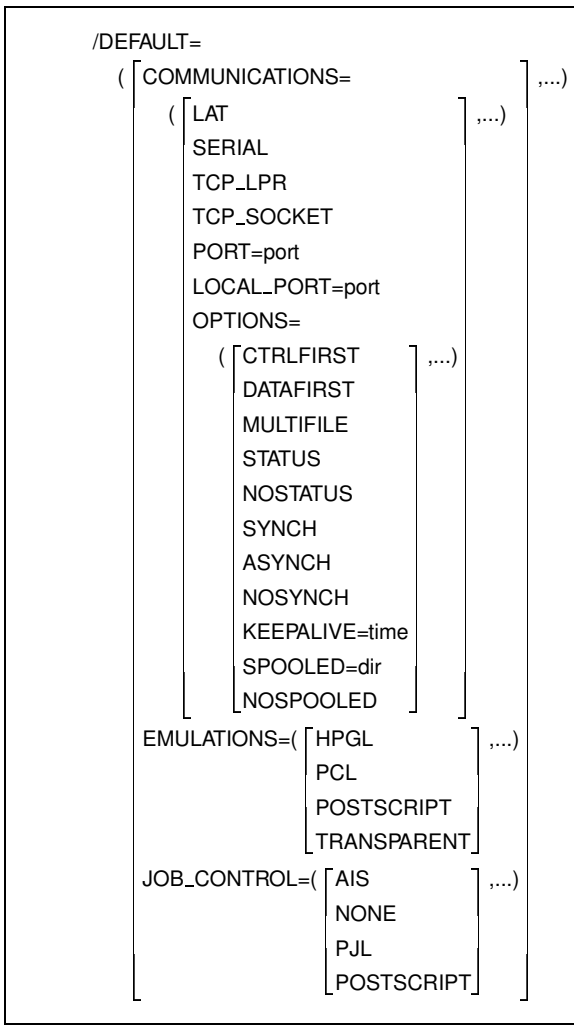
/REPLACE	Allow the medium specification to replace an existing specification.
/SIZE=name	The medium size.
/TYPE=name	The medium type.
/VERSION=version-number /NOVERSION	Assigns a version to the medium specification. Version numbers are described in Version numbers, page B.2.
/WEIGHT=weight	The medium weight. A weight specified as a simple number is treated as a BOND weight in units of pounds. Other units may be specified explicitly: number#-TAG number#-INDEX number#-BRISTOL number#-COVER number#-BOND number#-BOOK number#-GSM (grams/meter^2)

Table B-1, KITCP Commands (Continued)

ADD MODEL model-name	Make a new printer model known to PrintKit, supply PPD file. The model-name uniquely identifies the printer model. It is referred to by the queue /MODEL attribute. The model-name may be up to 63 characters long; if it contains lowercase letters, or non-alphanumeric characters (including blanks), enclose it in quotation marks ("").

Table B-1, KITCP Commands (Continued)

[1]
ADD MODEL KITCP command
HPGL data type
INPUT_TRAY parameter
OUTPUT_TRAY parameter
PCL data type
POSTSCRIPT data type
data type
device control library
Job Control Program
job status
LAT network protocol
PostScript Printer Definition (PPD) files
TCP/IP network protocol



Describe default characteristics of the printer model. These settings may be overridden by queue /COMMUNICATIONS and /PRINTER qualifier settings.

The EMULATIONS qualifier lists document data types directly supported by the printer model.

The JOB_CONTROL qualifier lists preferred job control languages for the printer model.

The COMMUNICATIONS qualifier describes the communications protocol characteristics of the printer model. The primary communications protocols, and the accompanying qualifiers used with each, are as follows:

LAT LAT protocol. Requires a queue /COMMUNICATIONS ADDRESS option that specifies a LAT node, and a PORT (here or as queue /COMMUNICATIONS option) that specifies LAT port or service name. The STATUS and SYNCH OPTIONS settings may also be specified.

SERIAL Direct serial connection. No required qualifiers. The STATUS and SYNCH OPTIONS settings may be specified.

Table B-1, KITCP Commands (Continued)

TCP_LPR TCP/IP lpr/lpd protocol. Requires a queue /COMMUNICATIONS ADDRESS option that specifies an Internet address, and a PORT (here or as queue /COMMUNICATIONS option) that specifies lpr/lpd queue name. The LOCAL_PORT qualifier and the CTRLFIRST, DATAFIRST, MULTIFILE, STATUS, SYNCH, and SPOOLED OPTIONS settings may also be specified.

If the PORT queue name contains lowercase letters or non-alphanumeric characters (including blanks), enclose it in quotation marks ("").

Port numbers for the LOCAL_PORT qualifier are specified as decimal values. A range of allowed port numbers may be specified by its first and last values, separated by a colon (:), enclosed in quotes (""). Default is to allow any non-privileged port as the client port.

TCP_SOCKET TCP/IP socket protocol. Requires a queue /COMMUNICATIONS ADDRESS option that specifies an Internet address. The PORT qualifier and the STATUS and SYNCH OPTIONS settings may also be specified.

The port number is a decimal value that specifies the port number of the socket. Default is 9100.

Table B-1, KITCP Commands (Continued)

Protocol OPTIONS settings are as follows:

CTRLFIRST, DATAFIRST, MULTIFILE Indicates organization of data and control information for a TCP_LPR job. CTRLFIRST places the job control information before the document data, DATAFIRST places the document data first, and MULTIFILE produces a single lpr/lpd job for all documents in an OpenVMS job (by transferring control information for all the documents as a single unit, after the data for each document has been transferred).

Default is CTRLFIRST if NOSPOOLED is set, and this is the only allowed setting. Otherwise, default is DATAFIRST.

STATUS Indicates whether PrintKit should process status information from the printer. NOSTATUS indicates printer or communications protocol restrictions inhibit status reporting. STATUS indicates status information should be processed. The STATUS setting is ignored if the communications protocol in use cannot provide status information. Default is STATUS.

SYNCH Indicates whether to synchronize job completion with completion of printing. SYNCH indicates jobs remain active, and status and errors are reported until they complete printing. ASYNCH causes jobs to complete as soon as all data is transferred (a less reliable, but sometimes faster, alternative). NOSYNCH indicates the protocol-specific default synchronization setting should be used. Default is NOSYNCH.

Table B-1, KITCP Commands (Continued)

	KEEPALIVE Indicates keepalive duration for TCP-based protocols, specified as a delta time value or the keyword FOREVER. SPOOLED Indicates whether TCP_LPR jobs are spooled before printing. The SPOOLED option should be used only for restrictive lpr/lpd interfaces that require accurate document data byte counts. Default is NOSPOOLED. The directory is the name of a directory in which temporary files are created to hold spooled jobs. It should be located on a device with enough free space to accomodate the largest print jobs that will be printed on the queue.
/FULL	Used with /INTERACTIVE to prompt for all attribute settings, not just the most common ones.
/INPUT_TRAY_SUBSTITUTION= ((tray-name,slot-name),...)	Translation from INPUT_TRAY parameter names to the input slot names used in the PostScript Printer Description file for the model. The following substitutions are provided by default: BOTTOM LOWER ENVELOPE_FEEDER ENVELOPE LCIT LARGECAPACITY TOP UPPER TOP TRAY1 TOP CASSETTE
/INTERACTIVE	Prompt interactively for model attribute settings, instead of taking their values from command qualifiers. By default, prompts for most common attributes and supplies default settings for others; use /FULL to prompt for all.

Table B-1, KITCP Commands (Continued)

/JOB_CONTROL= ([AIS | PJL | POSTSCRIPT] = [EMULATION | JOB | STATUS | SYNCH] ,...)	Job control languages provided by the printer model, and the capabilities of each. Job control capabilities as follows: EMULATION Can select emulation. JOB Can delimit print jobs. STATUS Can report job status. SYNCH Can synchronize with job progress.
/OUTPUT_TRAY_SUBSTITUTION= ((tray-name,bin-name),...)	Translation from OUTPUT_TRAY parameter names to the output bin names used in the PostScript Printer Description file for the model. The following substitutions are provided by default: FACE_UP REAR TOP UPPER TOP ONLYONE
/PPD=name	Name of the PostScript Printer Description file for the model. The file itself must be stored as a module in the PrintKit device control library with the name PPD$name.
/REPLACE	Allow the printer model to replace an existing model.
/VERSION=version-number /NOVERSION	Assigns a version to the printer model. Version numbers are described in Version numbers, page B.2.

