

# **Unison Software**

## **Formation User Guide**

**Electronic Forms Printing  
for the HP 3000**



Corporate Headquarters ■ Santa Clara, CA ■ 408/988-2800 ■ Fax 408/988-2236

Austin Division ■ Austin, TX ■ 512/478-0611 ■ Fax 512/479-0735

European Headquarters ■ Harpenden, Hertfordshire ■ +44(0)1582 462424 ■ Fax +44(0)1582 462425

## Notice

**Document No. FM-510000-9706-0**

Printed in USA June 1997

Unison Software  
811 Barton Springs Road  
Austin, Texas 78704 USA  
(512) 478-0611  
Fax (512) 479-0735

Copyright © 1990-1997 by Unison Software

The information in this document is subject to change without notice.

Unison Software makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Unison Software shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Unison Software assumes no responsibility for the use or reliability of its software on equipment other than as specified in the documentation.

This document contains proprietary information which is protected by copyright. All rights are reserved. No part of this document may be photocopied, or reproduced in any form, including electronic, without the prior written consent of Unison Software.

## Printing History

An **edition** of the manual represents a complete revision, in which all pages are replaced. An **update** contains additional and replacement pages to be inserted by the customer. Editions and updates are identified by a date. The date in the notice section above corresponds to that of the current edition or update. The software **version** corresponding to each edition or update is listed below; be aware, however, that not all version changes require changes to the manual.

First Edition..... August 1993 (Version Beta)  
Second Edition..... September 1993 (Version 3.5)  
Third Edition..... November 1993 (Version 3.51)  
Fourth Edition..... January 1994 (Version 3.51)  
Update..... August 1994 (Version 3.51)  
Update..... June 1997 (Version 3.51)

Business Session® for Windows is a registered trademark of Unison Software Corporation.

Microsoft® Windows™ is a registered trademark of Microsoft Corporation.

All other product names referenced herein are trademarks of their respective companies.

---

## Documentation Conventions

The following describes the notation used throughout this user guide to illustrate syntax. All syntax illustrations and examples are printed in Letter Gothic font.

### Literals

Literals are used for intrinsic and UDC names, and are printed in uppercase. LISTENV is the UDC name in the following example:

```
LISTENV envfile
```

### Parameters

Parameter names are printed in lowercase italics, and indicate the relative positions of user-supplied variables. In the example:

```
COMPILEENV envfile
```

the parameter *envfile* is a variable that you supply when you use the COMPILEENV UDC.

### Optional Parameters

Parameters that are not required are enclosed in brackets. In the example:

```
ADDFONT envfile, [num]
```

the parameter *num* is optional.

### Required Parameters

Parameters that must be supplied are not enclosed in brackets. In the example:

```
ADDPAGE envfile, formfile
```

the parameters *envfile* and *formfile* are required.



---

## About This User Guide

This user guide provides information about how to install and use Formation. The guide is organized as follows:

Chapter 1, **Introduction**, provides a high level description of Formation's capabilities and features.

Chapter 2, **Installation**, describes the installation procedures and related installation utilities.

Chapter 3, **Environment Files**, describes environment files, and how to create them using both Formation User Defined Commands (UDCs) and the Environment Manager. This chapter also describes the environment files supplied with Formation and pertinent utilities.

Chapter 4, **Print Options**, describes the different printing methods. This includes both the file equate facility and the LPRINT utility. This section also includes information about related utilities.

Chapter 5, **Fonts and Graphics**, describes character fonts and bar code fonts, their use, and conversion of fonts with Formation.

Chapter 6, **Intrinsics**, lists and explains Formation intrinsics and includes a sample program illustrating the use of Formation intrinsics.

Chapter 7, **HP 2680 Application Printing to LaserJets and HP 5000s**, describes how to print HP 2680 applications on the LaserJet and HP 5000.

Chapter 8, **Using Formation with ASK**, describes how to use Formation with ASK applications.

Appendix A, **TroubleShooting**, provides troubleshooting information, including a list of error messages for the intrinsics.

Appendix B, **Glossary**, contains a list of Formation printing terms and their definitions.

See the *Table of Contents* for a complete list of topics.



---

# Contents

Documentation Conventions .....	iii
About This User Guide .....	iv

**Contents .....** **v**

<b>1. Introduction .....</b>	<b>1-1</b>
Hardware / Software Requirements .....	1-1
Formation System Overview .....	1-2
Formation Print Terms .....	1-6
Formation's Documentation Set .....	1-6
<b>2. Installation .....</b>	<b>2-1</b>
Install the Software .....	2-2
Testing the Preliminary Installation .....	2-4
Quick Start for Demo Users .....	2-5
Install File Equate Facility .....	2-5
STOPFEQ Utility .....	2-6
Install Intrinsic .....	2-7
Upgrading Formation .....	2-7
LaserJet Serial Device Configuration .....	2-8
After Installing Formation .....	2-9
<b>3. Environment Files .....</b>	<b>3-1</b>
Creating Environment Files with UDCs .....	3-2
ADDFONT UDC .....	3-3
ADDPAGE UDC .....	3-5
COMPILEENV UDC .....	3-7
CREATEENV UDC .....	3-8
LISTENV UDC .....	3-10
MAKEENV UDC .....	3-11
Creating Environment Files with the Environment Manager .....	3-13
Environment File Menu .....	3-15
Main Menu .....	3-16
Physical Page Control .....	3-20
Multi-Copy Forms .....	3-25
Scale IDForm to LaserJet .....	3-27
Character Set Control .....	3-28
Logical Page Control .....	3-34
Logical Page Form Control .....	3-40
VFC Menu .....	3-43
Delete Character Font .....	3-44
Delete Logical Page .....	3-45
Compiling Environment Files .....	3-46

DOWNLENV Utility .....	3-47
Supplied Environment Files .....	3-48
<b>4. Printing Options .....</b>	<b>4-1</b>
File Equate Facility .....	4-2
LPRINT Utility .....	4-3
Printing with Pre-Space Carriage Control .....	4-4
Printing Simple Forms .....	4-4
Reduction Printing .....	4-6
Multi-Part Forms .....	4-7
Duplexed Forms .....	4-12
Resetting the Printer .....	4-13
Moving Data on the Page .....	4-13
Controlling Form Feeds .....	4-16
Printing PC Documents .....	4-17
Spool Queue Monitor .....	4-18
Creating SQM1DEV .....	4-19
Creating SQM1ENVD .....	4-20
SQM Operations .....	4-26
<b>5. Fonts and Graphics .....</b>	<b>5-1</b>
Understanding Fonts .....	5-2
Formation's Font Directory .....	5-4
Adding Fonts to the Directory .....	5-6
Converting PCL Fonts to PCELL Files .....	5-16
Convert Utility .....	5-17
Formation Conversion Utility Screen .....	5-18
Logo Management Screen .....	5-19
FIGPRINT Utility .....	5-21
Data Font Considerations .....	5-21
FONTDOWN Utility .....	5-26
Bar Code Fonts .....	5-27
2 of 5 Bar Code Checksum and Interleaving .....	5-30
Postnet with Formation .....	5-31
<b>6. Intrinsic .....</b>	<b>6-1</b>
Capabilities Available with Intrinsic .....	6-2
Converting Programs for Use with Formation Intrinsic .....	6-7
Calling Intrinsic .....	6-10
Communication Area (comArea) .....	6-12
Using Intrinsic In a Program .....	6-16
Intrinsic Sample Program .....	6-20
Intrinsic Listing .....	6-29
HP36580 .....	6-32
HP36580V .....	6-33
PACTIVATEPAGE .....	6-34
PBAR2OF5STRING .....	6-35
PBOX .....	6-36
PDEACTIVATEPAGE .....	6-37
PDELETERASTER .....	6-38
PDUPLEX .....	6-39
PENDSAVE .....	6-40
PERRMSG .....	6-41
PFIELDINFO .....	6-42

PFLASHRASTER .....	6-43
PFONTINFO .....	6-45
PFONTNUM .....	6-46
PFRONTPAGE .....	6-47
PINITDEVICE .....	6-48
PINITIALIZE .....	6-50
PJOBSEPARATE .....	6-51
PLOADRASTER .....	6-52
PLOGPAGEINFO .....	6-53
PMOVEPENABS .....	6-55
PMOVEPENREL .....	6-56
PNEWFORM .....	6-57
PNEWPAGE .....	6-58
PNEWPHYSPAGE .....	6-60
PNEWSUBFORM .....	6-61
PPRINTFIGURE .....	6-62
PPRINTRASTER .....	6-64
PPRINTSAVED .....	6-66
PRESET .....	6-67
PSAVEDATA .....	6-68
PSELECTPAGE .....	6-69
PSETCOPIES .....	6-70
PSPPLUSVERSION .....	6-71
PSTATEINFO .....	6-72
PSTRINGWIDTH .....	6-73
PSWITCHPAGE .....	6-74
PSWITCHPHYSPAGE .....	6-75
PUSEFONT .....	6-76
PWRITEFIELD .....	6-77
PWRITEFIELDJUST .....	6-78
PWRITEFIELDREL .....	6-79

<b>7. HP 2680 Applications Printing to LaserJets and HP 5000s .....</b>	<b>7-1</b>
Introduction .....	7-1
What is Migration? .....	7-2
Updating Applications for New Printers .....	7-5
Analyze Migration Requirements .....	7-5
Recompile Environment Files .....	7-7
Convert Programs .....	7-12
Modify / Update Job Streams .....	7-15
Modifying Forms .....	7-18
Multi-Part Forms on HP 2680 and HP 2688 Printers .....	7-25
Beyond Migration .....	7-28
<b>8. Using Formation with ASK .....</b>	<b>8-1</b>
Print Sample Forms with Your Data .....	8-2
Customize The Forms .....	8-3
Modify the Form .....	8-5
Compile and Print .....	8-7
Production Printing .....	8-9
Multi-Part Forms .....	8-9

---

<b>A. Troubleshooting</b> .....	<b>A-1</b>
Error Message 20 .....	A-1
Expanding the Environment File .....	A-1
Performance and Reduction Printing (Error 21) .....	A-2
Formation Error Messages .....	A-3
Printer Problems .....	A-3
Formation Error Codes .....	A-4
<b>B. Glossary</b> .....	<b>B-1</b>
<b>Index</b> .....	<b>index-1</b>



# Introduction

Formation is a complete electronic forms solution that lets HP 3000 users eliminate the high costs of using pre-printed forms. With Formation, forms and data can be merged electronically, and printed on centralized or departmental laser printers.

Formation provides:

- Tools for creating and printing electronic forms with HP 3000-based data on HP laser printers.
- Support for professional fonts, bar codes, logos, and signatures.
- No program changes—electronic forms replace pre-printed forms automatically at run-time.
- Complete library of procedures for form sequencing on HP laser printers.
- Intelligent forms capability, making applications independent of form design—when a form changes, the output is repositioned automatically.
- Customizable ASK forms allowing users to easily merge their company forms requirements with ASK application data.

Additional features available with the Formation/2680 option include:

- Converting and printing of forms created with HP's IDSSForm on HP LaserJets and high-speed printers.
- Support for printing Formation-created forms on HP 2680/88 printers.
- Support for creation of a single environment file which will print on any HP laser printer.

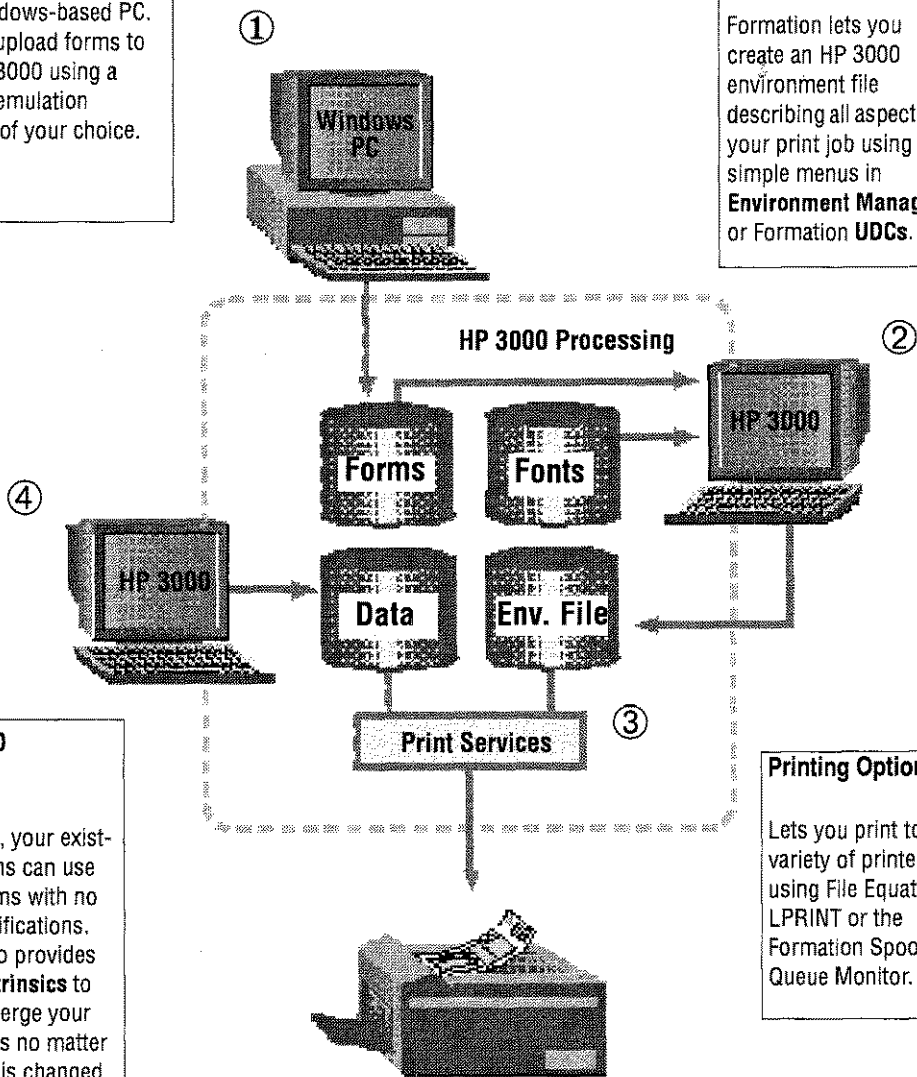
## Hardware/Software Requirements

- Any HP3000 system running the following operating system releases:
  - MPE-V release G.03.00 or later
  - MPE/XL release A.30.30 (2.0) or later
- Minimum PC configuration:
  - HP Vectra, IBM PC/AT, IBM PS/2, COMPAQ DESKPRO 386 or compatible system
  - Microsoft Windows 3.1 or later
  - 640K of memory (2Mb recommended)
  - Serial communications adapter, Ethernet adapter or Eicon Network Adapter Card
  - Any monitor and video adapter supported by Windows (e.g. Super VGA, VGA, EGA, Hercules, HP MultiMode, MCGA)

# Formation System Overview

**Formation Design**  
Lets you design forms on your Windows-based PC. You can upload forms to your HP 3000 using a terminal emulation program of your choice.

**Environment Files**  
Formation lets you create an HP 3000 environment file describing all aspects of your print job using the simple menus in **Environment Manager** or Formation UDCs.



**Your HP 3000 Application**  
In most cases, your existing applications can use electronic forms with no program modifications. Formation also provides application **intrinsic**s to intelligently merge your data and forms no matter how the form is changed.

**Printing Options**  
Lets you print to a variety of printers using File Equate, LPRINT or the Formation Spool Queue Monitor.

## Formation Tools

Formation gives you the tools you need to provide high quality print output for your users and eliminate the high costs and delays of using pre-printed forms. Formation includes:

- Formation Design, a forms drawing tool which runs in the Microsoft Windows environment.
- Environment Manager and User Defined Commands (UDCs) to create and compile environment files.
  - The Environment Manager is a menu-driven program that lets you compile your form design into a standard MPE environment file. You can then merge it with the data from your HP 3000 application with a simple file equation.
  - UDCs are simple commands that enable you to compile your form design into an environment file at the command line.
- Printing Options that enable you to send your form description and application data to your printer.
- Intrinsic, a complete library of procedures which you can integrate into your existing applications for advanced forms handling.

Formation also includes several utilities that perform functions complementary to other Formation tools.

## Formation Design

Formation Design is a professional forms design tool which runs under Microsoft Windows 3.1. With this powerful WYSIWYG drawing program you can create an electronic form in minutes on any PC. Creating or modifying forms with Formation Design takes a fraction of the time and cost of traditional forms design methods.

Using the mouse, you position the elements which make up your form: lines, boxes, grids, combs, and text. Formation Design provides a variety of line widths, styles, and shading patterns for graphic elements, and text can be formatted in different typefaces, sizes, and styles. To the basic layout, you can add logos, signatures, clip art, or other graphics imported from any Tag Image File Format (TIFF) file. TIFF files are produced by most scanners and PC-based graphics programs, so the possibilities are virtually unlimited.

Once your form is complete, you can copy it to your HP 3000 using the file transfer function of a terminal emulation product of your choice.

Forms created with Formation Design can be modified in minutes, instead of the weeks or months needed to update pre-printed forms. And one electronic form will work on all of your HP LaserJets and 2680/88 printers.

For more information about using Formation Design, read the *Formation Design User Guide*.

## Environment File Management

Formation Environment Manager, a menu-driven HP 3000 program, lets you list everything involved in a particular print job: forms, character fonts for data (including bar codes), characteristics of the physical and logical pages, form placement on page, and form sequence. This information is compiled into a standard HP 3000 environment file—which describes your specifications to your printers.

Formation/2680 adds IDSForm compatibility to the Environment Manager. With Formation/2680, IDSForm files can be recompiled and printed on any HP LaserJet, and Formation Design forms can be printed on a 2680/2688.

For more information, refer to "Creating Environment Files with the Environment Manager" on page 3-13.

In addition to the Environment Manager, Formation lets you to use UDCs and terminal type files (essentially configuration files for a particular printer) to format your output.

You can use Formation UDCs to quickly create, update, and compile environment files with a simple string of commands. Formation includes a number of sample UDC for your use. For more information refer to "Creating Environment Files with UDCs" on page 3-2.

### Formation Printing Options

Formation features the following printing options:

- File equate facility
- Print utilities
- Spool Queue Monitor
- Print intrinsics

Once your print job description has been captured in an environment file, this information must be passed to the printer to be merged with the data generated by your application. Formation's file equate facility takes care of this. You just enter an MPE file equation for your application's output file name, and form and data are merged at print time.

The Formation utilities provide additional useful functions for the printing environment. The LPRINT utility, for example, supports 2-up and 4-up printing of any host file on a LaserJet. This is useful for printing lengthy program and system listings.

The Spool Queue Monitor(SQM) enables you to monitor and redirect spool file printing and include environment file information when printing spool files to laser printers. SQM simplifies the printing process by allowing you to set-up printer "types" that perform special functions such as printing to device classes TWOUP, FOURUP, or INVOICES.

SQM includes the capability to do distributed printing from one application output file and to use environment file definitions for printing application data even when you cannot change the application to include an environment file.

The Formation intrinsics provide application developers with additional control over laser print jobs (see the following).

### Advanced Services Through Intrinsics

Formation provides a set of procedures, called intrinsics, you can include in applications to eliminate costly re-programming should your forms requirements change.

With pre-printed forms, the programmer who writes your applications has to know exactly what the form looks like to ensure that the output will line up properly. If the form changes, the programmer has to reformat the output. With Formation's "intelligent forms" capability, you can separate form layout from application printing specifications, so that users can design any number of custom forms without any changes to the application program.

For more information about using intrinsics, refer to Chapter 6, *Intrinsics*.

## Formation Utilities

Formation includes several utilities that perform various functions including printing files, downloading environment files to the printer, creating the font directory, and converting TIFF files to PCELL or raster.

The following table lists each of the available utilities in alphabetic order. The utilities are explained in the various chapters where they apply.

**Table 1-1: Utilities**

Name	Definition
CONVERT	Convert TIFF files to PCELL or raster format. For more information refer to "Convert Utility" on page 5-17.
DOWNLENV	Pre-download environment files to the printer. For more information refer to "DOWNLENV Utility" on page 3-47.
FDCOMP	Compile the font directory. For more information refer to "Formation's Font Directory" on page 5-4.
FIGPRINT	Print figure files from graphics packages. For more information refer to "FIGPRINT Utility" on page 5-21.
FONTDOWN	Download PCL fonts to the LaserJet. For more information refer to "FONTDOWN Utility" on page 5-26.
JETSTART	Convert an MPE V or compatibility mode program to use the Formation intrinsics. For more information refer to "JETSTART Utility" on page 6-7.
JETSTOP	Revert an MPE V or compatibility mode program to use HP PSP intrinsics after having been converted with JETSTART. For more information refer to "JETSTART Utility" on page 6-7.
LPRINT	Print Formation files on a LaserJet or HP 5000 printer. This utility features multiple options, including multi-part forms, duplex printing, and reduction printing. For more information refer to "LPRINT Utility" on page 4-3.
PCL2DCS	Convert PCL fonts to PCELL font files. For more information refer to "Converting PCL Fonts to PCELL Files" on page 5-16.
SIZELIST	Provide point size information about a PCELL font file. For more information refer to "Data Font Considerations" on page 5-21.
STARTFEQ	Activate the File Equate Facility on MPE XL systems. For more information refer to "STOPFEQ Utility" on page 2-6.
STOPFEQ	Deactivate the File Equate Facility on MPE XL systems. For more information refer to "STOPFEQ Utility" on page 2-6.
TFMTRANS	Read the Tagged Font Metric (TFM) file for a resident font, and create FDUSER text for each of the required point sizes and typefaces. For more information refer to "Extract Font Metrics From the TFM file" on page 5-14.

## Formation Print Terms

This section defines some of the more common terms used in this manual to describe aspects of forms design and printing. A glossary is included at the end of this user guide with additional terms and definitions.

<b>baseline</b>	An imaginary line along which the printer aligns character cells. The baseline runs along the bottom of most characters in a font, but is above the lowest part of letters with descenders (g, j, p, q, y).
<b>CPI</b>	Characters per inch. A measure of the spacing of adjacent characters on a page. Also called pitch.
<b>Dots or DPI</b>	Dots Per Inch. A measure of the resolution of a printed page. LaserJets print at 300 dpi and the 2680 printers print 180 dpi.
<b>font</b>	A family of print characters that share a single typeface, weight, style, and size that generally forms a complete set of alphabetic characters, numerals, and punctuation.
<b>HMI</b>	Horizontal motion index. Indicates horizontal movement across a page and is equivalent to pitch in fixed pitch font measures.
<b>logical page</b>	A rectangular area of any size within the dimensions of the physical page.
<b>LPI</b>	Lines per inch.
<b>orientation</b>	The direction of printing, either landscape or portrait.
<b>physical page</b>	A sheet of paper.
<b>VMI</b>	Vertical motion index. Indicates the vertical movement across a page and is equivalent to leading.

## Formation's Documentation Set

Formation's documentation is organized to help you design and produce electronic forms quickly and efficiently. Whether you are novice or expert, you'll benefit more from the documentation set if you're familiar with its pieces:

- *Formation Design User Guide*—describes the procedures for creating forms using Formation's MS-Windows-based forms design tool. A hands-on tutorial is included to teach you how to design forms and upload them to the HP 3000.
- *Formation User Guide*—contains the procedures for installing Formation, preparing the forms you have designed, and information about printing your forms, as well as instructions for solving common errors. In addition, this user guide contains reference information about adding intrinsics to applications.
- *Formation with HP 2680*—describes how to produce electronic forms on the LaserJet and HP 5000 printers from HP 2680 applications and is included in the *Formation User Guide*.
- *Formation with ASK*—describes how to quickly produce electronic forms with Formation using your ASK applications and is included in the *Formation User Guide*.
- *Formation File List*—contains a list of all Formation files in your package.

### Where Should I Start?

The following table provides some guidelines for using Formation tools and documentation at each stage of electronic forms production. The categories are grouped by task.

If you want to...	Formation provides...	At this stage you should read...
Design one or more forms,	Formation Design, a professional forms design tool which runs under Microsoft® Windows™ 3.1.	the <i>Formation Design User Guide</i>
Upload a PC form file,	N/A	Chapter 3 of the <i>Formation Design User Guide</i>
Compile the uploaded form,	Environment Manager and UDCs on the HP 3000 that enable you to include your form in an environment file, compile your form, and merge it with application data.	Chapter 3 of the <i>Formation User Guide</i>
Print to an HP printer,	Print Services—via LPRINT, the file equate facility, and Spool Queue Monitor—that allow you to print your form and data in one printing pass.	Chapter 4 of the <i>Formation User Guide</i>
Smart link application data and your forms,	Intrinsics to programmatically associate named items on forms with corresponding information in the application.	Chapter 6 of the <i>Formation User Guide</i> and Chapter 4 of the <i>Formation Design User Guide</i>

This document provides additional guidance in locating the appropriate sections to read at the end of Chapter 2, *Installation*. For more information, refer to "After Installing Formation" on page 2-9.





# *Installation*

This chapter describes the procedures required to install Formation on MPE V or MPE XL machines. The installation instructions are divided into the following sections:

- Installing the Software describes the procedure for installing the basic Formation software set on both MPE V and MPE XL machines.
- Testing the Installation provides two sample test procedures for LaserJet, HP 5000, and 2680 users.
- Installing the File Equate Facility contains the installation instructions for MPE V, MPE XL (version 4.0 or later) and MPE XL versions prior to 4.0.
- Installing the Formation intrinsics contains separate instructions for installing intrinsics on MPE V and MPE XL machines.
- Upgrading Formation describes what steps to take when upgrading Formation software.

Demo users have the option of stopping the installation process after testing the installation with the following considerations:

- Without installing the Formation intrinsics and file equate facility, users will be working with a subset of Formation's full capabilities.
- Users running MPE XL 4.0 or later need only activate the file equate facility to complete a preliminary installation.

## **Topics**

This chapter describes the following installation topics:

- Install the Software
- Testing the Preliminary Installation
- Quick Start for Demo Users
- Install File Equate Facility
- Install Intrinsics
- STOPFEQ Utility
- Upgrading Formation
- LaserJet Serial Device Configuration
- After Installing Formation

## Install the Software

To install the Formation software, perform the following steps on either your MPE V or MPE XL machines.

1. Log on as:

```
MANAGER.SYS
```

2. Set the job limit on your system to at least two.
3. Restore your product tapes as shown in the following:

```
FILE SYSLIST;DEV=LP
```

```
FILE TYMTAPE;DEV=TAPE
```

```
RESTORE *TYMTAPE;@.@.@;CREATE;SHOW
```

4. Use EDITOR to add passwords for MANAGER.SYS to line one of JSETUP.FORM.TYM.

If the TYM account already exists on your system, you may also need to add passwords to the line that streams JFONTDIR.FORM.TYM. (If you have purchased only Formation Print Services, this job is not streamed.)

The following instructions illustrate how to add passwords using EDITOR.

```
EDITOR
```

```
/T JSETUP.FORM.TYM;M1
```

```
FILE UNNUMBERED
```

```
!JOB JSETUP,MANAGER.SYS,PUB
```

(add any required passwords for MANAGER user and SYS account)

```
/F "JFONTDIR"
```

```
#JOB JFONTDIR,MANAGER.TYM,FORM
```

(add any required passwords for MANAGER user and TYM account)

```
/K $NEWPASS,UNN;:STREAM $OLDPASS (this avoids saving a file containing passwords)
```

```
$NEWPASS ALREADY EXISTS-RESPOND YES TO PURGE OLD AND KEEP NEW.
```

```
PURGE OLD?Y
```

```
#Jnn
```

```
/E
```

5. Install the Formation UDCs system-wide, using one of the following commands.

Enter the following command:

```
SETCATALOG FMUDCS.FORM.TYM;APPEND;SYSTEM
```

The APPEND keyword ensures that any current UDCs on your system remain active.

**MPE XL  
Users**

**MPE V  
Users**

If you do not currently have any system-wide UDCs, enter the following command:

```
SETCATALOG FMUDCS.FORM.TYM;SYSTEM
```

If you currently have system-wide UDCs, enter the SETCATALOG command. Specify each of your system wide UDCs separated by a comma, with the Formation UDC as the last UDC in the command.

```
SETCATALOG UDCNAME, ..., FMUDCS.FORM.TYM;SYSTEM
```

**Installation Validation for Demo Users**

The HP 3000 software you've installed is set to expire 30 days from the date you receive it. To find out how long you have left to test the software, you can run the following program at any time:

```
ENVMGR.FORM.TYM
```

The program displays one of the following two types of messages: "n DAYS LEFT IN DEMO PERIOD" or PROGRAM TIME HAS EXPIRED. If you decide to purchase Formation, call us to request an invoice. Once we receive your payment, we will validate your copy of the product so that it becomes a permanent version.

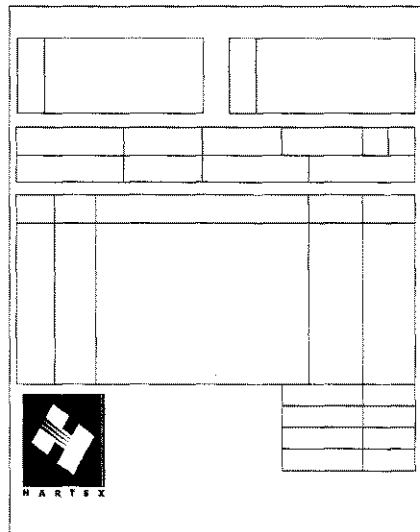
**If you have any questions or problems, call your Unison Software support representative.** You have unlimited phone-in consulting privileges throughout your demo period.

## Testing the Preliminary Installation

If your system includes LaserJets or HP 5000s, you can test your Formation installation and printer configuration by printing a sample electronic form.

To print a sample electronic form, enter the following commands. When using the following example, replace the *device* parameter with the device number or name of a LaserJet or HP 5000. These commands print three copies of the following invoice merged with data from the HP 3000

```
:HELLO MANAGER.TYM, FORM
:FILE LPIN=INVDATA
:FILE LPLIST;DEV=device;ENV=FILE2
:RUN LPRINT;LIB=G
```



**Note:** The LPRINT command does not print to the HP2680 or HP 2688 printers.

If you are using HP 2680/88 printers, you can use the following test instructions to print a sample electronic form:

```
:HELLO MANAGER.TYM, FORM
:FILE PRINTR;DEV=device;ENV=FILE2
:FCOPY FROM=INVDATA; TO=*PRINTR
```

The system will issue the following prompt:

```
CONTINUE OPERATION (Y OR N)
```

Enter the letter Y to print the electronic form with the sample data on your printer. You need to provide your printer's device name or number. You can replace the file name PRINTR with any name of your choice. However, the file name must be exactly the same on both lines.

## Quick Start for Demo Users

After performing the preliminary installation steps, demo users can start using some of the very basic features of Formation to print electronic forms and bypass the rest of the installation instructions.

Keep the following considerations in mind when running Formation without performing the full installation:

- Installation of the intrinsics and file equate facilities for MPE V or MPE XL provides added features not contained in the preliminary installation.
- While you can run some Formation features on 2680/88 printers without the full installation, the preliminary installation features work most effectively with LaserJet and HP 5000 printers.
- Formation automatically installs the file equate facility on MPE XL versions 4.0 or later.
- Simply activating the file equate facility makes it available for MPE XL (4.0) users.
- Users running MPE XL version 4.0 or later must activate the file equate facility to make use of its features.

### MPE XL Users

#### Activate File Equate Facility with STARTFEQ

During installation, if you are running MPE XL version 4.0 or later, activate the file equate facility by performing the following steps:

1. Log on as `MANAGER.SYS`.
2. Activate the file equate facility as follows:

```
RUN STARTFEQ.PUB.SYS
```

You must run the `STARTFEQ` program after each system start. To run `STARTFEQ` automatically, make the program part of a system start-up procedure or UDC.

---



---

### Demo Installation Complete!

At this point, demo users or others who wish to make preliminary use of Formation's electronic forms capabilities can skip the rest of the installation. The full installation is recommended for all Formation users and especially for HP 2680/88 users or users who are upgrading their Formation software.

---



---

## Install File Equate Facility

The next stage of the Formation installation is to install the file equate facility. MPE V and MPE XL systems require slightly different instructions.

**Note:** As indicated in the previous sections, Formation automatically installs the file equate facility on systems running MPE XL version 4.0 or later.

### MPE XL Users

#### MPE XL Versions Prior to 4.0

If you are installing Formation on an MPE XL version prior to 4.0, perform the following steps to install the file equate facility.

1. Add passwords for `MANAGER.SYS` to line one of `JINFEQXL.FORM.TYM`.
2. Keep the modified file as `$NEWPASS` and stream `$OLDPASS`.

This job creates two tapes:

- A backup SLT
- An SLT with the file equate facility

If you already have a backup SLT, reply to the first tape request with zero. The job warns that it is not creating a backup and proceeds with installation.

3. Restart your system using the second tape created by the `JINFEQXL` job stream.
4. If you are not already logged on as `MANAGER.SYS` (or any user with SM or OP capability), do so now. Activate the file equate facility as follows:

```
:RUN STARTFEQ.PUB.SYS
```

This step is necessary after each system start, and should be made part of a system start-up procedure or UDC.

5. Re-install the intrinsics and the file equate facility each time you update your operating system.

If you are running Formation on an MPE V CPU, perform the following steps:

**MPE V  
Users**

1. Add passwords for `MANAGER.SYS` to line one of `JINFEQV.FORM.TYM`.
2. Keep the modified file as `$NEWPASS` and stream `$OLDPASS`.

This job stream creates two cold load tapes:

- One contains a backup of the current configuration
- The second contains the file equate facility

You can run `JINFEQV` while the system is in use, although it locks the system Segmented Library (SL) for about 30 seconds while creating the backup cold load tape.

If you already have a backup cold load tape, you can reply to the first tape request with zero. After issuing a message that it is not creating a backup cold load tape, the job stream continues with the installation.

3. Perform a cold load from the second tape created by the `JINFEQV` job stream.
4. Re-install the intrinsics and the file equate facility each time you update your operating system.

**STOPFEQ Utility****MPE XL  
Users**

The `STOPFEQ` utility allows you to turn off the file equate facility. To run the utility:

1. Log on as `MANAGER.SYS` (or any user with SM or OP capability).
2. Deactivate the file equate facility as follows:

```
RUN STOPFEQ.PUB.SYS
```

## Install Intrinsic

The final stage of the installation process is to install the intrinsic. The following instructions apply to both MPE V and MPE XL machines.

1. Add passwords for MANAGER.SYS to line one of JSYSSL.FORM.TYM.
2. Keep the modified file as \$NEWPASS and stream \$OLDPASS.
  - Do not stream this job while any program that uses the Formation intrinsic or the HP PSP intrinsic is in use (e.g., the LPRINT utility).
  - If a program that calls the intrinsic is in use, some segments in which intrinsic are installed may be locked, resulting in a partial installation. With a combination of old and new intrinsic on your system, your software will not run correctly.
  - JSYSSL only checks HP software for use of HP PSP intrinsic; it does not check any other programs.
  - During the installation process, don't use or allocate any other applications programs containing HP PSP intrinsic.
3. Re-install the intrinsic in the system SL each time you update your operating system. To do this, just stream JSYSSL again.

Most users place the Formation Print Services intrinsic in the system SL because this is the most convenient place for users to access the software, and it is the place where most HP software expects to find it. However, if you cannot install in the system SL, install it in a group SL as follows:

1. Modify line one of JGRPSL.FORM.TYM to log on to the group and account in which you want to place the Formation intrinsic.
2. Add any passwords. If a file named SL does not already exist, this places a copy of SL.FORM.TYM in the group and account to which you modify JGRPSL.FORM.TYM. If an SL file already exists, the intrinsic are added to it.
3. Run your forms printing programs from the designated group, adding the LIB=G option to your :RUN command.

## Upgrading Formation

Users who are upgrading their Formation software need to re-install the:

- Formation intrinsic in the system SL by streaming JSYSSL
- File Equate Facility (if you are running MPE XL versions prior to 4.0).

Users running MPE XL 4.0 or later need to activate the file equate facilities with STARTFEQ to complete their upgrade tasks. For more information about installing the File Equate Facility, see "Install File Equate Facility" on page 2-5.

## LaserJet Serial Device Configuration

Terminal Type (TT) files are used to configure the initialization and behavior of the serial port on both MPE/V and MPE XL (MPE/iX) systems. Typically, PUB.SYS contains the TT files, although they may occasionally be found in other groups. Commonly used terminal types supplied by HP include:

- TT10 (the standard TT used for HP terminals)
- TT18 (generic non-HP serial device [either terminal or printer])
- TT22 (standard HP PCL printer requiring status checking hardware in the printer)
- TT26 (modified version of TT22 for use in some data communications environments such as over multiplexors which requires status checking hardware in the printer)

TT18 is a poor choice for LaserJet printers used with Formation because the printer is not reset before each spool file. Without a reset, unrelated text only spool files that follow a spool file with an associated environment file will print on electronic forms still set up for the printer from the prior spool file. A reset of the LaserJet is necessary before each spool file to clear the remaining electronic forms from the Formation environment file and to return the printer to the configured defaults.

On MPE/V, HP also provides TTPCL18 which does not require the status checking hardware but does reset the LaserJet printer before each spool file. On LaserJet printers which do not have the status checking hardware, TTPCL18 is the recommended terminal type file.

On MPE/iX, HP provides TTUTIL.PUB.SYS for users to create and edit terminal type files. Formation includes TT99.PUB.SYS which has been created with TTUTIL to provide equivalent functionality with TTPCL18 on MPE/V systems. It is the recommended terminal type file to use on LaserJet printers which do not provide status checking hardware. The terminal type file is specified by number in the profile of the serial device configuration in NMMGR.

The following table summarizes the terminal type files to use for your configuration.

If you have:	Then use:
Early LaserJet printers with status checking hardware (up to and including the LaserJet II and LaserJet 2000)	TT22 or TT26.
LaserJet printers made after the LaserJet 2000 with the optional status checking hardware installed (including the LaserJet IID, IIP, III & IV families),	TT22 or TT26.
LaserJet printers made after the LaserJet 2000 <u>without</u> status checking support (including the LaserJet IID, IIP, III & IV families),	TTPCL18, MPE V users. TT99, MPE XL or MPE iX users.



## After Installing Formation

After you have installed the Formation package on your system you are ready to begin designing and producing electronic forms. If you want to know where to begin by task, see "Where Should I Start?" on page 1-7. Otherwise, use the following table to familiarize yourself with the Formation product.

Table 2-2: Guide to Using the Documentation

If You Are...	Then read...	Topic Is	
		Required (General)	Optional (Advanced)
An ASK User	Chapter 8, <i>Using Formation with ASK Applications</i>	•	
	about LPRINT in Chapter 4, <i>Print Options</i>	•	
	the <i>Formation Design User Guide</i>	•	
	Chapter 3, <i>Environment Files</i>		•
Migrating from 2680/IFS	Chapter 7, <i>HP 2680 Applications Printing to LaserJets and HP 5000s</i>	•	
	the <i>Formation Design User Guide</i>	•	
	Chapter 3, <i>Environment Files</i> (especially Environment Manager)	•	
	Chapter 4, <i>Print Options</i>		•
Replacing Pre-Printed Forms	the <i>Formation Design User Guide</i>	•	
	Chapter 3, <i>Environment Files</i> (especially the section on the Environment Manager)	•	
	Chapter 4, <i>Print Options</i>	•	
	Chapter 5, <i>Fonts and Graphics</i>		•
	Chapter 6, <i>Intrinsics</i>		•
A Forms Designer	the <i>Formation Design User Guide</i>	•	
	Chapter 5, <i>Fonts and Graphics</i>		•
An Application Programmer using forms designed by someone else	Chapter 3, <i>Environment Files</i>	•	
	Chapter 4, <i>Print Options</i>	•	
	Chapter 6, <i>Intrinsics</i>		•
	Chapter 5, <i>Fonts and Graphics</i>		•
An Application Programmer designing "smart forms"	the <i>Formation Design User Guide</i>	•	
	Chapter 6, <i>Intrinsics</i>	•	
	Chapter 3, <i>Environment Files</i>	•	
	Chapter 4, <i>Print Options</i>	•	
	Chapter 5, <i>Fonts and Graphics</i>		•



## *Environment Files*

After designing a form using Formation Design and uploading it to an HP 3000, you must provide the printer with additional information to print the form and merge it with application data. Formation supplies the printer with this information by using environment files. Environment files contain all the printing information needed to process a page, including fonts, forms, margins and line spacing.

Producing an environment file requires the following steps:

1. Name the file.
2. Add data fonts.
3. Add physical page specifications.
4. Add logical page specifications.
5. Compile the environment file.

Formation provides two different methods to create an environment file:

- Formation UDCs offer a quick, command line-oriented method for creating an environment file. With most of the forms attributes defined through the Formation Design form setting capability, most of your environment file functions can be accomplished with UDCs.
- The Environment Manager features a menu-driven method for creating an environment file. Using the Environment Manager, you can:
  - Define physical page attributes such as number of copies and paper size.
  - Scale IDSForms to LaserJets.
  - Delete logical pages and data fonts.

Once created, you can compile an environment file using Formation UDCs or the Environment Manager. Compiling the environment file converts form specifications and information into printer commands (PCL or DCS).

### **Topics**

This chapter describes the following Environment File topics:

- Creating Environment Files with UDCs
- Creating Environment Files with the Environment Manager
- Compiling Environment Files
- DOWNLENV Utility
- Supplied Environment Files

## Creating Environment Files with UDCs

Formation supplies several environment file UDCs that you can use to:

- Create an environment file.
- Define basic logical page information.
- Add character fonts.
- Compile the environment file.

If you create environment files with Formation UDCs, you can also easily modify them using the Environment Manager. For more information refer to "Creating Environment Files with the Environment Manager" on page 3-13. The following table lists the Formation UDCs, and provides a brief description of how they are used. Full descriptions of the UDCs are provided on the pages that follow.

**Table 3-3: Environment File UDCs**

UDC Name	Description
ADDFONT	Add or replace a data font in an environment file. – Use ADDFONT to build and update multi-page environment files.
ADDPAGE	Add or replace a logical page in an environment file. Each logical page can include up to two forms. – Use ADDPAGE to build and update multi-page environment files.
COMPILEENV	Compile an environment file. – Use the CREATEENV UDC with environment files that require additional logical pages and fonts as it is more efficient to compile only once.
CREATEENV	Create an environment file, and define the first logical page and data font in the environment.
LISTENV	Display a listing of environment file information. – The listing can be printed to a terminal or a printer and includes printer, form file, physical page, character set, and logical page information. – The LISTENV UDC performs the same function as the <b>List Environment</b> command on the Environment Manager <b>Main Menu</b> .
MAKEENV	Create and compile an environment file. – Use the MAKEENV UDC to create simple environment files containing a single logical page and data font. This UDC creates and compiles the environment file in a single command.

When you run these UDCs, include the environment file name and the name of a form you created with Formation Design. Optional parameters include defining a secondary form file and a duplexing option. Form files included as parameters with the MAKEENV, CREATEENV, or ADDPAGE UDCs must have been created or updated with Formation Design version 1.7 or later.

### Job Streams

You can include Formation UDCs in job streams and command files to build or modify environment files. The CIERROR JCW allows you to check these jobs and command files for parameter and compilation errors. If CIERROR is set to 0, the job completed successfully. If CIERROR is set to any value other than 0, an error was encountered.

## ADDFONT UDC

### Purpose

Add or replace a data font in an existing environment file. To add a font, provide an unused font number when running the UDC, or allow Formation to assign the next available font number. To modify a font, enter an existing font number. The current information replaces existing font characteristics.

### Syntax:

```
ADDFONT envfile, [num], [orient], face, size, [wt], [style], [symset]
```

### Parameters

<i>envfile</i>	The name of the existing environment file to be modified. This is a required parameter.
<i>num</i>	The font number. Enter a number between 0 and 31. If this number is already in use, the existing information is overwritten. If a number is not provided, Formation adds a font and assigns it the next available font number.
<i>orient</i>	The orientation of the font. Enter P for portrait, or L for landscape. The orientation should match that of the form on which this data font will print. The default is P.
<i>face</i>	The name of a typeface in the Formation font directory (e.g., Courier). This typeface should be available on all printers for which the environment file is compiled. This is a required parameter.
<i>size</i>	The point size of the font. This is a required parameter.
<i>wt</i>	The weight of the font. Enter 0 for regular or 3 (or B) for bold. Formation attempts to provide the best match for the requested weight. The default is 0.  Note that for special printer circumstances, you can enter any integer value between -7 and +7. Refer to your printer manual for special weight considerations.
<i>style</i>	The style of the font. Enter U to indicate an upright font, or I to indicate an italic font. The default is U.
<i>symset</i>	The symbol set of the font. Enter R for Roman-8, or A for US ASCII. The default is A. If not already resident in the printer, the Roman-8 symbol set requires more printer memory and download time than other fonts.

### Description

Fonts added to the environment file can be modified with the Environment Manager's **Character Set Control** screen. Refer to page 3-28 for more information.

ADDFONT parameters are position dependent. Commas must be placed between required parameters to indicate any unnamed optional parameter. Trailing optional parameters may be omitted. For example:

```
ADDFONT INVENV,,,UNIVERS,14,,I
```

Character sets in point sizes larger than 30 points cannot be used with the LaserJet Plus or LaserJet 500+. Any character which is larger than the maximum cell size for these printers (128 dots) will not be printed.

#### For HP 2680 and HP 2688 users only:

**2680/2688  
Users**

Due to limitations of HP 2680/88 printers, Formation assigns two font numbers for each font that requires Roman 8 characters. If you are specifying font numbers when adding a Roman 8 font (for a 2680/88 environment file), do not use consecutive font numbers. For example, if you assign font number 2, Formation also automatically creates font number 3, and the next valid number is 4. If you assign consecutive numbers, the Roman 8 character set is overwritten. In addition, if the font from the form file is Roman 8, the environment file already contains at least two fonts (normally 0 and 1).

The following table lists the fonts available for use with the ADDFONT UDC.

**Table 3-4: Available Fonts**

Typeface	Pitch	Point Size	Weight	Slant	Symbol
Courier	Fixed	Range; 4-14	0	U	A,R
Prestige	Fixed	Range; 4-12	0	U	A,R
Letter Gothic	Fixed	Range; 4-12	0	U	A,R
BC30F9S	Small	Range; 4-36	0	U	A
BC30F9M	Medium	Range; 4-36	0	U	A
BC30F9W	Wide	Range; 4-36	0	U	A
USPOST	Fixed	Range; 5-36	0	U	A
Univers	Proportional	Range; 4-24	0,3 (B)	U,I	A,R
CGTimes	Proportional	Range; 4-24	0,3 (B)	U,I	A,R

### Examples

1. Add a 10 point Courier font to environment file INVENV, and assign it font number 1:

```
ADDFONT INVENV,1,,COURIER,10
```

2. Add a landscape 13.5 point italic Univers font and assign it the next unused font number. Specify that the full Roman-8 extended character set is to be downloaded to the printer:

```
ADDFONT INVENV,,L,UNIVERS,13.5,,I,R
```

## ADDPAGE UDC

### Purpose

Add or replace a logical page in an existing environment file. To add a logical page to the environment file, provide an unused logical page number, or allow Formation to assign the next available logical page number. To modify a logical page, enter that page's number.

### Syntax

```
ADDPAGE envfile, formfile, [secondaryform], [page-num]
```

### Parameters

<i>envfile</i>	The name of the existing environment file to be modified. This parameter is required.
<i>formfile</i>	The name of a Formation Design form file. This is the primary form on this logical page. Page size and position is determined by information contained in the form file. This is a required parameter.
<i>secondaryform</i>	The name of a second Formation Design form file. This optional parameter provides logical information specific to this page (e.g., routing information).
<i>page-num</i>	The logical page number. This number defines the position of the current page relative to other pages in the form set. If the number entered is already in use, then the previously defined information is overwritten. If you do not provide a page number, Formation uses the next available logical page number.

### Description

You must create or update the form files named as parameters with the ADDPAGE UDC using Formation Design version 1.7 or later. Formation extracts from the form file all the environment file processing information such as the type of printers for which to compile the environment, form size and position, and the default data font. Forms created with earlier versions of the software can be loaded and then saved under the newer version. For more information about creating forms, read the *Formation Design User Guide*.

The *formfile* parameter defines the primary form in a form set, while the *secondaryform* parameter contains information specific to the current logical page. For example:

- The primary form may contain the common layout of a multi-part invoice, and the secondary form might contain routing information such as 'Customer Copy.'
- When printed together on the same page, this process creates the customer copy of the form.
- The same primary form might be assigned to several logical pages with different secondary forms such as 'Sales Copy' and 'Accounts Copy.'

ADDPAGE parameters are position dependent. Place commas between parameters to indicate any unnamed optional parameters. Trailing optional parameters may be omitted. For example:

```
ADDPAGE INVENV,TCFORM,,2
```

Logical pages added to the environment file can be modified with the Environment Manager **Logical Page Control** screen. For more information refer to "Logical Page Control" on page 3-34.

### Examples

1. Add a page containing an existing form called TCFORM as the next page in environment file INVENV:

```
ADDPAGE INVENV,TCFORM
```

2. Place a form called INVFORM, and a secondary form called SALEFORM, on logical page 2 of the INVENV environment file:

```
ADDPAGE INVENV,INVFORM,SALEFORM,2
```



## COMPILEENV UDC

### Purpose

Compile the specified environment file.

### Syntax

```
COMPILEENV envfile
```

### Description

Before printing the form and data to your printing you must have defined an environment file. The environment file must then be compiled. Once compiled, the environment file is ready for printing.

For more information about printing a compiled environment file, refer to Chapter 4, *Printing Options*.

### Parameters

*envfile*            The name of the existing environment file to be compiled. This is a required parameter.

### Example

Compile environment file INVENV:

```
COMPILEENV INVENV
```

## CREATEENV UDC

### Purpose

Create a new environment file. Form processing details (data font, logical page, and printer information) are extracted from the primary form file assigned to the environment file with this UDC. After creating an environment file, compile it with the COMPILEENV UDC.

### Syntax

```
CREATEENV envfile, formfile, [secondaryform], [page-num], [duplex]
```

### Parameters

- envfile* The environment file name. The name must meet standard MPE naming conventions. If the file already exists, an error message is displayed, and the command is aborted. This is a required parameter.
- formfile* The name of a Formation Design form file. This is the primary form on this logical page. The data font named in this form file is assigned font number 0, and the processing information for the form file is assigned to the named environment. This is a required parameter.
- secondaryform* The name of a second Formation Design form file to be included on the current logical page. This form provides information specific to this page (e.g., routing information). Positioning is determined by information in the form file. This parameter is optional.
- page-num* The logical page number to be used as the first logical page in the environment file. The default is 0.
- duplex* Define a duplex or simplex environment.
- Enter S to define this environment as simplex (pages are printed single-sided-vertical binding).
  - Enter D or V for duplex (double-sided printing), where the binding will run vertically (down the long edge of the page).
  - Enter H for duplex printing with horizontal binding (short edge of the page). The default is S. Refer to page 3-20 for more information on duplex printing.

This parameter only affects printers which support duplex printing.

### Description

You must create or update the form files named as parameters with the CREATEENV UDC using Formation Design version 1.7 or later. Formation extracts from the form file all the environment file processing information such as the type of printers for which to compile the environment, form size and position, and the default data font.

---

Forms created with earlier versions of the software can be loaded and then saved under the newer version. For more information about creating forms, read the *Formation Design User Guide*.

The `formfile` parameter defines the primary form in a form set, while the `secondaryform` parameter contains information specific to the current logical page.

For example:

- The primary form may contain the common layout of a multi-part invoice, and the secondary form might contain routing information such as 'Customer Copy.'
- When printed together on the same page, this creates the customer copy of the form.
- The same primary form might be assigned to several logical pages with different secondary forms such as 'Sales Copy' and 'Accounts Copy.'

CREATEENV parameters are position dependent. Place commas between parameters to indicate any unnamed optional parameter. Trailing optional parameters may be omitted.

For example:

```
CREATEENV INVENV, INFORM, , 2
```

### Examples

1. Create environment file INVENV, and assign form INVFORM as logical page 0:

```
CREATEENV INVENV, INVFORM
```

2. Create environment file INVENV, assign form INVFORM as logical page 2 (the first in the environment file), and assign a duplex option with vertical binding:

```
CREATEENV INVENV, INVFORM, , 2, D
```

## LISTENV UDC

### Purpose

Display a listing of environment file information. The listing can be displayed on your terminal or printed to a named printer.

### Syntax

```
LISTENV envfile, [device]
```

### Parameters

- |                |                                                                                                                                                                   |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>envfile</i> | The name of the environment file to be listed. This is a required parameter.                                                                                      |
| <i>device</i>  | The printer device number or device class on which the environment file information is to be listed. If this parameter is not included, the default is \$STDLIST. |

### Description

LISTENV provides the same output as the Environment Manager's **List Environment File (LE)** command.

### Examples

1. Display information from environment file INVENV to the terminal screen or job \$STDLIST output.  

```
LISTENV INVENV
```
2. Print information from environment file INVENV on device 25.  

```
LISTENV INVENV,25
```
3. Print information from environment file INVENV on any printer in device class LP.  

```
LISTENV INVENV,LP
```

## MAKEENV UDC

### Purpose

Create and compile an environment file. Form processing details (data font, logical page, and printer information) are extracted from the form file assigned to the environment.

MAKEENV combines the functionality of the CREATEENV and COMPILEENV UDCs. This UDC is most often used for simple, single part forms. If you are creating a more complex form set, it is more efficient to define the forms with CREATEENV and ADDPAGE, and then compile once with COMPILEENV.

### Syntax

```
MAKEENV envfile, formfile, [secondaryform], [duplex]
```

### Parameters

- |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>envfile</i>       | The environment file name. The name must meet standard MPE naming conventions. If this file already exists, an error message is displayed and the command is aborted. This is a required parameter.                                                                                                                                                                                                                                                                                                                                                               |
| <i>formfile</i>      | The name of a Formation Design form file. This is the primary form on logical page zero. The data font named in this form file is assigned font number 0, and the processing information from the form file is assigned to the named environment. This is a required parameter. To add additional logical pages, use the ADDPAGE UDC.                                                                                                                                                                                                                             |
| <i>secondaryform</i> | The name of a second Formation Design form file to be included on the current logical page. This optional parameter provides information specific to this page (e.g., routing information). Positioning is determined by information in the form file.                                                                                                                                                                                                                                                                                                            |
| <i>duplex</i>        | Define a duplex or simplex environment. <ul style="list-style-type: none"> <li>– Enter S to define this environment as simplex (pages are printed single-sided).</li> <li>– Enter D or V for duplex (double-sided printing), where the binding will run vertically (down the long edge of the page).</li> <li>– Enter H for duplex printing with horizontal binding (short edge of the page). The default is S. Refer to page 3-20 for more information on duplex printing.</li> </ul> <p>This parameter only affects printers which support duplex printing.</p> |

### Description

Form files as parameters to the MAKEENV UDC must be created or updated with Formation Design version 1.7 or later. Formation extracts from the form file all the environment file processing information such as the type of printers for which to compile the environment, form size and position, and the default data font.

Forms created with an earlier version of the software can be loaded and saved under the newer version. For more information about creating forms, read the *Formation Design User Guide*.

The `formfile` parameter defines the primary form in a form set, while the `secondaryform` parameter contains information specific to the current logical page. For example:

- The primary form may contain the common layout of a multi-part invoice, and the secondary form might contain routing information such as 'Customer Copy.'
- When printed together on the same page, this process creates the customer copy of the form.
- The same primary form might be assigned to several logical pages with different secondary forms such as 'Sales Copy' and 'Accounts Copy.'

MAKEENV parameters are position dependent. Commas must be placed between parameters to indicate any unnamed optional parameter. Trailing optional parameters may be omitted. For example:

```
MAKEENV INVENV, INFORM, , 2
```

MAKEENV automatically assigns the number 0 to the first logical page in the environment. Logical page information can be modified in the Environment Manager **Logical Page Control** screen. Refer to page 3-34 for more information.

### Examples

1. Create environment file INVENV using a form file called INVFORM:

```
MAKEENV INVENV, INVFORM
```

2. Create environment file INVENV using form file INVFORM, and specify duplex printing with vertical binding:

```
MAKEENV INVENV, INVFORM, , D
```

## Creating Environment Files with the Environment Manager

The Environment Manager allows you to create, fine tune, and compile environment files using a set of V-PLUS screens. With these screens you can define:

- Physical page information
- Data fonts
- Where the logical pages print on any given physical page

To produce a Formation environment file using the Environment Manager,

1. Run the Environment Manager program.
2. Enter a name for the environment file on the **Environment File Menu**.

Once a name is assigned to the file, Formation displays the **Main Menu**. From the **Main Menu**, you can define:

- Data fonts using the **Character Set Control** screen.
- Physical page information using the **Physical Page Control** screen.
- Add any logical pages and their forms using the **Logical Page Control** screens.

If you choose not to define these items, the Environment Manager provides default values.

3. Compile the environment file. You must have write access to an environment file in order to change or compile it.

### Running the Environment Manager

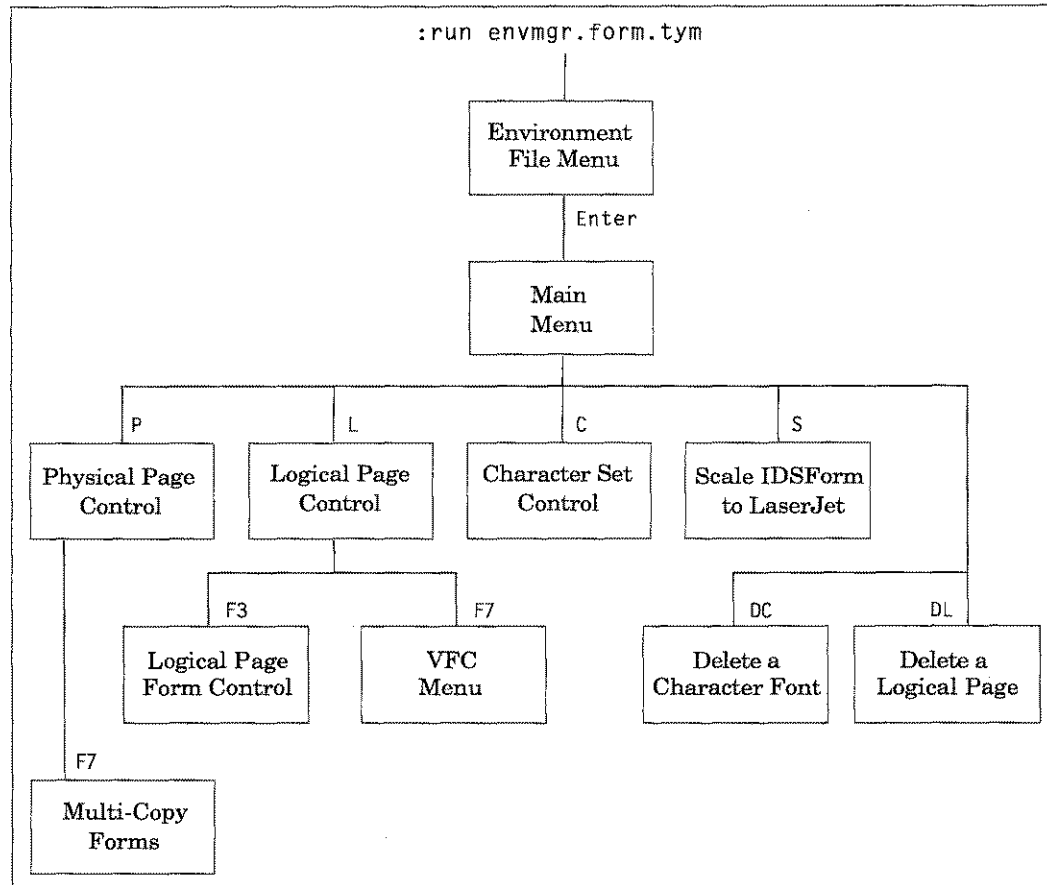
You can run the Environment Manager on any block mode terminal. To run the Environment Manager in interactive mode, enter the following command:

```
:RUN ENVMGR.FORM.TYM
```

Formation displays the **Environment File Menu**. From this menu you can select an environment file and display the Environment Manager **Main Menu**.

### Environment Manager Menus

The Environment Manager consists of a series of screens used to define environment file information. You can gain access to all Environment Manager screens by entering the appropriate keystroke(s) from the **Main Menu**. You can display any file for which you have read access. The following diagram illustrates a map of these screens.



The following table describes how to work with the Environment Manager menus.

To:	Press the:
Return to the <b>Main Menu</b> from any screen	8 function key
Exit the Environment Manager	8 key from any screen until the system displays the MPE colon prompt
Process the information you enter in the Environment Manager screen fields	e key or select the appropriate function key
Advance through the fields on any screen	t key
Move backward through the fields	s and t keys simultaneously
Edit information in any of the fields	Backspace key to clear out the entry in the field and enter the correct information



## Environment File Menu

The **Environment File Menu** is the first screen the Environment Manager displays. Use this screen to name the environment file being created or edited.

The screenshot shows a terminal window titled "Environment File Menu". The main area contains the text "Environment file name:" followed by a text input field containing "MPEFILE". At the bottom of the window, there are four buttons: "TermType Utility", "Refresh", and "Exit".

### Description

When Formation's Environment Manager displays this screen, enter an environment file name.

- To create a new environment file, enter a unique name and then press the `e` key twice. The Environment Manager displays the **Main Menu**.
- To edit an existing environment file, enter an existing file name and then press the `e` key once. The Environment Manager displays the **Main Menu**.

### Field Definitions

Environment file name

The name of a new or existing environment file. The name must meet standard MPE naming conventions. You may enter a fully-qualified file name, with an NS/3000 node name. You must have write access to a file in order to change or compile it, though you may examine any file to which you have read access.

### Function Keys

Key	Name	Description
1	Term Type Utility	Access the <b>Terminal Type</b> utility.
4	Refresh	Redisplay the menu and data from the last saved state.
8	Exit	Exit the Environment Manager and return to the MPE colon prompt.

## Main Menu

The **Main Menu** allows you to access Environment Manager screens, compile an environment file, or print a listing of the current environment file specifications.

The screenshot shows a terminal window titled "Main Menu" with "Environment file: NEWFILE" in the top right corner. The main area contains a list of options, each with a character in a box:
 

- P** Physical Page Control
- S** Scale IDSForm to LaserJet
- C** Character Set Control
- L** Logical Page Control
- DC** Delete Character Font
- DL** Delete Logical Page
- LE** List Environment File
- X** Compile
- E** Exit

 To the right of these options are several input fields:
 

- "LaserJet/HP 5000 Model:" with "LJ" entered.
- "HP 2680 Model:" (empty).
- "Optional Number:" (empty).
- "-or- Name:" (empty).
- "PCL printer memory (kb):" (empty).
- "COMAREA required (bytes):" (empty).
- "Comment:" followed by a large blacked-out area.

 At the bottom, there are two rows of buttons. The first row includes a "Refresh" button. The second row includes an "Env File Menu" button."/>

### Description

To access the **Main Menu**, enter a file name and press the **e** key at the **Environment File Menu**.

To select an option from the **Main Menu**:

1. Enter the appropriate character(s) in the selection box displayed in the top, left corner of the screen.
2. Supply a printer name in either the **LaserJet/HP 5000 Model** field or the **HP 2680 Model** field if one is not displayed. To create a dual environment, supply names in both fields.
3. If you are deleting a character font or logical page, enter an optional name or number for the font or logical page to be deleted in the **Optional Number** or **Name** fields.
4. Press the **e** key to process your selection.

### Dual Environments

When creating a dual environment by supplying names in both printer fields (**LaserJet/HP 5000 Model** and **HP 2680 Model**), Formation allows you to create environment files that can be used interchangeably on both HP2680/88 printers and any LaserJet family printers. You must specify all device-dependent measurements in terms of HP2680/88 printers. The Environment Manager automatically translates measurements internally when required.

When you create a dual environment, be aware that some of the items in subsequent menus are device-dependent. This means they may require different values in the fields, depending on the device. For example, a line height specified in dots is device-dependent, since the HP 2680 and LaserJets have different dot sizes.

## Field Definitions

- Selection Box** Enter menu selections in this field. The selection box is the field displayed in the upper left corner of the screen. The available menu selections are:
- P Display the **Physical Page Control** screen. This screen allows you to set physical page parameters such as number of copies and paper size.
  - S Display the **Scale IDSForm to LaserJet** screen. This screen scales IDS forms and PCELL fonts for printing on a LaserJet.
  - C Display the **Character Set Control** screen. This screen allows you to define data fonts.
  - L Display the **Logical Page Control** screen. This screen allows you to create logical page definitions and specify the actual forms to be used. The **Logical Page Form Control** screen and the **VFC Menu** are accessed from this menu.
  - DC Display the **Delete a Character Font** screen. This screen allows you to delete a previously specified character font from the environment file.
  - DL Display the **Delete a Logical Page** screen. This screen allows you to delete a previously specified logical page from the environment file.
  - LE Print a listing of all character fonts and logical pages used in the environment file. The listing is printed using formal designator ENVLIST. The default device is LP.
  - X Compile the environment file. When you select this function, the Environment Manager clears the screen and displays progress messages as each font and form is compiled for each of the specified printers.
  - E Exit the Environment Manager, and return to the MPE colon prompt.
- LaserJet/HP  
5000 Model** Enter a printer model value. The value entered must include all printers to which you intend to print the form.

Entering a value that does not include certain printers may restrict the form from being printed properly on those printers. However, when you enter a value that includes a large number of printers, the available features are reduced to accommodate the least capable printer. Therefore, it is recommended that you use a value that will allow you the most flexibility when printing your form.

The allowed printer values are listed in the following table:

Table 3-5: Printer Model Values

Value	Prints to
LJP1us	LaserJet Plus and 500+.
LJSII	LaserJet Series II.
LJIII	Any LaserJet III or 4.
LJ4	Any LaserJet 4.
LJET	LaserJet Series II or above and the HP 5000. Because extra code is required to support older LaserJets, forms containing TIFF images print more slowly on all printers. Compile for LJ2000 if you don't have a LaserJet Series II.
LJ2000	The LaserJet IID, IIP, all 2000s, all IIIs, 4s and the HP 5000. These provide better performance than LJET. (Does not include the LaserJet 500+, Plus or II.)
F100	HP 5000.
F100LP	HP 5000 when multiple logical pages are printed on a single physical page without using LPRINT or the intrinsics, or when recompiling any 2680/88 environment file supplied by HP for use with the F100. Environment files compiled for F100LP can only be printed using a file equation, not with LPRINT or the intrinsics. A dual environment file must be compiled with the 2680A as the system printer.

Character sets in point sizes larger than 30 points cannot be used with the LaserJet Plus or LaserJet 500+. Any character which is larger than the maximum cell size for these printers (128 dots) will not be printed.

**Note:** PCL printers other than the LaserJet series are not supported by Formation. Some printers which emulate LaserJet series printers may work, but Unison Software does not guarantee proper operation or attempt to resolve problems with printers that are not Hewlett-Packard printers.

**HP2680 Model** Enter 2680A for HP2680 printers, or 2688A for HP 2688 printers. Only users who have purchased the 2680 option have access to this field. Refer to appendix F for more information on creating environment files for these printers.

#### Optional Number

Enter the number of a character font or logical page to be accessed or deleted. The Environment Manager displays the corresponding **Character Set Control**, **Logical Page Control**, **Delete Character Font** or **Delete Logical Page** screen.

**Name** Enter the name of a character font to be deleted. Formation displays the **Delete Character Font** screen.

**PCL printer memory (kb)**

This field displays the approximate amount of memory, in kilobytes, the environment file uses when stored in the printer. This value is provided by the Environment Manager after the environment file is successfully compiled.

To actually print the environment can require more than the amount displayed in this field. This is particularly true for environment files compiled for the LaserJet Plus, which uses the least efficient representation of forms.

LaserJet family printers use about 115 kb of memory for internal functions: the rest is available for use by environment files as static and dynamic memory. For example, a standard LaserJet Plus or LaserJet Series II contains 512 kb of memory. The printer takes about 115 kb, leaving user-available memory of about 395 kb.

**COMAREA required (bytes)**

The size, in bytes, of the comArea array required by Formation intrinsics when they are used to access the current environment. Refer to chapter 6 for more information on comArea.

**Comment** Enter descriptive text to describe the environment file. This field is optional.

**Function Keys**

Key	Name	Description
4	Refresh	Redisplay the menu and data from the last saved state.
8	Env File Menu	Return to the <b>Environment File Menu</b> .
e	Enter	You must press <b>Enter</b> to record changes.

## Physical Page Control

The **Physical Page Control** screen allows you to assign an identification number to the environment, and to define page attributes in the environment file. These include such things as page size, number of copies and duplexing options.

```

Physical Page Control                               Environment file: NEMFILE
-----
0 Environment ID                                  Y Use printer resident fonts? Y/N
1 Number of copies                               Y Multi-copy forms? Y/N

11.25 Width
11.25 Length (in direction of paper motion)
-OR-
A4 Paper Size (A3, A4, L, LL, B)

I Units for page dimensions (I - inches, C - centimeters)

1 Initial primary character font number
-OR- 1 name

1 Initial secondary character font number
-OR- 1 name

E Default measurement system (E - English, M - Metric)

Duplexing: H (H - Horizontal; V - Vertical) Offset:
Default Form File: NEMFILE

Refresh Null Copy Main Menu
  
```

### Description

The number you specify in the **Number of Copies** field indicates how many copies of each physical page will print.

The **font number** fields allow you to assign the primary and secondary character fonts to be used in the current environment.

The **Paper Size** field allows you to designate the size of the paper on which the form will print. You can specify **Size** by entering one of the following:

- One of the supplied paper size codes
- The actual dimensions of the paper

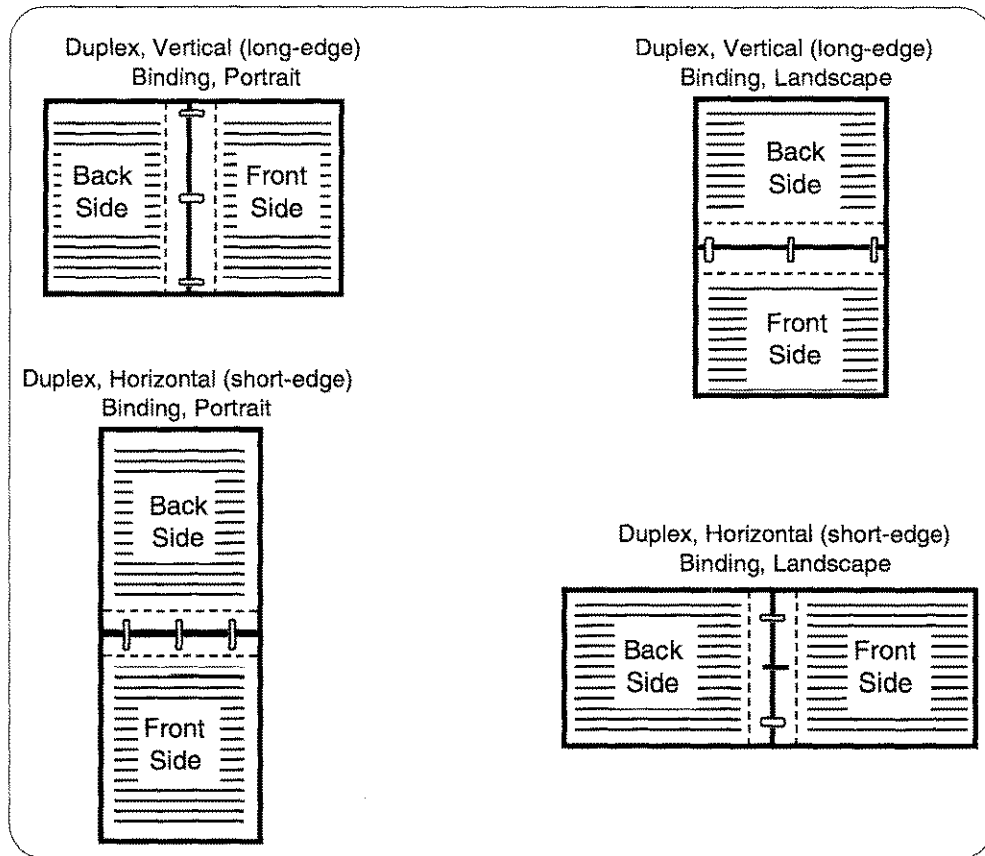
If the size you enter is not one of the standard sizes, the printer selects the smallest standard size that is at least as large as the size specified. For example, if you specify a page length of 11.25 inches, the printer automatically requests A4 paper, since A4 is the smallest paper with a length of at least 11.25 inches.