Table B-1, KITCP Commands (Continued)

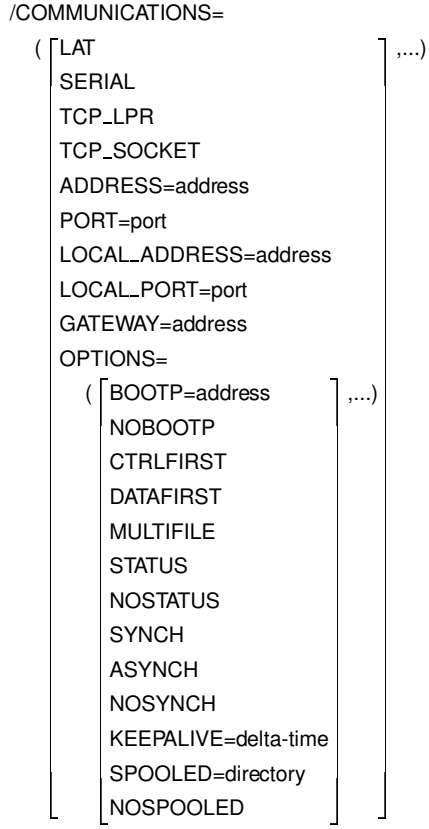
[1]
ADD QUEUE KITCP command
DATA_TYPE parameter
HPGL data type
PAGE_SIZE parameter
PCL data type
POSTSCRIPT data type
SHEET_SIZE parameter
autostart queue
data type
device control library
document specification
execution queue
flag page
generic queue
Job Control Program
LAT network protocol
logical queue
notification messages
sheet package specification
symbiont
TCP/IP network protocol
trailer page

ADD QUEUE queue-name	Create a new queue. The queue-name may be up to 31 characters long. It consists of letters (upper and lower case are equivalent), digits, dollar sign ($), and underscore (_), and must include at least one letter.
/AUTOSTART_ON=(node::device:,...)	Designate an autostart queue. The Job Control Program will start the queue automatically, on the first available node::device: (multiple nodes provide failover when nodes are removed from a cluster). See the *OpenVMS System Manager's Manual* for more information. See the /ON qualifier for a description of the node::device: value. The /AUTOSTART_ON and /ON qualifiers cannot be used together.
/BASE_PRIORITY=priority	Set the base priority of the PrintKit symbiont process for the queue. The priority is a number representing a valid process priority.
/BLOCK_LIMIT=([lower-limit,]upper-limit) /NOBLOCK_LIMIT	Limit the size of jobs that will be processed by the queue. For a job to print, its total size, in blocks, must lie within the specified limits; other jobs remain pending until the limits are changed to allow them to print. Default is /NOBLOCK_LIMIT; print regardless of size. If only an upper limit is specified, the lower limit remains zero.

Table B-1, KITCP Commands (Continued)

/CHARACTERISTICS=(characteristic,...) /NOCHARACTERISTICS	Specify characteristics supplied by the queue. For jobs requiring particular characteristics to print, a queue must supply all the characteristics; otherwise, the jobs remain pending. Characteristics are installation-specific. You can specify a characteristic as a number in the range 0 to 127, or using a name defined by the DEFINE/CHARACTERISTIC command. Default is /NOCHARACTERISTICS.
/CLOSE	Prevent jobs from being entered in the queue. No new jobs are accepted by a closed queue, but jobs already in the queue continue to be processed normally. Default is /OPEN; accept jobs.

Table B-1, KITCP Commands (Continued)



Describe the printer communications configuration. The /COMMUNICATIONS option settings are a superset of those specified with the printer model /DEFAULT COMMUNICATIONS options. Unspecified options take their values from the model referenced by the queue /PRINTER MODEL option.

Communications options in addition to the printer model are as follows:

ADDRESS Printer address. For the LAT protocol, a LAT node name; for TCP/IP-based protocols, an Internet address.

An Internet address may be specified as a symbolic node name or a numeric address – four decimal values, separated by periods (d.d.d.d). Use of node names requires standard OpenVMS TCP/IP Services or a third-party service with a compatible implementation of name translation.

GATEWAY Gateway Internet address for routing printer communications. Allowed with any of the TCP/IP protocols, but *only* when the TCP/IP support is provided with PrintKit's built-in PEP TCP/IP interface. Specified as a numeric address – four decimal values, separated by periods (d.d.d.d).

LOCAL_ADDRESS Local Internet address for printer communications. Required if TCP/IP support is provided with PrintKit's built-in PEP TCP/IP interface; ignored otherwise. Specified as a numeric address – four decimal values, separated by periods (d.d.d.d).

Table B-1, KITCP Commands (Continued)

	Communications OPTIONS settings in addition to the printer model are as follows: BOOTP Provide BOOTP services for the printer. Allowed with any of the TCP/IP protocols, but *only* when the TCP/IP support is provided with PrintKit's built-in PEP TCP/IP interface. The address is the Ethernet address of the printer, expressed as six two-digit hexadecimal values separated by hyphens (xx-xx-xx-xx-xx-xx).

Table B-1, KITCP Commands (Continued)

/DEFAULT=([BURST=keyword | NOBURST | DOCUMENT=document-name | FEED | NOFEED | FLAG=keyword | NOFLAG | FORM=form-type | SHEET_SIZE=paper-size | TRAILER=keyword | NOTRAILER] ,...)
/NODEFAULT

Supply defaults for PRINT command options.

BURST Control printing of burst pages. The keyword is one of: ALL (default), print a burst page before each document the job; ONE, print a burst page before only the first document in the job. Default is NOBURST.

DOCUMENT Select default document specification, which supplies default values for document printing attributes. PRINT parameters may replace the default document specification with another specification, or may supersede individual attribute settings.

FEED Control whether a new document page is started automatically when the output exceeds the output form line limit. Default is FEED.

FLAG Control printing of flag pages. Values for keyword are the same as BURST. Default is FLAG.

FORM Select default form for queue. See FORM_MOUNTED for a description of form-type. Valid for execution, not generic or logical, queues. Default is FORM=DEFAULT.

SHEET_SIZE Select default paper size for queue. Used when no SHEET_SIZE or PAGE_SIZE parameter is specified for a job. Default is LETTER.

TRAILER Control printing of trailer pages. Values for keyword are the same as BURST. Default is NOTRAILER.

Table B-1, KITCP Commands (Continued)

/DESCRIPTION="string" /NODESCRIPTION	Extended queue description for users and operators. The string consists of up to 255 characters; if it contains lowercase letters, or non-alphanumeric characters (including blanks), enclose it in quotation marks ("").
/ENABLE_GENERIC /NOENABLE_GENERIC	Specify whether queue can accept jobs from generic queues. Valid for execution, not generic or logical, queues.
/FORM_MOUNTED=form-type	Select mounted form. For a job to print, its form (either explicit, or taken from the default form for the queue) must specify a stock that matches the stock of the mounted form; otherwise, the job remains pending until the mounted form is changed to supply a matching stock. Forms are installation-specific. You can specify a form as a number or a name defined by the DEFINE/CHARACTERISTIC command. Valid for execution, not generic or logical, queues.
/FULL	Used with /INTERACTIVE to prompt for all attribute settings, not just the most common ones.
/GENERIC=(queue-name,...) /NOGENERIC	Define a generic queue, and specify the execution queues to which it transfers jobs for printing. A generic queue lacks most queue attributes – they are provided by its associated execution queues. However, a generic queue can supply a default document specification.
/INTERACTIVE	Prompt interactively for queue attribute settings, instead of taking their values from command qualifiers. By default, prompts for most common attributes and supplies default settings for others; use /FULL to prompt for all.

Table B-1, KITCP Commands (Continued)

/LIBRARY=(library-name/DATA_TYPE=type,...)	Specify device control libraries for queue. Libraries must be located in SYS$LIBRARY:, and must be text libraries with a file type of .TLB. Each specification consists of a library filename and a /DATA_TYPE qualifier that specifies the data type of the library contents. Data types are the same as those allowed for the PrintKit PRINT DATA_TYPE parameter.
/LOGICAL=queue-name	Define a logical queue, and specify the execution queue to which it transfers jobs for printing. Like a generic queue, a logical queue lacks most queue attributes, but may supply a default document specification.
/ON=[node::]device:	Specify the device used by the execution queue for communication with the printer, and, optionally, the node on which the queue is located (where the PrintKit symbiont for the queue executes). For PrintKit, the device depends on the communications protocol in use. Serial and lat protocols require the name of the port connected to the printer. Protocols based on TCP/IP require either an Ethernet or TCP/IP template device, or the keyword TCPIP, in which case PrintKit locates an appropriate template device. Default is /ON=queue-name; use the queue name as the name of the communications device, and locate the queue on the node from which it is started. The /ON and /AUTOSTART_ON qualifiers cannot be used together.
/OPEN	Allow jobs to be entered in the queue. Default is /OPEN; accept jobs.

Table B-1, KITCP Commands (Continued)

/OWNER_UIC=uic	Specify the owner User Identification Code (UIC) for the queue. Default is /OWNER_UIC=[1,4].
/PRINTER= ([EMULATIONS=([HPGL PCL POSTSCRIPT TRANSPARENT] ,...) ,...) JOB_CONTROL=([AIS NONE PJL POSTSCRIPT] ,...) MODEL=model-name OPTIONS=([DUPLEX] ,...)]	Describe the printer configuration. The MODEL qualifier assigns a printer model to the queue, using a model-name that specifies a defined printer model (the SHOW MODEL command lists models). If the model name contains lowercase letters or non-alphanumeric characters (including blanks), enclose it in quotation marks ("""). The EMULATIONS qualifier lists document data types directly supported by the printer. Syntax is the same as for the printer model /DEFAULT EMULATIONS option, and, if omitted, the value from the printer model assigned to the queue is used. The JOB_CONTROL qualifier lists preferred job control languages for the printer. Syntax is the same as for the printer model /DEFAULT JOB_CONTROL option, and, if omitted, the value from the printer model assigned to the queue is used. The OPTIONS qualifier selects printer options to use.
/PROCESSOR=file-name	Specify the symbiont for the execution queue. Symbionts must be located in SYS$SYSTEM:, and must have a file type of .EXE. Default is /PROCESSOR=PRINTKIT.