- On a LaserJet printer, if the requested paper size is loaded in the printer, it is selected automatically. If the requested paper size is not available, the paper size is requested from the operator.
- On the HP 5000, paper size errors are ignored by the printer and the current paper stock is used.

Duplexing options are also defined in the **Physical Page Control** screen. A simplex page is single sided. Environments destined for printers configured to print in duplex mode can be configured for simplex printing by entering S in the **Duplexing** field. A duplex page has data printed on both sides. Duplexing options are available for both horizontal and vertical binding.

The following diagram illustrates the duplex options available with Formation.

**Figure 3-1. Formation Duplex Options**



Use the following procedure to assign page size.

1. Enter a P in the selection box on the **Main Menu** and press the **e** key, to access the **Physical Page Control** screen. The Environment Manager displays the screen.
2. Fill in the fields.
3. Press the **e** key to process the information.
4. Compile the environment file to activate any changes made on this screen.

### Field Definitions

**Environment ID** Enter a number between 0 and 31 to define an identification number for the environment file. Assigning a value of 0 defines a temporary environment. Temporary environments remain in the printer until a reset escape sequence removes them from the printer's memory. Assigning a value between 1 and 31 allows the environment file to be downloaded as a permanent file.

Do not assign the same ID number to different environments. The environment ID is used by the DOWNLENV utility when pre-downloading files to the printer. See "DOWNLENV Utility" on page 3-47 for more information.

Number of copies

Enter the number of copies to be printed. Form sets that have more than one page are printed in a decollated fashion. This means that all copies of page 1 will print first, followed by all copies of page 2, and so on. The following considerations apply to this field:

- The maximum number of copies you can specified for LaserJet printers is 99.
- The maximum number of copies you can specified for HP2680 and HP2688 printers is 999.
- In dual environments, enter the value desired for the HP2680/88 (if greater than 99), and the environment manager will adjust the LaserJet value to 99.

When using a serial printer, Formation is faster than the MPE spooler to print multiple copies, especially if the document contains graphics or uses a very large environment file.

Use printer resident fonts? Y/N

Enter N in this field if you do not want the printer to use any printer resident fonts when printing the form and/or data. In all other instances, enter Y. The default is Y.

For more information on printer resident fonts, refer to page 5-5.

Multi-copy forms? Y/N

Enter Y in this field if the environment file is being compiled for a multi-copy (also known as a multi-part form) form on either the HP 2680 or the HP 2688. Enter N in all other instances. The default is N.

Width

If you are not using a standard paper size (as defined in Table 3-6: "Paper Sizes," on page 3-23) enter the width of your paper. The width should be in portrait orientation for LaserJet Printers, or in the direction of paper motion for HP2680 printers. The width must be defined in the unit of measure specified in the *Units for page dimensions* field.

Length

If you are not using a standard paper size, enter the length of your paper. The length should be in portrait orientation for LaserJet printers, or in the direction of paper motion for HP2680 printers. The length must be defined in the unit of measure specified in the *Units for page dimensions* field.

When entering paper size dimensions (width and length), the printer actually selects the smallest standard size that is closest in size (without going under) to the dimensions you specified.

Paper Size

Enter a standard paper size code. When Formation tries to locate the specified paper size in the printer, it looks first to the default bin. If paper of the correct size is not located in this bin, Formation tries to find the correct paper in another bin. Refer to page 3-37 for information on bin switching.



The following table lists the Laserjet codes for standard paper sizes.

**Table 3-6: Paper Sizes**

Letter	L	8.5 x 11 in.	8.0 x 10.5 in.
Legal	LL	8.5 x 14.0 in.	8.0 x 13.5 in.
Ledger	B	11 x 17 in.	10.5 x 16.5 in.
Executive	EX	7.25 x 10.5 in.	6.75 x 10.0 in.
A4	A4	210 x 297 mm	197 x 284 mm
A3	A3	297 x 420 mm	284 x 407 mm

Units for page dimensions (I - inches, C - centimeters)

Enter I to specify page dimensions in inches, or C to specify page dimensions in centimeters. Dimensions are rounded to the nearest 1/4 inch when used by the HP 2680 and HP 2688.

Initial primary character font number -OR- name

Enter the name or number of the primary character font. The specified font is used to print form data. Specify the name and number you assign to the font in the **Character Set Control** screen.

The data automatically starts printing in the primary font and will be used until a shift-out is encountered. To return to the primary font, issue a shift-in control character.

Initial secondary character font number -OR- name

Enter the name or number of a character font to be used as the secondary font. This font prints form data when a shift-out escape sequence is encountered.

Default measurement system (E-English, M-Metric)

Enter E to define environment measurements in inches. Enter M to define them in metric units of measure (millimeters, etc.). The unit type specified is then used as a default in other Environment Manager screens when defining dimensions and distances.

Duplexing

Enter one of the following values in this field:

- S to define simplex
- D for duplex printing
- H to define a duplex page with horizontal binding
- V to define a duplex page with vertical binding

Offset

Enter the number of inches (or millimeters) the printer is to move the page image right or left (vertical binding) or up and down (horizontal binding). Specify a positive value to move the image on the front of a page to the right or down. Specify a negative value to move the image left or up. The printer will move in the opposite direction on the back side of the paper, leaving room for binding both the front and the back.

When using the offset field, remember to account for the size of the logical page.

**Default Form File**

Enter the name of a default form file if you are using an IDS form. This file will be used whenever you do not specify a form file on the **Logical Page Forms** screen. If you are not using an IDS form, leave this field blank.

**Function Keys**

Key	Name	Description
4	Refresh	Redisplay the menu and data from the last saved state.
7	Multi Copy	Display the <b>Multi-Copy Forms</b> screen. This screen is for 2680 users only.
8	Main Menu	Return to the <b>Main Menu</b> .
e	Enter	You must press <b>Enter</b> to record changes.

## Multi-Copy Forms

2280/2688  
Users

The **Multi-Copy Forms** screen is used to define multi-part forms for HP 2680/88 users. Laserjet users can implement multi-copy (multi-part) forms using Formation's LPRINT utility.

Multi-copy Forms Environment file: NEWFILE

Logical page that will be used to write to these forms

Form	Form File (if not default)	Copy	Scale? Y/N
		Copy 1	Y
		Copy 2	
		Copy 3	
		Copy 4	
		Copy 5	
		Copy 6	
		Copy 7	
		Copy 8	

Refresh Cancel Main Menu

### Description

For each form part (individual page in the multi-part form), you can specify up to two forms.

- The first form is usually your primary form containing elements which are identical from one copy to the next.
- The secondary form(s) contains variable information that is different for each page.

For example, the first form might contain the main portion of an invoice and the secondary form might contain routing information. Formation overlays the second form on the first, centering each form on the associated logical page. To ensure proper placement, make sure both forms are the same size and shape.

To access the **Multi-Copy Forms** screen, press the 7 key on the **Physical Page Control** screen.

### Field Definitions

Logical page that will be used to write to these forms

Use this optional field to specify the ID number of the logical page used to write data to the forms. The logical page defines the position where data is printed, as well as the data's orientation. Data from all subsequent forms is rotated to match that of the logical page. If you specify scaling, forms are scaled to fit the logical page.

If you do not specify a logical page, the logical page with the lowest ID number in the environment is used. If no logical pages have been defined, the laser printer's default logical page is used.

**Form** Use this field only if the form was created with Hewlett-Packard's IDSSForm software. It is used to specify the name of the form within the file indicated in the next (**Form File**) field. Since each form created with Formation Design resides in a separate file, this field is not used if your form was created with Formation Design. Specify both the base form and an overlay for each copy.

**Form File** Use this field to specify a fully-qualified MPE name for the form file. Include a lockword, if needed. If no file is specified, the default form file specified on the **Physical Page Control** screen is assumed. The form specified here is compiled into the environment file.

**Scale?** To scale your forms to fit the logical page specified by the ID number in the **Logical page** field, type a Y here. All forms are scaled to fit that logical page, and all printing is restricted to the printing area it defines. To use scaling, you must specify a logical page.

Lines and boxes are scaled so that all elements fit within the logical page defined. Fonts are scaled by selecting from the sizes available in the font defined in your form. If the smallest size is too large for the reduced form, type will extend beyond the boundaries of the logical page.

**Function Keys**

Key	Name	Description
4	Refresh	Redisplay the menu and data from the last saved state.
7	Cancel	Return to the <b>Physical Page Control</b> screen without processing any of the current entries on the <b>Multi-Copy Forms</b> screen.
8	Main Menu	Return to the <b>Main Menu</b> .
e	Enter	You must press <b>Enter</b> to record changes.

## Scale IDSForm to LaserJet

2680  
Users

The **Scale IDSForm to LaserJet** screen is used to scale forms created for an HP 2680 so they will print properly on a LaserJet.

### Description

While the HP 2680 printers print to the edges of the page, the print region on LaserJet printers is limited to 8" by 10.5". To print HP 2680 forms on a LaserJet without clipping, you can scale IDS forms during recompilation by 92% or 96% of the original size.

1. To access the **Scale IDSForm to LaserJet** screen, enter S at the **Main Menu**.
2. Enter a scaling factor in the **Percentage by which to scale** field. Use the following table to determine the scaling factor to enter.

Table 3-7: IDSForm Scaling

Scaling	Used When...
92	The original form prints edge-to-edge in either direction.
96	There is at least 1/8" margin all the way around the original form.
100	You need to return the environment file to its original unscaled form.

3. Press the **e** key.

### Function Keys

Key	Name	Description
4	Refresh	Redisplay the menu and data from the last saved state.
8	Main Menu	Return to the <b>Main Menu</b> .
e	Enter	You must press the Enter key to record changes.

## Character Set Control

The **Character Set Control** screen allows you to define data fonts to be used with the current environment. These fonts can be PCL fonts, fonts from PCELL files, internal fonts or cartridge fonts. The fonts defined in the **Character Set Control** screen are included as part of the environment file, and are automatically downloaded to the printer when necessary.

```

Character Set Control                               Environment file: NEWFILE

Character Font:
Number 0
Local Name ██████████ (user defined)
File ██████████
-OR- Name UNIVERS Weight N Style J
Download ██████████ A - Always; N - Never.

Orientation
P
L - Landscape
P - Portrait
RL - Reverse Landscape
RP - Reverse Portrait

Point size 10 -OR-
Cell height in dots -OR-
Height of Capital letters
Units: I - in
M - mm

Match size
N - Nearest
S - Smaller
L - Larger

ASCII Character code: ██████ Lowest ██████ Highest
-----
Proportional spacing: Y
--System Printer--
Point   Ht of
Size    Capitals
Actual size to be used ..... 10      0.14444
Next Larger size available .....
Next smaller size available .....

----- LaserJet -----
Point   Ht of
Size    Capitals
Actual size to be used ..... 9.95999 0.14333
Next Larger size available .....
Next smaller size available .....

First Font  Last Font  Refresh  Prev Font  Next Font  Main Menu
    
```

### Description

To define a font, assign it a name and number. The name and number are used to reference the font in the **Physical Page Control** screen, or with the intrinsics. Choose the actual font by selecting either a PCL or PCELL file, or by entering a font name from the font directory and an optional weight and style.

You can enter the size of the font in any of the following measurements:

- Points
- Dots
- Height of capital letters

PCELL files can contain several sizes of fonts for different devices. The Environment Manager automatically selects the correct font based on the specifications provided in the **Character Set Control** screen. If the Environment Manager cannot match the size you request, it selects an alternate based on the match option you specify.

If you are constructing a dual environment file, the Environment Manager tries to find a suitable match for both the HP 2680/88 and the LaserJet. If no suitable LaserJet font can be found, the system scales the 2680 font.

For information about advanced uses of fonts and bar codes, refer to Chapter 5, *Fonts and Graphics*.

You can define a maximum of 32 fonts in any environment. Use the following procedure to define a font:

1. Access the **Character Set Control** screen by entering a C in the selection box on the **Main Menu**, and then press the  $\epsilon$  key.
2. When the Environment Manager displays the screen, fill in the fields to define a font, and then press the  $\epsilon$  key to process the information.
3. To define the next font, press the 6 (Next Font) key. The Environment Manager displays the new blank font screen.  
 To scroll forward through the previously defined fonts, use the 6 (Next Font) key. To scroll backward, use the 5 (Prev Font) key.
4. After defining and updating fonts, compile the environment file to activate any changes made on this screen.

Once a font is defined in the environment, you can easily change its characteristics by maintaining the font's name and number, and adjusting the characteristics using the **Character Set Control** screen.

It is not necessary to specify the fonts included in the actual form on the **Character Set Control** screen. These fonts are automatically included by the Environment Manager.

**Field Definitions**

**Number** Enter a font number. When the Environment Manager displays the **Character Set Control** screen, the screen displays a value in this field. The number is either the current value of a previously defined font, or a default value supplied by the Environment Manager for a new font. To define a number for a new font, leave the default value, or enter a new number. To modify an existing font, enter a previously defined number.

**Orientation** Enter a code to define the orientation of the font. The following table lists the allowed codes.


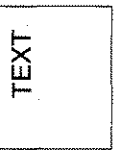


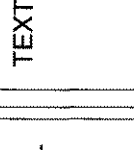
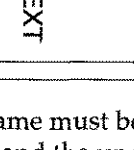
**Table 3-8: Font Orientation Codes**

Code	Definition
L	Landscape - The font prints upright in the direction of the long edge of the paper.
P	Portrait - The font prints upright in the direction of the short edge of the paper.
RL	Reverse Landscape - The font prints upright in the direction of the short edge of the paper. This results in the font being rotated from the rest of the form (printing up and down the page instead of across). This value can only be specified for rotated fonts on the LaserJet III, LaserJet 4, and the HP 2680.
RP	Reverse Portrait - The font prints upright in the direction of the long edge of the paper. This results in the font being rotated from the rest of the form (printing up and down the page instead of across). This value can only be specified for rotated fonts on the LaserJet III, LaserJet 4, and the HP 2680.

Rotated text is printed using the intrinsics. Refer to Chapter 6, *Intrinsics* for more information.

The following table illustrates how to define font orientation.

**Table 3-9: How to Define Font Orientation**

Form Orientation	Text To Go...	Font Orientation
Portrait	Across 	P (portrait)
	Up 	L (landscape)
	Down 	RL (reverse landscape)
Landscape	Across 	L (landscape)
	Up 	P (portrait)
	Down 	RP (reverse portrait)

- Local Name      Enter a font name. The name must begin with a letter, and can contain letters, numbers, and the underscore character. The name entered in this field is the name used to reference this font.
  
- File              Enter the fully-qualified name of a PCL or PCELL file where the font resides (or you can enter a font name in the **Name** field).
  
- Name             Enter the name of the font as defined in the font directory (e.g., univers, courier, CG Times). This field is required if the **File** field is empty.  
  
 When using Roman 8 (8 bit) data fonts, you must enter the name of a file pair in the font directory (one file for character codes 32 through 127, and the corresponding extension for character codes 128 through 255).



- 
- Weight** Enter the weight of the font, or leave the field blank to default to the standard character weight. The allowed values are: B for bold, N for normal, or a value between -7 and 7, with -7 being the lightest and 7 the darkest. If the weight you specify is not available, the Environment Manager provides the closest match. This field is only valid for fonts defined in the **Name** field.
- Style** Enter the style of the font. The allowed values are: I for italic and U for upright. The default is U. If the style you specify is not available, the Environment Manager provides the closest match. This field is only valid for fonts defined in the **Name** field.
- Download** Enter A to always download the font to the printer, or N to never download the font to the printer. You should only enter N if you are certain that the required fonts already reside in the printer. The default is A.
- Point Size** Enter a point value to define the size of the font in points. Formation attempts to match this point size as closely as possible. The actual size used is displayed in the **Actual size to be used** field at the bottom of the screen. Some font restrictions apply for LaserJet+ and 500 printers. For more information see page 3-18.
- Cell height** Define the font size in dots by entering the total height of the font cell. Character fonts specified by cell height must match exactly to be considered the same font. Defining font size by cell height is available for compatibility with IFS, but is not recommended for normal use. Differences in dots sizes occur for different printers. For more information refer to "Dual Environments" on page 3-16.
- Height of Capital letters**  
Define the font size in dots by entering the total height of the font from the baseline to the top of a capital letter. Formation attempts to match this size as closely as possible. The actual size used is displayed in the **Actual size to be used** field at the bottom of the screen.
- Match size** If the Environment Manager cannot find the exact font size you request, it selects a match based on the entry in this field. Enter an N, and the Environment Manager selects the nearest size to the one specified. Enter L and the Environment Manager selects the next largest size. Enter S and the Environment Manager selects the next smallest size. The default is N. The available sizes are displayed at the bottom of the screen.
- ASCII Character code**  
These fields are used to defined the range of the symbol set to be used. The values you enter in these fields limit the number of characters that must be downloaded to the printer. To get all possible characters (necessary for Roman 8 fonts) enter 32 as the lowest value and 255 as the highest value. These fields are only necessary for downloadable (PCL or PCELL) fonts, and will only work if the font directory specifies extensions for this font. Refer to your printer manual for actual character set values.

**Lowest** Enter a value for the lowest symbol set character you want downloaded to the printer. For Roman 8 fonts, enter 32.

**Highest** Enter a value for the highest symbol set character you want downloaded to the printer. For Roman 8 fonts, enter 255.

The bottom half of the **Character Set Control** screen displays information about the actual font selected by Formation.

### Proportional spacing

A Y is displayed in this field if the selected font is proportionally spaced (spacing based on the width of the characters). An N is displayed in this field if the selected font is monospaced (every character is the same width).

### Actual size to be used

The entry under the System Printer column displays the actual font size used for the HP 2680. The entry under the LaserJet column displays the actual font size used for LaserJet printers. The size displayed may be different from the one you specified if the Environment Manager is unable to find the specified size. If the Environment Manager is unable to find a LaserJet font that corresponds to the selected HP2680 font, an asterisk is displayed in the field on the right side of the screen.

### Next larger size available

If the Environment Manager selects a larger size font than the one specified in the size fields, information about the selected font size is displayed in this field.

### Next smaller size available

If the Environment Manager selects a smaller size than the one specified in the size fields, information about the selected font size is displayed in this field.

## Usage Notes

1. PCELL files have size limitations that make them unsuitable for very large fonts. (Character cells in PCELL files can be no more than 255 dots on a side.) Since these fonts cannot normally be used on the HP 2680 or HP 2688, the Environment Manager allows you to use them in their PCL form. A PCL font file contains only a single size of one font for a single device. The Environment Manager displays the size of the font contained in the file, but does not allow you to change it.
2. The following fonts are included with Formation as Roman 8 fonts:
  - Univers 4 - 24 point in regular, **bold**, *italic*, and **bold italic**
  - CG Times 4 - 24 point in regular, **bold**, *italic*, and **bold italic**
  - Courier 4 - 14 point, regular only
  - Letter Gothic 4 - 14 point, regular only
  - Prestige Elite 4 - 14 point, regular only

3. When modifying a font in an HP 2680/88 environment file created with UDCs, you might notice a -2 entry in the **Lowest ASCII Character code** field. This tells Formation to compile the current font for the Roman 8 special characters.

### Function Keys

Key	Name	Description
1	First Font	Display information about the lowest numbered font in the environment. If no fonts are defined, Formation displays a blank screen with a default value of 0 in the <b>Number</b> field.
2	Last Font	Display information about the highest numbered font defined in the environment. If no fonts are defined, Formation displays a blank screen with a default value of 0 in the <b>Number</b> field.
4	Refresh	Redisplay the menu and data from the last saved state.
5	Prev Font	Display information about the font that is numbered 1 less than the current font. For example, if font 3 is currently displayed, pressing this function key displays font 2. If the current font is 0, font 31 is displayed.
6	Next Font	Display information about the next sequentially numbered font. This key is used to define the next sequential font. For example, if the current font is 5, pressing 6 displays font 6. If font 31 is displayed, pressing 6 displays font 0.
8	Main Menu	Return to the <b>Main Menu</b> .
e	Enter	You must press <b>Enter</b> to record changes.

## Logical Page Control

The **Logical Page Control** screen allows you to define the logical pages in the current environment. There can be a maximum of 32 logical pages in an environment file.

Logical Page Control		Environment file: NEWFILE	
<input type="checkbox"/> 0	Logical page number	<input checked="" type="checkbox"/>	Initially active?
<input checked="" type="checkbox"/> 1	Orientation: L - Landscape, P - Portrait	<input type="checkbox"/>	RL - Reverse Landscape, RP - Reverse Portrait
<input checked="" type="checkbox"/>	Change Forms or UFC?	Units I-in, C-cm, M-mm, D-dots -OR-	
<input type="checkbox"/> 8	Width	<input type="checkbox"/> I	NC - number of characters
<input type="checkbox"/> 11	Height	<input type="checkbox"/> I	NL - number of lines
<input type="checkbox"/>	Distance from left	<input type="checkbox"/> I	
<input type="checkbox"/>	Distance from top	<input type="checkbox"/> I	
<input type="checkbox"/>	Overriding line spacing	<input type="checkbox"/> I	-OR-
<input type="checkbox"/>	Left margin	<input type="checkbox"/> I	LI - lines/inch, LC - lines/cm
<input type="checkbox"/>	Font number for UFC	<input type="checkbox"/>	NC - number of characters
<input type="checkbox"/>	Side (F - front, B - back)	<input type="checkbox"/>	name
<input type="checkbox"/>	Bin (U - upper, L - Lower, M - Manual, P - Paper deck, E - envelope)	<input type="checkbox"/>	Overriding Page Size
Actual spacing information:			
dots character width		43	dots line height
First Log Page	Last Log Page	Change Forms	Refresh
Prev Log Page	Next Log Page	Change UFC	Main Menu

### Description

Using the **Logical Page Control** screen, you can define such logical page attributes as: size, location on the physical page, line spacing, duplex options, paper bin, and the actual form to be used.

This screen is also used to access the **Logical Page Form Control** screen and the **VFC Menu** screen. The **Logical Page Form Control** screen is used to define the actual forms to be included on the logical page.

Each logical page in the environment file must be assigned a unique number. Once a number has been assigned to a logical page, it remains in the environment file until it is deleted with the **Delete Logical Page** option on the **Main Menu**.

Use the following procedure to change the number assigned to a logical page:

1. Enter a new number in the **Logical Page Number** field, and press **e**. The existing values on the screen are assigned to the new number.
2. You can then delete the old logical page definition using the **Delete Logical Page** option on the **Main Menu**.

Use the following procedure to define the logical page:

1. Enter an **L** in the selection box on the **Main Menu**, and then press **e**, to access the **Logical Page Control** screen.
2. When the screen appears, fill in the fields to define the logical page, and then press **e** to process the information.
3. Press the **F6** (Next Log Page) key to display a new **Logical Page Control** screen.
4. Compile the environment file to activate any changes made on this screen.

## Field Definitions

### Logical Page Number

The logical page number identifies the logical page to the printer and to the intrinsics. The logical page number is automatically assigned by the environment manager and should not be modified.

### Initially active?

Enter Y to flag the logical page as initially active. Enter N to control printing with the intrinsics. Refer to chapter 6 for more information on controlling printing with the intrinsics. The default is Y.

Only initially active pages are used by the LPRINT utility. At least one logical page per environment file must be flagged as initially active.

### Orientation

Enter a code to define the logical page orientation. When creating an environment file for the LaserJet only, the orientation is the *logical* orientation. The allowed codes are listed in the following table.

**Table 3-10: Logical Page Orientation Codes**

L	<input type="text" value="TEXT"/>	Landscape - The logical page prints in the direction of the long edge of the paper.
P	<input type="text" value="TEXT"/>	Portrait - The logical page prints in the direction of the short edge of the paper.
RL	<input type="text" value="TEXT"/>	Reverse Landscape - The logical page prints reversed in the direction of the long edge of the paper. Note that LaserJet printers do not support this option.
RP	<input type="text" value="TEXT"/>	Reverse Portrait - The logical page prints in the direction of the short edge of the paper. Note that LaserJet printers do not support this option.

**2680/2688  
Users**

#### For HP 2680/88 users only:

When using page sizes larger than standard letter size, the orientation codes will be reversed from their normal 2680 values. If you are defining a dual LaserJet/2680A environment, use the orientation that you would use for the 2680.

### Change Forms or VFC

Leave this field as N unless a Unison Software support representative advises you to change it.

### Width

Enter the width of the logical page. Define the unit of measurement in the corresponding **Units** field. Width is a required field.

Units I-in, C-cm, M-mm, D-dots, P-points

These units of measure can be used in the fields identified in the following table:

Unit		Use any unit in fields
I	Inches	Width
C	Centimeters	Height
M	Milimeters	
D	Dots. For dual environments, the dots default to the HP2680 or 2688 dot size.	Distance from left
		Distance from top
P	Points	Overriding line spacing
		Left margin

NC - number of characters

The number of characters that can fit across the logical page (characters per inch). The font defined in the **Font number for VFC Name** field determines the character width.

- If no font is specified in the **Font number for VFC Name** field, the primary font defined in the **Physical Page Control** screen is used.
- If no primary font is defined in the environment, the printer's default font is used.

This is an additional unit of measure for width.

Height

Enter the height of the logical page. Define the unit of measurement in the corresponding **Units** field. Height is a required field.

NL- number of lines

The number of lines that can fit down the page. The font defined in the **Font number for VFC Name** field determines the character height. If no font is specified in the **Font number for VFC Name** field, the primary font defined in the **Physical Page Control** screen is used. If no primary font is defined in the environment, the printer's default font is used. This is an additional unit of measure for height.

Distance from left

Enter the distance between the left edge of the paper to where the logical page should begin.

If this field is left blank, the logical page is centered horizontally on the page.

Distance from top

Enter the distance from the top edge of the paper to where the logical page should begin.

If this field is left blank, the logical page is centered vertically on the page.

- Overriding line spacing**  
Enter a value to define the spacing to be used between lines of data. Normally, line spacing is controlled by the character font. This option is useful for such things as forcing a specific number of lines per page and reducing or increasing the amount of white space between lines.
- LI - lines/inch, LC - lines/cm**  
The number of lines of data that can fit in an inch (or centimeter). When a value of LI (or LC) is entered, overriding line spacing is calculated in number of lines.
- Left Margin** Enter the amount of space to be left between the left edge of the logical page, and where the data begins.
- NC - number of characters**  
The number of characters from the left edge of the paper that the logical page should be placed. When a value of NC is entered, the left margin is determined using characters.
- Font number of VFC**  
Enter the number of the font Formation should use to calculate number of characters and number of lines. This field relates to any field using NL or NC as a measurement value, and is optional.
- Name** Enter the name of the font Formation should use to calculate number of characters and number of lines. This field relates to any field using NL or NC as a measurement value, and is optional. The name entered must be the name defined for the font in the **Character Set Control** screen.
- Do not use this field if you are providing a font number.
- Side** Enter F to print the logical page on the front side of a page, or B to print the logical page on the back side of a page. If you leave this field blank, the printer will use the next side if switching to this logical page would cause a page eject. If a value is specified, the printer ejects a page or changes sides if necessary to accommodate the request. This field is only valid if duplex printing is specified on the **Physical Page Control** screen.
- Overriding Page Size**  
Enter a page size for this logical page. The page sizes allowed are listed in Table 3-6: "Paper Sizes," on page 3-23. When the printer switches to this logical page, it will select the page size you requested. If you specify a page size, and the printer is currently using a different page size, the current physical page is ejected.
- This field has no effect on HP 2680 and HP 2688 printers.
- Bin** Enter the code for the paper bin which contains the paper on which the current logical page should print.

The following table lists the paper bin codes.

**Table 3-11: Paper Bin Codes**

<b>Code</b>	<b>Definition</b>
U	Upper bin.
L	Lower bin.
M	Manual feed (if supported by the printer). The LaserJet 2000 does not support manual feed and will ignore this specification.
P	High capacity paper deck (if available).
E	Envelope feeder (if available) or manual feed (if envelope feeder not available). This selection is ignored by the LaserJet 2000.
1 - 9	A digit from 1 to 9 indicating the bin selection number for the printer. This option is provided for new and non-HP laser printers.

The printer always ejects the current physical page if this option is specified on a logical page that becomes the current one. If both the paper size and the bin are specified, the paper size selection overrides the bin selection.

To use paper from the default bin, leave this field blank. The default bin contains the paper specified in the **Physical Page Control** screen.

**Actual spacing information:**

These fields define the logical spacing on this logical page. If they are left blank, Formation fills them in based on the specified data fonts. These information only fields are primarily useful for HP 2680 and HP 2688 printers.

**dots character width**

Display the actual character width in dots. Differences in dot sizes occur for different printers. See "Dual Environments" on page 3-16 for more information.

**dots line height**

Display the actual line spacing in dots.



## Function Keys

Key	Name	Description
1	First Log Page	Display the first logical page in the environment. If no logical pages are defined, Formation displays logical page zero (0).
2	Last Log Page	Display the last logical page defined in the environment. If no logical pages are defined, Formation displays logical page zero (0).
3	Change Forms	Display the <b>Logical Page Form Control</b> screen for this logical page.
4	Refresh	Redisplay the menu and data from the last saved state.
5	Prev Log Page	Display information about the logical page that is numbered 1 less than the current logical page. For example, if logical page 4 is currently displayed, pressing this function key displays logical page 3. If the current logical page is 0, logical page 31 is displayed.
6	Next Log Page	Display information about the next sequentially numbered logical page. This key is used to define the next sequential logical page. For example, if the current logical page is 5, pressing the F6 key displays logical page 6. If logical page 31 is displayed, pressing the F6 key displays logical page 0.
7	Change VFC	Display the <b>VFC Menu</b> for the current logical page.
8	Main Menu	Return to the <b>Main Menu</b> .
e	Enter	You must press <b>Enter</b> to record changes.

## Logical Page Form Control

The **Logical Page Form Control** screen identifies the forms to be used on the current logical page. The forms defined on this screen can be either Formation Design or IDS forms.

### Description

To access the **Logical Page Form Control** screen, press 3 (Change Forms) on the **Logical Page Control** screen.

You can specify up to two forms for each logical page. Enter a form name in the **Form** field (for IDSForms) or a form file name in the **Form File** field to define the form.

If you would like to scale the forms, enter a Y in the **Scale? Y/N** field. (For IDS Forms it is also necessary to define the form within the form file.)

There are two options for positioning your forms on the physical page. The **Position on Page** option is for automatic positioning. The manual positioning option allows you to precisely position each form.

Once form specifications are defined, press e to process your changes.

### Field Definitions

#### Change VFC specification

Leave this field as N unless a Unison Software support representative advises you to change it. (This field is supplied solely for compatibility with IDS users.)

#### Form 1 (or Form 2)

Enter the name of a form created in IDSSForm in these fields. Forms specific to a particular page (e.g., routing information) should be assigned to Form 2. If you are using Formation Design forms, leave these fields blank.

---

**Form File 1 (or Form File 2)**

Enter the name of a Formation Design form file. General forms used on other pages should be assigned to Form File 1, while forms specific to a particular page (e.g., routing information) should be assigned to Form File 2. If you are using a form created with IDSSForm, you may enter the name of the IDSSForm file if it is different from the default that you specified on the **Physical Page Control** screen.

**Scale? (Y/N)** Leave as N. This field is provided for compatibility with IFS only.

**Position on Page**

This option allows you to quickly determine the placement of the form on the logical page. Enter an X in the box corresponding to the position in which you would like to place the form. For example, to center the form on the logical page, place the X in the center box.

If you do not enter an X in any of the position boxes, the Environment Manager assumes that you want to position the form manually using the **Distance to move right** and **Distance to move down** fields. In this case, the starting position is assumed to be centered on the logical page.

**Distance to move right**

Enter the number of inches (or other measurement unit) to move the form right from the position on the page marked with an X. Enter a negative value to move the form left. Choose values carefully as this feature allows forms to be positioned off the logical page. For example, if you want the form positioned .5 inches from the left edge of the paper, place an X in the upper left **Position on page** box, and then specify .5 in the **Distance to move right** field.

If you have not specified a position on the page, the distance from right is calculated from the center of the logical page, and you cannot place the form off the edge of the logical page.

When calculating the position of the form on the page, the Environment Manager uses the *design size* of the form. In IDSSForm, the design size is specified when you create the form, and cannot be changed. In Formation Design, the size can be changed by selecting **Set Form Options** from the **Settings** menu.

**Distance to move down**

Enter the number of inches (or other measurement unit) to move the form down from the position on page marked with an X. Enter a negative value to move the form up. Choose values carefully as this feature allows forms to be positioned off the logical page. For example, if you want the form positioned .5 inches from the top of the paper, place an X in the upper left **Position on page** box, and then specify .5 in the **Distance to move down** field.

If you have not specified a position on the page, the distance to move down is calculated from the center of the logical page, and you cannot place the form off the edge of the logical page.

- Units
- Define the unit of measure to be used when moving the form on the logical page. The allowed values are:
- I Define movement in inches.
  - C Define movement in centimeters.
  - M Define movement in millimeters.
  - D Define movement in dots per inch (dpi). If you are creating a dual HP 2680/LaserJet environment file, displacements specified in dots are calculated using the HP 2680 dot size (180 dpi).

**Function Keys**

Key	Name	Description
1	First Log Page	Display the first logical page in the environment. If no logical pages are defined, Formation displays logical page zero (0).
2	Last Log Page	Display the last logical page defined in the environment. If no logical pages are defined, Formation displays logical page zero (0).
3	Change Log Page	Display the <b>Logical Page Control</b> screen.
4	Refresh	Redisplay the menu and data from the last saved state.
6	Log Page Menu	Display the <b>Logical Page Control</b> screen.
7	Change VFC	Display the <b>VFC Menu</b> for the current logical page.
8	Main Menu	Return to the <b>Main Menu</b> .
e	Enter	You must press <b>Enter</b> to record changes.

## VFC Menu

The **VFC Menu** enables HP 2680/88 users of IFS/3000 to specify vertical forms control parameters for the logical page. Each logical page can have a different VFC if required.

```

VFC Menu                               Environment file: NEWFILE

S Type of VFC.

S - Standard, automatically computed VFC
  █ lines before first line
  █ lines after last line

F - Special VFC from a file
  VFC File
  █

N - No VFC (space according to character height)
  █ Top margin
  █ Units I - IN, C - CM, M - MM, D - Dots

First Log Page  Last Log Page  Change Log Page  Refresh
Next Log Page  Prev Log Page  Change Forms  Main Menu
  
```

### Description

The Environment Manager allows the creation of three different kinds of VFCs: standard (automatically computed), custom (user defined), none (character height used for spacing). The VFC Menu is provided solely for backwards compatibility with IFS/3000 system use. The screen fields are not supported.

### Function Keys

Key	Name	Description
1	First Log Page	Display the first logical page in the environment. If no logical pages are defined, logical page 0 is displayed.
2	Last Log Page	Display the last logical page defined in the environment. If no logical pages are defined, logical page 0 is displayed.
3	Change Log Page	Display the <b>Logical Page Control</b> screen.
4	Refresh	Redisplay the menu and data from the last saved state.
6	Prev Log Page	Display the next lower numbered logical page, or logical page 31 is logical page 0 is currently displayed.
6	Next Log Page	Display the next higher numbered logical page, or logical page 0 is logical page 31 is currently displayed.
7	Change Forms	Display the <b>Logical Page Form Control</b> screen
8	Main Menu	Return to the <b>Main Menu</b> .
e	Enter	You must press Enter to record changes.

## Delete Character Font

The **Delete Character Font** screen deletes fonts from the current environment. The following diagram illustrates the **Delete Character Font** screen.

```

Delete Character Font                                Environment file: NEWFILE

Character Font:
Number 0
Local Name _____ (user defined)
File _____
-OR- Name UNIVERS Weight N Style U
Download _ A - Always; N - Never.

10 Point size -OR-
Cell height in dots -OR-
Height of Capital letters
I Units: I - in M - mm

Orientation
L - Landscape
P - Portrait
RL - Reverse Landscape
RP - Reverse Portrait

Match size
N - Nearest
S - Smaller
L - Larger

ASCII Character code: ___ Lowest ___ Highest
-----
Proportional spacing: Y
--System Printer--
Point Size Ht of Capitals
LaserJet
Point Size Ht of Capitals
Actual size to be used ..... 10 0.14444 9.95999 0.14333
Next Larger size available .....
Next smaller size available .....

Delete Font Refresh Cancel

```

### Description

1. To display the **Delete Character Font** screen, enter DC in the selection box on the **Main Menu**.
2. Enter the name or number of the font to be deleted in the optional **Number** or **Name** fields.
3. Press **e** and Formation displays the **Delete Character Font** screen with font information. To delete a font, press the **1** (Delete Font) key. Formation deletes the font from the environment file, and returns to the **Main Menu**. You can use the **8** (Cancel) key at any time to return to the **Main Menu** without deleting the font.

When you press the **1** key, the character font is immediately deleted from the control information stored in the environment file. However, the Environment Manager still downloads the font to the printer until the next time the file is compiled.

As always, after making any change to the environment file, you must recompile the environment file for the changes to take effect.

### Function Keys

Key	Name	Description
1	Delete Font	Delete the currently displayed font.
4	Refresh	Redisplay the menu and data from the last saved state.
8	Cancel	Return to the <b>Main Menu</b> without deleting the current font.
e	Enter	You must press Enter to record changes.

## Delete Logical Page

The **Delete Logical Page** screen deletes logical pages from the current environment.

Delete Logical Page		Environment file: NEWFILE	
<u>1</u> Logical page number	<u>Y</u> Initially active?		
<u>E</u> Orientation: L - Landscape, P - Portrait,	RL - Reverse Landscape, RP - Reverse Portrait		
<u>N</u> Change Forms or UFC?	Units I-in, C-cm, M-mm, D-dots		
<u>8</u> Width	<u>I</u> NC - number of characters		
<u>11</u> Height	<u>I</u> NL - number of lines		
Distance from left	<u>I</u>		
Distance from top	<u>I</u>		
Overriding line spacing	<u>I</u> LI - lines/inch, LC - lines/cm		
Left margin	<u>I</u> NC - number of characters		
Font number for UFC	-OR- name		
Side (F - front, B - back)	Overriding Page Size		
Bin (U - upper, L - Lower, M - Manual, P - Paper deck, E - envelope)			
Actual spacing information:			
dots character width	49 dots line height		
Delete Log Page	Refresh	Cancel	

### Description

- To display this screen, enter DL in the selection box on the **Main Menu**.
- Enter the name or number of the logical page to be deleted in the **Number** field.
- Press **e** to display the **Delete Logical Page** screen.

To delete a logical page, press the **1** (Delete Log Page) key. The logical page is deleted from the environment file, and the **Main Menu** is re-displayed.

Deleting a logical page also deletes any forms associated with the logical page from the environment (unless the forms are used on some other logical page within the same environment file.)

The logical page is not actually deleted from the environment until you recompile, although you will no longer be able to access the page through the Environment Manager's menus.

As always, after making any change to the environment file you must recompile the environment file.

### Function Keys

Key	Name	Description
1	Delete Log Page	Delete the currently displayed logical page.
4	Refresh	Redisplay the menu and data from the last saved state.
8	Cancel	Return to the <b>Main Menu</b> without deleting the current logical page.
e	Enter	You must press <b>Enter</b> to record changes.

## Compiling Environment Files

Formation environment files must be compiled to convert specifications and information into printer commands. This information is then stored in the environment file for future use.

Formation features three ways to compile an environment file:

- Select the **Compile** option on the Environment Manager **Main Menu**.
- Use the `COMPILEENV` UDC (described on page 3-7). UDCs are the recommended method for compiling an environment file at the command line.
- Run Environment Manager with an `INFO` string specifying the name of the environment file to be compiled.

### Compile Option from Main Menu

Environment files are compiled from the **Main Menu** by entering an `X` in the selection box and pressing `e`. The Environment Manager keeps track of the total number of bytes that are generated and displays the total in the **PCL printer memory** field when compilation is finished.

### COMPILEENV UDC

To compile an environment file using the `COMPILEENV` UDC, enter:

```
:COMPILEENV envfile
```

where *envfile* is the name of an existing environment file. Once compiled, the environment file is ready for printing and the system re-displays the colon prompt. You can also include this UDC in job streams and command files.

### Run the Environment Manager

If you have not installed the UDCs, you can compile an environment file using the Environment Manager `RUN` statement. To do so enter:

```
:RUN ENVMGR.FORM.TYM;INFO=filename
```

where *filename* is the name of an existing environment file.

After this command is run, the Environment Manager automatically compiles the named file, and re-displays the colon prompt.

### Notes on Environment File Compilation

Different LaserJet printers have different capability sets: each one uses a slightly different version of PCL. The printer type that you specify on the **Main Menu** is used during compilation to determine the most efficient sequence of PCL commands that can be generated. When you compile for the least capable printer supported by Formation, the LaserJet Plus, you will not be able to take advantage of some of the features available in others (e.g., the LaserJet 2000 (LJ 2000)). Conversely, if you compile for the LJ2000, you may be unable to use the resulting environment on the LaserJet Plus.

Laser printer capabilities are described in Table 3-5: "Printer Model Values," on page 3-18.



## DOWNLENV Utility

### Purpose

The DOWNLENV utility lets you to download environment files to the printer prior to use.

### Downloading Environment Files

Normally, environment files are automatically downloaded as they are used. In some instances, however, it is more convenient to download the file in advance. For example, downloading a very large environment file can be time consuming. If you use the file frequently, it makes sense to pre-download it, and avoid waiting for the download every time you use the file.

Before downloading a file, it must be compiled. In addition, the file should have a unique environment ID (1-31) as specified on the Physical Page screen of the Environment Manager. Environment files that are assigned duplicate ID's are overwritten by the last file that was downloaded with the ID number. You may need to coordinate environment IDs with your system manager if others are also using pre-downloaded environment files.

### Running DOWNLENV

To run DOWNLENV, enter the following:

```
:FILE ENVIR=environmentfile
:FILE LASERJET;DEV=laserjet
:RUN DOWNLENV.FORM.TYM
```

Where:

*environmentfile*

The name of an existing environment file to be downloaded to the printer.

*laserjet*

The name of the device to which the environment file should be downloaded.

The environment will be stored in the printer for future use.

### Using Pre-Downloaded Environments

To use a pre-downloaded environment, the JCW FMCART must be set to a value of 101. If FMCART is not set to 101, the environment file will still be downloaded to the printer automatically at print time, even though it was downloaded with DOWNLENV.

The system cannot check to see that the environment you are using has actually been downloaded. Setting FMCART to 101, and using an environment with a nonzero ID ensures that the process of loading the environment will always be skipped. If you receive a printout that appears to be formatted properly, but without forms and in the default font check the following:

- Have you set FMCART to 101?
- Was the environment file downloaded?

The downloaded environment file should be listed on a font printout. You can obtain a front printout using the control panels on the LaserJet printer.

**Note:** All downloaded environment files are deleted from the printer's memory when the printer is turned off.

## Supplied Environment Files

Formation comes equipped with several pre-defined environment files that can be used to print your application data. Each environment file name is eight characters in length.

The supplied environment file naming convention is as follows:

`R7sccc0E.ENV.TYM`

Where:

- R            The environment file is for a report.
- 7            The number of logical pages per printed sheet.
- s            The style of the form. The possible styles are B (bordered) and G (gray bar).
- ccc         The number of characters per line.
- o            The orientation of the printed sheet. The possible values are P (portrait) and L (landscape).
- E            The current file is an environment file.

For example, the following environment file name:

`R2G085LE.ENV.TYM`

defines a landscape report environment with 2 logical pages per printed sheet. The report prints gray bar with 85 characters per line.

The following table lists environment files in `ENV.TYM` that are 66 lines per page. The first column in the table illustrates the layout of the environment. The next column provides the environment file name. Column three lists the style of the output: bordered or gray bar. Finally, the number of characters per line, character font and point size are displayed.

**Table 3-12: 66 Line Environment File Layouts**

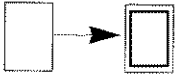
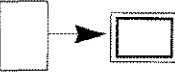
Layout	Environment File	Style	Chars/Line	Font	Point Size
1 portrait image per portrait sheet 	<code>R1B085PE.ENV.TYM</code>	Bordered	85	Courier	11.25
	<code>R1B104PE.ENV.TYM</code>	Bordered	104	Courier	9.25
	<code>R1B141PE.ENV.TYM</code>	Bordered	141	Courier	7.00
1 landscape image per landscape sheet 	<code>R1B143LE.ENV.TYM</code>	Bordered	143	Courier	9.00
	<code>R1B175LE.ENV.TYM</code>	Bordered	175	Courier	7.25
	<code>R1B242LE.ENV.TYM</code>	Bordered	242	Courier	5.25
	<code>R1B287LE.ENV.TYM</code>	Bordered	287	Courier	4.50

Table 3-12: 66 Line Environment File Layouts

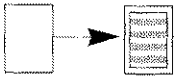
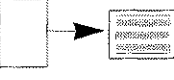
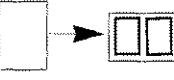
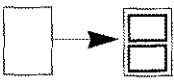
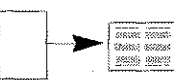

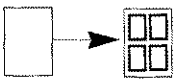
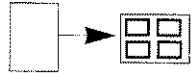
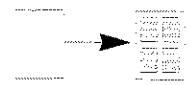
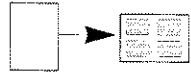
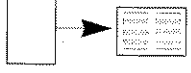
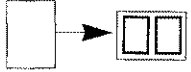
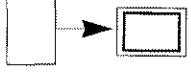
Layout	Environment File	Style	Chars/ Line	Font	Point Size
1 portrait image per portrait sheet. 	R1G085PE.ENV.TYM	Gray Bar	85	Courier	11.25
	R1G104PE.ENV.TYM	Gray Bar	104	Courier	9.25
	R1G141PE.ENV.TYM	Gray Bar	141	Courier	7.00
1 landscape image per landscape sheet 	R1G143LE.ENV.TYM	Gray Bar	143	Courier	9.00
	R1G175LE.ENV.TYM	Gray Bar	175	Courier	7.25
	R1G242LE.ENV.TYM	Gray Bar	242	Courier	5.25
	R1G287LE.ENV.TYM	Gray Bar	287	Courier	4.50
2 portrait images per landscape sheet 	R2B085LE.ENV.TYM	Bordered	85	Courier	7.25
	R2B102LE.ENV.TYM	Bordered	102	Courier	6.00
	R2B139LE.ENV.TYM	Bordered	139	Courier	4.50
2 landscape images per portrait sheet 	R2B140PE.ENV.TYM	Bordered	140	Courier	7.00
	R2B239PE.ENV.TYM	Bordered	239	Courier	4.00
2 portrait images per landscape sheet 	R2G085LE.ENV.TYM	Gray Bar	85	Courier	7.25
	R2G102LE.ENV.TYM	Gray Bar	102	Courier	6.00
	R2G139LE.ENV.TYM	Gray Bar	139	Courier	4.50
2 landscape images per portrait sheet 	R2G140PE.ENV.TYM	Gray Bar	140	Courier	7.00
	R2G239PE.ENV.TYM	Gray Bar	239	Courier	4.00
4 portrait images per portrait sheet 	R4B089PE.ENV.TYM	Bordered	89	Courier	5.25
	R4B106PE.ENV.TYM	Bordered	106	Courier	4.25

Table 3-12: 66 Line Environment File Layouts

Layout	Environment File	Style	Chars/Line	Font	Point Size
4 landscape images per landscape sheet 	R4B140LE.ENV.TYM	Bordered	140	Courier	4.50
4 portrait images per portrait sheet 	R4G089PE.ENV.TYM	Gray Bar	89	Courier	5.25
	R4G106PE.ENV.TYM	Gray Bar	106	Courier	4.25
4 landscape images per landscape sheet 	R4G140LE.ENV.TYM	Gray Bar	140	Courier	4.50

Three additional environment files are available that are 90 lines per page. The following table lists each of these files, using the format described previously.

Table 3-13: 90 Line Environment File Layouts

Layout	Environment File	Style	Chars/Line	Font	Point Size
2 portrait images per landscape sheet 	GREYLSTE.ENV.TYM	Gray Bar	85	Courier	7.25
2 portrait images per landscape sheet 	PROGLSTE.ENV.TYM	Bordered	85	Courier	7.25
1 landscape image per landscape sheet 	SHOWDATA.ENV.TYM	Graph	185	Special	

**HP Environment Files**

The following environment files are supplied for backward compatibility with applications such as HP Word. They should only be used for earlier versions of LaserJet printers. For all other printers, Unison Software recommends using the environment files from ENV.TYM.

Table 3-14: Compatibility Environment Files

Environment File Name	Character Fonts	Point Size	Orientation
COUR86.HPENV.SYS	Courier regular Courier bold Courier italics	12 12 12	Portrait
COUR86A.HPENV.SYS	Same as COUR86		Portrait
COUR86L.HPENV.SYS	Same as COUR86		Landscape
GOTH86.HPENV.SYS	Gothic regular Gothic bold Gothic italics	12 12 12	Portrait
GOTH86A.HPENV.SYS	Same as GOTH86		Portrait
GOTH86L.HPENV.SYS	Same as GOTH86		Landscape
PRES86.HPENV.SYS	Elite regular Elite bold Elite italics	12 12 12	Portrait
PRES86A.HPENV.SYS	Same as PRES86		Portrait
PRES86L.HPENV.SYS	Same as PRES86		Landscape
PROP86.HPENV.SYS	Proportional regular Proportional bold Proportional italics	12 12 12	Portrait
PROP86A.HPENV.SYS	Same as PROP86		Portrait
PROP86L.HPENV.SYS	Same as PROP86		Landscape
SCRIPT86.HPENV.SYS	Script ASCII Script Roman Ext.	12 12	Portrait
SCRIP86A.HPENV.SYS	Same as SCRIPT86		Portrait
BAR3986.HPENV.SYS	Gothic regular 3 of 9 Bar Code	12 17.5	Portrait
BAR3986L.HPENV.SYS	Gothic regular 3 of 9 Bar Code	12 17.5	Landscape
GBAR86L.HPENV.SYS	Line Printer	8	Landscape Gray Bar
GBAR86P.HPENV.SYS	Line Printer	8	Portrait Gray Bar
GRAF2686.HPENV.SYS	None. Use for graphics		
GRAF86A4.HPENV.SYS	None. Use for graphics with A4 paper		

Table 3-14: Compatibility Environment Files

Environment File Name	Character Fonts	Point Size	Orientation
LP86.HPENV.SYS LP86A4.HPENV.SYS	Line Printer ASCII Line Printer Ext.	8 8	Landscape
LP862L.HPENV.SYS LP862LA4.HPENV.SYS	Line Printer	6	Landscape 2:1 reduction
LP862P.HPENV.SYS LP862PA4.HPENV.SYS	Line Printer	6	Portrait 2:1 reduction
LP864L.HPENV.SYS LP864LA4.HPENV.SYS	Line Printer	4	Landscape 4:1 reduction
LP864P.HPENV.SYS LP864PA4.HPENV.SYS	Line Printer	4	Portrait 4:1 reduction
EZREAD86.HPENV.SYS	Line Printer	8	Landscape with form
EZRD86D.HPENV.SYS	Line Printer	8	Landscape duplex with form

## Printing Options

Formation offers standard methods for printing form layouts merged with data generated by printing applications. The printing methods provided by Formation include:

- Direct printing, which combines the form with the data at the time it is produced by the application.
- Running the LPRINT utility to retrieve the data and merge it with the form. This method provides more formatting capabilities than the LaserJet printers can do automatically.
- Monitoring and redirecting spool file printing using Spool Queue Monitor (SQM). SQM provides additional control over forms printing and spool file activity and allows you to include environment file information when printing spool files to laser printers.

### Topics

This chapter discusses the following printing topics:

- File Equate Facility
- LPRINT Utility
- Printing with Pre-Space Carriage Control
- Printing Simple Forms
- Multi-Part Forms
- Duplexed Forms
- Resetting the Printer
- Moving Data on the Page
- Controlling Form Feeds
- Printing PC Documents
- Spool Queue Monitor
- Creating SQM1DEV
- Creating SQM1ENVD
- SQM Operations

## File Equate Facility

### Purpose

The file equate facility allows you to print forms with data directly.

### Syntax

```
:FILE outputfile;DEV=device;ENV=environmentfile  
:RUN application
```

### Parameters

*outputfile*        The file name to which your application writes its data.

*device*            The device number or name of a LaserJet, HP 5000, or HP 2680/2688 printer to which your output should be directed.

*environmentfile*    The name of the environment file that contains the printing specifications for this job.

### Description

The file equate facility can be used with LaserJet, HP 5000 and HP 2680/88 printers. To print your forms and data directly,

1. Set a file equation specifying the environment file and the output device as illustrated in the "Syntax" section.
2. Run your application.

Formation loads the environment file into the printer before the data and routes the final product to the printer specified. An environment file specified in a file equation only takes effect for the print file in which it is specified. Subsequent print jobs are not affected.

For production use, the file equate facility provides some advantages over LPRINT. By merging the form and data on the fly, and sending the output directly to your LaserJet, the file equate facility eliminates the extra steps of creating and processing an output file. This streamlines CPU and disk space.

LPRINT should be used for advanced formatting. Use LPRINT to print to print multi-part forms or forms that include multiple logical pages on a single physical page to the LaserJet or HP 5000. The file equate facility is not designed to print these kinds of form sets on LaserJets or HP 5000s.



## LPRINT Utility

### Purpose

LPRINT takes data that has already been output by your printing application, and formats it according to the specifications in your environment file. The LPRINT utility provides a number of printing capabilities on a LaserJet or HP 5000 printer. However, it does not print to the HP 2680 or HP 2688. LPRINT's advanced capabilities include:

- String search
- Reduction printing
- Data positioning
- Duplexing
- Printing multi-part forms

### Syntax

```
:RUN LPRINT.FORM.TYM [,PRESPACE][;INFO="p[[ p][ ...]]"]
```

### Parameters

PRESPACE	Enable printing of forms with input files that use pre-space carriage control.
<i>p</i>	Can be any of the following: M[ <i>n</i> [, <i>n</i>  0][, ... ]] Print a multi-part form. Each optional <i>n</i> parameter defines the number of logical pages to print on each physical page. The 0 parameter can be included for pages with no data. Refer to page 4-9 for more information.
D	Print a duplexed form merged with data. Refer to page 4-12 for more information.
L[+ -] <i>n</i>	Move the data right (+) or left (-) <i>n</i> spaces. Refer to page 4-13 for more information.
T[+ -] <i>n</i>	Add or remove <i>n</i> lines from the top of each page of data. Refer to page 4-14 for more information.
B [+ -] <i>n</i>	Add or remove <i>n</i> lines at the bottom of each page of data. Refer to page 4-14 for more information.
X	Reset the printer after printing is complete. Refer to page 4-13 for more information.
F	Ignore carriage control form feeds found in the first record of the data file. Refer to page 4-16 for more information.
'string'@ <i>nn</i> =[ <i>newString</i> ][ <i>x</i> ][D]	When the specified string is encountered at column <i>nn</i> , LPRINT substitutes the <i>newString</i> text and/or carries out one of six actions. Refer to page 4-16 for more information.

## Description

The LPRINT utility includes the capability to do all of the following:

- Print forms with pre-space carriage control.
- Print simple forms—single-part forms merged with data or forms without data.
- Print reduced size listings (2-up and 4-up printing).
- Print multi-part forms merged with data. Multi-part forms can be printed 2-up (or more) per page and duplexed with standard text or forms containing data on the back.
- Print duplexed forms merged with data.
- Reset the printer.
- Shift data to the left, right, up, or down on the page with no changes to the printing application.
- Control aspects of the form feed function.
- Print PC documents uploaded to the HP 3000.

The following sections describe LPRINT's capabilities.

## Printing with Pre-Space Carriage Control

The PRESPEC parameter can be included with the LPRINT command for input files that use pre-space carriage control. This parameter allows you to accurately print forms with input files in which carriage control positioning is done before the text.

The syntax for the command is:

```
:RUN LPRINT.FORM.TYM,PRESPEC
```

Additional INFO string parameters can be included after the PRESPEC parameter.

## Printing Simple Forms

### Printing Single-Copy Forms with Data

1. To print your form with LPRINT, send your application data to an output file on disk.  
:FILE *outputfile=tempfile.group.acct*  
:RUN *application*
2. Specify the name of this file in a file equation for LPIN, LPRINT's input file.  
:FILE LPIN=*tempfile.group.acct*
3. Specify the device number or name of your LaserJet, and the name of your environment file in a file equation for LPRINT's output file, LPLIST.  
:FILE LPLIST;DEV=*device*;ENV=*environmentfile*
4. Run LPRINT. For example, to print a single-copy, single-sided form, enter:  
:RUN LPRINT.FORM.TYM
5. Remember to purge the output file if you no longer need it:  
:PURGE *tempfile.group.account*

---

### Printing Forms without Data

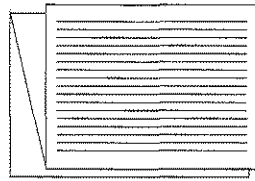
1. To print a form without data, specify \$NULL as the input file:  
:FILE LPIN=\$NULL  
:FILE LPLIST;DEV=*device*;ENV=*environmentfile*  
:RUN LPRINT.FORM.TYM

A single page is printed containing all active logical pages.

2. To print each active logical page on a separate sheet, include the INFO="M" string with the :RUN~ command:  
:FILE LPIN=\$NULL  
:FILE LPLIST;DEV=*device*;ENV=*environmentfile*  
:RUN LPRINT.FORM.TYM;INFO="M"

## Reduction Printing

LPRINT gives you the ability to print source code listings, compiler listings, and other system listings and reports in a reduced typeface, grouped two or four to a sheet of 8 1/2" x 11" paper.



*When archiving listings, it is useful to print two or four reduced page images per sheet.*

Unison Software ships four reduction printing environment files providing the following reduction capabilities:

- 2 portrait-oriented images per sheet
- 2 landscape-oriented images per sheet
- 4 portrait-oriented images per sheet
- 4 landscape-oriented images per sheet.

For more information refer to "Supplied Environment Files" on page 3-48.

The landscape-oriented files will print 132-column data. For example, to print a 4-up listing of 132-column data in a file called LISTING, enter the following:

```
:FILE LPIN=LISTING
:FILE LPLIST;DEV=device;ENV=LP864L.HPENV.SYS
:RUN LPRINT.FORM.TYM
```

To print four images on each side of the paper (eight pages of data per sheet), add the INFO="D" parameter to your :RUN command.

To reposition each page of data with respect to the black outline drawn around it, read the section "Moving Data on the Page" on page 4-13.

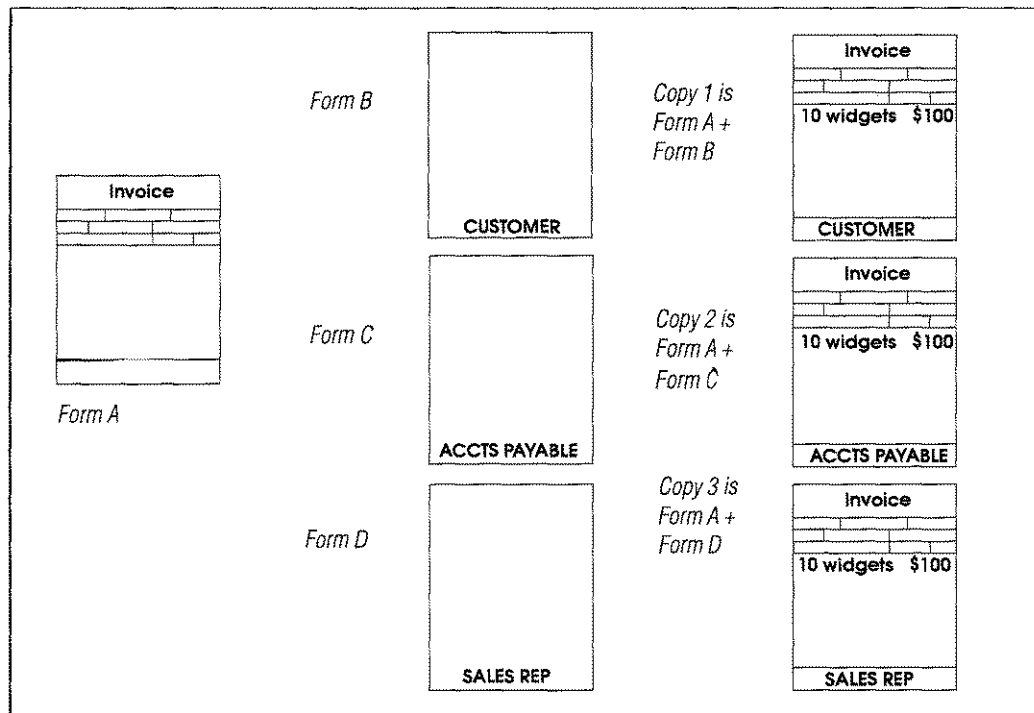
If your printing application does not completely fill each page with data, but uses form feeds to control the pagination of the output, add the CCTL parameter to the file equation for your listing file. For example, the following commands create a COBOL II compiler listing that can be printed in reduced format with the same pagination it would have on a line printer:

```
:FILE LIST,NEW;DEV=DISC;REC=-132,1,V,ASCII;CCTL;DISC=20000
:COBOLII MYPROG,*,*LIST
```

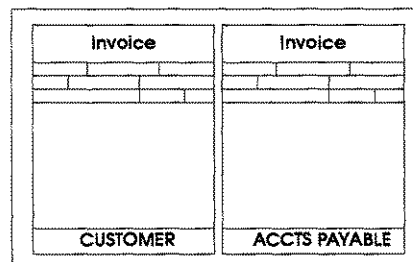
## Multi-Part Forms

The primary purpose of Formation's multi-part forms capability is to simulate the production of carbon copies in the electronic forms environment. For example, you may need to print three copies of an invoice; one for the customer, one for your accounts payable department, and one for the sales representative. Each copy is identical except for a label area indicating who receives this copy. You can create up to 32 such versions of a form on a LaserJet with LPRINT.

To do this, create a basic form containing the elements that go on all copies (Form A in the following example). Then create additional forms containing routing information for each copy in the form set (forms B, C, and D in the following example). In production, each page of data is printed on every page in the form set.

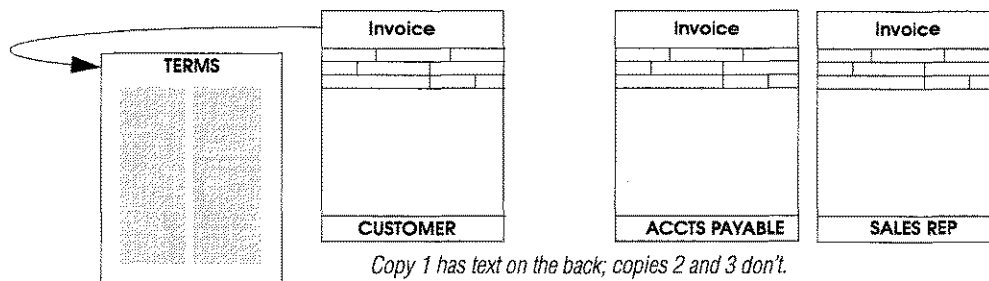


LPRINT also provides control over the formatting of each set of copies. For example, you can print two or more copies of the form per sheet of output:



*Copy 1 and Copy 2 printed 2-up.*

You can also print standard terms and conditions on the backs of some or all copies in your form set.



### Setting Up The Environment File

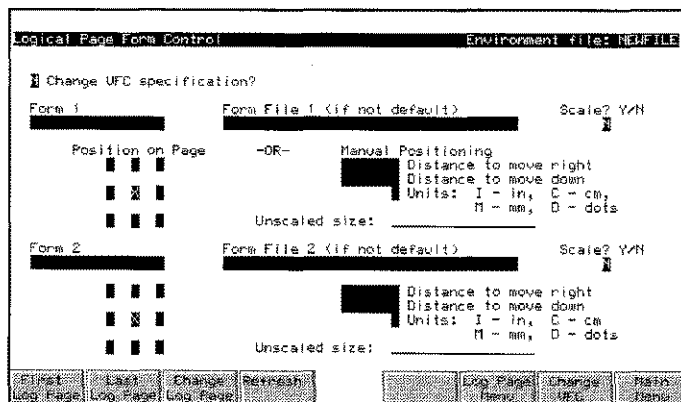
To print multi-part forms on a LaserJet, define a sequence of logical pages, each of which represents one copy of the form. LPRINT cycles through these pages, printing each set of data on all logical pages in the environment file.

The first step in setting up the form file is to create the necessary forms. Put the body of the form in one form file, and the identifying label for each copy (CUSTOMER, ACCTS PAYABLE, etc.) in a separate form file. Assign forms and logical pages to the environment file using Formation UDCs or with the Environment Manager.

Forms and logical pages are assigned to an environment file using the CREATEENV and ADDPAGE UDCs. CREATEENV defines the first logical page in the environment, and ADDPAGE defines any subsequent pages. See "CREATEENV UDC" on page 3-8 and "ADDPAGE UDC" on page 3-5.

To assign forms and logical pages with the Environment Manager:

1. Run the Environment Manager.  
:RUN ENVMGR.FORM.TYM
2. Enter the environment file name at the **Environment File Menu**, and then press e .
3. Type L at the **Main Menu**. The **Logical Page Control** screen is displayed.
4. Enter or verify the dimensions, orientation, and other basic characteristics of the logical page. Press F3 to access the **Logical Page Forms Control** screen.



5. For Form 1, specify the form file containing the main body of the form. For Form 2, specify one of the form files containing the variable information. Press `e` to process the data. Press `e` again to return to the **Logical Page Control** screen.

6. Press F6 to set up the next copy of the form.

**Note:** LPRINT prints each logical page in the environment file in numerical order; Logical Page 0, Logical Page 1, Logical Page 2, etc.

7. When all logical pages are defined, return to the **Main Menu**.

8. Compile the environment.

For more information on how to define logical pages with the Environment Manager, see "Logical Page Control" on page 3-34.

**Note:** You can also create a single form file for each copy in the form set which includes both the body of the form and the identifying label, but this takes up a larger amount of memory, and is more time consuming when downloading the files.

### Printing Multi-Part Forms

The `INFO="M"` parameter allows you to print environment files set up for multi-part printing.

```
:FILE LPIN=output.group.account
:FILE LPLIST;DEV=device;ENV=environmentfile
:RUN LPRINT.FORM.TYM;INFO="M"
```

When the `INFO="M"` parameter is specified, each page of data in your output file is repeated on each logical page. For example, if you have three logical pages in your environment file, and two pages of data in your data file, you get six pages of output — each of the logical pages printed with page 1 of the data, and each of the logical pages printed with page 2 of the data.

For special capabilities such as 2-up printing, you can specify a string of numbers separated by commas after the letter M on the `INFO` string.

```
INFO="Mn[,n][,n][,n]..."
```

The value entered for `n` indicates how many logical pages to print on a single physical page. LPRINT always prints each logical page in the environment file in sequence. The current page of data is printed on each logical page. The commas in the `INFO` string signify page ejects.

For example, in the command,

```
:RUN LPRINT.FORM.TYM;INFO="M2"
```

the 2 means, "Print the next two logical pages in the environment file on one sheet, and fill out each logical page with the current page of data." This would most likely be used when there are two logical pages in the environment file.

The following diagram illustrates the sample output.

First page of output. Contains two copies of the form (the first two logical pages in the environment file), each with the first page of data.

Invoice		Invoice	
10 widgets	\$100	10 widgets	\$100
Logical page 0		Logical page 1	
CUSTOMER		ACCTS PAYABLE	

Second page of output. Since there are no other logical pages in the environment file, LPRINT cycles back to Logical Page 0 and Logical Page 1. This time, each is filled with the second page of data.

Invoice		Invoice	
4 widgets	\$40	4 widgets	\$40
Logical page 0		Logical page 1	
CUSTOMER		ACCTS PAYABLE	

If you have four logical pages defined in your environment file, and you want to group two on each physical page, your string would look like this:

```
:RUN LPRINT.FORM.TYM;INFO="M2,2"
```

The first 2 means, "Print the next two logical pages in the environment file on one sheet, and fill out each of them with the current page of data." The comma means "Go to the next physical page." The second 2 prints the next two logical pages in the environment file on one sheet, and fills each with the current page of data.

First page of output. Contains the first two logical pages in the file, each with the first page of data.

Invoice		Invoice	
10 widgets	\$100	10 widgets	\$100
Logical page 0		Logical page 1	
CUSTOMER		ACCTS PAYABLE	

Second page of output. Contains the next two logical pages in the file, each with the first page of data.

Invoice		Invoice	
10 widgets	\$100	10 widgets	\$100
Logical page 2		Logical page 3	
SALES REP		FILE	

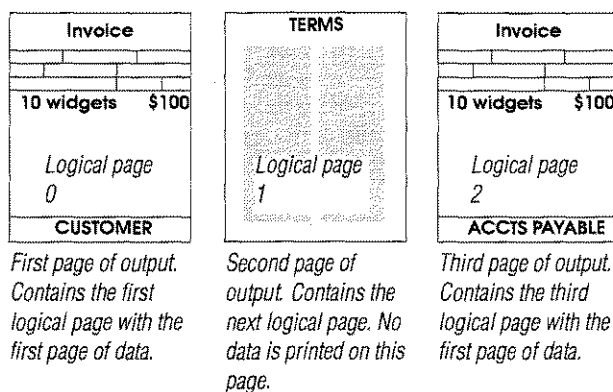
You can also intersperse logical pages with no data in your output by placing zeros in the string. Each zero indicates that the current logical page should be printed without application data.



For example, to print two copies of the invoice, with a page describing payment terms (that has previously been defined as the second logical page in the form set), enter:

```
:RUN LPRINT.FORM.TYM;INFO="M1,0,1"
```

0 means "Print the next logical page *with no data*." The output from this command might look like this:

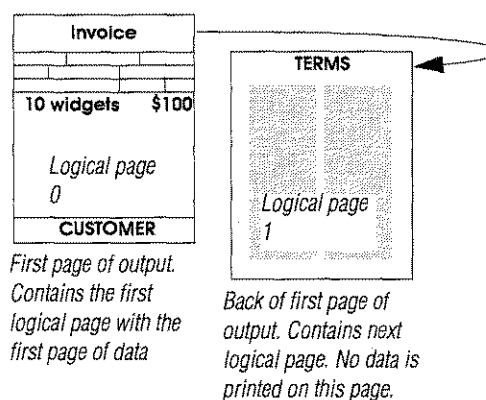


*This sequence would be repeated for each page of data.*

To print terms and conditions on the back of a form, combine duplexing capabilities with INFO="M" specifications. Duplexing is defined by entering a "D" parameter in the info string, and is further explained on page 4-12. When you combine these parameters, a page can be printed with application data on one side, and static form data on the other side.

For example, to print the invoice with a static page on the back, enter:

```
:RUN LPRINT.FORM.TYM;INFO="D M1,0"
```

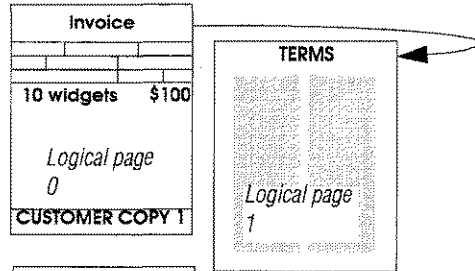


To print the same back page on multiple form copies, create additional logical pages referencing the back page form file. These are placed in the environment file after each front logical page. (There is no performance penalty for this. No matter how many logical pages reference a form, it is only compiled and downloaded once.)