Table B-1, KITCP Commands (Continued)

/PROTECTION=(ownership[:access],...)	Specify the protection of the queue. The ownership is one of SYSTEM, OWNER, GROUP, WORLD; access is a combination of R (read), W (write), E (execute), and D (delete). (On older VAX systems, access is a combination of R (read), S (submit), M (manage), and D (delete).) Default is /PROTECTION=(S:E,O:D,G:R,W:W).
/PROTOCOL	Deprecated qualifier; see /COMMUNICATIONS.
/REPLACE	Allow the queue definition to replace an existing OpenVMS queue. In this case, KITCP uses the existing queue specification to provide defaults for the PrintKit queue definition.
/REQUEST_TO=(operator-class,...)	Select operator classes to receive status and error notification. Operator classes are CENTRAL, PRINTER, TAPES, DISKS, DEVICE, CARDS, NETWORK, CLUSTER, SECURITY, SOFTWARE, LICENSE, OPER1, OPER2, OPER3, OPER4, OPER5, OPER6, OPER7, OPER8, OPER9, OPER10, OPER11, and OPER12. Default is PRINTER.
/RETAIN=[ALL | ERROR] /NORETAIN	Select job retention policy. ALL (default) holds all jobs in the queue after execution; ERROR holds only jobs that fail. Default is /NORETAIN; remove all jobs on completion.
/SCHEDULE=[SIZE | NOSIZE]	Select job scheduling policy. SIZE causes short jobs to print before longer ones of the same priority; NOSIZE causes jobs of the same priority to print in the order they were queued. Default is /SCHEDULE=SIZE; print in order of size.

Table B-1, KITCP Commands (Continued)

/SEPARATE=(BURST | NOBURST | FLAG | NOFLAG | TRAILER | NOTRAILER | RESET=(module,...) | SHEET_PACKAGE= sheet-package-name ,...) /NOSEPARATE	Specify mandatory job separation options for an execution queue. These settings cannot be overridden by individual print jobs. BURST Control printing of burst page at beginning of job. FLAG Control printing of flag page at beginning of job. TRAILER Control printing of trailer page at end of job. RESET Specify device control modules to reset the printer. They are transferred to the printer after each job. SHEET_PACKAGE Select media used for separator pages. The sheet-package-name refers to a sheet package specification. Default is DEFAULT.
/START /NOSTART	Start the queue. Default is /NOSTART.

Table B-1, KITCP Commands (Continued)

[1]
ADD SHEET_PACKAGE KITCP command
flag page
medium specification
sheet package specification
trailer page

ADD SHEET_PACKAGE sheet-package-name	Create a new sheet package specification. The sheet-package-name may be up to 31 characters long. It consists of letters (upper and lower case are equivalent), digits, dollar sign ($), and underscore (_), and must include at least one letter.
/DESCRIPTION=text	Provide a text description of the sheet package specification.
/FULL	Used with /INTERACTIVE to prompt for all attribute settings, not just the most common ones.
/INTERACTIVE	Prompt interactively for sheet package attribute settings, instead of taking their values from command qualifiers. By default, prompts for most common attributes and supplies default settings for others; use /FULL to prompt for all.
/REPLACE	Allow the sheet package specification to replace an existing specification.

Table B-1, KITCP Commands (Continued)

/SHEETS= (BURST | DEFAULT | FILE_BURST | FILE_FLAG | FILE_TRAILER | FILE | FLAG | JOB_BURST | JOB_FLAG | JOB_TRAILER | JOB | LOG | TRAILER =(MEDIUM=medium-name),...)	Select medium for separator pages as indicated by sheet selection assignments. Each assignment specifies a medium to be used in printing a particular type of separator pages. The medium-name refers to a medium specification. The separator sheet-type is one of: FILE_BURST file burst page FILE_FLAG file flag page FILE_TRAILER file trailer page JOB_BURST job burst page JOB_FLAG job flag page JOB_TRAILER job trailer page LOG log page FILE file burst, flag and trailer pages JOB job burst, flag and trailer pages BURST file and job burst pages FLAG file and job flag pages TRAILER file and job trailer pages DEFAULT pages not selected by other assignments
/VERSION=version-number /NOVERSION	Assigns a version to the sheet package specification. Version numbers are described in Version numbers, page B.2.

Table B-1, KITCP Commands (Continued)

[1]
ADD DOCUMENT KITCP command
COPY DOCUMENT KITCP command
document specification

COPY DOCUMENT old-name new-name	Create a new document specification using an existing document to supply default attributes. All ADD DOCUMENT qualifiers may also be used with COPY DOCUMENT to override defaults from the existing document.
/DATABASE=file-name	The PrintKit configuration database containing the existing document specification. The definition is copied from this database to the current database. When a database is specified, you may omit the new document name if it is the same as the old one, and you may use wildcards in the old name to copy multiple specifications.
COPY FINISHING old-name new-name	Create a new finishing specification using an existing specification to supply default attributes. All ADD FINISHING qualifiers may also be used with COPY FINISHING to override defaults from the existing specification.
/DATABASE=file-name	The PrintKit configuration database containing the existing finishing specification. The definition is copied from this database to the current database. When a database is specified, you may omit the new finishing name if it is the same as the old one, and you may use wildcards in the old name to copy multiple specifications.

Table B-1, KITCP Commands (Continued)

[1]
ADD FINISHING KITCP command
COPY FINISHING KITCP command
ADD IMPOSITION KITCP command
COPY IMPOSITION KITCP command
imposition specification

COPY IMPOSITION old-name new-name	Create a new imposition specification using an existing specification to supply a default attributes. All ADD IMPOSITION qualifiers may also be used with COPY IMPOSITION to override defaults from the existing specification.
/DATABASE=file-name	The PrintKit configuration database containing the existing imposition specification. The definition is copied from this database to the current database. When a database is specified, you may omit the new imposition name if it is the same as the old one, and you may use wildcards in the old name to copy multiple specifications.
COPY MEDIUM old-name new-name	Create a new medium specification using an existing specification to supply default attributes. All ADD MEDIUM qualifiers may also be used with COPY MEDIUM to override defaults from the existing specification.
/DATABASE=file-name	The PrintKit configuration database containing the existing medium specification. The definition is copied from this database to the current database. When a database is specified, you may omit the new medium name if it is the same as the old one, and you may use wildcards in the old name to copy multiple specifications.

Table B-1, KITCP Commands (Continued)

[1]
ADD MEDIUM KITCP command
COPY MEDIUM KITCP command
medium specification
ADD MODEL KITCP command
COPY MODEL KITCP command

COPY MODEL old-name new-name	Create a new printer model using an existing model to supply default attributes. All ADD MODEL qualifiers may also be used with COPY MODEL to override defaults from the existing model.
/DATABASE=file-name	The PrintKit configuration database containing the existing printer model. The definition is copied from this database to the current database. When a database is specified, you may omit the new model name if it is the same as the old one, and you may use wildcards in the old name to copy multiple models.
COPY QUEUE old-name new-name	Create a new queue using an existing queue to supply default attributes. All ADD QUEUE qualifiers may also be used with COPY QUEUE to override defaults from the existing queue.
/DATABASE=file-name	The PrintKit configuration database containing the existing queue definition. The definition is copied from this database to the current database. When a database is specified, you may omit the new queue name if it is the same as the old one, and you may use wildcards in the old name to copy multiple queues.

Table B-1, KITCP Commands (Continued)

[1]
ADD QUEUE KITCP command
COPY QUEUE KITCP command
ADD SHEET_PACKAGE KITCP command
COPY SHEET_PACKAGE KITCP command
sheet package specification

COPY SHEET_PACKAGE old-name new-name	Create a new sheet package specification using an existing specification to supply default attributes. All ADD SHEET_PACKAGE qualifiers may also be used with COPY SHEET_PACKAGE to override defaults from the existing specification.
/DATABASE=file-name	The PrintKit configuration database containing the existing sheet package specification. The definition is copied from this database to the current database. When a database is specified, you may omit the new sheet package name if it is the same as the old one, and you may use wildcards in the old name to copy multiple specifications.
EXIT	Exit from KITCP.
MODIFY DOCUMENT document-name	Modify the attributes of an existing document specification. All ADD DOCUMENT qualifiers may also be used with MODIFY DOCUMENT.
MODIFY FINISHING finishing-name	Modify the attributes of an existing finishing specification. All ADD FINISHING qualifiers may also be used with MODIFY FINISHING.
MODIFY IMPOSITION imposition-name	Modify the attributes of an existing imposition specification. All ADD IMPOSITION qualifiers may also be used with MODIFY IMPOSITION.
MODIFY MEDIUM medium-name	Modify the attributes of an existing medium specification. All ADD MEDIUM qualifiers may also be used with MODIFY MEDIUM.
MODIFY MODEL model-name	Modify the attributes of an existing printer model. All ADD MODEL qualifiers may also be used with MODIFY MODEL.