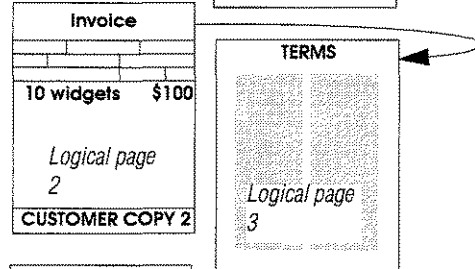
The command to print a job like this might be:

```
:RUN LPRINT.FORM.TYM;INFO="D M1,0,1,0,1,0"
```

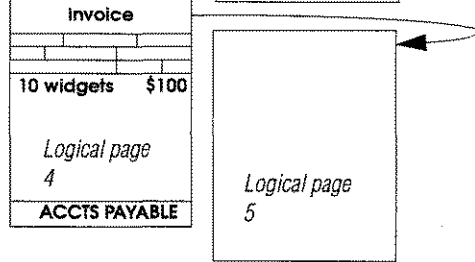
The first page of output contains the first logical page with the first page of data. On the back is the next logical page with no data.



The second page of output contains the next logical page with the first page of data. Since logical page 3 references the same form as logical page 1, the back of this page is the same as the back of the first.



The third page of output contains the next logical page with the first page of data. Since logical page 5 is blank, its back is blank.



## Duplexed Forms

To print duplexed forms (printing on front and back), add the INFO="D" parameter to your :RUN command:

```
:RUN LPRINT.FORM.TYM;INFO="D"
```

Both sides of each sheet, with the possible exception of the last side, contain all active logical pages filled with data. In some instances, the last side printed may not contain all logical pages. For example, if you have three logical pages, and LPRINT comes to the end of the data in your data file while filling out logical page 2 on the last sheet, logical page 3 will not appear on that sheet. To print different logical pages on the front and back of each sheet, see "Multi-Part Forms" on page 4-7.

## Resetting the Printer

LPRINT provides a way for you to reset your printer after a print job has completed. Normally this is not required because your default printer configuration (including font, orientation, bin, etc.) is contained in a TERMTYPE file. This file is used by MPE to automatically reset the printer after each print job. If you don't have a TERMTYPE file that does this automatically, specify INFO="X" with the LPRINT command to reset to the defaults specified on the front panel of the printer after printing. For example:

```
:FILE LPIN=output.group.account
:FILE LPLIST;DEV=device;ENV=environmentfile
:RUN LPRINT.FORM.TYM;INFO="X"
```

The X can be used with other INFO parameters, and can be placed anywhere in the string, as long as it is delimited by spaces. For example:

```
:RUN LPRINT.FORM.TYM;INFO="M2,2 X"
:RUN LPRINT.FORM.TYM;INFO="X D M1,0"
:RUN LPRINT.FORM.TYM;INFO="L-10 X D"
```

## Moving Data on the Page

LPRINT provides a way for you to change where your data prints on the logical page without modifying your printing application. This is especially useful when converting applications that used to print on other printers. These programs may pad data records with blank spaces to position the left margin, and send carriage returns to print blank lines at the top and bottom of the page for vertical alignment. On the LaserJet, printing does not start at the edge of the paper, so these extra lines and spaces may cause misalignment.

Several INFO parameters are available with LPRINT that remove or add characters at the beginning of each line, and remove or add lines at the beginning and end of each page.

Another way to change where data prints, without having LPRINT add or remove characters and/or lines, is to modify the margins set up for your form using the Environment Manager's **Logical Page Control** screen.

### Left and Right

To move data to the left or right, add the INFO="L" parameter to the LPRINT RUN command as follows:

```
:RUN LPRINT.FORM.TYM;INFO="L[+|-]n"
```

L stands for left margin and n is the number of spaces (a space is the size of a character in the current data font) to shift the left margin. If the number is positive (plus sign or no sign), spaces are added to each data record, moving the data to the right. If the number is negative (minus sign), spaces are removed, and the data is moved to the left. For example, if you are using a 10 pitch data font, and your form is printing about one inch too far to the right, you might specify the following:

```
:RUN LPRINT.FORM.TYM;INFO="L-10"
```

The following diagram illustrates positioning data using LPRINT.

The data on this form is printing too far to the right. This is because the printing application is placing blank spaces at the beginning of each record.

Name
John Jones
Address
1701 Oak St.
Kansas City

By running LPRINT with INFO="L-2", two spaces are removed at the start of each record. The data is now positioned correctly.

Name
John Jones
Address
1701 Oak St.
Kansas City

A positive value can be used if the data is printing too far to the left. For example:

```
:RUN LPRINT.FORM.TYM;INFO="L+3"
```

This re-positions the data as shown in the following illustration.

The data on this form is printing too far to the left.

Name
John Jones
Address
1701 Oak St.
Kansas City

By running LPRINT with INFO="L+3", three spaces are added to the start of each record. The data is now positioned correctly.

Name
John Jones
Address
1701 Oak St.
Kansas City

### Up and Down

The INFO="T" and/or INFO="B" parameters move data up or down on a page. The syntax for the :RUN command follows:

```
:RUN LPRINT.FORM.TYM;INFO="T[+|-]n"
:RUN LPRINT.FORM.TYM;INFO="B[+|-]n"
:RUN LPRINT.FORM.TYM;INFO="T[+|-]n B[+|-]n"
```

T stands for top, and n is the number of lines to add or remove at the top of each page of data. If the number is positive (plus sign or no sign), blank lines are added at the beginning of the page. If the number is negative (minus sign), lines are removed at the beginning of the page.

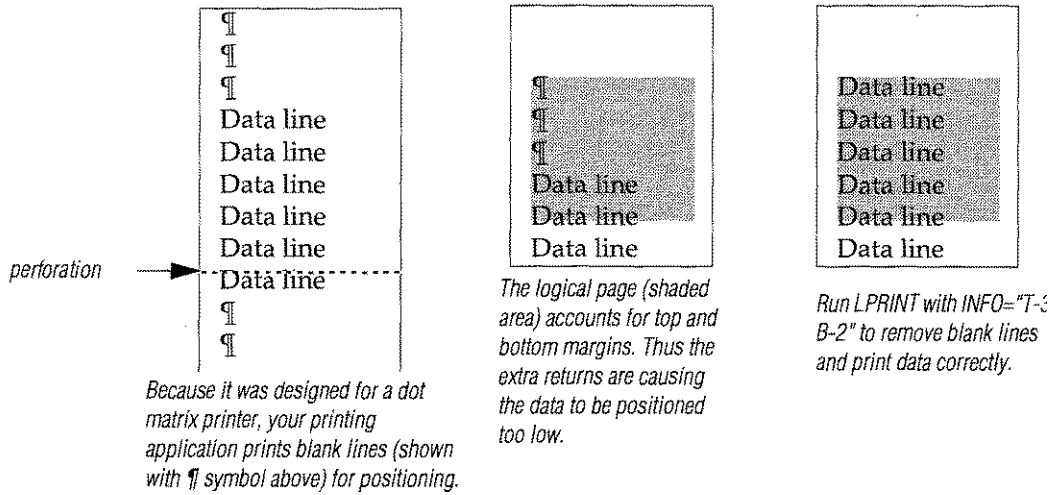
B stands for bottom, and n is the number of lines to add or remove at the bottom of each page of data. If the number is positive (plus sign or no sign), blank lines are added at the end of the page. If the number is negative (minus sign), lines are removed at the end of the page.

To manipulate both the top and bottom of the page, specify T and B together. To specify multiple INFO parameters for LPRINT, list them one after another, optionally separated by a space.

The most common use of this capability is illustrated by the following example. Say your printing application was set up for a dot matrix printer. To create a top margin, it sends three carriage returns at the start of each page of data. It prints the data, then sends two more carriage returns to move down to the perforation which marks the top edge of the next piece of paper. These extra carriage returns appear in your data file as blank lines. It then sends three more returns for the top margin, and so on.

To tell LPRINT to remove these blank lines, use a command like this:

```
:RUN LPRINT.FORM.TYM;INFO="T-3 B-2"
```

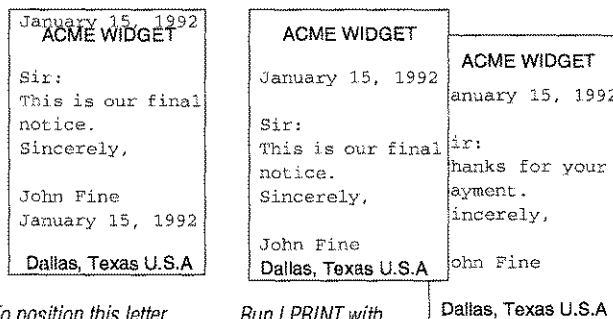


For applications which send carriage returns at the start of the first page of data only, set T to a negative number equal to the number of lines to take out. Set B to a positive number to nullify the effect on later pages. For example:

```
:RUN LPRINT.FORM.TYM;INFO="T-3 B+3"
```

An example where you need to add blank lines to your data file would be an application which prints letters. The form itself includes the company logo and address line. If your data file does not include blank lines, the data starts printing right at the top of the logical page and runs to the end of it. To tell LPRINT to add the blank lines needed to format this correctly, use a command like this:

```
:RUN LPRINT.FORM.TYM;INFO="T+2 B+3"
```



To position this letter correctly, you need 2 blank lines at the top and three blank lines at the bottom.

Run LPRINT with INFO="T+2 B+3" to position the first and subsequent letters properly.

## Controlling Form Feeds

### Ignore Initial Form Feed

Applications written for line printers often have a form feed request at the start of the data to ensure that the paper was lined up at the top of the form. This is unnecessary for laser page printers, and can cause unwanted blank pages or forms to be printed at the start of the job. For this reason, LPRINT provides the option to ignore any carriage control form feed found in the first record of the data file. To do this, specify `INFO="F"` with the LPRINT command.

```
:RUN LPRINT.FORM.TYM;INFO="F"
```

If the first record does not contain a form feed, the option has no effect.

### String-Sensitive Form Feeds

If your data contains strings which signify the need to perform a form feed — for example, the string "MONTHLY TOTAL" in a year-to-date sales report — LPRINT can watch for up to three such strings and perform a page eject when one is encountered. It can also replace the existing string with another string. This capability is accessed through the `INFO` string as shown below:

```
INFO="'string'@nn=['newstring'][x][D]"
```

When the specified `string` is encountered at column `nn` (note that the search is case-sensitive), LPRINT substitutes the text specified by `newstring` and/or carries out action `x`. The `x` parameter can be any of the following:

P	New logical page before current record
Q	New logical page after current record
F	New front page before current record (used when duplexing)
G	New front page after current record (used when duplexing)
H	New physical page before current record
I	New physical page after current record

`newstring` cannot be longer than `string`. If it is shorter, the extra characters of `string` will be replaced with spaces.

D indicates that the record should be deleted (i.e., not printed on the output page). Note that there must be no space between the `x` and the D if both are specified.

For example, to cause a physical page eject after every line containing the string MONTHLY TOTAL at column 23, enter the following command:

```
RUN LPRINT.FORM.TYM;INFO="'MONTHLY TOTAL'@23=I"
```

## Printing PC Documents

LPRINT enables you to print documents created by any PC application (Word, Excel, WordPerfect, Lotus, etc.) on a LaserJet connected to your HP 3000.

To print PC application documents perform the following steps:

1. Write the output from your PC application to a print output file.
2. Upload the file to the HP 3000 as a binary file using a terminal emulator.
3. Run LPRINT with no environment file:

```
:FILE LPIN=filename  
:FILE LPLIST;DEV=device  
:RUN LPRINT.FORM.TYM
```

This technique doesn't work with Macintosh-based documents, as Macintosh applications use PostScript rather than PCL to control the printer.

## Spool Queue Monitor

Formation's Spool Queue Monitor (SQM) enables you to monitor and redirect spool file printing, providing additional control over forms printing and other spool file activity. SQM allows you to include environment file information when printing spool files to laser printers and adds a number of additional printing capabilities to Formation:

- You can send spool files and their corresponding environment files directly to a printer for standard processing.
- Formatting parameters can be included that allow you to print multi-part forms, print duplexed forms, adjust data layout, and print data in reduced format.
- Multiple copies of a single spool file can be printed on several different printers with a single print command.

Spool Queue Monitor is designed to operate on MPE XL versions 2.2 or later.

### Monitoring Process

When started, Spool Queue Monitor runs through a continuous cycle of monitoring activities. The following table describes the spool queue monitoring process:

At This Stage: Spool Queue Monitor:	
Monitor	Monitors the spool queues of the devices listed in the SQM1DEV file. SQM looks for suspect spool files on devices. Suspect files are spool files in READY mode queued on monitored devices.
Check	Checks each suspect spool file against the selection criteria for each definition listed in the in the SQM1ENV.D file. The selection criteria includes the spool file name, job/session name, user, account, and the original target device.
Match occurs	When a matching spool file is found, SQM creates and directs a new spool file to the named output device. Based on the PRINT parameters specified, the spool file is either printed directly to an output device, or formatted with LPRINT. If a processing statement contains several AND keywords, the spool file is printed multiple times.
Output file	Deletes the original spool file and continues the monitoring cycle.  <b>Note:</b> If a print error occurs (such as printing to a non-existing device), SQM sets the original spool file to a deferred status pending user intervention.

### Configuring SQM

To configure the Spool Queue Monitor for use, you must perform the following steps:

1. Add OP capability to the TYM account and manager.tym user or add another user with OP capability.
2. Create the following files:
  - SQM1DEV.FORM.TYM defines the devices to be monitored.
  - SQM1ENV.D.FORM.TYM defines how to process the spool files.
3. Run the SQMSTART command to compile SQM1ENV.D and start SQM. Changes to the SQM1DEV or SQM1ENV.D files do not take effect until SQMSTART is executed.



## Creating SQM1DEV

The file SQM1DEV.FORM.TYM contains a list of all laser printer devices to be monitored. Device (Class) statements should be added and deleted from this file as appropriate. MPE handles spool files destined to any devices not listed in SQM1DEV.

SQM1DEV can contain two statement types: comment statements and class statements. Comment entries describe the contents of the file. Class statements define the devices to be monitored by the Spool Queue Monitor.

### Comment Statement

Comment statements allow you to document the SQM1DEV file. A statement beginning with an asterisk(\*) in column one, followed by at least one space, is treated as a comment. The syntax for the comment statement is:

```
* [comment]
```

### Class Statement

Class statements define the devices to be monitored. A statement beginning with a C in column one, followed by at least one space is treated as a class statement. A class statement contains the class name followed by at least one space and a list of the logical device numbers of the printers that comprise the class. The logical device numbers must be separated by at least one space. The syntax for a class statement is:

```
C class-name ldn [ldn] [...]
```

### Example

The following is a sample SQM1DEV file:

```
* Monitored laser printer device information
*
C LP 6 11
C LJETWEST 101 102 103
C LJETEAST 104 105 108
```

## Creating SQM1ENV D

The second file you must create, SQM1ENV D . FORM . TYM, defines how to process incoming spool files. The Spool Queue Monitor (SQM) uses the environment definitions in the SQM1ENV D file to monitor and match suspect spool files. After creating the file, you must compile it for use by SQM. Run the following program to compile SQM1ENV D into the compiled SQM1ENV C file.

```
:RUN EDCOMP . FORM . TYM ; INFO="SQM1ENV D SQM1ENV C"
```

Running SQMSTART also compiles the SQM1ENV D file, however, SQMSTART includes other steps such as checking to see if SQM is already running and starting SQM monitoring.

To create the SQM1ENV D file you must include statements defining your selection criteria. The two statement types allowed in the SQM1ENV D . FORM . TYM file are:

- Comment statements that describe the contents of the file.
- Processing statements that define which spool files are to be processed, how they are to be formatted, and the printers to which they are sent.

### Comment Statement

Comment statements allow you to document the SQM1ENV D file. There are two possible formats for a comment statement.

Any text that falls between a COMMENT and an ENDCOMMENT statement is treated as a comment. Double quote (") and single quote (') characters are allowed, but a mismatched set will cause the compile to fail. The syntax is as follows:

```
COMMENT [comment]  
ENDCOMMENT
```

A statement beginning with an asterisk (\*) in column one, followed by at least one space, is also treated as a comment. The syntax for this comment statement is:

```
* [comment]
```

### Processing Statements

Processing statements define spool file processing information. They identify spool files, and provide formatting and printing instructions. All processing statements begin with a WHEN keyword. Processing statements are concluded by the next WHEN keyword, a COMMENT keyword, or an End-Of-File.

**Note:** Do not enter LPLIST in the SQM1ENV D file. Although SQM includes LPRINT, you cannot process LPRINT files twice. (LPLIST is the formal designator for LPRINT.)

There are three syntax formats. You can use any or all of these syntax formats in the SQM1ENV D file. The syntax you select depends on your processing needs. The syntax statements are described in the following paragraphs and are followed by an explanation of the keywords and parameters used.

```

WHEN "filename" FROM "[jsname,]user.acct" [TO "device"]
  PRINT FORMATTED [PRESPACE] [WITH "info-string"]
  ON "out-string"
  [AND [WITH "info-string"] ON "out-string"]*

```

```

WHEN "filename" FROM "[jsname,]user.acct" [TO "device"]
  PRINT DIRECT ON "out-string"
  [AND ON "out-string"]*

```

```

WHEN "filename" FROM "[jsname,]user.acct" [TO "device"]
  PRINT [FORMATTED [PRESPACE] [WITH "info-string"]]
  ON "out-string"
  [AND [WITH "info-string"] ON "out-string"]*
  DIRECT ON "out-string"
  [AND ON "out-string"]*

```

Where:

- WHEN** This keyword marks the beginning of a processing statement, and identifies the spool file(s) to be processed.
- filename* The name of the spool file to be processed. The name is limited to 8 alphanumeric characters. A trailing @ wildcard can be used.
- FROM** Identifies the spool file creator.
- jsname, user.acct* The creator of the spool file to be processed. Trailing @ wildcards can be used to specify the job/session name, user or account. If the job or session name is omitted, only files created with no job/session name will match. To match all jobs and sessions, use an @ wildcard in the form:
- @, @. @.
- TO** Identifies the device to which the spool file is directed.
- device* The number or class name of any device that is included in the SQM1DEV file. This is the original device number or class name to which the spool file was directed. If the TO clause is omitted, files on all monitored devices that match the other criteria are processed. Wildcards are not permitted.
- PRINT** Indicates the start of the formatting/printing statements. The PRINT keyword must be followed by either a FORMATTED or a DIRECT keyword.
- FORMATTED** Use LPRINT to process the named spool file. Both the PRESPACE and WITH formatting keywords can be specified.

PRESpace	Identifies the input spool file as a pre-space file. Pre-space files include COBOL programs using Write-After-Advancing, as well as many RPG and FORTRAN programs.
WITH	Specify LPRINT parameters.
<i>info-string</i>	<p>A valid LPRINT parameter string. Multiple parameters can be specified, but must be separated by a space. Valid parameter values are:</p> <p>D Print the spool file duplexed (two-sided).</p> <p>M[<i>n</i>[ , <i>n</i> 0][ , ... ]]</p> <p>Print the spool file as a multi-part form. Each optional <i>n</i> parameter defines the number of logical pages to print on each physical page. The 0 parameter can be included for pages with no data.</p> <p>L[+ -]<i>n</i> Move the data right (+) or left (-) <i>n</i> spaces.</p> <p>T[+ -]<i>n</i> Add (+) or remove (-) <i>n</i> lines from the top of each page of data.</p> <p>B [+ -]<i>n</i> Add (+) or remove (-) <i>n</i> lines at the bottom of each page of data.</p> <p>X Reset the printer after printing is complete.</p> <p>F Ignore carriage control form feeds found in the first record of the data file.</p> <p>'string'@<i>nn</i>=[ '<i>newString</i>' ][x][D]</p> <p>When the specified string is encountered at column <i>nn</i>, substitute the <i>newString</i> text and/or carry out one of six actions.</p> <p>For more information on LPRINT parameters, see "LPRINT Utility" on page 4-3.</p>
DIRECT	<p>Print the spool file directly to the named output device. This statement is useful when:</p> <ul style="list-style-type: none"> <li>- LPRINT is not needed for formatting.</li> <li>- Directing one or more copies of the spool file to an HP2680/88.</li> <li>- Directing a copy of a spool file to a line printer or other device that does not support page formatting (environments).</li> </ul>
ON	Identifies the output device to which the output spool file will be directed.
<i>out-string</i>	Specifies the output device. Begin with the DEV= keyword, followed by file equation printing parameters. For spool files that will be formatted with LPRINT, an environment file must be included using the ENV= keyword. The ENV= keyword is optional for DIRECT spool files. For more information on output device specifications, refer to <i>Usage Notes</i> .

AND                    Send the spool file to multiple output devices. Statements can include differing LPRINT parameters for each device. Up to 15 AND statements can be included in a single processing entry.

### Usage Notes

1. Original spool file information can be referenced with the DEV= keyword using any of the following parameters:

```
!DEVICE, !PRIORITY, !COPIES.
```

When these parameters are named, the Spool Queue Monitor inserts the device number, priority, and number of copies specified with the original spool file. For example:

```
DEV=!DEVICE, 8, 1
```

```
DEV=25, !PRIORITY, !COPIES
```

The !PRIORITY parameter includes an additional value that allows SQM to elevate the priority of the original spool file before inserting the priority into the out-string. The format of the parameter is:

```
!PRIORITY+n
```

where *n* is a single digit from 1-9. For example:

```
DEV=!DEVICE, !PRIORITY+5, !COPIES
```

2. When specifying an output device, the *out-string* can include additional MPE file equate parameters for a spool file such as FORM= or SPSAVE. For example:

```
DEV=101, , !COPIES; ENV=PROGLSTE.PUB.SYS; FORM=forms_msg
```

```
DEV=!DEVICE; ENV=MYRPRT.PUB.SYS; SPSAVE
```

3. Any single processing entry can include a maximum of 16 ON keywords (16 output devices).
4. Both FORMATTED and DIRECT keywords can be included in a single processing statement.
5. When processing spool files, the Spool Queue Monitor looks through the SQM1ENV C file starting with the first statement in the file, and proceeding to the last. Each spool file is processed only once, based on the first matching statement found.
6. When SQM1ENV D is compiled, statements are placed in alphabetical order by spool file name. The sorting order is numbers first, alphabetic characters next and wildcards last. For example:

```
TWO
```

```
TWO1
```

```
TWOUP
```

```
TWO@
```

When SQM1ENV C is compiled, the spool file user, account and device names are not alphabetized. Therefore, FROM statements should be placed in the file in descending order, from most qualified to least qualified. Refer to example 5 below for a sample of ordering processing statements.

7. If PRES SPACE is included in a processing statement, all subsequent ON parameters use pre-spacing until the next FORMATTED or DIRECT keyword is encountered.
8. To print spool files in special formats, you can configure special device classes on your system to which users can direct spool files. Create processing entries that name these device classes in their TO statements and specify any special processing instructions in their PRINT statements. Refer to examples 3 and 4 below for a sample of special device class configuration.

## Examples

### Redirect Printing

1. Create a processing entry that prints a spool file on device 33 when the Spool Queue Monitor notices a spool file called SFILE2 from user JOB1, JOE.ACCT destined for device LJET:
 

```
WHEN "SFILE2" FROM "JOB1,JOE.ACCT" TO "LJET"
PRINT DIRECT ON "DEV=33,4,2;ENV=MYFORM.ENV.TYM"
```
2. Create a processing entry that notices spool files named PODATA from any session and user in the ACCT account, and uses LPRINT with pre-spacing to print the file to device 25 as a multi-part form using the POFORM environment. The file will also be printed on the OPSLJET device class printer using LPRINT with the duplex option and the TW OUP environment file. Finally, print the spool file directly to device 101 without any page formatting:
 

```
WHEN "PODATA" FROM "@,@.ACCT"
PRINT FORMATTED PRES SPACE WITH "M"
ON "DEV=25;ENV=POFORM.PUB.SYS"
AND WITH "D" ON "DEV=OPSLJET;ENV=TW OUP.PUB.SYS"
DIRECT ON "DEV=101"
```

### Special Device Monitoring

3. Create a processing entry that monitors spool files destined for special device class TW OUP. Any spool file found will automatically print on device 100 using environment file TW O E . PUB . SYS:
 

```
WHEN "@" FROM "@,@.@" TO "TW OUP"
PRINT FORMATTED ON "DEV=100;ENV=TW O E . PUB . SYS"
```
4. Create a processing entry that monitors spool files destined for special device class TW OUP. If the creator of the spool file is from account PAY, send the spool file to device 100 using the TW O E environment. If the creator of the spool file is from account SLS, send the spool file to device 101 using the TW O E environment. In all other instances, send the spool file to device 102 using the TW O E environment.
 

```
WHEN "@" FROM "@,@.PAY" TO "TW OUP"
PRINT FORMATTED ON "DEV=100;ENV=TW O E . PUB . SYS"

WHEN "@" FROM "@,@.SLS" TO "TW OUP"
PRINT FORMATTED ON "DEV=101;ENV=TW O E . PUB . SYS"

WHEN "@" FROM "@,@.@" TO "TW OUP"
PRINT FORMATTED ON "DEV=102;ENV=TW O E . PUB . SYS"
```

### Processing Parameters

5. The following example shows how to order FROM parameters in a processing statement:

```
WHEN "TWO@" FROM "@,@.PAYROLL"
    PRINT FORMATTED ON "DEV=!DEVICE;ENV=ENV1.PUB.SYS"
```

```
WHEN "TWO@" FROM "@,@.LEDGER"
    PRINT FORMATTED ON "DEV=!DEVICE;ENV=ENV2.PUB.SYS"
```

```
WHEN "TWO@" FROM "@,MGR.@"
    PRINT FORMATTED ON "DEV=!DEVICE;ENV=ENV3.PUB.SYS"
```

```
WHEN "TWO@" FROM "@,@.@" TO "25"
    PRINT FORMATTED ON "DEV=LJ3;ENV=STANDARD.PUB.SYS"
```

In the prior example, any spool file name matching TWO@:

- From the PAYROLL account will be printed to the original device using environment file ENV1.PUB.SYS.
- From the LEDGER account is printed to the original device using environment file ENV2.PUB.SYS.
- From the MGR user, not from PAYROLL or LEDGER accounts, will be printed to the original device using environment file ENV3.PUB.SYS.
- To device 25, that is not from the MGR user or from the PAYROLL or LEDGER accounts, will be printed using environment file STANDARD.PUB.SYS.

Had the fourth entry been placed first, any spool file name matching TWO@ to device 25, regardless of account or user, would be printed to LJ3 using environment file STANDARD.PUB.SYS.

In this example, SQM would not process spool files named TWO@, if the spool files do not match one the four previously listed FROM definitions.

6. Create a processing entry that notices spool files named PROGLIST and prints them with their original processing parameters using environment file PROGLSTE.ENV.TYM.

```
WHEN "PROGLIST" FROM "@,@.@"
    PRINT FORMATTED
    ON "DEV=!DEVICE,!PRIORITY,!COPIES;ENV=PROGLSTE.ENV.TYM"
```

### SQMSAMP

Formation Spool Queue Monitor includes a file named SQMSAMP. This file is a sample environment definition file that includes preset spool file definitions for reduction printing formats. Use any portion or all of this file just as you would the SQM1ENV.D file.

## SQM Operations

When started, Spool Queue Monitor begins looking for any suspect spool files. Suspect spool files are spool files in a READY state in specified device queues that match the selection criteria set up in the SQM1ENV environment definition file.

**Note:** Spool Queue Monitor will not look at any files from job/session names prefixed with SQM (such as SQM1 or SQMTEST). This restriction preserves print data across SQM restarts.

Spool Queue Monitor provides several commands to control spool file monitoring. The following table describes the Spool Queue Monitor commands. Each command is described in further detail in the sections that follow.

Command	Description
SQMSTART	Start Spool Queue Monitor process. Starts SQM as background job.
SQMSTOP	Stop Spool Queue Monitor process (job or session).
SQMPAUSE	Changes the wait time of the running Spool Queue Monitor.
SQMKICK	Directs the Spool Queue Monitor to exit the current wait state and look again for suspect spool files.
SQMSHOW	The SQMSHOW command displays information about spool files that have already been processed.
SQM1SESS	Runs Spool Queue Monitor from a session instead of in a background job.
SQM1JOB	Runs Spool Queue Monitor as a background job.

### Starting/Stopping SQM

Use SQMSTART following a system restart, and when any changes have been made to either the SQM1ENV or SQM1DEV files.

**Note:** DO NOT stream SQM1JOB file directly.

1. To start the Spool Queue Monitor, enter the SQMSTART command. If the Spool Queue Monitor is already running, SQMSTART displays an error message.
2. To stop the Spool Queue Monitor at any time, issue the SQMSTOP command. The SQMSTOP command stops the SQM after processing any in-process spool files.

### SQM Commands and Jobs

#### SQMPAUSE

The SQMPAUSE command changes the wait time of the running SQM program. The syntax for the SQMPAUSE command is as follows:

```
SQMPAUSE n
```

Where:

*n* Specifies a wait time in seconds from 1 to 32767 or a maximum of 9 hours, 6 minutes, 7 seconds.



**SQMKICK**

The SQMKICK command directs the Spool Queue Monitor to exit the current wait state and look for suspect spool files again. This command has no parameters.

**SQMSHOW**

The SQMSHOW command displays information about spool files that have already been processed. This command has no parameters. Use SQMSHOW to see spool files that were processed by SQM but have not printed yet. The following diagram illustrates the use of the SQMSHOW command.

```

:sqmshow
SPOOLID  JOBNUM  FILEDES  PRI COPIES DEV  STATE  RSPFN OWNER
#01E392  S32     TWQUP    6    1 2330  READY  MANAGER.TYM
#01E394  S32     TWQUP    6    1 2330  READY  MANAGER.TYM
#01E396  S32     TWQUP    6    1 2330  READY  MANAGER.TYM

INPUT SPOOL FILES          OUTPUT SPOOL FILES
ACTIVE = 0;                CREATE = 0;                READY = 3;
OPEN = 0;                  DEFER = 0;                SELECTED = 0;
READY = 0;                 DELPND = 0;               SPSAVE = 0;
                                PRINT = 0;                XFER = 0;
                                PROBLM = 0;

TOTAL IN FILES = 0;        TOTAL OUT FILES = 3;
IN SECTORS = 0;           OUT SECTORS = 96;

OUTFENCE = 7
OUTFENCE = 13 FOR LDEV 6
OUTFENCE = 13 FOR LDEV 26
OUTFENCE = 13 FOR LDEV 25
:

```

**SQM1SESS and SQM1JOB**

SQM1SESS and SQM1JOB perform the same function—they run SQM. Both use two parameters for setting wait time and message levels.

SQM1JOB is a job file that runs the SQM as a background job. The *waittime* and *msglevel* options are set as variable in the job file. If you are using a version of MPE XL prior to 4.0, you must include your passwords in the job before streaming SQM1JOB.

SQM1SESS is a command file that runs the SQM from a session instead of in a background job. If started, SQM occupies the terminal while running, but is excellent for evaluating definitions, debugging and performance tuning. To use SQM1SESS, you must log on to your system as follows:

```
MPE XL:HELLO SQM1,MGR.TYM,FORM
```

The syntax of the SQM1SESS command is as follows:

```
SQM1SESS [waittime] [msglevel]
```

Where:

- waittime* Specifies the time in seconds that the SQM waits when it does not find a spool file to process, before looking again for a spool file to process. The time range is 1-32767 seconds (9 hours, 6 minutes., 7 seconds maximum).
- msglevel* Specifies the level of messages the program displays while running. Each level includes all lower level messages. The messages displayed are shown in the following tables:

Production Modes		Description
0	QUIET	No messages, except fatal errors
100	MATCH	Display a message for each matched spool file (includes spool file number).
200	OUTPUT	Display an output record (formatted out-string) for each processed spool file.
225	CLICK	Display a single character (>) at each pause (most applicable when running in session mode).
250	PAUSE	Display message during pause, including time to pause.

The following messages provide more levels of detail but reduce system performance.

Set-Up and Debug Modes		Description
300	SUSPECT	Display a table for each qualified suspect spool file.
400	ENTRY	Display a table for each entry when compared to a suspect spool file.
500	FULL DISPLAY	Display comments while comparing suspect spool files to entries. Used for checking environment definitions for accuracy.

### Setting Spool File Priorities

For best results, any files SQM might attempt to monitor (suspect files) should remain below the device outfence. This ensures that all non-exempt files will be analyzed by the SQM. Files exempt from SQM are those created by the SQM1JOB job.

To use Spool Queue Monitor to analyze spool files before they are printed, you have the following options:

- Set the priority of all spool files that match processing statements below the MPE outfence setting for the device. The priority of the spool file can then be raised using a processing statement, so that after SQM processing the spool file will print.
- Raise the device outfence to be greater than the system default priority. This ensures that all spool files are analyzed, but restricts printing to processed spool files only. Spool files that do not match processing statements never print.

Another option is to create a pseudo printer device on your system containing multiple device classes. The pseudo device should not equate to any physical device. You can use the device classes associated with the pseudo device to print output from applications for processing by the SQM. To use this option, you must:

- Add statements in the SQM1ENV.D file to have SQM look for any file directed toward certain device classes and print the file to a physical printer. The SQM1ENV.D file must specify the target physical printer in the ON statement and include whatever formatting statements are desired for that file.
- Prevent spool files from printing to non-existing printers by using the DOWN function in MPE on the pseudo device or set the pseudo-device with a high outfence.

Use the following commands to DOWN a device:

SPOOLER *device*;STOP;OPENQ;SHOW

DOWN *device*

Where:

*device* is the name of the pseduo device.

STOP stops the spooler for the specified device.

OPENQ allows the spool queue to remain open.

DOWN *device* makes the specified device unavailable.

Once you have DOWNed a device, files can continue to enter the spool queue for that device but MPE XL consideres the device for that spool queue as unavailable.



## Fonts and Graphics

Formation forms consist of two kinds of text: form text and data. Form text makes up the headers, titles and labels that are part of the form layout. Form text (and the corresponding fonts) are assigned to a form using Formation Design. Data is the information that is filled in by your HP 3000 printing application. Data fonts are assigned using Formation UDCs, the Environment Manager **Character Set Control** screen, or the **PUSEFONT** intrinsic.

Formation also includes the capability to print imported graphics on your forms. The Formation package includes utilities to convert TIFF (Tagged Image File Format) files to either PCELL or raster files and to print figure files from graphics packages.

### Topics

This chapter describes the following Formation font and graphics topics:

- Understanding Fonts
- Formation's Font Directory
- Adding Fonts to the Directory
- Converting PCL Fonts to PCELL Files
- Convert Utility
- FIGPRINT Utility
- Data Font Considerations
- FONTDOWN Utility
- Bar Code Fonts
- 2 of 5 Bar Code Checksum and Interleaving
- Postnet with Formation

## Understanding Fonts

A font is a set of characters that have similar characteristics such as typeface (shape of characters), weight (boldness), point size (height), and spacing (fixed or proportional). Using Formation, you will assign fonts to the different text elements in your forms.

### Font Types

The following form text fonts are available with Formation Design:

- CG Times (same as Times Roman)
- Univers (same as Helvetica)
- Courier
- Bar 3 of 9 15 mm Bar Code
- US Postnet Bar Code

**Note:** If you use this font to print envelopes for mailing, print a sample envelope and have the placement of the bar code checked by the U.S. Post Office.

Each of these fonts is shipped with Formation Design, except for Courier, which is provided as part of Microsoft® Windows™. These fonts are also provided in font files on the HP 3000.

### How Fonts Are Accessed

1. When you compile forms on the HP 3000, the Environment Manager checks Formation's font directory.
2. The font directory cross-references PC fonts to the corresponding font files on the HP 3000.
3. The Environment Manager then brings in the HP 3000 versions of the fonts used for your form text.

For more information refer to "Formation's Font Directory" on page 5-4. For more information about assigning fonts with Formation Design, refer to the *Formation Design User Guide*.

### Data Fonts

In Formation, data fonts are classified as follows:

- **Soft fonts** are contained in a file (PCELL or PCL), and are always bitmapped.
- **Hard fonts** (also called **resident fonts**) are contained in the hardware and can be either scalable or bitmapped soft fonts.
- **Scalable fonts** use a printer escape sequence to print and are designed to be printed at any point size.
- **Bitmapped fonts** point to a PCELL file containing font information. Each bitmapped font is a single point size.

**Note:** If a bar code is used as part of your form text, the same code is printed on all copies of the form. To place a unique bar code on each copy of the form, use the bar code as a data font rather than as form text. (If you need bar codes that are not provided with Formation, please contact Unison Software's technical support department.)

The following table provides samples of the different types of font.

**Table 5-1: Type of Fonts**

Font	Hard (Resident)	Soft
<b>Bitmap</b>	Courier 12 point** Prints on LJII, LJIII, and LJ4 printers	PCELL file: UNJ7R3.FONTS.TYM **  PCL file: Custom 8-bit
<b>Scalable</b>	Univers** Available on LJIII and LJ4 printers	

\*\*Fonts contained in font directory.

**Note:** A PCL file contains only a single size of one font for a single device. A PCELL file can contain multiple point sizes and resolutions, of a font. PCL fonts can only be used on LaserJet printers, not on 2680 printers.

### Soft Fonts

Formation uses three types of soft fonts:

1. Soft fonts in the group CHARSETS.SYS. This group contains Hewlett-Packard fonts. You may already have fonts in this group that were supplied with other HP products on your system. A number of HP fonts (Courier, Elite, Line Printer, two 3 of 9 bar codes, and others) are shipped with Formation and placed in this group during installation.

Formation/2680 users: If you have HP's IFS/3000, your reference guide for that product lists the files in CHARSETS.SYS and gives a sample of each font.

2. Soft fonts in the group FONTS.TYM. This group contains fonts created by Unison Software — the HP 3000 versions of CG Times, Univers, Courier, Letter Gothic, Prestige, PostNet and 3 of 9 15 mm. Proportionally spaced fonts like CG Times and Univers are more useful for printing form text rather than data, because each letter occupies a different width and this makes it difficult to align columns of numbers or other data. Monospaced fonts like Courier, Prestige, or Letter Gothic provide better column alignment for data fonts. The bar codes in this group are often used as data fonts.
3. Any other soft font in a PCELL file containing HP 2680 or 2688 fonts, or in a PCL font file.

### PCL Resident Fonts

These fonts also break into three types:

1. Internal fonts. These are hard fonts built into the printer. On a LaserJet III, for example, the internal fonts include Univers (scalable), CG Times (scalable), Courier (bitmapped) and Line Printer (bitmapped).
2. Cartridge fonts. These hard fonts are contained in a ROM cartridge you plug into the printer. They may be scalable or bitmapped.
3. Permanent soft fonts. These are soft fonts that are downloaded to the printer and identified as permanent. These are bitmapped.

Scalable fonts are not supported on the LaserJet Plus, any LaserJet II model, or the HP 5000.

## Formation's Font Directory

The Formation font directory contains detailed character width metrics for each type of font (resident fonts). The Environment Manager requires this character metric information to actually write text on your form.

**Note:** The PCELL file contains the character width metric information, therefore this information does not need to be in the directory for soft fonts.

The font directory also tells Formation the location of:

- The HP 3000 font files that correspond to the fonts used on your forms.
- The file containing the font you've described, when defining fonts using the Environment Manager's character **Font Control** screen. This screen allows you to specify a general typeface name, weight, style, and size (e.g. 12 point italic Universe).

The font directory is a KSAM file pair, FONTDIR.FORM.TYM and FONTDIRK.FORM.TYM. These files are created by a special compiler program called FDCOMP. FDCOMP is run automatically when you install Formation, and can be run manually to add fonts to the directory at any time.

To create the directory, FDCOMP reads the information in the following files:

- FDTYM.FORM.TYM. This is a list of the fonts supplied by Unison Software or resident in various LaserJet models. **PLEASE DO NOT EVER MODIFY THIS FILE.**

The FDTYM.FORM.TYM file:

- Matches the typeface names used by Formation Design (Univers, CG Times, and Courier) to the HP 3000 files that contain these fonts in all their various weights, styles, and sizes.
  - Contains descriptions of some of the internal hard fonts contained in the various LaserJet models. These are listed in the section titled "Resident Fonts in FDTYM" on page 5-4.
- FDUSER.FORM.TYM. This is a user created text file used to add fonts to the directory. For more information on adding fonts, refer to "Adding Fonts to the Directory" on page 5-6.

### Resident Fonts in FDTYM

Some of the fonts contained in the FDTYM file are resident in LaserJet IIs, IIIs and 4s, though the scalable fonts are resident only in IIIs and 4s. Each of these fonts is automatically indexed in the font directory. You can access the fonts from the Environment Manager's **Character Set Control** screen using the typeface name.

**Note:** Included with Formation are files that contain the font definitions for other fonts resident on specific LaserJets and the HP 5000. You can add these entries to your FDUSER file. See "FDUSER Entry Format" on page 5-6.

The following table lists the resident fonts in the FDTYM file.



Table 5-2: Fonts in the Formation directory

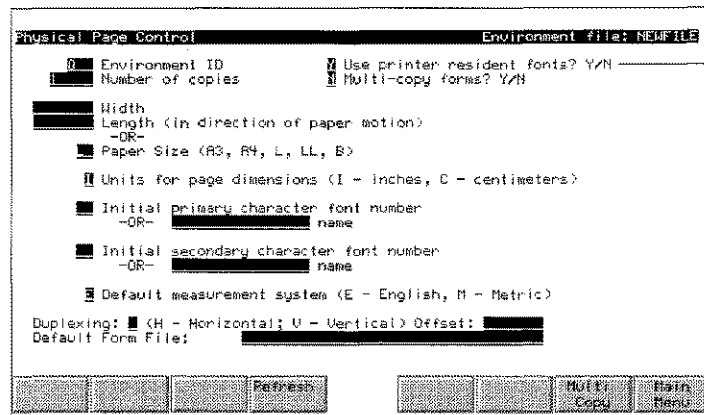
Resident Fonts in FDTYM.FORM.TYM	PCELL Fonts in FDTYM.FORM.TYM
Courier 10cpi 12pt Bold 10cpi 12pt	Univers Medium Bold Italic Bold Italic
CGTimes Medium Scalable Bold Scalable Italic Scalable Bold Italic Scalable	CGTimes Medium Bold Italic Bold Italic
LinePrinter 16.66cpi 8.5pt	Bar Codes 3 of 9 15 mm PostNet
Univers Medium Scalable Bold Scalable Italic Scalable Bold Italic Scalable	MICR  Courier (Medium) Prestige (Medium) Letter Gothic (Medium)

### Duplicate Font Names

The font directory may contain more than one entry for some font names. This is because several fonts are available in soft, resident/bitmapped and resident/scalable forms, and because you may add fonts to FDUSER with the same names as those in FDTYM. Univers, for example, is supplied both as a soft font in FONTS.TYM and as a resident scalable font in LaserJet IIIs and 4s. If you specify Univers on the **Character Set Control** screen, Formation must decide which of the Univers fonts to use. Here are the rules it follows to make this decision.

1. If you have specified an HP 2680/88 printer as one of your target printers on the **Main Menu**, Formation always chooses the soft PCELL font. Resident fonts are not supported by the HP 2680/88. Even if you are using a dual environment file and are printing to a LaserJet, the soft PCELL font is used.
2. If your target printer is not an HP 2680/88, the following rules are applied:
  - If a resident scalable font is available, it is used. (Resident scalable fonts are found only on LaserJet IIIs and 4s).
  - If no resident scalable font is available, it looks for a resident bitmapped font.
  - If no resident bitmapped font is available, it uses a soft PCELL font.

If you don't want to use resident fonts at all, you can override this order of precedence using the **Use printer resident fonts?** field on the Environment Manager's **Physical Page Control** screen. To specify that resident fonts should not be used, change the value in this field to N.



Set this field to N if you don't want to use fonts resident in your printer.

**Note:** When you open an environment file created with a pre-3.4 version of Formation, the Use printer resident fonts? field is set to N. If you want to use resident fonts with these files, change the value to Y and recompile.

## Adding Fonts to the Directory

Adding fonts to the Formation font directory requires the following steps:

1. To add a font to the Formation font directory, create a text file called FDUSER.FORM.TYM. If you already have an FDUSER file, append to that file.
2. Run FDCOMP to update the font directory after creating or modifying this file. If you install a Formation upgrade, FDCOMP automatically reads FDUSER during the installation process, and includes the information FDUSER contains in the new version of FONTDIR.

Detailed step by step instructions for adding resident fonts to FDUSER are provided in the following section of this chapter.

FDUSER is basically a list of font definitions. If the font definition refers to a resident proportional font, it is associated with a metric definition. (Resident fixed-width fonts do not require metric definitions.)

Formation includes a special utility, TFMTRANS, to automatically extract the information needed for this entry from the TFM file provided with your font.

### FDUSER Entry Format

You can enter font information to FDUSER using three possible formats. The format you select will depend upon the type of font:

- Soft PCELL font
- Resident bitmapped font
- Resident scalable font

The font formats are described in the following sections. Entries may consist of multiple lines with no restriction on where line breaks are placed.

## Soft PCELL Fonts

The syntax for soft font entries is as follows:

```
font typeface_name [face][style][weight]
pcell "mpe_file_name"
[extension "mpe_file_name"][hmi hmi_value][usebestmetrics]
```

Where soft font entries are defined by using one or more of the following:

*font typeface\_name*

Specify the font keyword and a typeface name.

*face*

The *face* keyword is optional (not required); it is supported for backward compatibility only.

*style*

Specify a style from the following list:

*italic* *condensed* *compressed* *expanded*  
*inline* *outline* *shadowed*

*weight*

Specify a weight from the following list:

*ultrathin*|*extrathin*|*thin*|*extralight*|*light*|  
*demilight*|*semilight*|*medium*|*semibold*|*demibold*|  
*bold*|*extrabold*|*black*|*extrablack*|*ultrablack*

*pcell mpe\_file\_name*

Specify the *pcell* keyword and the name of a PCELL file containing the font character set.

*extension mpe\_file\_name*

Specify the *extension* keyword and the MPE file name of a PCELL file containing the top half of the Roman-8 character set. In general PCELL files are limited to 128 characters, so two must be combined together to create full Roman-8 character sets. The *extension* keyword is only needed in this case.

*hmi hmi\_value*

Specify the *hmi* keyword and an *hmi\_value*. Use the value only for proportional fonts used in form text (as opposed to data). The value is expressed in 1/120 of an inch. This value ensures that Formation Design and the Environment Manager use the same values when calculating the position of words on the form. Obtain the correct value from a Tagged Font Metric file using the TFMTRANS program described on page 5-14.

*usebestmetrics*

Specify *usebestmetrics* to force the Environment Manager to use the most accurate character size metrics available in the PCELL file. If the PCELL file contains both 180 dpi HP 2680 characters and 300 dpi LaserJet/HP 2688 characters, the Environment Manager uses the 300 dpi metrics to calculate font sizes for the 2680. This is a finer, more accurate measurement, though the character bitmap itself is still 180 dpi.

The following examples illustrate some PCELL font definitions:

```
font Univers pcell "UNR7R3.FONTS.TYM"  
    extension "UNR XR3.FONTS.TYM"  
  
font Univers face bold pcell "UNB7R3.FONTS.TYM"  
    extension "UNB XR3.FONTS.TYM"
```

### Resident Bitmapped Fonts

The syntax for bitmapped entries is as follows:

```
font typeface_name [face][style][weight]  
internal [metricname metric_name]  
size point_size hmi hmi_value vmi vmi_value  
[symbolset "symbol_set"] typeface typeface_value  
spacing {f | p} baseline baseline_offset
```

Where bitmapped font entries are defined by using one or more of the following:

- font** *typeface\_name* Specify the font keyword and typeface name.
- face** The face keyword is not necessary; it is supported for backward compatibility only.
- style** Specify a style from the following list:  
    *italic* *condensed* *compressed* *expanded*  
    *inline* *outline* *shadowed*
- weight** Specify a weight from the following list:  
    *ultrathin*|*extrathin*|*thin*|*extralight*|*light*|  
    *demilight*|*semilight*|*medium*|*semibold*|*demibold*|  
    *bold*|*extrabold*|*black*|*extrablack*|*ultrablack*
- internal** Specify this keyword to define an internal font.
- metricname** *metric\_name* Specify a *metricname* as the key to the associated metric definition located elsewhere in the file. Specify the parameter only for proportional fonts. For fixed pitch fonts, the character widths are derived automatically from *hmi\_value*, so *metric\_name* is not required.
- size** *point\_size* Specify the size keyword and the height of the font in points. The parameter can include up to 4 decimal places.
- hmi** *hmi\_value* Specify the hmi keyword and a value expressed in 1/120 of an inch, matching the values used in the PCL font selection sequence. The *hmi\_value* parameter can include up to 4 decimal places.

`vmi` *vmi\_value* Specify the `vmi` keyword and a value expressed in 1/48 of an inch, matching the values used in the PCL font selection sequence. The *vmi\_value* value can include up to 4 decimal places.

`symbolset` "*symbol\_set*"

Specify the `symbolset` keyword and a symbol set for the font. Defaults to "8U" or Roman-8 if not specified.

`typeface` *typeface\_value*

Specify the `typeface` keyword and a number indicating the design of the font as found in a font selection string (e.g., 52 for Univers, or 4148 if the Agfa family value of 4096 is included.) Both values select Univers.

`spacing` *f|p* Specify the keyword and either F for fixed or P for a proportionally spaced font.

`baseline` *baseline\_offset*

Specify the `baseline` keyword and the offset of the baseline from the character cell top. The parameter is required for correct positioning of characters.

The following example illustrates a bitmapped font definition:

```
font Courier internal size 12.00 hmi 12.00 vmi 9.60
  typeface 4099 spacing f baseline 47
```

### Resident Scalable Fonts

The entries for scalable resident fonts are almost the same as for bitmapped resident font entries, except that the keyword `scalable` replaces `size`. Express the `hmi`, `vmi` and `baseline` values as if for a 100 point font. The Environment Manager scales the fonts as appropriate.

The syntax for scalable entries is as follows:

```
font typeface_name [face][style][weight]
  internal [metricname metric_name]
  scalable hmi hmi_value vmi vmi_value
  [symbolset "symbol_set"] typeface typeface_value
  spacing {f | p} baseline baseline_offset
```

Where scaleable font entries are defined by using one or more of the following:

`font` *typeface\_name*

Specify the font keyword and the typeface name.

`face` This keyword is not necessary; it is supported for backward compatibility only.

`style` Specify a style from the following list:

- italic condensed compressed expanded  
inline outline shadowed
- weight* Specify one of the following:
- ultrathin|extrathin|thin|extralight|light|  
demilight|semilight|medium|semibold|demibold|  
bold|extrabold|black|extrablack|ultrablack
- metricname metricname*
- Specify the keyword and the *metricname* key to the associated metric definition located elsewhere in the file. This parameter is required only for proportional fonts. For fixed pitch fonts, the character widths are derived automatically from *hmi\_value*, so *metricname* is not required.
- hmi hmi\_value* Specify the *hmi* keyword and a value expressed in 1/120 of an inch, matching the values used in the PCL font selection sequence. This value may include up to 4 decimal places, and represents the hmi for a 100 point font.
- vmi vmi\_value* Specify the *vmi* keyword and value expressed in 1/48 of an inch, matching the values used in the PCL font selection sequence. This value may include up to 4 decimal places, and represent the VMI for a 100 point font.
- symbolset symbol\_set*
- Specify the *symbolset* keyword and a symbol set for the font. Defaults to "8U" or Roman-8 if not specified.
- typeface typeface\_value*
- Specify the *typeface* keyword and a value that indicates the design of the font as found in a font selection string (e.g., 52 for Univers, or 4148 if the Agfa family value of 4096 is included.) Both values select Univers.
- spacing f|p* Specify the *spacing* keyword and either F for fixed or P for a proportionally spaced font.
- baseline baseline\_offset*
- Specify the *baseline* keyword and the offset of the baseline from the character cell top. This value is required for correct positioning of characters, and represents the baseline for a 100 point font.

A sample scalable font definition:

```
font CGTimes internal metricname m_CGTimes100 scalable
  hmi 49.1696 vmi 79.6540 symbolset "8U" typeface 4101
  spacing p baseline 376
```

## Metric entries

```
metric metric_name
  laserjet [scalable] startchar first_char_value
  delta centi_point_width_list
```

The following example illustrates a metric font definition:

```
metric metric_name
```

Specify the keyword and a string to provide the link between the metric entry and the associated font entry.

*laserjet* Specify *laserjet* to indicate that these metrics are for a LaserJet or PCL printer. In the future, other keywords will allow for PostScript and other printers, which may have slightly different character width metrics for the same font.

*scalable* Specify *scalable* to indicate that the metrics are for a scalable typeface.

*first\_char\_value* Specify the decimal value of the first character for which character width metrics are provided. Typically this will be 32 for the space character. Lower valued characters in the Roman-8 symbol set are reserved as control characters.

```
delta centi_point_width_list
```

Specify the *delta* keyword and a list of character width values. The following considerations apply to this field:

- Each of these widths must be specified in centi-points (i.e., 1/100 of a point or 1/7200 of an inch).
- For scalable fonts, use widths for a 100 point font.
- For bitmapped fonts, use the metrics for the point size specified in the associated font entry.

## Adding Resident Fonts to FDUSER

To add a resident font to FDUSER, follow the steps listed below.

1. Determine whether the resident font is predefined by Unison Software in the files FD2000, FDIIXIII, FD5000, or FD4.
2. If it is, append all or part of one of these files to an existing FDUSER file.
3. If FDUSER does not exist, rename the desired file FDUSER. The available fonts are shown in the table that follows.

Once a resident font is added to FDUSER, Formation attempts to use the font whenever it prints an environment file that calls for that font. This can create problems if you use a single environment file with multiple types of LaserJets, and some of them don't have the particular resident font. To solve this problem, try the following possible workarounds:

1. Compile two versions of the environment file: one with the **Use printer resident fonts?** field set to Y, and one with it set to N. At print time, select whichever version is appropriate for the LaserJet in use.
2. Create a private font directory in your own group. This involves copying FDTYM into your group, creating an FDUSER file, and compiling them with FDCOMP to create a FONDIR file in your group. When you print from this group, this version of FONDIR is used.

**Note:** When you install a new release of Formation, recompile any private font directories you have created that are located outside the FORM.TYM group. Most users don't need to do this, as their information files (FDUSER and FDTYM) are in FORM.TYM, and those are recompiled automatically during the installation process.

## Using CheqSys Fonts

The file FDCHEQ.FORM.TYM contains the default definition for the CheqSys micr cartridge font used for check printing. You may append this file to FDUSER as with other predefined fonts.

In most cases you will need to use the extra thin font weight specified in the FDCHEQ file. If you have been advised to use another weight to match your printer tolerances, you may need to change the extrathin keyword. For more information refer to "FDUSER Entry Format" on page 5-6.

The following table lists the available resident fonts separated by printer type. The names correspond to what you type in the Name field on the **Character Set Control** screen. Note that these are not all the fonts resident in each printer, as some of the fonts are already in FDTYM.

**Note:** The Courier typeface in FD4.FORM.TYM is scalable. Do not use this entry when compiling for a LaserJet III, which uses a bitmapped Courier typeface rather than a scalable one. Some of the Courier point sizes displayed in FD4.FORM.TYM may not be available on the LaserJet III.



Table 5-3: Available Resident Fonts

FD2000.FORM.TYM Fonts resident in LaserJet 2000	FD11XIII.FORM.TYM Fonts resident in LaserJets IID and later	FD5000.FORM.TYM Fonts resident in HP 5000	FD4.FORM.TYM** Fonts resident in LaserJet 4
Prestige	Courier	Courier	Albertus
16.66cpi 7pt	Italic 10cpi 12pt	Italic 10cpi 12pt	Regular, Extra Bold
12cpi 10pt	12cpi 10pt	12cpi 10pt	Antique Olive
Bold 12cpi 10pt	Bold 12cpi 10pt	Bold 12cpi 10pt	Regular, Bold
Italic 12cpi 10pt	Italic 12cpi 10pt	Italic 12cpi 10pt	Italic
Courier		Prestige	Arial
12cpi 10pt		16.66cpi 7pt	Regular, Bold
Italic 10cpi 12pt		12cpi 10pt	Italic, Bold Italic
TmsRmn		Bold 12cpi 10pt	Clarendon Condensed
prop 8pt		Italic 12cpi 10pt	Bold
prop 10pt		LinePrinter	Coronet
Bold prop 10pt		12.91cpi 10pt	Italic
Italic prop 10pt		*CGTimes	Courier
Helv		prop 10pt	Regular, Bold
Bold prop 14pt		Bold prop 10pt	Italic, Bold Italic
		Italic prop 10pt	Garamond
		prop 12pt	Regular, Bold
		Bold prop 12pt	Italic, Bold Italic
		Italic prop 12pt	Letter Gothic
		*Univers	Regular, Bold
		Bold prop 14pt	Italic
			Marigold
			Regular
			CG Omega
			Regular, Bold
			Italic, Bold Italic
			Times New Roman
			Regular, Bold
			Italic, Bold Italic
			Univers Condensed
			Regular, Bold
			Italic, Bold Italic

\* Fonts marked with an asterisk are not scalable even though they are from the same family as those in the LaserJet III.

\*\* All FD4.FORM.TYM fonts are scalable, except the line printer font.

## Extract Font Metrics From the TFM file

Formation provides a utility for fonts not pre-defined by Unison Software. The PC-based, TFMTRANS . EXE utility:

- Reads the Tagged Font Metric (TFM) file for your resident font.
- Creates FDUSER text for each of the point sizes and typefaces you need.

This utility is placed in the same directory as Formation Design during installation.

TFM files are provided by Hewlett-Packard and other font suppliers for internal, cartridge and soft fonts available for LaserJet and HP 5000 printers. These files are also used by Type Director®, and are generally placed in the directory C:\AUTOFONT whenever a new font is added.

1. To create font and metrics definitions for a scalable typeface, enter the following command at the DOS prompt:

```
TFMTRANS S tfm_file_name output_file_name [facename]
```

For a bitmapped typeface, use this form:

```
TFMTRANS B tfm_file_name output_file_name [facename]
```

If the TFM file contains a typeface name string, by default TFMTRANS uses that name in the FDUSER entries it generates. However, some older TFM files do not contain typeface name strings. In this case, the *facename* parameter must be used. Also, some TFM files add characters to the typeface name to differentiate between bold, medium and italic. For example, if TFMTRANS is run on the Univers regular TFM file, it generates a typeface name of "UniversMd." You may want to use the *facename* parameter to identify the font as "Univers."

2. To create FDUSER text for a scalable CG Palacio font from the associated TFM file:

```
tfmtrans s \autofont\par0000s.tfm par.txt CGPalacio
```

This results in the following text being placed in the file PAR.TXT:

```
FONT CGPalacio INTERNAL METRICNAME m_CGPalacio100 SCALABLE
HMI 55.3111 VMI 79.6540 TYPEFACE 4111 SPACING p BASELINE 392

METRIC m_CGPalacio100 LASERJET STARTCHAR 32 DELTA
3319 3319 4610 4979 4979 8851 7744 3319 3319 3319 4979 8851
3319 3319 3319 3319 4979 4979 4979 4979 4979 4979 4979 4979
4979 4979 3319 3319 9957 8851 9957 4425 8851 7744 6085 7192
7744 6085 5532 8297 8297 3319 3319 7192 6085 9404 8297 8297
6085 8297 6638 6085 6085 7744 7192 9957 6638 6638 6638 3319
3319 3319 4979 4979 3319 4979 5532 4425 6085 4979 3319 5532
6085 2765 2765 5532 2765 8851 6085 5532 6085 5532 3872 4425
3319 6085 5532 8297 4979 5532 4979 4425 4979 4425 9957 9957
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 8851
```

---

```
8851 4979 3319 3319 3319 8851 4979 4979 0 7744 7744 6085
6085 6085 3319 3319 4979 4979 4979 4979 4979 7744 7744 4979
4979 6638 5532 4979 7192 4425 8297 6085 3319 4425 4979 4979
4979 4979 4979 4979 4979 4979 5532 6085 4979 4979 5532 6085
4979 4979 5532 6085 4979 4979 5532 6085 7744 2765 8297 10142
4979 2765 5532 7929 7744 2765 8297 7744 6085 2765 5900 8297
7744 7744 4979 7744 5532 3319 3319 8297 8297 8297 5532 6085
4425 7744 6638 5532 6085 6085 1107 5532 4979 8851 8851 8851
8851 3872 3872 4240 7192 4240 8851 0
```

### Update FDUSER and Recompile

Combine the entries generated using TFMTRANS in a file called FDUSER. FORM.TYM, or append them to the existing file. Then log on in FORM.TYM, and recompile the font directory:

```
:RUN FDCOMP
```

## Converting PCL Fonts to PCELL Files

PCL is the font file format used by LaserJets and HP 5000s. A PCELL font is an HP 3000 font file format which can contain multiple point sizes and resolutions. Because PCL fonts cannot be printed on the HP 2680/88, PCELL font files provide greater flexibility when using a 2680 or 2688 in addition to LaserJets and 5000s. Because PCELL files can contain multiple font sizes, they provide greater convenience, even for those who use only LaserJets and 5000s.

To convert PCL fonts to PCELL font files, use the utility PCL2DCS.FORM.TYM, as follows:

1. Upload your PCL font to the HP 3000.
2. Enter the following commands to convert the file:

```
:FILE CHARSET=sourcePCLfilename
```

```
:FILE PCELL=targetPCELLfilename
```

```
:RUN PCL2DCS.FORM.TYM[;PARM=1] Use PARM=1 for 180 dpi PCL fonts only.  
Do not use for 300 dpi fonts.
```

You can run PCL2DCS multiple times, adding a single PCL font (one point size in one resolution) at a time. To save time, set up the commands in a job stream.

In general, a single PCELL file should only contain fonts of the same typeface, weight and style, with only the point size varying. For a given resolution you can only store one font of each point size in a PCELL file. You could not place 300 dpi 12 point italic and 12 point bold in the same file.

If you want to be able to reference the font by its typeface name and characteristics, add a font definition to FDUSER.

## Convert Utility

Formation's `Convert` utility converts TIFF (Tagged Image File Format) files to either PCELL or raster files. Converting TIFF files to PCELL files allows you to:

- Create logos for `IDSForm`.
- Create signatures that print as data characters.
- Create HP 3000 PCELL files for use in electronic forms.

Converting TIFF files to raster files allows you to:

- Create graphics for use with the raster graphics intrinsics.
- Create HP 3000 raster format files for inclusion in reports and documents. You can use the resulting raster or PCELL file like any other HP 3000 raster or PCELL file.

TIFF is a standard file format for raster graphics. Most scanner software can produce output in TIFF format. Some personal computer "paint" programs and several Macintosh programs can save output in TIFF format.

You can use many different scanners to digitize images for placement in forms or documents. When scanning a logo for both an HP 2680 and an HP 2688/LaserJet, scan once at 180 dots per inch and also at 300 dots per inch.

Both logos can be placed back into the same PCELL file for easy reference.

To use many of the conversion features offered by the `Convert` utility, you need to upload the soft font, form, or scanned image from your personal computer to the HP 3000. To do this, use a terminal emulation program of your choice.

**Note:** Formation places a few restrictions on the file that will be created on the HP 3000. The file must have fixed-length records, and must have a record length no longer than 1024 bytes. The optimum record size is 256 bytes.

### Converting Logos to Character Formats

To print a logo or signature as data on a document, convert it into a PCELL file. Logos print much faster on LaserJet printers, when they are stored as characters in a PCELL file, than when they are stored as raster images. Logos must be stored as character images if you are going to use them with `IDSForm`.

Use the **Logo Management** screen to convert logos to character formats. Based on specifications you define, Formation takes a logo and splits it into smaller units. Each unit is assigned a character in the PCELL file.

To print the logo, specify the PCELL file as a font in an environment file. The characters defined at conversion can then be included as application data, and printed to create the logo.

### Running the Convert Utility

To run the `Convert` utility, enter the following:

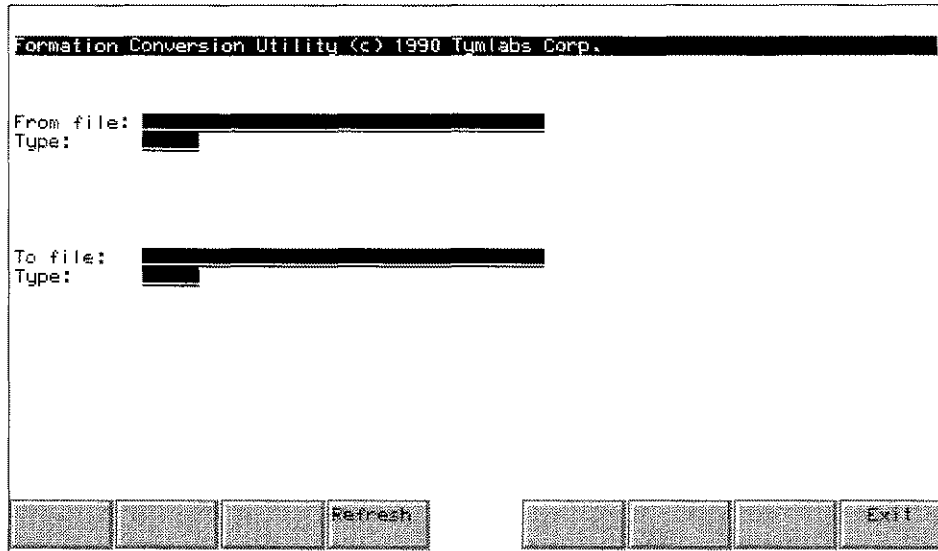
```
:RUN CONVERT.FORM.TYM
```

The `Convert` utility banner appears, followed by the Formation Conversion Utility screen.

## Formation Conversion Utility Screen

### Purpose

The Formation Conversion Utility screen specifies a TIFF file to be converted, and defines the resulting PCELL or raster file. Enter the name of the original TIFF file in the From file field, and TIFF in the Type field. Enter a name for the new file in the To file field, and either RASTR or PCELL in the Type field.



### Description

To convert to a raster file, fill in the fields and press **e**. For HP 2680 files, use a 180 dpi TIFF file, and make certain the HP 2680 graphics option is available. For LaserJet files, use a 300 dpi TIFF file. If you are converting to a PCELL file, press **e**. The **Logo Management** screen is displayed. If you are converting to a raster file, press **e** and the conversion is completed. If the **From** file you enter does not exist, an error message is displayed.

### Field Definitions

- From File** Enter the name of the TIFF file to be converted. Use an existing file name.
- Type** Enter TIFF to indicate a TIFF file.
- To File** Enter a file name. This is the name of the new PCELL or raster file created during the conversion.
- Type** Enter PCELL if you are converting the file to a PCELL file. Enter RASTR if you are converting the file to a raster file.

### Function Keys

Key	Name	Description
4	Refresh	Redisplay the menu and data from the last saved state.
8	Exit	Exit the Convert utility and return to the MPE colon prompt.

## Logo Management Screen

### Purpose

The **Logo Management** screen converts TIFF files into characters. To access this screen, enter a TIFF source file and a PCELL destination file in the **Formation Conversion Utility** screen, and then press **e**.

```

Formation Conversion Utility (c) 1990 Tymlabs Corp.

Image size: 307x304 Point size: 72.96001
Large cells needed: 4 (2x2)
Small cells needed: 9 (3x3)

----- Conversion Options -----
Convert to: 2680 80 - 2680
            88 - 2688 or LaserJet
First character: 1
Logo file? 111
Use small cell? 111 Y - logo/cell will be used on LaserJet Plus
                   N - logo/cell will be used on newer printers
                   (or on 2680/2688).

Refresh Main Menu
  
```

### Description

The top portion of the **Logo Management** screen displays information about the TIFF file you are converting. This includes the actual image size of the logo, the total point size (height in points) of the logo, and how many character cells are needed to place the logo on a form. Logos are always printed as a rectangular array of characters.

The **Conversion Options** section of the screen contains fields to define how to convert the file. Using these fields, you tell Formation which printer to convert the file for, which characters to use in this file, and what size cells are required.

Some printers have a limit to the size of a single character. This is why the logo must be split into cells. For HP 2680/88 printers, LaserJet Series II family printers, and the LaserJet 2000 family printers, the limit is 255 characters. Files created for these printers can use large cells. LaserJet Plus and LaserJet 500+ printers have a maximum cell size of 128 dots, and therefore, require small cells.

A logo may consist of no more than 95 cells. That limits the maximum size of a LaserJet Plus/500+ logo to about 25 square inches (162 cm<sup>2</sup>).

Once you convert the file to PCELL, and assign the necessary characters, you can place the logo on a form. To do this, define a font in an environment that references this PCELL file. Activate the font within your application, and then enter the logo characters as data. The logo is printed on the form in place of the data. When you enter the characters to make up the logo, use the dimensions specified in the number of cells needed field (e.g., 3 x 4 indicates that you must enter 4 rows of 3 characters each.)

You can place multiple logos or signatures into the same PCELL file. If you do this, start each new logo with a subsequent character. If you use the same character as another logo, you will overwrite the original logo.

To convert your logo to a PCELL file, fill in the fields and press **e**.

### Field Definitions

**Image size**      Displays the width and the height of the logo in scanned dots.

**Point size**      Displays the point size of the logo in scanned dots.

**Large cells needed**  
 The number of large cells that are needed to define the entire logo. The value displayed is the number of cells that will be created if you enter N in the **Use small cell?** field. The dimensions of how the characters should be entered is also displayed (e.g., 3x4 indicates 4 rows of 3 characters each).

**Small cells needed**  
 The number of small cells that are needed to define the entire logo. The value displayed is the number of cells that will be created if you enter Y in the **Use small cell?** field. The dimensions of how the characters should be entered is also displayed (e.g., 3 x 4 indicates 4 rows of 3 characters each).

**Convert to**      Enter 80 to convert the logo to 180 dpi characters (HP 2680), or 88 to convert the logo to 300 dpi characters (HP 2688 or a LaserJet). The **Convert** utility provides a default value in this field based on the scanning resolution of the TIFF file.

**First Character**  
 Enter a character to assign to the first (upper left) or only cell in the file. This can be any printable character, but it is recommended you begin with A for consistency. The **Convert** utility then checks to be sure that there are enough consecutive characters to fit the logo, and assigns the necessary characters to the remaining cells. For example, if the **Convert** utility tells you that the logo requires 12 cells as a 3 x 4 logo, and you enter A as the first character, the following sequence prints the logo:

```

ABC
DEF
GHI
JKL
    
```

**Logo file?**      Enter N in this field unless a Unison Software customer support representative advises otherwise.

**Use small cell?** Enter Y in this field to convert the logo into small cells (LaserJet Plus or LaserJet 500+ printers only). Enter N to convert the logo into large cells (all other printers).

### Function Keys

Key	Name	Description
4	Refresh	Redisplay the menu and data from the last saved state.
8	Main Menu	Return to the <b>Formation Conversion Utility</b> screen.



---

## FIGPRINT Utility

### Purpose

The FIGPRINT utility prints figure files from graphics packages. A figure file can contain more than one figure. The default figure name of most packages is the same as the name of the file.

In the following instructions, the examples shown assume that you saved your chart as MYCHART. Use the following instructions to print a figure from one of these packages.

### Running FIGPRINT

You can then run the FIGPRINT utility as follows:

```
:FILE LASERJET;DEV=laserjet  
:RUN FIGPRINT.FORM.TYM;INFO="MYCHART:MYCHART"
```

Your figure is sent to the printer that you specified in the file equation.

The FIGPRINT utility can be used to print any figure in any accessible figure file. In general, you can run FIGPRINT as follows:

```
:RUN FIGPRINT.FORM.TYM;INFO=figfile:figname
```

### Example

Formation supplied a sample figure file for testing the FIGPRINT utility. To print the figure PLANT1 in the HP-supplied figure file PLANTS.FIGURE.SYS, enter:

```
:RUN FIGPRINT.FORM.TYM;INFO=PLANTS.FIGURE.SYS:PLANT1
```

## Data Font Considerations

When selecting a data font consider the readability of the data font in relation to form elements, font spacing, and printer limitations. In addition, keep the following design considerations in mind:

- Use a different font for headings and data. Using different fonts separates the data from static form elements.
- Monospaced (fixed space) fonts allow you to easily calculate how much space is required on the form to fit the data. Each character is exactly the same width.
- Font choice is limited by the font capability of your printer. You can select any of the supplied fonts, or any fonts you add to the font directory.
- Formation Design allows you to design your form around a selected font. For more information about the Formation Design program, read the *Formation Design User Guide*.