Table B-1, KITCP Commands (Continued)

[1]
EXIT KITCP command
ADD DOCUMENT KITCP command
MODIFY DOCUMENT KITCP command
document specification
ADD FINISHING KITCP command
MODIFY FINISHING KITCP command
ADD IMPOSITION KITCP command
MODIFY IMPOSITION KITCP command
imposition specification
ADD MEDIUM KITCP command
MODIFY MEDIUM KITCP command
medium specification
ADD MODEL KITCP command
MODIFY MODEL KITCP command
ADD QUEUE KITCP command
MODIFY QUEUE KITCP command

MODIFY QUEUE queue-name	Modify the attributes of an existing queue. All ADD QUEUE qualifiers may also be used with MODIFY QUEUE.
MODIFY SHEET_PACKAGE sheet-package-name	Modify the attributes of an existing sheet package specification. All ADD SHEET_PACKAGE qualifiers may also be used with MODIFY SHEET_PACKAGE.
RELINK LIBRARY old-name new-name	Replace all references to a specified device control library with a new library. Each occurrence of the old library name in the /LIBRARY list of a PrintKit queue will be replaced with the new library name. If the old library name contains wildcards, all matching names are replaced.
RELINK MODEL old-name new-name	Replace all references to a specified printer model with a new model. Each occurrence of the old model name as the /PRINTER MODEL parameter of a PrintKit queue will be replaced with the new model name. If the old model name contains wildcards, all matching names are replaced.
REMOVE DOCUMENT document-name	Remove document specifications. If the document name contains wildcards, all matching document specifications are removed.
REMOVE FINISHING finishing-name	Remove finishing specifications. If the finishing name contains wildcards, all matching finishing specifications are removed.
REMOVE IMPOSITION imposition-name	Remove imposition specifications. If the imposition name contains wildcards, all matching imposition specifications are removed.

Table B-1, KITCP Commands (Continued)

[1]
ADD SHEET_PACKAGE KITCP command
MODIFY SHEET_PACKAGE KITCP command
sheet package specification
RELINK LIBRARY KITCP command
device control library
RELINK MODEL KITCP command
REMOVE DOCUMENT KITCP command
document specification
REMOVE FINISHING KITCP command
REMOVE IMPOSITION KITCP command
imposition specification
REMOVE MEDIUM KITCP command
medium specification

REMOVE MEDIUM medium-name	Remove medium specifications. If the medium name contains wildcards, all matching medium specifications are removed.
REMOVE MODEL model-name	Remove printer models. If the model name contains wildcards, all matching model specifications are removed.
REMOVE QUEUE queue-name	Remove queues. If the queue name contains wildcards, all matching PrintKit queues are removed.
REMOVE SHEET_PACKAGE sheet-package-name	Remove sheet package specifications. If the sheet package name contains wildcards, all matching sheet package specifications are removed.
SET DATABASE file-name	Set PrintKit configuration database file.
/CREATE_IF	If specified database file does not exist, create it.
SHOW DATABASE	Display name of current PrintKit configuration database file.
SHOW DOCUMENT [document-name]	Display document specifications. The document-name may contain wildcards; all document specifications are shown if no name is specified.
/APPEND	Used with /OUTPUT, appends the display to the end of the output file.
/DCL	Full display of attributes, formatted as KITCP ADD and MODIFY commands.
/FULL	Full display of attributes.
/OUTPUT=file-name	Specifies output file to receive the displayed specification.

Table B-1, KITCP Commands (Continued)

[1]
REMOVE MODEL KITCP command
KITCP Administration Utility
REMOVE QUEUE KITCP command
REMOVE SHEET_PACKAGE KITCP command
sheet package specification
SET DATABASE KITCP command
SHOW DATABASE KITCP command
SHOW DOCUMENT KITCP command
document specification
SHOW DOCUMENT KITCP command
SHOW FINISHING KITCP command
finishing process specification

SHOW FINISHING [finishing-name]	Display finishing process specifications. The finishing-name may contain wildcards; all finishing processes are shown if no name is specified. All SHOW DOCUMENT qualifiers may also be used with SHOW FINISHING.
SHOW IMPOSITION [imposition-name]	Display imposition specifications. The imposition-name may contain wildcards; all imposition specifications are shown if no name is specified. All SHOW DOCUMENT qualifiers may also be used with SHOW IMPOSITION.
SHOW MEDIUM [medium-name]	Display medium specifications. The medium-name may contain wildcards; all medium specifications are shown if no name is specified. All SHOW DOCUMENT qualifiers may also be used with SHOW MEDIUM.
SHOW MODEL [model-name]	Display printer models. The model-name may contain wildcards; all models are shown if no name is specified. All SHOW DOCUMENT qualifiers may also be used with SHOW MODEL.
SHOW QUEUE [queue-name]	Display printer queues. The queue-name may contain wildcards; all queues are shown if no name is specified. Default format is a brief display of principal queue attributes and status. All SHOW DOCUMENT qualifiers may also be used with SHOW QUEUE.

Table B-1, KITCP Commands (Continued)

[1]
SHOW DOCUMENT KITCP command
SHOW IMPOSITION KITCP command
imposition specification
SHOW DOCUMENT KITCP command
SHOW MEDIUM KITCP command
medium specification
KITCP Administration Utility
SHOW DOCUMENT KITCP command
SHOW MODEL KITCP command
SHOW DOCUMENT KITCP command
SHOW QUEUE KITCP command
SHOW DOCUMENT KITCP command
SHOW SHEET_PACKAGE KITCP command
sheet package specification

SHOW SHEET_PACKAGE [sheet-package-name]	Display sheet package specifications. The sheet-package-name may contain wildcards; all sheet package specifications are shown if no name is specified. All SHOW DOCUMENT qualifiers may also be used with SHOW SHEET_PACKAGE.
SHOW VERSION	Display the version of KITCP in use.
START QUEUE [queue-name]	Start printer queues.
/ALL	Starts all queues in the PrintKit configuration database.
STOP QUEUE [queue-name]	Stop printer queues.
/ALL	Stops all queues in the PrintKit configuration database.

Table B-1, KITCP Commands (Continued)

[1]
SHOW VERSION KITCP command
START QUEUE KITCP command
STOP QUEUE KITCP command

Appendix C
ANSI-PPL3 Initial State and Fonts

C.2 Initial state settings
C.6 Font sets
Standard font sets

[1]
ANSI data type

Initial state settings

The ANSI-PPL3 translator begins each document with all state values in a predetermined configuration. The forms characteristics and the default paper size and orientation for the job affect the settings of several of the characteristics.

[1]
ANSI data type
configuration
landscape orientation
portrait orientation

State Variable	Value
position	origin (1,1)
vertical spacing	font-dependent
horizontal spacing	font-dependent
horizontal tab stops	every eight columns
vertical tab stops	every line
size unit	decipoints
graphics left character set (GL)	G0
graphics right character set (GR)	G2
graphics 0 character set (G0)	ASCII
graphics 1 character set (G1)	ASCII
graphics 2 character set (G2)	user preference
graphics 3 character set (G3)	user preference
user preference character set	DEC Supplemental
autowrap mode	matches /WRAP setting of VMS form used for job.
control representation mode	unimplemented
carriage return / new line mode	reset (no implicit line feed)
justify mode	unimplemented
line feed / new line mode	reset (no implicit line feed)
origin placement mode	reset (origin in printable area)
pitch select mode	reset (use DECSHORP pitch)
position unit mode	reset (column, line units)
proportional space mode	reset (monospaced printing)
C1 receive mode	eight-bit
graphic size (GSS)	10 pts
graphic size modification (GSM)	100, 100

Table C-1, Initial State Settings

SGR attributes	none
SGR fonts	
10 DBULTN1	DEC Builtin1 type family
11 RCOURIR	Courier type family
12 RELITE0	Elite 12 type family
13 RCOURIRJ02SK00GG	Courier 12 pt, 10 pitch coll. plus
14 RELITE0L02SK00GG	Elite 12 10 pt, 12 pitch coll. plus
15 RCOURIR101VK00GG	Courier 6.7 pt, 13.6 pitch coll. plus
16 RCOURIR202SK00GG	Courier 10 pt, 10.3 pitch coll. plus
17 DBULTN1	DEC Builtin1 type family
18 DBULTN1	DEC Builtin1 type family
19 DBULTN1	DEC Builtin1 type family

Table C-1, Initial State Settings (Continued)

Page Size	Orientation	Format[2]	SGR Font
A3	portrait	DECVPFS[1]	16
	landscape	DECVPFS[1]	15
A4	portrait	PFS ?22	16
	landscape	PFS ?23	15
A5	portrait	DECVPFS[1]	16
	landscape	DECVPFS[1]	15
B4	portrait	DECVPFS[1]	16
	landscape	DECVPFS[1]	15
B5	portrait	DECVPFS[1]	16
	landscape	DECVPFS[1]	15
Executive	portrait	DECVPFS[1]	10
	landscape	DECVPFS[1]	15
Legal	portrait	PFS ?24	10
	landscape	PFS ?25	15
Ledger (B)	portrait	PFS ?26	16
	landscape	PFS ?27	15
Letter (A)	portrait	PFS ?20	10
	landscape	PFS ?21	15

[1] The DECVPFS parameters are calculated from the initial page size and orientation as follows: left, right, top and bottom margins are all positioned 18 points in from the edge of the page; line home and end, and page home and end are all positioned 36 points in from the edge of the page.

[2] Initial settings may be superseded by forms qualifiers as follows: form width, if non-zero, replaces the right margin setting; form length, if non-zero, replaces the bottom margin setting; finally, form margins are subtracted from the initial margin settings. All form dimensions are interpreted in units of columns or lines, as appropriate, and are converted to physical dimensions using the initial horizontal and vertical spacing (which in turn are based on SGR font settings and page size).

Table C-2, Initial Page State Settings

Font sets

ANSI-PPL3 processing provides a standard set of 80 font files. Definitions for additional font sets are provided as setup modules.

Standard font sets

The standard set of font files provides six character sets (ASCII, DEC Supplemental, ISO Latin-1 Supplemental, ISO Latin-9 Supplemental, DEC Technical, and DEC Special Graphics), in four style variants (regular, bold, slanted, and bold slanted) for each of the following four font collections:

RCOURIR J 02S K	Courier, 10 pt, 10 pitch
RCOURIR 2 02S K	Courier, 10 pt, 10.3 pitch
RELITE0 L 02S K	Elite 12, 10 pt, 12 pitch
RCOURIR 1 01V K	Courier, 6.7 pt, 13.6 pitch

The font files may also be accessed as members of two default type families:

DBULTN1	DEC Built-In Type Family
D000000	PI Type Family

In this guise, the ASCII, DEC Supplemental, ISO Latin-1 Supplemental, and ISO Latin-9 Supplemental character sets belong to the Built-In type family, and the DEC Technical and DEC Special Graphics character sets belong to the PI type family.

[1]
ANSI data type
font set
setup modules
ANSI data type
font set

Pitch	Size	Character Set	Font File ID
DEC Builtin1 Type Family			
10	10	ASCII	DBULTN1J02SK00GG0001UZZZZ02F000
		DEC Supplemental	DBULTN1J02SK00GG00245ZZZZ02F000
		ISO Latin-1 Supplemental	DBULTN1J02SK00GG006DDZZZZ02F000
		ISO Latin-9 Supplemental	DBULTN1J02SK00GG006EAZZZZ02F000
10.3	10	ASCII	DBULTN1202SK00GG0001UZZZZ02F000
		DEC Supplemental	DBULTN1202SK00GG00245ZZZZ02F000
		ISO Latin-1 Supplemental	DBULTN1202SK00GG006DDZZZZ02F000
		ISO Latin-9 Supplemental	DBULTN1202SK00GG006EAZZZZ02F000
12	10	ASCII	DBULTN1L02SK00GG0001UZZZZ02F000
		DEC Supplemental	DBULTN1L02SK00GG00245ZZZZ02F000
		ISO Latin-1 Supplemental	DBULTN1L02SK00GG006DDZZZZ02F000
		ISO Latin-9 Supplemental	DBULTN1L02SK00GG006EAZZZZ02F000
13.6	6.7	ASCII	DBULTN1J02SK00GG0001UZZZZ02F000
		DEC Supplemental	DBULTN1J02SK00GG00245ZZZZ02F000
		ISO Latin-1 Supplemental	DBULTN1J02SK00GG006DDZZZZ02F000
		ISO Latin-9 Supplemental	DBULTN1J02SK00GG006EAZZZZ02F000

Table C-3, Built-In Fonts

Courier Type Family			
10	10	ASCII	RCOURIR J02SK00GG0001UZZZZ02F000
		DEC Supplemental	RCOURIR J02SK00GG00245 ZZZZ02F000
		DEC Technical	RCOURIR J02SK00GG0001QZZZZ02F000
		DEC Special Graphics	RCOURIR J02SK00GG0001C ZZZZ02F000
		ISO Latin-1 Supplemental	RCOURIR J02SK00GG006DDZZZZ02F000
		ISO Latin-9 Supplemental	RCOURIR J02SK00GG006EA ZZZZ02F000
10.3	10	ASCII	RCOURIR 202SK00GG0001UZZZZ02F000
		DEC Supplemental	RCOURIR 202SK00GG00245 ZZZZ02F000
		DEC Technical	RCOURIR 202SK00GG0001QZZZZ02F000
		DEC Special Graphics	RCOURIR 202SK00GG0001C ZZZZ02F000
		ISO Latin-1 Supplemental	RCOURIR 202SK00GG006DDZZZZ02F000
		ISO Latin-9 Supplemental	RCOURIR 202SK00GG006EA ZZZZ02F000
13.6	6.7	ASCII	RCOURIR 101VK00GG0001UZZZZ02F000
		DEC Supplemental	RCOURIR 101VK00GG00245 ZZZZ02F000
		DEC Technical	RCOURIR 101VK00GG0001QZZZZ02F000
		DEC Special Graphics	RCOURIR 101VK00GG0001C ZZZZ02F000
		ISO Latin-1 Supplemental	RCOURIR 101VK00GG006DDZZZZ02F000
		ISO Latin-9 Supplemental	RCOURIR 101VK00GG006EA ZZZZ02F000
Elite 12 Type Family			
12	10	ASCII	RELITE0 L02SK00GG0001UZZZZ02F000
		DEC Supplemental	RELITE0 L02SK00GG00245 ZZZZ02F000
		DEC Technical	RELITE0 L02SK00GG0001QZZZZ02F000
		DEC Special Graphics	RELITE0 L02SK00GG0001C ZZZZ02F000
		ISO Latin-1 Supplemental	RELITE0 L02SK00GG006DDZZZZ02F000
		ISO Latin-9 Supplemental	RELITE0 L02SK00GG006EA ZZZZ02F000

Table C-3, Built-In Fonts (Continued)

PI Font Type Family			
10	10	DEC Technical	D000000J02SK00GG0001QZZZZ02F000
		DEC Special Graphics	D000000J02SK00GG0001CZZZZ02F000
10.3	10	DEC Technical	D000000202SK00GG0001QZZZZ02F000
		DEC Special Graphics	D000000202SK00GG0001CZZZZ02F000
13.6	6.7	DEC Technical	D000000101VK00GG0001QZZZZ02F000
		DEC Special Graphics	D000000101VK00GG0001CZZZZ02F000
12	10	DEC Technical	D000000L02SK00GG0001QZZZZ02F000
		DEC Special Graphics	D000000L02SK00GG0001CZZZZ02F000

Table C-3, Built-In Fonts (Continued)

Appendix D
ANSI-PPL3 Page Format Controls

D.2 Introduction
- Coding control sequences

D.3 Page coordinate system
- Setting the page orientation
- Changing the margins
- Changing column and line spacing
- Changing the origin

D.7 Selecting fonts
- Selecting an SGR parameter
- Selecting font size
- Assigning fonts to an SGR parameter

[1]
ANSI data type

Introduction

The ANSI-PPL3 Printing Protocol is an encoding for printable documents. It uses a stream of coded characters to represent text, graphics, and other control information. An ordinary text file is a simple ANSI-PPL3 document – it consists of printable characters from the ASCII character set, along with control characters such as carriage returns, line feeds, and form feeds to organize the printed text into lines and pages. More complex ANSI-PPL3 documents may represent text using other character sets, and may use other control characters and control sequences.

The following sections present a simplified view of a few of the ANSI-PPL3 control sequences – just enough to accomplish the more common formatting adjustments. These include control sequences to specify how text is placed on the page (orientation, margins, and column and line spacing), and control sequences to select the font used for your document text.

You can use these control sequences to create setup modules that specify the document formats you commonly use. Once the modules are defined and installed in the device control library for a PrintKit queue, you can refer to them when you print. Since the document format is established by the setup module, it need not be specified in the document file itself, which might be a simple text file. The PRINT command /SETUP qualifier lets you refer to setup modules directly; they can also be specified as part of an OpenVMS print form definition and referenced indirectly with the /FORM qualifier. Details of the procedure for defining setup modules are defined in Managing device control libraries, page 4.20.

The descriptions that follow intentionally omit some of the more esoteric features of the ANSI-PPL3 Printing Protocol to allow a clearer and more compact presentation. For the complete definition, see the *Digital ANSI-Compliant Printing Protocol Level 3 Programming Reference Manual*.

Coding control sequences

Control sequences consist of a special control character, followed by one or more printable characters that specify the control operation. The following notations are used to indicate special characters in the control sequence examples that follow:

ESC Escape character (27_{10}).

CSI Control Sequence Introducer character (155_{10}).

DCS Device Control Sequence (144_{10}).

ST String Terminator (156_{10}).

[1]
/FORM qualifier
/SETUP qualifier
ANSI data type
device control library
setup modules

␣ Space character. Represented as a visible character to make its presence more obvious.

⟨text⟩ Bracked text indicates a location in the control sequence where you supply a parameter value.

Since ESC, CSI, DCS, and ST are not printable characters, you will have to enter them using their numeric codes. Check the documentation for your editor to find out how to enter special characters; EDT, for instance, has a SpecIns command (GOLD-KP3) that lets you insert a character using its decimal code.

Page coordinate system

A page coordinate system provides a way of specifying positions on the page being printed. In the ANSI-PPL3 emulator provided by PrintKit, this "printed page" may correspond directly to the printed sheet. However, PrintKit also provides layup controls that allow the ANSI-PPL3 page to be scaled and repositioned on the printed sheet. In the discussion that follows, *page image* refers to the unaltered ANSI-PPL3 page, and the coordinate system specifies positions within the page image.

In the ANSI-PPL3 coding, positions on the page are specified according to their horizontal and vertical coordinates. The coordinate system has the following characteristics:

> Coordinates may be set so that the horizontal direction is parallel to the short edge of the page (portrait orientation), or parallel to the long edge (landscape orientation).

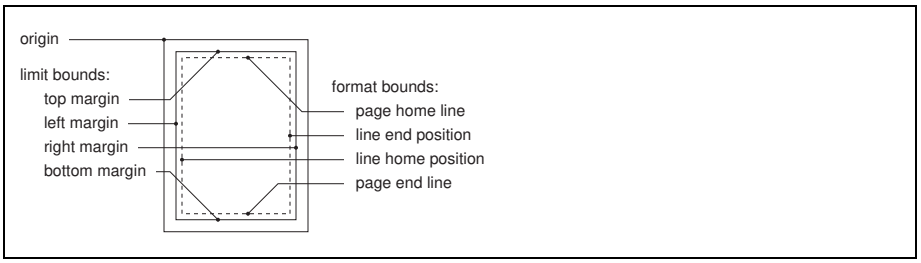


Figure D-1, Page Coordinate Bounds

[1]
ANSI data type
landscape orientation
page layout
portrait orientation

> The upper-leftmost position is the *origin* of the coordinate system. It has coordinates (1,1) – *not* (0,0); be careful! The origin position is normally inset 1/4 inch from the top left corner of the page, but it can also be set to match the corner of the page.