The available fonts and their allowed sizes are listed in the tables that follow.

Table 5-4: Courier and Prestige Fonts

Point Size	Characters Per Inch (Pitch)		Lines Per Inch (Typical)	
	300 dpi	180 dpi	300 dpi	180 dpi
4.00	30.00	30.00	15.00	15.00
4.50	30.00	30.00	13.64	13.85
5.00	27.27	25.71	12.00	12.00
5.25	25.00	25.71	11.54	11.25
5.50	23.08	22.50	11.11	11.25
6.00	21.43	22.50	10.00	10.00
6.50	20.00	20.00	9.38	9.47
7.00	18.75	18.00	8.57	8.57
7.25	17.65	18.00	8.33	8.18
7.50	16.67	16.36	8.11	8.18
8.00	15.79	16.36	7.50	7.50
8.50	15.00	15.00	7.14	7.20
9.00	14.29	13.85	6.67	6.67
9.25	13.64	13.85	6.52	6.43
9.50	13.04	12.86	6.38	6.43
10.00	12.50	12.86	6.00	6.00
10.50	12.00	12.00	5.77	5.81
11.00	11.54	11.25	5.45	5.45
11.25	11.11	11.25	5.36	5.29
11.50	10.71	10.59	5.26	5.29
12.00	10.34	10.59	5.00	5.00
12.50	10.00	10.00	4.84	4.86
13.00	9.68	9.47	4.62	4.62
13.25	9.38	9.47	4.55	4.50
13.50	9.09	9.00	4.48	4.50
14.00	8.57	8.57	4.29	4.29

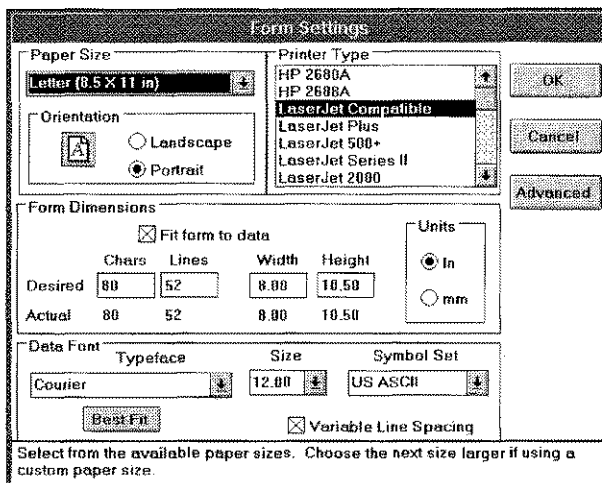
Table 5-5: Letter Gothic Font

Point Size	Characters Per Inch (Pitch)		Lines Per Inch (Typical)	
	300 dpi	180 dpi	300 dpi	180 dpi
4.00	37.50	36.00	15.00	15.00
4.50	33.33	30.00	13.64	13.85
5.00	30.00	30.00	12.00	12.00
5.50	27.27	25.71	11.11	11.25
6.00	25.00	25.71	10.00	10.00
6.25	23.08	22.50	9.68	9.47
6.50	21.43	22.50	9.38	9.47
7.00	20.00	20.00	8.57	8.57
7.50	18.75	20.00	8.11	8.18
8.00	17.65	18.00	7.50	7.50
8.50	16.67	16.36	7.14	7.20
9.00	15.79	16.36	6.67	6.67
9.50	15.0	15.00	6.38	6.43
10.00	14.29	15.00	6.00	6.00
10.50	13.64	13.85	5.77	5.81
11.00	13.04	12.86	5.45	5.45
11.50	12.50	12.86	5.26	5.29
12.00	12.00	12.00	5.00	5.00
12.50	11.54	11.25	4.84	4.86
13.00	11.11	11.25	4.62	4.62
13.50	10.71	10.59	4.48	4.50
14.00	10.34	10.59	4.29	4.29

### Assigning a Data Font

The font in which your data prints can be selected in any of the following ways:

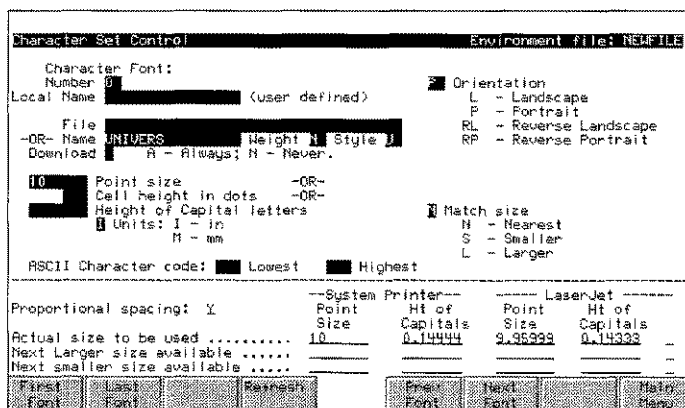
1. Select the data font in the Formation Design **Form Settings** dialog box as you're designing the form.



Select a typeface and font size from the drop down lists to define a data font.

Upload the form to the HP 3000. Create an environment file with either the MAKEENV or CREATEENV UDC that includes the form file as the primary form. When the form is compiled, Formation automatically assigns font number 0 to the named data font.

2. Use the ADDFONT UDC to add a font as described in "ADDFONT UDC" on page 3-3.
3. Use the Environment Manager **Character Font Control** screen as shown in the following figure.



Specify your data font by entering a fully-qualified file name in the File field or using a typeface name in the Name field. If you specify a typeface name, you can also enter values for weight, style and point size.

If you specify your data font using a typeface name in the Name field, you can also specify values in the following fields:

Table 5-6: Data Font Values

Field	Value	Meaning
Weight	N	Normal
	-7 to 7	- 7 = lightest 7 = darkest
Style	B	Bold
	U	Upright
	I	Italic

### Selecting an HP 3000 Data Font

Your Formation shipment includes a chart (*Formation File List*) listing all HP 3000 files shipped with the product. You may wish to choose a data font from those shipped in CHARSETS.SYS. File names which contain an "88" are fonts which work well on the LaserJet or HP 2688 printers. The others are either HP 2680 fonts or bar codes.

The HP 2680 prints 180 dots per inch (dpi), and the only fonts you can use with this printer are those designed for that resolution. The HP 2688 and LaserJet printers print 300 dpi. Though you get the most predictable results when you use 300 dpi fonts with these printers, LaserJet and HP 2688 users can also use fonts designed for 180 dpi (2680). Be aware that the scaling required may affect the appearance of the letters.

Some of the fonts in the *Formation File List* are described in terms of dpi, point size, and pitch (characters per inch).

### SIZELIST Utility

You can use the utility program SIZELIST.FORM.TYM, to find out what point sizes are in any PCELL font file.

1. Log on in the group where your fonts reside.
2. Run the SIZELIST utility:  
:RUN SIZELIST.FORM.TYM
3. Supply the name of any font file at the prompt.  
File? crr7r3.fonts.tym
4. The utility returns the point sizes in that file for 2680 fonts and/or 2688 fonts. Although not stated in the listing, the 2680 fonts are 180 dpi and the 2688 fonts are 300 dpi.

## FONTDOWN Utility

### Purpose

The FONTDOWN utility downloads PCL fonts to LaserJet printers. Once fonts are downloaded to the printer with this utility, they do not need to be downloaded with the environment file. This saves time when downloading environment files, and is useful for frequently used fonts.

### Running FONTDOWN

To use the FONTDOWN utility for a single PCL file:

```
:FILE PRINTER;DEV=laserjet
:FILE FONTFILE=pclname
:RUN FONTDOWN;PARM=fontid
```

where:

*laserjet* Any Formation LaserJet printer value (e.g., LJIII). The allowed values are listed in the table on page 3-18.

*pclname* The name of the PCL font file to be downloaded.

*fontid* The internal font number recognized by the printer as the specified font.

To use the FONTDOWN utility to download multiple PCL files:

```
FILE PRINTER;DEV=laserjet
RUN FONTDOWN;INFO="filename"
```

where:

*laserjet* Any Formation LaserJet printer value (e.g., LJIII). The allowed values are listed in the table on page 3-18.

*filename* The name of a text file containing the PCL font file name and id number of each font to be downloaded:

```
pclname1    fontid1
pclname2    fontid2
...
```

Once a font is downloaded to the printer, set the **Download** field in the **Character Set Control** screen to N. This tells the Environment Manager not to download the current font as it already resides in the printer.

## Bar Code Fonts

A bar code is a printed machine-readable code that is made up of parallel bars of varied width and spacing. The Formation package includes three standard bar code fonts:

- 3 of 9 bar code (code 39)
- 2 of 5 bar code
- Postnet

The supplied bar code fonts are described in detail in the following sections. The following diagram illustrates a 3 of 9 bar code.



### Code 39 Specification

Code 39 is a discrete, variable length, self-checking alphanumeric symbology employing two-element widths. Every character has five bars and four spaces. Three of the elements in any given character are wide, and six are narrow. Every symbol starts and ends with a unique start/stop character. A summary of Code 39's main characteristics is as follows:

Character Set:	26 uppercase letters 10 digits 7 special characters Extendable to full 128 ASCII character set using a two-character precedence code scheme.
Symbol Length:	Variable
Check Characters:	Optional
Overhead Characters:	2 per symbol
Other Features:	Concatenation ability
Net Data Density:	Maximum of 9.8 characters per inch when printed using a 7.5 mil X dimension.

### Code 39 Symbol Description

Each Code 39 symbol consists of:

1. Leading quiet zone
2. Start character
3. Data characters
4. Optional check character
5. Stop character
6. Trailing quiet zone

### Code 39 Check Character

An optional check character is defined for code 39 applications requiring higher levels of data security. When used, the check character is positioned between the final data character and the stop character.

The check character is determined as follows:

1. Using Table 5-7, "Code 39 Character Set," on page 5-29, determine the numerical value of each of the data characters.
2. Sum the values of all the data characters.
3. Find the module 43 remainder of the sum from step 2.
4. The check character is that character whose value from Table 5-7, "Code 39 Character Set," on page 5-29 is equal to the remainder from step 3.

For example, if the data characters are "TEST," the values would be summed as:

$$29 + 14 + 28 + 29 = 100$$

Divide 100 by 43 to get 2, with a remainder of 14. The letter "E" has a value of 14, and is therefore the check character.

### Code 39 with Formation

The code 39 bar code fonts included with Formation for the LaserJet/HP5000 printers are:

B39W0988.CHARSETS.SYS(9 point)

B39W1288.CHARSETS.SYS(17 Point)

By creating an environment file that contains your selected "human readable" character set as font #0, and your selected code 39 character set as font #1, you can use the Shift-Out, Shift-In control codes to switch between the desired data fonts.

An example of a combined human readable and machine readable bar code would be printing the part number 1234 in both human readable and code 39 on the same line, as follows:

1234 [Shift-Out]\*1234\*[Shift-In]

Where:

- |             |                                                                                                                                              |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| 1234        | Prints in the human readable data font in the environment file (Font #0).                                                                    |
| [Shift-Out] | Is Control-N (Decimal 14). This key sequence causes the printer to switch to the bar code font in the environment file (Font #1).            |
| *           | Is the starting and ending character for the included bar code fonts.                                                                        |
| [Shift-In]  | Is Control-O (Decimal 15). This key sequence causes the printer to switch back to the human readable font in the environment file (Font #0). |



Table 5-7: Code 39 Character Set

Character	Value	Character	Value	Character	Value
0	0	E	14	S	28
1	1	F	15	T	29
2	2	G	16	U	30
3	3	H	17	V	31
4	4	I	18	W	32
5	5	J	19	X	33
6	6	K	20	Y	34
7	7	L	21	Z	35
8	8	M	22	-	36
9	9	N	23	.	37
A	10	O	24	Space	38
B	11	P	25	\$	39
C	12	Q	26	/	40
D	13	R	27	+	41
				%	42

**Note:** Some of the tabular information is from: *The Bar Code Book* by Roger C. Palmer, Helmers Publishing, Inc., 1989.

## 2 of 5 Bar Code Checksum and Interleaving

The following instructions assume that the 2 of 5 bar code characters are:

- A Narrow space (NS)
- B Wide space (WS)
- C Narrow bar (NB)
- D Wide bar (WB)

The 2 of 5 encoding interleaves pairs of characters with black bars representing the first character in the pair, and white bars for the second. Optionally, there may be a single character check sum digit added to the right end of the string. The check sum is calculated as follows:

1. Assume that the right-most end character of the string is in an even position.
2. Sum the odd digits.
3. Sum the even digits and multiply by 3.
4. Add the two sums.
5. The check digit is the smallest number which when added to the result of step 4, results in a multiple of 10. It is always a single digit:

$$\text{digit} = 10 - (\text{result mod } 10)$$

The interleaving requires that an even number of characters be encoded. If the data string plus the checksum digit is an odd number of characters, a leading zero (0) character is added.

Each character is represented by five bars. The string is divided into pairs. The bar code is assembled by taking the first bar of the first character, then the first bar of the second character, then the second bar of the first character, then the second bar of the second character, etc. Each bar is printed by using one of the four 'characters' in the small bar code character set. The bar code characters to use for each encoded digit are as follows:

Table 5-8: Bar Code Character Set

Digit	Pattern	Odd Pos. (1st char)	Even Pos. (2nd char)
0	00110	CCDDC	AABBA
1	10001	DCCCD	BAAAB
2	01001	CDCCD	ABAAB
3	11000	DDCCC	BBAAA
4	00101	CCDCD	AABAB
5	10100	DCDCC	BABAA
6	01100	CDDCC	ABBAA
7	00011	CCCDD	AAABB
8	10010	DCCDC	BAABA
9	01010	CDCDC	ABABA

The character pair 23 combines CDCCD with BBAAA to give CBDBCACADA. Lastly, a start and stop marker must be added to the output string. The start marker is "CACA" and the stop marker "DAC".

## Postnet with Formation

The Postnet bar code font included with Formation for the LaserJet/HP5000 printers is:

UP07A1.FONTS.TYM

By creating an environment file that contains your selected "human readable" character set as font #0, and the postnet bar code as font #1, you can use the simple Shift-Out and Shift-In control codes to switch between the desired data fonts. An example of this would be printing the part number 12345 in both human readable and postnet on the same line as follows:

12345      [Shift-Out]/123455:[Shift-In]

Where:

12345	Prints in the human readable data font in the environment file (Font #0).
[Shift-Out]	Is Control-N (Decimal 14). This will cause the printer to switch to the bar code font in the environment file (Font #1).
/	Is the starting character and : is ending character in this font.
[Shift-In]	Is Control-O (Decimal 15). This will cause the printer to switch back to the human readable font in the environment file (Font #0).

**Note:** The second 5 digit is the check digit required in the use of this font. The check digit is calculated as the difference of the sum of the 5 or 9 zip code digits from the next higher multiple of 10. If the sum of the digits is a multiple of 10, the value of the check digit is zero.

### Example

$1+2+3+4+5=15$

$20-15=5$

For complete information on the use of the Postnet bar code symbology in postal applications, request the Official U.S.P.S. guide from your local post office.



## *Intrinsics*

The Formation programmatic interface allows you to write application programs that exercise close control over the operation of the LaserJet. Once you create an environment file, or choose one supplied by Unison Software (or Hewlett-Packard), you can supply additional instructions to the LaserJet by calling a series of *intrinsics* within your program.

**Note:** You do not have to use the Formation intrinsics to print data on forms. You need only specify the environment file name in the file equation. The intrinsics programmatic interface provides additional control over the operation of the printer.

Using the programmatic interface, you can control the printing process based on conditions you determine at run-time. You can do any of the following:

- Select various forms for each page.
- Select a font in which to print a variable based on the value of that or some other variable.
- Fill out different blanks on a form.

When your program fills out a form, it can do so symbolically by specifying the name of a field and the value that is to be placed in it. This makes your forms independent of your data and application. You can change your form without having to change the application program. You can also print graphics, as long as you have HP or other graphics software.

Many of the Formation intrinsics perform functions similar to HP's PSP intrinsics. In these cases, their names are the same as the HP intrinsics. Other intrinsics are unique to Formation; they have no corresponding PSP intrinsics. Refer to "Using Intrinsics In a Program" on page 6-16 for an example on how to use Formation intrinsics within a program.

### **Topics**

This chapter describes the Formation intrinsics and provides some guidelines for incorporating them into your program. Topics include:

- Capabilities Available with Intrinsics
- Converting Programs for Use with Formation Intrinsics
- Calling Intrinsics
- Using Intrinsics In a Program
- Intrinsics Listing
- Intrinsics HP36580 through

## Capabilities Available with Intrinsic

Formation intrinsic provide greater flexibility in manipulating printer output. Capabilities that are available using the intrinsic include:

- Specifying multiple fonts within form data
- Controlling the physical page
- Using smart fields in your forms
- Controlling the logical page
- Rotating form text
- Printing graphics

The following is a brief explanation of how to use some of the intrinsic. Each intrinsic is explained in detail later in this chapter. Refer to the appropriate section header for information on individual intrinsic.

### Initializing Intrinsic

An initialization intrinsic must be called before using any other Formation intrinsic. There are two such intrinsic available: `PINITDEVICE` and `PINITIALIZE`. Both of these intrinsic establish the `comArea` and define the calling language. `PINITDEVICE` also allows you to specify an output device when using a LaserJet. `PINITIALIZE` is provided for compatibility with earlier systems.

### Multiple Fonts within Data

Using the intrinsic, you can have a maximum of 32 data fonts in any given form set. Two fonts are initially named in an environment file. To change between these fonts, use the shift-in (primary font) and shift-out (secondary font) escape sequences.

The `PUSEFONT` intrinsic allows you to change to a different primary and secondary font. By issuing a series of `PUSEFONT` commands throughout your program, you can change the data font as desired. If you do not know the actual number assigned to a font, use the `PFONTNUM` intrinsic to determine the number based on the name.

### Controlling the Physical Page

Several Formation intrinsic are provided that allow you to control the attributes of the physical page. This includes forcing page ejects, defining number of copies, resetting the LaserJet printer, separating bin output, and controlling duplexing options from within your program.

The `PNEWPHYSPAGE` intrinsic returns to the first active logical page in an environment, and forces a page eject. The `PSETCOPIES` intrinsic allows you to define the number of copies to print of each page. The `PJOBSEPARATE` intrinsic controls the output to the paper bin on an LJIIIsi. The `PDUPLEX` intrinsic allows you to control whether output on a LaserJet is duplex or simplex.

### FMDUPLEX JCW

The `FMDUPLEX JCW` allows users and job streams to override the default duplex status of an application which uses the Formation intrinsic. It may be used to force a simplex application to print duplex, or a duplex application to print simplex. It is also used to specify long edge or short edge binding, something which was not possible through the Formation intrinsic prior to version 3.4.

The setting of FMDUPLEX is overridden by any calls to the PDUPLEX intrinsic.

**Table 6-6: FMDUPLEX JCW Values**

Value	Meaning
0	Prevent duplexing
1	Duplex with long edge binding
2	Duplex with short edge binding

All other values are ignored.

**Note:** Duplex printing is not supported by all printer models. If FMDUPLEX is used in conjunction with a printer which does not support duplex printing, it has no effect.

### Smart Fields

Formation smart fields allow you to change the characteristics of your forms without modifying your application. This is because data automatically prints to a given field, regardless of the field's location on the form. Therefore, the field can be moved to different areas of the form, and the corresponding data is always printed in that field.

Smart fields are placed on your Formation forms using the smart field tool of Formation Design. The name you assign to the smart field is the name you will reference when using the intrinsic to place data in that field. Smart fields on different forms can be given the same name to reference the same application data. Refer to the *Formation Design User Guide* for more information on creating forms with smart fields.

To write data to a smart field, use one of the following intrinsic:

- PWRITEFIELD
- PWRITEFIELDREL
- PWRITEFIELDJUST

The PWRITEFIELD intrinsic tells Formation to write data into the field with the specified name. The PWRITEFIELDREL intrinsic tells Formation to write data into the field relative to the specified field. The PWRITEFIELDJUST intrinsic tells Formation to write data into the specified field with right, left or centered justification.

Smart fields can be divided into smaller units called subfields. Separate PWRITEFIELD intrinsic can be included in your program to write different bits of data into each subfield. Include the name of the smart field and the number of the subfield into which information should be written with the intrinsic. The data will automatically print in the desired area on the form. Subfields are useful for such things as address boxes, where you need different parts of the data printed on individual lines.

### Controlling the Logical Page

Formation provides several intrinsic that can be used to control logical pages in your environments. The PACTIVATEPAGE intrinsic is used to activate a specified logical page. Logical pages can be deactivated with the PDEACTIVATEPAGE intrinsic.

The PNEWPAGE, PSELECTPAGE, and PSWITCHPAGE intrinsic are used to switch to a new logical page. PNEWPAGE is used to advance to the next logical page. PSELECTPAGE is used to activate and advance to a specified logical page. Finally, PSWITCHPAGE is used to switch to a logical page without performing a physical page eject.

## Graphics Capabilities

Formation's graphic intrinsic can be used to print raster images. The PLOADRASTER and PPRINTRASTER intrinsic load a raster image into printer memory and then print the image. PFLASHRASTER loads, prints, and then deletes raster images from the printer's memory. PPRINTFIGURE converts figures to raster, and then prints the raster image.

## Printing Rotated Text

Rotated text is often used to label areas of a form, as shown in the following example.

SHIP TO:

Name and Address	Mr. John Smith Director of Manufacturing Acme Company MS 560K 12000 Technology Blvd. Dallas, Texas 75299
------------------	-------------------------------------------------------------------------------------------------------------------------

Rotated text can be included on a form as a TIFF image or as part of the data text. You can only print rotated data text when printing to a LaserJet III, LaserJet 4 (or other PCL5 printer), an HP 2680 or an HP 2688. However, if you have form files created with IDForm that contain rotated form text, these files can be recompiled with Formation and printed on any printer.

To print rotated text, use the Environment Manager to orient a font so that it goes up or down the page instead of across as follows:

- Run the Environment Manager.  
:RUN ENVMGR.FORM.TYM  
The **Environment File Menu** is displayed.
- Type the environment file name and press **e**.
- Type **C** at the **Main Menu** to access the **Character Set Control** screen. The new font number is displayed in the **Number** field. Keep track of this number as you will need it for the PUSEFONT intrinsic in your print program.

Font number  
For rotated text, set up a font with an orientation opposite to the rest of the form. See chart next page.


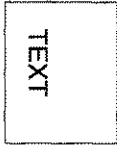

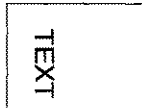
The screenshot shows the 'Character Set Control' window with the following settings:

- Character Font: Number 0
- Local Name: (user defined)
- File: (redacted)
- Weight: (redacted)
- Style: (redacted)
- Download: A - Always; N - Never.
- Point size: (redacted)
- Cell height in dots: (redacted)
- Height of Capital letters: (redacted)
- Units: I - in, H - mm
- ASCII Character code: (redacted) Lowest, (redacted) Highest
- Orientation: L - Landscape, P - Portrait, RL - Reverse Landscape, RP - Reverse Portrait (circled in red)
- Match size: N - Nearest, S - Smaller, L - Larger
- Proportional spacing: -
- System Printer: Point Size, Ht of Capitals
- LaserJet: Point Size, Ht of Capitals
- Actual size to be used: .....
- Next Larger size available: .....
- Next smaller size available: .....
- Buttons: First Font, Use Font, Refresh, Prev Font, Next Font, Exit

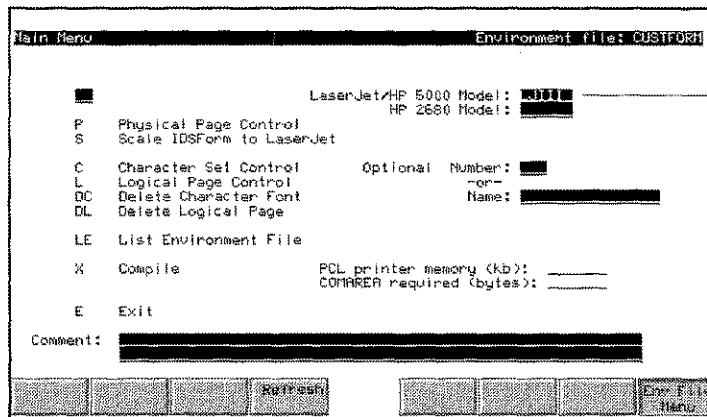


- Place a letter in the **Orientation** field to indicate the desired orientation. Use the chart below to determine the correct letter.

Table 6-7: Page Orientation

Form Orientation	Text To Go...	Font Orientation
Portrait	Up 	L (landscape)
	Down 	RL (reverse landscape)
Landscape	Up 	P (portrait)
	Down 	RP (reverse portrait)

- Return to the **Main Menu**, and then compile the environment file.



*Make sure LJIII or LJ4 is entered here.*

You must tell the Formation print routines that you are using a LaserJet III, IIID or 4. You can do this in your program with the PINITDEVICE intrinsic, or you can set FMFORCELJ at run-time to 3000, 3001, 13000, 13001, 4000, 4001, or 14000. See "JCW Setting for Converted Programs" on page 6-9. In your program, use the PUSEFONT intrinsic to call the rotated font you set up in your environment file. Use the PWRITEFIELD intrinsic or the write statement of your programming language to position the cursor and write the data. If you are familiar with setting up rotated data text for the HP 2680/88, this is the same procedure used with IFS.

The following code fragment shows how rotated text might be produced in a COBOL program.

```
*
*Initialize the intrinsic and specify that the target device is a LaserJet III.
*
    Move 2000 to AreaLen.
    Move Zero to Language.
    Move "LJSIII " to OutputDev.
    Call "PINITDEVICE" using ComArea,
                          AreaLen,
                          Language,
                          PrintFile,
                          OutputDev.

    Perform Check-For-Error.
Intrinsic-Exit.
Exit.
*
* Before you write the rotated text, switch your primary font to the font number
* of the appropriately-oriented font in the environment file. For instance, here
* font #2 has been specified as landscape in the environment file. It prints
* as rotated text on a portrait form.
*
Form-Print.
*
* Select primary font 2; don't change secondary font
*
    Move 2 to PrimaryFont.
    Move -1 to SecondaryFont.
    Call "PUSEFONT" using ComArea,
                          PrimaryFont,
                          SecondaryFont.
*
* Write to the fields on the form.
*
    Move 1 to SubFieldNum
    Move Spaces to FieldName
    Move "FIELD1" to FieldName
    Move Spaces to Info
    Move "Name and Address" to Info
    Move 17 to InfoLen
    Call "PWRITEFIELD" using Comarea,
                          FieldName,
                          SubFieldNum,
                          Info,
                          InfoLen.

    Perform Check-For-Error.
```

## Converting Programs for Use with Formation Intrinsic

Once installed, most Formation intrinsic appear in your system SL with an apostrophe as the next to last character in their names. Your compiled programs must use these Formation intrinsic rather than the HP PSP intrinsic. To have your programs do this, they must be converted for Formation as follows:

### MPE V Users

Run the following command against your program to change the external reference list in your program to call the Formation intrinsic:

```
JETSTART.FORM.TYM
```

### MPE XL Users

For compatibility mode programs, follow the previous instructions for MPE V. For native mode programs, add the `XL="PSPLUXL.FORM.TYM"` parameter to your RUN command. This causes your program to call the Formation intrinsic, and is described on page 6-8. Because Formation uses the same intrinsic names as HP, it is easy to convert your programs to use Formation instead of PSP, as well as to migrate from MPE V to MPE XL.

### JETSTART Utility

The JETSTART utility converts MPE V or compatibility mode programs to use the Formation intrinsic. JETSTART converts the program file rather than the source code. Converting the source code would make conversion to MPE XL more difficult.

JETSTART.FORM.TYM can be used on compatibility mode MPE XL programs. However, the Formation XL, PSPLUXL.FORM.TYM makes this conversion unnecessary for native mode programs. JETSTART can be run on multiple program files at once, and has no effect on files that do not already use the HP PSP intrinsic.

**Note:** JETSTART adds PH and DS capabilities to the program file. Once JETSTART has run, the program file will need to be in a group and account that has PH and DS capabilities. If you receive the message `ILLEGAL CAPABILITY` when you try to run the converted program, check that the group (and account) that the program file resides in has PH and DS capability.

To run JETSTART:

1. Log on in the group and account that contain the program file or files to be converted.
2. Enter the following command:  
:RUN JETSTART.FORM.TYM
3. At the fileset prompt, enter a file or fileset name.

For example, to convert an MPE V program called PINVOICE, enter the following:

```
:RUN JETSTART.FORM.TYM
Formation JETSTART Utility 3.41 © 1990-92 Tymlabs Corp.
Fileset to convert (<CR> to end)? PINVOICE
PINVOICE.PROG.AR successfully converted.
Fileset to convert (<CR> to end)? r
END OF PROGRAM
:_
```

You can also convert multiple program files at one time with JETSTART. At the fileset prompt, enter a fileset rather than a single file name.

```
:RUN JETSTART.FORM.TYM

Formation JETSTART Utility 3.41© 1990-92 Tymlabs Corp.

Fileset to convert (<CR> to end)? P@
PDUN.PROG.AR converted to use Formation.
PINVOICE.PROG.AR converted to use Formation.
PSTATMNT.PROG.AR converted to use Formation.

Fileset to convert (<CR> to end)? r

END OF PROGRAM

:_
```

**Note:** If you recompile a previously converted program, you must re-convert it with JETSTART to continue using the Formation intrinsic.

### JetStop Utility

The JETSTOP utility returns programs converted with the JETSTART utility to their pre-Formation status.

To return converted programs to their pre-Formation status, run JETSTOP as follows:

```
:RUN JETSTOP.FORM.TYM
```

**Note:** The JETSTOP program does not remove PH or DS capability from the program even if these were added by JETSTART. If you recompile a program previously converted with JETSTART, you must convert it again with JETSTART to continue using Formation.

### Using Intrinsic in MPE XL Programs

To use Formation intrinsic with an MPE XL or native mode program that references the HP PSP intrinsic, use the XL PSPLUXL.FORM.TYM. This XL is supplied with your Formation installation tape.

To convert a pre-existing program, supply the name of the XL on the RUN command as follows:

```
:RUN programname;XL="PSPLUXL.FORM.TYM"
```

**Note:** JETSTOP will not remove PH or DS capabilities from the program file even if these were added by JETSTART.

The name of the XL must be in quotation marks. The XL name can also be added to other XL's currently in use:

```
:RUN programname;XL="PSPLUXL.FORM.TYM,XL.PUB.DISC"
```

Unison Software also supplies switch stubs for Hewlett-Packard's PSP intrinsic. These are located in the XL called PSPXL.FORM.TYM, and may be used the same way that the Formation XL is used. These are only needed for native-mode programs using the HP PSP intrinsic on MPE XL 2.1 or earlier. Beginning with MPE XL 2.2, the switch stubs are built into the operating system.

### JCW Setting for Converted Programs

Once a program has been converted, Formation software may need to be "notified" that it should override the program's normal selection of HP 2680 or HP 2688 printer features. To do this set the JCW FMFORCELJ before running the program.

When the converted program is run, Formation checks the JCW. If set to the proper value, Formation sends the output to the specified LaserJet printer, while appearing to the application program as an HP 2680 or HP 2688.

The following table lists allowed FMFORCELJ values and their corresponding printers:

Table 6-8: FMFORCELJ values

Target Printer	FMFORCELJ Value		Notes
	Program Written for 2688	Program Written for 2680	
LaserJet 500+	500	10500	
LaserJet Plus, II	1	10001	
LaserJet IID, IIP, 2000, 2000D	2000	12000	
	2001	12001	For duplexing
LaserJet III, IIID, IIP, IIISi	3000	13000	
	3001	13001	For duplexing
LaserJet 4	4000	14000	
	4001	14001	For duplexing
HP 5000	5000	15000	
HP 2680	N/A	2680	These values are used when Formation cannot automatically recognize the 2680 or 2688, either because it belongs to a device class containing multiple device types or because a third-party product which creates "pseudo-devices" is in use.
HP 2688	2688	N/A	

Depending on your printer type, you may need to set the FMFORCELJ JCW before running a program that calls the Formation intrinsic. Programs that are converted to use the Formation intrinsic instead of the PSP intrinsic generally contain either a PINITIALIZE or PINITDEVICE statement for a 2680 or 2688. Once converted to use Formation, such a program can send output to any printer. Instead of modifying the program code, set FMFORCELJ at run-time (or in the job stream that runs the program) to the appropriate value. This value is determined by whether your program expects a 2680 or a 2688, and by the model of your new target printer.

For example, a job stream that runs a native mode print program written for the 2688 but now printing to a LaserJet IIISi might look like this:

```
!JOB JPRINT,MANAGER.ACCOUNT
!SETJCW FMFORCELJ=3000
!FILE OUTPUT;DEV=25;ENV=MYENV
!RUN MYPROG;XL="PSPLUXL.FORM.TYM"
!EOJ
```

## Calling Intrinsic

To invoke an intrinsic, call it from within a program, subprogram, or procedure. To call an intrinsic, you must know the:

- Number of required parameters
- Data type of each parameter
- Order of the parameters
- Functional return value (if applicable)

For all programming language references in this user guide, the following rules apply to the intrinsics and their parameters:

- All parameters are passed by reference. This means the parameter's address in the caller's data area is made available to the called procedure.
- Most intrinsics require a word array, called the comArea, to be used for communication between the intrinsics, and between the subsystem and your program.
- No condition codes are returned. The status of the call is returned in the first word of the comArea.
- All parameters are required, though you can pass designated default values for certain parameters.

Most programs that call Formation intrinsics require one or more special capabilities, especially if you use graphics. All programs should be prepared with DS capability. In addition, if you use graphics, you should prepare your programs with PH and MR capabilities as well.

The following table shows the format of calls to Formation intrinsics from each of the supported languages.

**Table 6-9: Intrinsic Call Format**

Language	Intrinsic Call Format
COBOL	CALL <i>intrinsicname</i> USING <i>parm1</i> , <i>parm2...</i>
FORTRAN	CALL <i>intrinsicname</i> ( <i>parm1</i> , <i>parm2...</i> )
BASIC	<i>nnnn</i> CALL <i>intrinsicname</i> ( <i>parm1</i> , <i>parm2...</i> )
SPL, Pascal	<i>intrinsicname</i> ( <i>parm1</i> [, <i>parm2...</i> ]);
C	<i>intrinsicname</i> (& <i>parm1</i> [,& <i>parm2...</i> ]);

Where:

- intrinsicname* The name of a Formation intrinsic.
- parm* A list of the parameters required by the named intrinsic.
- nnnn* A line number in the program.

---

### COBOL and Real Parameters

Several of the graphics intrinsics included with Formation require real or floating-point parameters. Because COBOL does not support this data type, Formation provides an additional intrinsic interface for these procedures. These additional intrinsics are distinguished by an A appended to the intrinsic name. Thus, the COBOL interface intrinsic for PPRINTFIGURE is PPRINTFIGUREA.

In these intrinsics, the real parameter is replaced with a DISPLAY-type item that contains the display representation of the real number. This method works satisfactorily for cases where you know the value at compile time. For run-time, you may want to do some additional calculation and formatting. Contact your Unison Software support representative for additional information.

### Character Strings

For languages other than BASIC, character strings passed to or returned from Formation intrinsics may be either fixed or variable length. Fixed-length strings have a maximum length that is determined by the string's usage. For example, a form name or a character string name is always 16 characters long. Such strings can be terminated by a blank, by a byte with a value of 255 (called an EOS), or by filling out the array completely. Variable-length strings have no special termination. Intrinsics that take variable-length strings (e.g., PWRITEFIELD and PSTRINGWIDTH) also take a parameter that gives the number of characters in the string.

For BASIC programmers, a string has a length that is intrinsic to the string itself, so no special termination is necessary. When you pass variable-length strings to an intrinsic from BASIC, you must supply a dummy parameter for the length. Formation will use the BASIC string length instead.