> Horizontal coordinates increase to the right, vertical coordinates increase downward.

> Coordinates are specified in terms of either a *character cell model*, or a *point model*. In the character cell model, a position is a cell on the page that can be occupied by a character, and positions are designated in terms of rows and columns. In the point model, a position is a location on the page, and positions are designated by horizontal and vertical coordinates, in dimensional units of decipoints (1/720 inch), centipoints (1/7200 inch), or pixels (1/300 inch).

Within the page, there are two sets of bounds which constrain various operations on the page:

format bounds The vertical format bounds are the page home and end lines, and the horizontal format bounds are the line home and end positions. The format bounds designate the normal boundary of text on the page. Lines begin at the line home bound; the first line of text is placed at the page home bound, and a new page starts when lines exceed the page end bound.

limit bounds The vertical limit bounds are the top and bottom margin positions, and the horizontal limit bounds are the left and right margin positions. The limit bounds constrain all positioning commands. It is possible to use positioning commands to position text outside the format bounds, but not outside the limit bounds.

The format bounds are always within the limit bounds.

The page bounds are summarized in Figure D-1, Page Coordinate Bounds, page D.3.

The following discussion focuses on the format bounds. By default, the format and limit bounds are the same, and, to keep matters simple, the examples that modify the format bounds also modify the limit bounds to match.

Setting the page orientation

The PFS (Page Format Select) control sequence sets orientation, size, and bounds for standard paper sizes.

Here are PFS commands to establish the default coordinate systems in portrait and landscape orientations for Letter and A4 page sizes:

CSI ?20␣J Letter portrait

[1]
landscape orientation
portrait orientation

CSI ?21␣J Letter landscape
CSI ?22␣J A4 portrait
CSI ?23␣J A4 landscape

These commands do not change the text size. The initial state settings for portrait and landscape orientations set the text size for landscape smaller than portrait, but when you change orientation with PFS, the size remains unchanged.

Changing the margins

The DECSLRM (Set Left and Right Margins) control sequence sets the left and right margin positions, and it sets the line home and end positions to match. Likewise, the DECSTBM (Set Top and Bottom Margins) control sequence sets the top and bottom margin positions, and page home and end positions.

Here are the control sequences:

CSI ⟨left⟩ ; ⟨right⟩ s
CSI ⟨top⟩ ; ⟨bottom⟩ r

By default, the margins are in units of lines and columns. Specify a value of zero to leave a margin unchanged.

To adjust margins by a fraction of a line or column, you need to first switch coordinates to the point model and select a unit of measure using the PUM (Positioning Unit Mode) and SSU (Select Size Unit) control sequences.

CSI 11h CSI 2␣I decipoints (1/720 inch)
CSI 11h CSI 7␣I pixels (1/300 inch)
CSI 11h CSI ?1␣I centipoints (1/7200 inch)
CSI 11l line and columns (reset to cell model)

(The first three control sequences end with an uppercase I, and the final one ends with a lowercase L.)

The DECSLRM control sequence does not change tab stops, so, unless you move the line home position by a multiple of eight columns, they will not line up properly with the new margins. If your documents use TAB characters for formatting, you must use the DECSHTS control sequence to reset the tab stops:

CSI ⟨position⟩ ;...; ⟨position⟩ u

(Specify as many tab stops as necessary, separating them with semicolons.) Like the DECSLRM and DECSTBM control sequences, DECSHTS positions are in units of columns unless you select a different unit of measure.

Here is an example that sets the left margin to 1/4 inch and resets the tab stops to match the new margin. Since the origin is already inset 1/4 inch, this places the margin 1/2 inch from the edge of the page. The tab stops are reset assuming a character spacing of 10 per inch.

CSI 11h CSI 2␣I	select decipoints
CSI 181;0s	set left margin (180/720 inch)
CSI 757;1333;1909;2485;3061;3637;4213;4789;5365u	tabs at columns 9, 17, ..., 73
CSI 11l	reset to cell model

Changing column and line spacing

If you don't specify column and line spacing explicitly, they are taken from the normal spacing values for the current font. So one way to change column and line spacing is to select a font with different dimensions; that method is described below in Selecting fonts.

To change the spacing without changing fonts, you use the SPI (Spacing Pitch Increment) command to override the default spacing provided by the font:

CSI ⟨vertical⟩ ; ⟨horizontal⟩ ␣G

The spacing values are in units of decipoints. Specify either value as zero to use the corresponding spacing value from the current font.

Changing the origin

The initial coordinate system origin is inset by 1/4 inch from the top left corner of the page. You can move it to the corner using the DECOPM (Origin Placement Mode) control sequence. Doing this can often make other coordinate system settings more intelligible, since they relate directly to the page, rather than being offset by 1/4 inch. However, the ability to specify coordinates out to the edge of the page doesn't mean you can print there – many printers have an unprintable border around the edge of the page.

Here are the control sequences to reposition the origin:

CSI ?52h	origin at corner
CSI ?52l	origin inset 1/4 inch from corner

(The last control sequence ends with a lowercase L.)

Selecting fonts

A *typeface* is the full range of type (the letters, numerals, and symbols that are the visual representations of characters) of a single design, such as Courier, Helvetica, or Times. A *font* is a complete set of type of a given typeface for a particular set of visual characteristics. The characteristics that determine a font include spacing (proportional, monospaced), size, type style (normal or italic), weight (normal or bold), and proportion (normal, expanded, condensed). A *type family* is the set of fonts for a given typeface – the fonts in a type family share a common design, but each has a different set of visual characteristics.

PrintKit's ANSI-PPL3 emulation provides an initial set of fonts for use in a document, and additional fonts may be defined by the document itself. Font selection relies on a table of *SGR parameters*. There are ten entries in the table, numbered 10 through 19 – *not* 0 through 9; watch out! Fonts, or sets of fonts, are assigned to entries in the table, then are selected for printing by specifying their table entry. The set of fonts assigned to a SGR parameter is one of:

Type family All fonts belonging to a particular typeface.

Font collection All fonts belonging to a particular typeface and having a particular combination of spacing and size, but differing in type style, weight, and proportion.

Font Collection Plus Specifies the same set of fonts as a font collection, but also specifies a default type style, weight, and proportion.

Font A single font belonging to a particular typeface and having a particular combination of spacing, size, type style, weight, and proportion.

Selecting an SGR parameter

If the font or font set you want to use has already been assigned to an SGR parameter, you can select it for printing by selecting that SGR parameter.

The initial set of SGR parameter assignments is described in Table C-1, Initial State Settings, page C.3. For example, SGR parameter 10 is the default for letter portrait pages (at 10-pitch, it gives 80 columns with standard margins). SGR 16 selects the default for A4 portrait pages (its 10.3-pitch fits 80 columns into the slightly narrower page). SGR 15 is default for both letter and A4 landscape (13.6-pitch fits 132 columns).

The SGR selection control sequence is:

CSI ⟨index⟩ m

The index is an SGR parameter index in the range 10 to 19.

[1]
ANSI data type
font set
landscape orientation
portrait orientation

Here, for example, is the control sequence to select SGR parameter 10:

CSI 10m

Although this is typically the default setting for landscape orientation, the smaller font it selects can also be useful in portrait orientation to fit additional lines or columns of output on the page.

Selecting font size

If a type family, font collection, or font collection plus is assigned to the current SGR parameter, you can select a font by using the GSS (Graphic Size Selection) control sequence to set the font size.

The font size selection control sequence is:

CSI ⟨size⟩ ␣C

The size is in units of decipoints.

Assigning fonts to an SGR parameter

The DECATFF (Assign Type Family or Font) control sequence assigns a type family, font collection, font collection plus, or font to an SGR parameter. It uses a font identification string to specify the set of fonts being assigned to the parameter. A complete font identification string is 31 characters long, and it fully specifies a particular font. When used with the DECATFF control sequence, the font identification is abbreviated to specify only the information required for a particular type of assignment.

The built-in and downloadable fonts available with PrintKit's ANSI-PPL3 emulation, along with their font identification strings, are listed in Appendix C, ANSI-PPL3 Initial State and Fonts.

Here are the DECATFF commands for each type of SGR assignment:

DCS 1;⟨index⟩}⟨id⟩ ST	font collection plus (16-character id)
DCS 2;⟨index⟩}⟨id⟩ ST	type family (7-character id)
DCS 3;⟨index⟩}⟨id⟩ ST	font (16-character id)
DCS 4;⟨index⟩}⟨id⟩ ST	font collection (12-character id)

The index is an SGR parameter index in the range 10 to 19. The id is an abbreviated font identification string.

Once you have assigned a new value to a SGR parameter, remember to use the SGR control sequence to select it for printing, as described previously.

[1]
ANSI data type
landscape orientation
portrait orientation

Here is an example that assigns to SGR parameters 10 and 15. Parameter 10 is used as the default for portrait orientation for most paper sizes, 15 is the landscape default for most sizes. The assignments replace the default Courier with a CG Times typeface, available as a downloadable Soft Font.

DCS 2;10}RTIMES0 ST	assign type family to parameter 10
DCS 1;15}RTIMES0002SK00GG ST	assign font collection plus to parameter 15

Appendix E
Forms Overlays

E.2 Introduction
Forms overlay definition requirements
Creating forms overlays

Introduction

Forms overlays are a flexible replacement for preprinted paper stock. PrintKit allows you to use them with PostScript documents, as well as ANSI-PPL3 documents (which PrintKit translates to PostScript). To use a forms overlay:

> Create a definition for the overlay and store it in the PrintKit device control library;

> Define a medium specification that specifies the overlay;

> Print using the overlay medium, either as the default medium, or for selected pages of the document.

When PrintKit prints a page using the overlay medium, it begins by displaying the page overlay on the page. To do this, it establishes the coordinate system for the page, and then executes the commands from the overlay definition. The document text is then displayed on top of the overlay, completing the page.

Forms overlay definition requirements

Forms overlays are coded as PostScript Level 2 Form resources. A Form resource is essentially a block of PostScript code that draws a graphic; its distinction is that it can be executed very efficiently multiple times. For forms overlays, this means that complex overlays can be used while still maintaining document printing speed.

See the *PostScript Language Reference Manual*, Second Edition, for a complete description of Form resources.

When a Form resource is used with PrintKit, the Form resource name must be the same as the module name used to store the definition in the PrintKit device control library. It should not contain any uppercase letters (use lowercase instead). PrintKit uses the module name (in lowercase) as the name of the Form resource when it displays the overlay. If the two names do not match, the PostScript interpreter on your printer will report an undefinedresource error when you try to use the overlay.

Creating forms overlays

Simple forms overlay definitions can be coded by hand, if you are familiar with the PostScript language, but this is a tedious task for a form of any complexity. In most cases, it is easiest to use a graphics application to draw the form, and then adjust resulting PostScript output to be in the format required by PrintKit.

Check the Northlake web site, www.nls.com, for technical notes on creating forms overlays for use with PrintKit. There is information on manually creating forms overlays, an on tools and services that can help with the task.

[1]
ANSI data type
device control library
medium specification
overlay
POSTSCRIPT data type
overlay

A

ACCVIO message, 6.8
ADD DOCUMENT KITCP command, 4.11, B.4, B.29, B.33
ADD FINISHING KITCP command, 4.15, B.7, B.30, B.33
ADD IMPOSITION KITCP command, 4.14, B.8, B.30, B.33
ADD MEDIUM KITCP command, 4.13, B.11, B.31, B.33
ADD MODEL KITCP command, B.12, B.31, B.33
ADD QUEUE KITCP command, xiii, 4.6–4.8, B.17, B.32–B.33
ADD SHEET_PACKAGE KITCP command, 4.16, B.27, B.32, B.34
/AFTER qualifier, 3.3
ALTERNATE layup option, B.8
ANSI data type, 1.2, 1.6–1.8, 2.3, 2.6, 2.8–2.10, 2.13–2.15, 4.18, 4.20–4.22, A.23, B.4, C.1–C.2, C.6, D.1–D.3, D.7–D.8, E.2
ASCII data type, 2.3, B.4
ATTRERR message, 6.8
authorization key *see* License PAK
AUTOGEN program, 5.3
AUTOMATIC data type, xiv, 2.3, 4.4, B.4
autostart queue, xiv, 4.8, B.17

B

/BACKUP qualifier, 3.3
BADLAYDEF message, 6.9
BADLOGIC message, 6.9
/BEFORE qualifier, 3.3
binding orientation, 1.7, 2.5, 2.12, B.7
BOOTPERR message, 6.9
BORDERS parameter, 2.11, B.8
broadcast messages, 6.2–6.3
BUGCHECK message, 6.9
/BURST qualifier, 3.2
/BY_OWNER qualifier, 3.3

C

CFGQUEUPD message, 6.9
/CHARACTERISTICS qualifier, 3.3
CLOSEIN message, 6.9
CLOSEOUT message, 6.9
CONATMPT message, 6.10
CONFAIL message, 6.10
configuration, ix, xii–xiv, 1.7, 2.3, 2.6, 2.8, 2.14, 3.2, 4.2–4.9, 4.11, 4.16–4.17, 4.20, 5.3–5.6, 6.2, 6.5, 6.8–6.13, 6.16, C.2
/CONFIRM qualifier, 3.3
copies, 1.7, 2.4–2.5, B.4
/COPIES qualifier, 2.4
COPY DOCUMENT KITCP command, 4.11, B.29
COPY FINISHING KITCP command, 4.15, B.30
COPY IMPOSITION KITCP command, 4.14, B.30
COPY MEDIUM KITCP command, 4.13, B.31
COPY MODEL KITCP command, B.31
COPY QUEUE KITCP command, 4.7–4.8, B.32
COPY SHEET_PACKAGE KITCP command, 4.16, B.32

D

data type, 1.2, 1.6–1.8, 2.3, 2.9, 2.13–2.14, 3.4, 4.8, 4.18, 4.20–4.22, B.4, B.12, B.17
DATA_TYPE parameter, 2.3, B.4, B.17
DECprint Supervisor, 1.2, 1.5, 2.2, A.2
DECwindows FileView Print Dialog, 1.4–1.5
DEFAULT_MEDIUM parameter, 2.6, B.4
DELETE/ENTRY command, 6.3
device control library, 2.14–2.15, 3.2, 4.20–4.23, 5.2, 5.6, 6.2, 6.7, 6.9, 6.11, 6.13, 6.15, B.11–B.12, B.17, B.34, D.2, E.2
document specification, 2.13, 4.2, 4.8–4.11, 4.13–4.14, B.2, B.4, B.17, B.29, B.33–B.35
duplex printing, 1.6, 2.5

[1]
authorization key *see* License PAK
FileView Print Dialog *see* DECwindows FileView Print Dialog
FUSION *see* TCP/IP network protocol
folding *see* finishing process specification
forms overlay *see* OVERLAY parameter
horizontal orientation *see* landscape orientation
lpr/lpd protocol *see* TCP/IP network protocol
one-sided printing *see* simplex printing
layout, page *see* page layout
LN03 *see* ANSI data type
magnification, page *see* page scaling
MultiNet *see* TCP/IP network protocol
orientation, binding *see* binding orientation
orientation, page *see* PAGE_ORIENTATION parameter
PathWays *see* TCP/IP network protocol
PEP (Printer Ethernet Protocol) *see* TCP/IP network protocol
PPL3 data type *see* ANSI-PPL3 data type
punching *see* finishing process specification
reconfiguration *see* configuration, changing
scaling, page *see* page scaling
packet protocol *see* TCP/IP network protocol
stapling *see* finishing process specification
TCPware *see* TCP/IP network protocol
two-sided printing *see* duplex printing
vertical orientation *see* portrait orientation

E
Encapsulated PostScript (EPSF), 1.6, 2.3
ENDABORT message, 6.10
error messages, 1.7, 2.6, 3.4, 5.5, 6.6, 6.9–6.10, 6.13–6.16
/EXCLUDE qualifier, 3.3
execution queue, 2.3, 3.2, 4.2, 4.6–4.8, 4.18, 4.20, 4.22, 5.5–5.6, 6.2–6.3, 6.5–6.6, 6.9–6.12, 6.14–6.16, B.7, B.11, B.17
EXIT KITCP command, B.33
EXLICENSE message, 6.10
/EXPIRED qualifier, 3.3

F
FileView Print Dialog *see* DECwindows FileView Print Dialog
FINISHING_INCLUDES_DOCUMENT parameter, 2.12, B.4
FINISHING parameter, B.4
finishing process specification, 4.9, 4.14–4.15, B.7, B.35
FIRSTPAGE layup option, B.8
flag page, 2.4, 3.2, 6.7, B.17, B.27
/FLAG qualifier, 2.4, 3.2
folding *see* finishing process specification
font set, 1.7, 3.2, C.6, D.7
FONTS_USED parameter, 3.2, B.4
/FORM qualifier, 2.15, 3.3, D.2
forms overlay *see* OVERLAY parameter
FUNNOTSUP message, 6.10
FUSION *see* TCP/IP network protocol

G
generic queue, 4.8, B.17
GRID layup option, B.8

H
/HOLD qualifier, 3.3, 6.6
horizontal orientation *see* landscape orientation
HPGL data type, 2.3, B.4, B.12, B.17

I
imposition specification, 2.12, 4.9, 4.13–4.14, B.4, B.8, B.30, B.33–B.34, B.36
INCCONFIGVER message, 6.11
INITIAL_VALUE_DOCUMENT parameter, 2.13
INITIALIZE/QUEUE command, 4.2, 4.20, 5.3
INPUT_TRAY parameter, 1.6, 2.6–2.8, B.4, B.12
INSFMEM message, 6.11
installation, ix–xi, 1.5, 1.7, 5.2–5.4
Installation Verification Procedure (IVP), x–xi, xiv
INVLICENSE message, 6.11

J
Job Control Program, 1.2, 4.2, 6.10, B.12, B.17
/JOB_COUNT qualifier, 2.4
job status, 6.2, 6.5–6.7, B.12

K
KITCP Administration Utility, ix, xii–xiv, 1.7, 2.12–2.13, 4.2, 4.7, 4.9–4.10, 4.17, 4.22, 5.3, 6.7–6.9, 6.12–6.16, B.2, B.35–B.36

L
landscape orientation, 2.5, 2.10–2.11, C.2, D.3–D.4, D.7–D.8
LaserJet printers, 1.2
LAT network protocol, xiv, 1.2, 4.4, 4.6, 5.5, B.12, B.17
layout, page *see* page layout
LAYUP_DEFINITION parameter, 4.9, B.4, B.8
LICENSE_LEVEL message, 6.11
License PAK, ix, 6.5, 6.10–6.11
LIST data type, 1.2, 1.6–1.7, 2.3, 2.10, 2.15, A.23, B.4
LN03 *see* ANSI data type
logical name, 5.6
logical queue, 2.3, 4.6, 4.8, B.17

lpr/lpd protocol *see* TCP/IP network protocol

M

magnification, page *see* page scaling
managing queues, xi, 2.15, 3.1–3.2, 4.18, 4.20, 5.3
MARGINS layup option, B.8
medium specification, 2.6, 4.2, 4.9, 4.12–4.13, 4.15, B.4, B.11, B.27, B.31, B.33–B.34, B.36, E.2
MEDIUM_SUBSTITUTION parameter, 2.6, 2.8, B.4
message log file, 6.3, 6.7
messages, x–xii, 1.7, 2.6, 3.3–3.4, 5.4–5.5, 6.1–6.3, 6.5–6.7, 6.12–6.16, B.4
MESSAGES parameter, 3.4, B.4
/MODIFIED qualifier, 3.3
MODIFY DOCUMENT KITCP command, 4.11, B.33
MODIFY FINISHING KITCP command, 4.15, B.33
MODIFY IMPOSITION KITCP command, 4.14, B.33
MODIFY MEDIUM KITCP command, 4.13, B.33
MODIFY MODEL KITCP command, B.33
MODIFY QUEUE KITCP command, xiii, 4.7–4.8, 4.20, 4.23, B.33
MODIFY SHEET_PACKAGE KITCP command, 4.16, B.34
MODNOTFND message, 6.11
MultiNet *see* TCP/IP network protocol
multiple file jobs, 2.4

N

NOCONFIG message, 6.12
NOLICENSE message, 6.12
NOMATCHOBJ message, 6.12
NOREFOBJ message, 6.12
NOSUCHDEV message, 6.12
NOSUCHOBJ message, 6.12
/NOTE qualifier, 3.2
notification messages, 3.4, 5.4, 6.2–6.3, 6.7, A.9, B.17
/NOTIFY qualifier, 3.4, 5.4, 6.2–6.3, 6.6–6.7, A.9
NOWILD message, 6.12
NUMBER_UP parameter, 2.5, 2.10–2.11, B.4

O

OBJEXISTS message, 6.13
OBJNOTADDED message, 6.13
OBJNOTCOPIED message, 6.13
OBJNOTMODIFIED message, 6.13
OBJNOTREMOVED message, 6.13
OBJNOTSHOWN message, 6.13
one-sided printing *see* simplex printing
OPCOM (Operator Communications Facility), 6.2
OPENIN message, 6.13
OPENOUT message, 6.14
operator messages, xii, 5.4–5.5, 6.2, 6.5, 6.7, 6.15
/OPERATOR qualifier, 3.3
orientation, binding *see* binding orientation; page *see* PAGE_ORIENTATION parameter
OUTPUT_TRAY parameter, B.4, B.12
overlay, 1.6, 2.6, B.11, E.2

P

page layout, 1.2, 1.6–1.8, 2.9–2.10, 2.12, 6.9, 6.16, B.4, B.8, D.3
PAGE_LIMIT parameter, A.23, B.4
PAGE_MEDIA_SELECT parameter, A.23, B.4
PAGE_ORIENTATION parameter, 2.5, 2.10–2.11, B.4
page scaling, 1.7, 2.10
page selection expression, A.23, B.4
PAGE_SIZE parameter, 2.10, B.4, B.17
PAGEORDER layup option, B.8
PAGESPERSHEET layup option, B.8
paper handling, 1.2, 1.6
/PARAMETERS qualifier, 1.5, A.2

PARSEFAIL message, 6.14
PathWays *see* TCP/IP network protocol
PCL data type, 1.2, 1.6, 1.8, 2.3, 2.9–2.10, 2.13, 4.18, B.4, B.12, B.17
PEP (Printer Ethernet Protocol) *see* TCP/IP network protocol
portrait orientation, 2.5, 2.11, C.2, D.3–D.4, D.7–D.8
POSTSCRIPT data type, 1.6, 2.3, 2.5, 2.10, 2.14, 3.4, 4.8, 4.12, 6.10, 6.14, A.23, B.4, B.11–B.12, B.17, E.2
PostScript Printer Definition (PPD) files, 1.2, 2.14–2.15, 4.17, 4.21, B.7, B.12
PPDFMTERR message, 6.14
PPL3 data type *see* ANSI-PPL3 data type
PRINT command, 1.4–1.5, 2.2, 2.15, 3.3, A.1–A.2, A.9
PRINTKIT023.TLB Device Control Library, 4.20, 5.6
/PRIORITY qualifier, 3.3
PROTOCOL message, 6.14
punching *see* finishing process specification

Q

QUALMISS message, 6.14
QUENOTSTART message, 6.14
QUENOTSTOP message, 6.15
QUESTART message, 6.15
QUESTOP message, 6.15
/QUEUE qualifier, 2.3
queue status, 4.3, 5.6, 6.2, 6.5–6.7

R

READERR message, 6.15
reconfiguration *see* configuration, changing
related documentation, 1.8
release notes, x, 1.7
RELINK LIBRARY KITCP command, B.34
RELINK MODEL KITCP command, B.34
REMOVE DOCUMENT KITCP command, 4.11, B.34
REMOVE FINISHING KITCP command, 4.15, B.34
REMOVE IMPOSITION KITCP command, 4.14, B.34
REMOVE MEDIUM KITCP command, 4.13, B.34
REMOVE MODEL KITCP command, B.35
REMOVE QUEUE KITCP command, 4.7–4.8, B.35
REMOVE SHEET_PACKAGE KITCP command, 4.16, B.35
removing the PrintKit software, 4.23
REQUEST_TO parameter, 6.2
RESTARTINGQUE message, 6.15
retained job, 6.2–6.3, 6.6–6.7, A.9

S

scaling, page *see* page scaling
SET DATABASE KITCP command, 6.12, B.35
setup modules, 2.9, 2.13–2.15, 4.18, 4.20–4.23, 5.2, 5.6, 6.3, 6.7, 6.9, 6.11, 6.13, 6.15, C.6, D.2
/SETUP qualifier, 2.15, D.2
SHEET_COUNT parameter, 2.4, B.4
sheet package specification, 4.2, 4.15–4.16, B.17, B.27, B.32, B.34–B.36
SHEET_SIZE parameter, 2.6–2.7, 2.10, B.4, B.17
SHOW DATABASE KITCP command, B.35
SHOW DOCUMENT KITCP command, 4.11, B.35–B.36
SHOW ENTRY command, 4.18, 6.2, 6.6, A.9
SHOW FINISHING KITCP command, B.35
SHOW IMPOSITION KITCP command, B.36
SHOW MEDIUM KITCP command, B.36
SHOW MODEL KITCP command, 4.7, 4.17, B.36
SHOW QUEUE command, xiii, 4.3, 4.20, 5.3, 6.2, 6.5
SHOW QUEUE KITCP command, 4.3, 4.8, 4.18, 6.2, 6.6, A.9, B.36

SHOW SHEET_PACKAGE KITCP command, 4.16, B.36
SHOW VERSION KITCP command, 4.3, B.37
SIDES parameter, 2.5, 2.10–2.11, B.4
simplex printing, 2.5
/SINCE qualifier, 3.3
socket protocol *see* TCP/IP network protocol
stapling *see* finishing process specification
START/QUEUE command, xiv, 4.20
START QUEUE KITCP command, B.37
startup command file, 5.6, 6.14
status messages, 5.4, 6.1–6.2, 6.7
status of print job, 3.4
STOP/ABORT command, 6.3
STOP/NEXT command, 5.5, 6.5
STOP/QUEUE command, 4.22, 5.5, 6.2–6.3, 6.5
STOP QUEUE KITCP command, 6.15, B.37
STOP/RESET command, 5.5, 6.3, 6.5, 6.9, 6.11, 6.16
STOPPINGQUE message, 6.15
symbiont, 1.2, 4.2, 4.6, 5.6, 6.5, 6.8, 6.10–6.11, 6.13, 6.15, B.17
SYMDEL message, 6.15
SYNTAX message, 6.15
SYSGEN parameters, ix, 3.3, 5.3
SYSQUECRE message, 6.16

T

TAB parameter, B.4
TABLEOVER message, 6.16
TCP/IP network protocol, 1.2, 4.4–4.6, 6.9, B.12, B.17
TCPware *see* TCP/IP network protocol
trailer page, 2.4, 3.2, B.4, B.17, B.27
/TRAILER qualifier, 2.4, 3.2
troubleshooting, xii, 1.7, 4.3, 5.1–5.2, 5.4–5.5
tumble duplex printing, 2.5
two-sided printing *see* duplex printing

U

UNSUPPORTED message, 6.16
/USER qualifier, 3.3
USESETDATA message, 6.16

V

version, software, 1.7, 4.3, 5.2–5.3, 6.2, 6.11; specification, 6.2, B.2
vertical orientation *see* portrait orientation

W

WRITEERR message, 6.16

Z

ZEROAREA message, 6.16