## Communication Area (comArea)

Formation uses the comArea for internal communication. The Formation communication area (comArea):

- Is an array which your program must supply to most Formation intrinsics.
- Contains information for your program to use.
- Has a fixed length during execution of the program. The amount of space required varies from environment file to environment file.

To determine the comArea size required for any particular file, access the Environment Manager **Main Menu** for that file, or use IFS.

**Note:** Do not confuse the comArea length with the "printer memory requirement" that is also shown on the **Main Menu** screen.

Advanced applications may also get comArea length information programmatically, as indicated in the following steps:

1. You must specify a comArea must at least 158 words long.
2. If your program calls PINITIALIZE or PINITDEVICE with this minimum-length comArea, the intrinsic:
  - Returns with an error.
  - Provides required comArea length in the third word of the comArea returned.
3. You can then allocate the right amount of space and call the initialization intrinsic a second time.

**Note:** This method only works with languages that provide dynamic allocation of space (e.g., SPL, PASCAL, and C).

Your program will be concerned with the seven elements of the comArea listed in the following table. A discussion of each element follows.

**Table 6-10: Elements of the comArea**

Word	Type	Name	Contents
0	Integer	comStatus	Status word
1	Integer	errorNumber	Error/warning number
2	Integer	errorParm1	Error details
3	Integer	errorParm2	Error details
4	Integer	msgCatalog	MPE file number
5	BA	stringParm	Error details
27	LA	Reserved	Reserved



**ComStatus**

The first word of the comArea contains the status of the call. It will always have a value of 0, 1, or 2.

- 0 The call was successful.
- 1 A warning was issued.
- 2 An error occurred.

When comStatus has a value of 1 or 2, the next few words provide information about the cause of the problem.

**ErrorNumber**

The error number provides your program with information about the cause of a problem. Formation sets the error number both for errors and for warnings.

You should check error status words after each call to the intrinsics. Refer to appendix I for a list of error messages.

**ErrorParms 1 and 2**

These two numbers provide additional detail about an error. For example, in the case of error 901 (Not enough space in the comArea), error parameter 1 contains the amount of comArea space that is required by the current environment.

**MsgCatalog File Number**

This is the file number assigned to the message catalog by the File System. If you have PMSGCAT.PUB.SYS, the IFS/3000 message catalog, this will be the file that is open. Otherwise, PSPMCnnn.FORM.TYM will be opened. (nnn is the language number as specified by your system.)

**StringParm**

This is additional information about the error that occurred. The maximum length of the string is 80 characters, including an EOS. The string parameter is used to return such items as an invalid character or file name.

**System Area**

Formation intrinsics use the system area to record information from the environment file and about the current state of the laser printer. You should not change any values in this area.

### Communication Area Examples

The following examples illustrate comArea definitions. Examples are provided for COBOL, FORTRAN, BASIC, SPL, Pascal, and C languages. In all the examples, *nnnn* is the comArea length as required by the environment file that will be used.

#### COBOL, COBOL II, COBOL 84

```
DATA DIVISION.  
01    PSP-COM-AREA.  
      05 Com-Status          PIC S9(4) COMP.  
      05 Com-Error-Number    PIC S9(4) COMP.  
      05 Com-Error-Parm-1    PIC S9(4) COMP.  
      05 Com-Error-Parm-2    PIC S9(4) COMP.  
      05 Com-Message-File    PIC S9(4) COMP.  
      05 Com-Error-String    PIC X(40).  
      05 Com-System-Info     PIC S9(4) COMP OCCURS nnnn TIMES.
```

#### FORTRAN

```
INTEGER ComArea(nnnn)  
INTEGER Status, ErrorNumber, ErrorParm1, ErrorParm2  
INTEGER MsgFileno  
CHARACTER ErrorString*80  
EQUIVALENCE (ComArea(1), Status), (ComArea(2), ErrorNumber  
&           (ComArea(3), ErrorParm1), (ComArea(4), ErrorParm2),  
&           (ComArea(5), MsgFileno), (ComArea(6), ErrorString)
```

#### BASIC

```
100 REM --> C[*] is the Formation COMAREA  
110 REM --> C[1] is Status  
120 REM --> C[2] is ErrorNumber  
130 REM --> C[3] is ErrorParm1  
140 REM --> C[4] is ErrorParm2  
150 REM --> C[5] is MsgFileno  
160 REM --> C[6] is ErrorString  
170 REM -->  
180 INTEGER C[nnnn]
```

**SPL**

```

array ComArea(0:nnnn);
integer array iComArea(*) = comArea;
define comStatus          = iComArea#,
      comErrorNumber      = iComArea(1)#,
      comErrorParm1      = iComArea(2)#,
      comErrorParm2      = iComArea(3)#,
      comMsgFileno       = iComArea(4)#,
      comErrorString     = comArea(5)#;

```

**Pascal**

```

type
      shortint = -32768..32767;

var
      comArea = record
      Status:          shortint;
      ErrorNumber:    shortint;
      ErrorParm1:     shortint;
      ErrorParm2:     shortint;
      MsgFileno:      shortint;
      ErrorString:    PACKED ARRAY [1..80] OF char;
      SystemArea:    ARRAY [1..nnnn] OF shortint;
      END;

```

**C**

```

#define comStatus          comArea[0]
#define comErrorNumber    comArea[1]
#define comErrorParm1     comArea[2]
#define comErrorParm2     comArea[3]
#define comMsgFileno      comArea[4]
#define comStringParm     (char *)&comArea[5]

int comArea[nnnn];

```

## Using Intrinsic In a Program

This section describes the requirements for using intrinsic and includes an intrinsic programming sample written in COBOL.

### Intrinsic Programming Requirements

Before coding any intrinsic calls, you must include a few essential preparatory steps to make sure that your intrinsic instructions are effective. You must perform the following steps prior to using any intrinsic:

1. Name a file and specify a file equation to a print device. You must specify the ENV=FILE for the intrinsic to work. Also, do not specify any device other than a printer.

```
FILE INVOICE4;DEV=(LaserJetwest);ENV=FILE
```

2. Issue an OPEN instruction in your program to open the output file specified in the file equation.
3. Issue an initialization instruction using the PINITDEVICE intrinsic, before calling any other Formation intrinsic. The PINITDEVICE intrinsic establishes the comArea, defines the calling language, and allows you to specify the output device.
4. To provide for error information, use PERRMSG after every intrinsic call.

### Intrinsic Programming Example

Using a COBOL program example, the following pages illustrate the use of the Formation intrinsic to print an invoice statement on the front side of a page, and terms and conditions on the back of the page. The example includes:

- A sample environment file listing.
- Form INVINT, the sample invoice form.
- Form TANDC, containing the data printed on the back of the form.
- Form TANDC2, containing data printed on the back of the form under specific conditions.
- CSAMPLE.SAMPLES.TYM, the sample COBOL program listing.

Keep the following in mind when viewing the sample:

1. Formation "smart fields" were included when the form was designed with Formation Design. These fields allow the program to use the PWRITEFIELD intrinsic, making the program independent from the form layout. The fields can be repositioned without changing the program.
2. Paragraph 4 on the back of the page (Form TANDC2) is only printed under certain conditions. The program could be designed to print a different form for other conditions. This would be accomplished by testing a condition in the program (around line 445), and selecting the logical page that contained the desired form.
3. To keep the sample program a reasonable length, not all fields are printed. In a real application environment, you would print the entire form.
4. This program is written to process only one customer record. In a real application environment, your program would more likely be structured as follows:

```

OPEN output file
LOOP:
Read customer record
Write data to form using FWRITE and/or Formation intrinsic
If more customer records, go to LOOP
CLOSE output file

```

- For print and space forms, you do not need to modify your application to use the PWRITEFIELD intrinsic.

### The Environment File

Form set specifications must be included in an environment file before the form can be processed with the intrinsic. The following diagram illustrates a sample environment file listing:

Figure 6-1. Environment File Listing

```

Formation Environment Manager                                     Page 1
    Environment file SAMPLE.FORM.FORM created FRI , MAR 5, 1993

121 records (0 free) ; 4092 max.                               HP 2680/88 printer: None
Currently compiled? YES                                       LaserJet printer: 2684D
Default form file:
Measurement system: English

                Physical Page Information

Physical page width: in.                                     Number of copies: 1
Physical page height: in.                                  Multi-copy forms: NO
Page size: L

                Character Sets

C#   Name           File Name or Font Name   OR   ReqSize  ActSize  Atr
-----
0    COUR.CHARSETS.SYS    P    12.000  12.000
1    COURITAL.CHARSETS.SYS P    12.000  12.000

                Logical Pages

PG OR A   Width   Height  From Left  From Top  Base Font  S B PO
-----
0  P Y    8 I    10.5 I
Form 1: Formation form from INVINT
1  P Y    8 I    10.5 I
Form 1: Formation form from TANDC
2  P N    8 I    10.5 I
Form 1: Formation form from TANDC2

```

### The Form

The following diagram illustrates a sample form set without the data.

Figure 6-2. Form INVINT on logical page 0

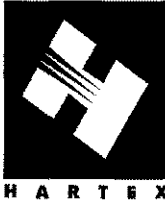
B I L L T O	<i>BILLTO field with 4 subfields</i>	S H I P T O	<i>SHIPTO field with 4 subfields</i>	
Invoice Number <i>INVNUM</i>	Invoice Date <i>INVDATA</i>	Due Date <i>DUEDATE</i>	Customer Number Rep Page	
Purchase Order #	Ship Date	Ship Via	Waybill	
Item	Quantity	Description	Price	Total
<i>ITEM</i>	<i>QTY</i>	<i>DESC</i>	<i>PRICE</i>	<i>TOTAL</i>
<div style="display: flex; justify-content: space-between; align-items: center;"> <span>←</span> <span><i>All fields with 13 subfields</i></span> <span>→</span> </div>				
Please remit to: 			Subtotal Tax Shipping Charge Total	

Figure 6-3. Form TANDC on logical page 1

Terms and Conditions

1.LICENSE  
By its acceptance hereof, Unison Software grants to Licensee a non-exclusive, non-transferable limited license to use the PRODUCTS for a period of twenty one (21) years commencing on the EFFECTIVE DATE.

2.PAYMENT  
LICENSEE shall pay to Unison Software the License Fee(s) listed on the first page hereof within thirty (30) days following delivery of PRODUCTS to LICENSEE, and:OCEMSEE shall pay all import duties, use taxes, sales taxes, value added taxes, property taxes, and excise taxes levied on the licensing, shipping or use of the products.

3.LOCATION OF USE  
The products are to be used by, for, or on behalf of licensee on the single HP 3000 CPU identified on the first page hereof.  
Upon written consent on Unison Software, which shall not be unreasonably withheld, licensee may purchase licenses for use of the products on additional HP 3000 CPUs operated by licensee.Licensee agrees to pay to Unison Software, within thirty (30) days from the date of invoice, the License Fee(s) for the products as specified on Unison Software' then current price list for each such CPU.  
Licensee's use of the products on all additional CPUs shall be governed by the terms and conditions of this Agreement.

Figure 6-4. Form TANDC2 on logical page 2

4.TITLE TO SOFTWARE  
Licensee acknowledges that the products, including all standard releases thereof, and any related patents, trademarks, or copyrights are, and shall remain, the exclusive property of the owners and licensee shall have no right, title or interest to the products except as expressly set forth herein.

## Intrinsic Sample Program

The following code is a sample program included in the Formation package in CSAMPLE.SAMPLES.TYM.

```
* Sample program in COBOL for Formation intrinsic
*
$CONTROL LINES=80
Identification Division.
Program-Id.          FORMATION.
Author.              J.Frels.
Date-Written.        10 Oct 1992.
Remarks.             Illustrate use of Formation Intrinsic.

Environment Division.
Configuration Section.
Source-Computer.     HP-3000.
Object-Computer.     HP-3000.

Input-Output Section.
File-Control.

Select PrintFile assign to "SPOOLOUT"
                    access is sequential.

Data Division.
File Section.

FD  Printfile
    label records are omitted.
01  PrintFileRec          pic X(132).

Working-Storage Section.

77  Language              pic S9(4) comp.
77  XPosition             pic X(16).
77  YPosition             pic X(16).
77  PrimaryFont           pic S9(4) comp.
77  SecondaryFont         pic S9(4) comp.
77  AreaLen               pic S9(4) comp.
77  FieldName             pic X(17).
77  SubFieldNum           pic S9(4) comp.
77  Info                  pic X(255).
77  InfoLen               pic S9(4) comp.
77  OutputDev             pic X(6).
77  XLoc                  pic S9(4) comp.
77  YLoc                  pic S9(4) comp.
77  LogPageNum            pic S9(4) comp.
77  FigureName            pic X(17).
77  FigureFileName        pic X(36).
77  RasterFileName        pic X(36).
77  Height                pic X(16).
77  Rotation              pic S9(4) comp.
77  Units                 pic S9(4) comp.
77  RasterNumber          pic S9(4) comp.
77  PositionMode          pic S9(4) comp.
```



```

77  ErrorMessage          pic X(80).
77  ErrorMessageLen      pic S9(4) comp.
77  MsgLen               pic S9(4) comp.
77  DisplayErrorNum     pic X(5).

01  ComArea.
    03  IFSStatus        pic S9(4) comp.
    03  ErrorNum         pic S9(4) comp.
    03  ErrorParm1      pic S9(4) comp.
    03  ErrorParm2      pic S9(4) comp.
    03  MsgFileNum      pic S9(4) comp.
    03  Filler          pic S9(4) comp
                          occurs 1995 times.

```

Procedure Division.

Main-Process Section.

Mission-Control.

```

Open Output, Printfile.
Display "Formation Sample Program V2.00 JKF Oct 91".
Perform Form-Print.
Close Printfile.
Stop Run.

```

Generate-Report Section.

Form-Print.

```

*
* Initialize the Formation Intrinsic
* The Language is 0; COBOL
* A "2684D" printer is a LaserJet 2000 with the duplex option
*
Move 2000 to AreaLen.
Move Zero to Language.
Move "2684D " to OutputDev.
Call "PINITDEVICE" using ComArea,
                          AreaLen,
                          Language,
                          PrintFile,
                          OutputDev.

Perform Check-For-Error.
*
* Select font pair 0 and 1
*
Move 0 to PrimaryFont.
Move 1 to SecondaryFont.
Call "PUSEFONT" using ComArea,
                          PrimaryFont,
                          SecondaryFont.
Perform Check-For-Error.

*
* Write to the fields on the form.
*

```

## Using Intrinsic In a Program

---

```
*
* First of all, write out the items purchased.
* The items purchased are designed in the form as subfields,
* or an array of fields. We initially want to write to
* the first item in each of the arrays. Therefore,
* SubFieldNum is set to 1.
*
```

```
Move 1 to SubFieldNum
```

```
Move Spaces to FieldName
Move "ITEM" to FieldName.
Move Spaces to Info.
Move "111" to Info.
Move 3 to InfoLen.
Call "PWRITEFIELD" using Comarea,
                        FieldName,
                        SubFieldNum,
                        Info,
                        InfoLen.
```

```
Perform Check-For-Error.
```

```
Move 1 to SubFieldNum
```

```
Move Spaces to FieldName
Move "QTY" to FieldName.
Move Spaces to Info.
Move "1000" to Info.
Move 4 to InfoLen.
Call "PWRITEFIELD" using Comarea,
                        FieldName,
                        SubFieldNum,
                        Info,
                        InfoLen.
```

```
Perform Check-For-Error.
```

```
Move 1 to SubFieldNum
```

```
Move Spaces to FieldName
Move "DESC" to FieldName.
Move Spaces to Info.
Move "Gadgets" to Info.
Move 7 to InfoLen.
Call "PWRITEFIELD" using Comarea,
                        FieldName,
                        SubFieldNum,
                        Info,
                        InfoLen.
```

```
Perform Check-For-Error.
```

```
Move 1 to SubFieldNum
```

```
Move Spaces to FieldName
Move "PRICE" to FieldName.
Move Spaces to Info.
Move "$ 1.00" to Info.
```

```
Move 7 to InfoLen.  
Call "PWRITEFIELD" using Comarea,  
                        fieldName,  
                        SubFieldNum,  
                        Info,  
                        InfoLen.
```

```
Perform Check-For-Error.
```

```
Move 1 to SubFieldNum
```

```
Move Spaces to fieldName  
Move "TOTAL" to fieldName.  
Move Spaces to Info.  
Move "$1000.00" to Info.  
Move 8 to InfoLen.
```

```
Call "PWRITEFIELD" using Comarea,  
                        fieldName,  
                        SubFieldNum,  
                        Info,  
                        InfoLen.
```

```
Perform Check-For-Error.
```

```
*
```

```
* Next we'll write to the second subfield of each field to  
* fill the second row of the purchase order. SubFieldNum  
* is set to 2.  
*
```

```
Move 2 to SubFieldNum
```

```
Move Spaces to fieldName  
Move "ITEM" to fieldName.  
Move Spaces to Info.  
Move "222" to Info.  
Move 3 to InfoLen.
```

```
Call "PWRITEFIELD" using Comarea,  
                        fieldName,  
                        SubFieldNum,  
                        Info,  
                        InfoLen.
```

```
Perform Check-For-Error.
```

```
Move 2 to SubFieldNum
```

```
Move Spaces to fieldName  
Move "QTY" to fieldName.  
Move Spaces to Info.  
Move " 10" to Info.  
Move 4 to InfoLen.
```

```
Call "PWRITEFIELD" using Comarea,  
                        fieldName,  
                        SubFieldNum,  
                        Info,  
                        InfoLen.
```

```
Perform Check-For-Error.
```

```
Move 2 to SubFieldNum
Move Spaces to FieldName
Move "DESC" to FieldName.
Move Spaces to Info.
Move "Devices" to Info.
Move 7 to InfoLen.
Call "PWRITEFIELD" using Comarea,
                        fieldName,
                        SubFieldNum,
                        Info,
                        InfoLen.
```

```
Perform Check-For-Error.
```

```
Move 2 to SubFieldNum
```

```
Move Spaces to FieldName
Move "PRICE" to FieldName.
Move Spaces to Info.
Move "$ 5.00" to Info.
Move 7 to InfoLen.
Call "PWRITEFIELD" using Comarea,
                        fieldName,
                        SubFieldNum,
                        Info,
                        InfoLen.
```

```
Perform Check-For-Error.
```

```
Move 2 to SubFieldNum
```

```
Move Spaces to FieldName
Move "TOTAL" to FieldName.
Move Spaces to Info.
Move "$ 50.00" to Info.
Move 8 to InfoLen.
Call "PWRITEFIELD" using Comarea,
                        fieldName,
                        SubFieldNum,
                        Info,
                        InfoLen.
```

```
Perform Check-For-Error.
```

```
*
* Next, fill in the BILL TO and SHIP TO information, using
* subfields again. In a real application environment, we
* would fill in all four rows of the address.
*
```

```
*
* NOTE: The BILL TO and SHIP TO fields appear above the
* ITEM fields on the form printed. The order you use in your
* program to write to fields on a single form has no effect.
*
```

```
Move 1 to SubFieldNum
```

```
Move Spaces to FieldName
Move "BILLTO" to FieldName.
Move Spaces to Info.
Move "Hartex" to Info.
Move 6 to InfoLen.
Call "PWRITEFIELD" using Comarea,
                        FieldName,
                        SubFieldNum,
                        Info,
                        InfoLen.

Perform Check-For-Error.
```

```
Move 2 to SubFieldNum
```

```
Move Spaces to FieldName
Move "BILLTO" to FieldName.
Move Spaces to Info.
Move "Accounts Payable" to Info.
Move 16 to InfoLen.
Call "PWRITEFIELD" using Comarea,
                        FieldName,
                        SubFieldNum,
                        Info,
                        InfoLen.

Perform Check-For-Error.
```

```
Move 1 to SubFieldNum
```

```
Move Spaces to FieldName
Move "SHIPTO" to FieldName.
Move Spaces to Info.
Move "Hartex" to Info.
Move 6 to InfoLen.
Call "PWRITEFIELD" using Comarea,
                        FieldName,
                        SubFieldNum,
                        Info,
                        InfoLen.

Perform Check-For-Error.
```

```
Move 2 to SubFieldNum
```

```
Move Spaces to FieldName
Move "SHIPTO" to FieldName.
Move Spaces to Info.
Move "4159 Industrial Pkwy" to Info.
Move 20 to InfoLen.
Call "PWRITEFIELD" using Comarea,
                        FieldName,
                        SubFieldNum,
                        Info,
                        InfoLen.

Perform Check-For-Error.
```

\*

\* Fill in various other fields. Only a few are filled in for  
\* illustration purposes.

\*

Move -1 to SubFieldNum

Move Spaces to FieldName  
Move "INVNUM" to FieldName.  
Move Spaces to Info.  
Move "101013" to Info.  
Move 6 to InfoLen.  
Call "PWRITEFIELD" using Comarea,  
                          FieldName,  
                          SubFieldNum,  
                          Info,  
                          InfoLen.

Perform Check-For-Error.

Move -1 to SubFieldNum

Move Spaces to FieldName  
Move "INVDATA" to FieldName.  
Move Spaces to Info.  
Move "09/30/91" to Info.  
Move 8 to InfoLen.  
Call "PWRITEFIELD" using Comarea,  
                          FieldName,  
                          SubFieldNum,  
                          Info,  
                          InfoLen.

Perform Check-For-Error.

Move -1 to SubFieldNum

Move Spaces to FieldName  
Move "DUEDATE" to FieldName.  
Move Spaces to Info.  
Move "09/30/91" to Info.  
Move 8 to InfoLen.  
Call "PWRITEFIELD" using Comarea,  
                          FieldName,  
                          SubFieldNum,  
                          Info,  
                          InfoLen.

Perform Check-For-Error.

\*

\* Change the primary font to write out a special notice.

\*

Move 1 to PrimaryFont.  
Move 0 to SecondaryFont.  
Call "PUSEFONT" using ComArea,  
                          PrimaryFont,  
                          SecondaryFont.

Perform Check-For-Error.

Move -1 to SubFieldNum

```
Move Spaces to FieldName
Move "SPECNOTE" to FieldName.
Move Spaces to Info.
Move "Thank you for paying your account." to Info.
Move 34 to InfoLen.
Call "PWRITEFIELD" using ComArea,
                        FieldName,
                        SubFieldNum,
                        Info,
                        InfoLen.

Perform Check-For-Error.
```

```
*
* Once we have filled in the front of the form, we want
* to go to the back of the form (or the next page for
* non-duplex printers) to print the terms and conditions.
*
* To do that, we must deactivate the form we have just printed and
* eject the physical page. We will now be on the back of the page
* with logical page 1 active because it is declared initially
* active in the environment file. There is no data on the back
* page, only a form. To make it print, simply eject the physical
* page.
*
```

```
* Deactivate logical page zero. We do not want to print
* it on the back.
```

```
Move Zero to LogPageNum.
Call "PDEACTIVATEPAGE" using ComArea,
                        LogPageNum.

Perform Check-For-Error.
```

```
* Go to the back side of the page. Logical page 1 will print
* because we have declared it initially active in the env file.
```

```
Call "PNEWPHYSPAGE" using ComArea.
Perform Check-For-Error.
```

```
*
* Now, suppose that under some circumstances we want to print
* logical page 2, another special "terms and conditions" form,
* on the back of the page along with logical page 1.
* We would need to test the conditions and if true, activate
* logical page 2 and print it. Assume here the condition is true.
*
```

```
* Activate logical page 2 - special terms and conditions
Move 2 to LogPageNum.
Call "PACTIVATEPAGE" using ComArea,
                        LogPageNum.
```

```
Perform Check-For-Error.
```

```
* Force logical page 2 to print on the back of the page with
* logical page 1.
  Call "PNEWPAGE" using ComArea.
  Perform Check-For-Error.
```

```
*
* Now we are done setting up the back of the page. Before
* we print it, we need to set up for the next pass through
* our data processing.
```

```
*
* Activate logical page 0 to be ready for next form!
* If we are printing more than one form, this will
* cause logical page 0 to print the next time around.
```

```
*
  Move Zero to LogPageNum.
  Call "PACTIVATEPAGE" using ComArea,
  LogPageNum.
  Perform Check-For-Error.
```

```
* Eject the physical page (with the back forms) to cause it
* to exit the printer.
```

```
  Call "PNEWPHYSPAGE" using ComArea.
  Perform Check-For-Error.
```

```
Generate-Exit.
  Exit.
```

```
Check-For-Error Section.
```

```
*
* This routine checks the ComArea's IFSSTATUS to see if an error
* occurred. If so, then PERRMSG is called to report the error and
* the program is terminated.
```

```
*
  If IFSStatus of ComArea Not = 0 then
    Move Spaces to DisplayErrorNum
    Move ErrorNum of ComArea to DisplayErrorNum
    Display "(IFS Error ", DisplayErrorNum, ")"
    Move Spaces to ErrorMessage
    Move 80 to ErrorMessageLen
    Move Zero to MsgLen
    Call "PERRMSG" using ComArea,
    ErrorMessage,
    ErrorMessageLen,
    MsgLen

    Display ErrorMessage
    Stop Run.
```

```
Check-Exit.
```

```
  Exit.
```



## Intrinsics Listing

The following sections list the Formation intrinsics in alphabetic order. Each of the Formation intrinsics are described in detail. A quick reference to each intrinsic is provided in Table 6-12: "Formation Intrinsics List," on page 6-30.

### Syntax Description

The syntax statement for each intrinsic includes the calling sequence of the parameters. Intrinsic parameters must be separated by commas. The syntax for each intrinsic is listed as follows:

```

      dt      dt
intrinsicname( parm[, parm ...])
      i/o    i/o
```

### Parameters

- intrinsicname* The name of the Formation intrinsic.
- parm* The parameter string that goes along with the intrinsic. All parameters named with the intrinsic are required.
- dt* The data types allowed for the various parameters. The documentation uses SPL terminology. Table 6-11: "Data Types," on page 6-29 lists the data types, and provides a cross-reference between SPL types and the types used by the language of your choice.
- i/o* Indicates whether the parameter is an input parameter, an output parameter or both. Input parameters pass values into the intrinsic routine, and are denoted by *in*. Output parameters return values from the intrinsic routine to the calling routine, and are denoted by *out*. I/O parameters pass values into and return values from the intrinsic routine and are denoted by *in/out*.

Table 6-11: Data Types

Data Type	COBOL	FORTTRAN	Pascal	SPL	BASIC	C
BA	DISPLAY PIC X (n)	CHARACTER	PACKED ARRAY of CHAR	byte array	String	char []
LA	PIC S9(4) COMP OCCURS n	LOGICAL ARRAY	ARRAY OF shortint	LOGICAL ARRAY	INTEGER array	int []
I	PIC S9(4) COMP	INTEGER*2	shortint	INTEGER	INTEGER	int
R	See text	REAL	REAL	REAL	REAL	float

## Formation Intrinsic Groups

The Formation intrinsics can be divided into four groups:

- Initialization
- Information
- Printer Control
- Graphics

The following tables list all programmatic intrinsics by function and group.

**Table 6-12: Formation Intrinsics List**

Function	Intrinsic	Description
Initialization	PINITIALIZE PINITDEVICE	Initialize the comArea in preparation for calling other Formation intrinsics.
Information	HP36580	Print HP PSP version information
	HP36580V	Return HP PSP version
	PERRMSG	Retrieve an error message for the current error condition
	PFIELDINFO	Return information about a smart field.
	PFONTINFO	Get information about a particular font
	PFONTNUM	Retrieve the font number for a font name
	PLOGPAGEINFO	Retrieve dimensions and position information for a logical page
	PSPPLUSVERSION	Print Formation version information
	PSTATEINFO	Save current printer state information
	PSTRINGWIDTH	Calculate the width of a string as printed in a given font Printer Control
Printer Control	PACTIVATEPAGE	Activate a particular logical page
	PBAR2OF5STRING	Translate a string into 2 of 5 bar code and return the string ready to be printed with a 2 of 5 bar code font
	PDEACTIVATEPAGE	Deactivate a particular logical page
	PDUPLEX	Control duplex printing
	PENDSAVE	Stop saving data
	PFRONTPAGE	Switch to the first logical page and a new front side physical page
	PJOBSEPARATE	Identify one print job from another in the printer's output tray
	PMOVEPENREL PMOVEPENABS	Change the current print position
	PNEWPAGE	Move to the next logical page
	PNEWPHYSPAGE	Switch to the first logical page and a new physical page
	PNEWFORM	Select form 1 or 2 on the same logical page

Table 6-12: Formation Intrinsics List

Function	Intrinsic	Description
Printer Control	PNEWSUBFORM	Switch to a new subform in the current form
	PRESET	Reset the printer
	PPRINTSAVED	Print data saved by PSAVE DATA
	PSAVEDATA	Begin saving data sent to the printer
	PSELECTPAGE	Select and activate a logical page
	PSETCOPIES	Set number of copies to print of each page
	PSWITCHPAGE	Switch to the next logical page without printing
	PSWITCHPHYSPAGE	Switch to the next physical page without printing
	PUSEFONT	Specify a primary and secondary font
	PWRITEFIELD	Write data into a field on a form
	PWRITEFIELDJUST	Write data to a field with right, left or centered justification
	PWRITEFIELDREL	Write data at a location relative to the position of a field or subfield
	Graphics	PBOX
PDELETERASTER		Delete a raster image from the printer's memory
PFLASHRASTER		Print a raster image
PLOADRASTER		Load a raster image into the printer's memory
PPRINTFIGURE		Print a figure from a figure file
PPINTRASTER		Print a raster image that has been stored in laser printer memory

## HP36580

Print the version number of the IFS/3000 intrinsics currently installed on your system, or of the Formation substitute intrinsics if you do not have IFS/3000.

### Syntax

HP36580
---------

### Description

The intrinsic prints the following message to \$STDLIST.

The current version of the IFS/3000 intrinsics is *v.uu.ff*

## HP36580V

Return the current version of IFS intrinsics available to the calling program. If IFS is not installed, it returns the compatible version of the Formation substitute intrinsics.

### Syntax

```
LA
HP36580V(version)
out
```

### Parameter

*version* A logical array to which the current version of IFS or the Formation intrinsics is returned. For the intrinsics, the value returned will be HP36580A.02.00. The string will be terminated by an EOS, followed by the character P. This trailing character permits an application program to distinguish between Formation intrinsics and the real IFS/3000 intrinsics.

### Description

The value returned will be in the form

```
HP36580v.uu.ff
```

followed by an EOS.

## PACTIVATEPAGE

Activate the specified logical page.

### Syntax

```
LA          I  
PACTIVATEPAGE(comArea, pagenum)  
in/out     in
```

### Parameters

*comArea*            The Formation communication area.

*pagenum*            An integer containing the number of the logical page to be activated. If the value is less than 0, greater than 31, or does not represent a defined logical page, Formation returns an error.

### Description

Application programs use this intrinsic to control which pages will be printed. PACTIVATEPAGE is used in conjunction with the PNEWPAGE intrinsics to print the current logical page.

The PACTIVATEPAGE and PDEACTIVATEPAGE intrinsics can be used together to control the printing sequence of logical pages in the current environment. Refer to the PNEWPAGE intrinsic for more information.

## PBAR2OF5STRING

Translate a string of numeric characters into 2 of 5 bar code, and return the encoded string ready to be printed with a 2 of 5 bar code font.

### Syntax

LA	BA	I16	BA	I16	I16
PBAR2OF5STRING( <i>comArea</i> , <i>instrng</i> , <i>inlnth</i> , <i>outstrng</i> , <i>outlnth</i> , <i>checksum</i> )					
in/out	in	in	out	in/out	in

### Parameters

<i>comArea</i>	The Formation communication area.
<i>instrng</i>	A numerical data string to be translated into 2 of 5 bar code. This string may not exceed 250 characters.
<i>inlnth</i>	The length of the string.
<i>outstrng</i>	The character array of the encoded data.
<i>outlnth</i>	The length of the encoded string. The length required is always less than 5 times the length of the input string, plus 18.
<i>checksum</i>	0 = no check sum to be used. 1 = check sum to be calculated and added to end of <i>instrng</i> .

2 of 5 encoding interleaves pairs of digits with black bars representing the first digit in the pair and white bars representing the second. A single checksum digit can be added to the right end of the string, using the standard 2 of 5 checksum algorithm. Since interleaving requires that an even number of characters be encoded, if the data string plus the checksum digit makes an odd number of characters, a leading 0 is added.

When an error occurs, the *outlnth* is set to 0, and an error number is returned to the *comArea*. Possible errors include:

Table 6-13: PBAR2OF5STRING errors

Error	Meaning
2000	The input string is too large.
2001	The <i>outlnth</i> specified is too short for the encoded string.
2002	The input string contains a non-numeric character.

## PBOX

Draw a box or a line at the current cursor position. To draw a line, the line width must exceed the height or width specified.

PBOX can only be used on LaserJets. It has no effect on DCS printers such as the HP 2680 or HP 2688.

### Syntax

	LA	I16	I16	I16	I16	I16
	PBOX( <i>comArea</i> ,	<i>width</i> ,	<i>ht</i> ,	<i>linewidth</i> ,	<i>linestyle</i> ,	<i>shading</i> )
	in/out	in	in	in	in	in

### Parameters

<i>comArea</i>	The Formation communication area.
<i>width</i>	An integer defining the width of the box in dots (pixels).
<i>ht</i>	An integer defining the height of the box in dots (pixels).
<i>linewidth</i>	An integer defining the thickness of the box outline. Enter a negative value to define the thickness in points. The maximum point value allowed is 32. Enter a positive value to define the thickness in printer dots. The maximum dot value is 150.
<i>linestyle</i>	Enter 0 to indicate a solid outline. Other values are reserved for future use.
<i>shading</i>	An integer defining the shading percentage for the center of the box.

### Description

Cursor position prior to calling this intrinsic can be established by issuing a PMOVEPENREL, PMOVEPENABS, or PWRITEFIELDREL with no data specified. The PBOX intrinsic does not affect current cursor position.

PBOX can be used to highlight a field or subfield when used in conjunction with PWRITEFIELDREL and PFIELDINFO.



---

## PDEACTIVATEPAGE

Deactivate the specified logical page.

### Syntax

LA	I
PDEACTIVATEPAGE( <i>comArea</i> , <i>pagenum</i> )	
in/out	in

### Parameters

<i>comArea</i>	The Formation communication area.
<i>pagenum</i>	The number of the logical page to be deactivated. If the page number specified is less than 0 or greater than 31, Formation returns an error.

### Description

Your application program can prevent a particular logical page from printing by calling this intrinsic. At least one logical page must be active at all times. If your program attempts to deactivate the last active page, a warning is returned in the *comArea*, and the default logical page is used.

The PACTIVATEPAGE and PDEACTIVATEPAGE intrinsics can be used together and with PNEWPAGE to control the printing sequence of logical pages in the current environment. Refer to the PNEWPAGE intrinsic for more information.

## PDELETERASTER

Delete a previously-stored raster image from the printer's memory.

### Syntax

LA	I
PDELETERASTER( <i>comArea</i> , <i>imagenum</i> )	
in/out	in

### Parameters

<i>comArea</i>	The Formation communication area.
<i>imagenum</i>	An integer variable containing the number of the image to be deleted. This is the number specified in PLOADRASTER when the image was loaded. It must be between 0 and 31 inclusive. A Formation error occurs if the image has not previously been loaded.

### Description

The PLOADRASTER and PDELETERASTER intrinsics are the most efficient way to print multiple copies of a single graphics image. They are also the most efficient way to print multiple copies of a group of raster images when printing on an HP2680. See the discussion under PLOADRASTER for further information on ways to use stored raster images.

## PDUPLEX

Control duplex printing on printers that offer duplexing capabilities. This intrinsic is ignored when printing to an HP 2680 or HP 2688.

### Syntax

LA	I16	I16	I16	I16
PDUPLEX( <i>comArea</i> ,	<i>flags</i> ,	<i>duplex</i> ,	<i>topreg</i> ,	<i>leftreg</i> )
in/out	in	in	in	in

### Parameters

<i>comArea</i>	The Formation communication area.
<i>flags</i>	A bit-significant flag word indicating which parameters are to be applied. To indicate changes in multiple settings at the same time, enter the sum of all applicable values: <ol style="list-style-type: none"> <li>1 change duplex state</li> <li>2 change top registration</li> <li>4 change left registration</li> </ol>
<i>duplex</i>	An integer defining the duplex printing mode. The allowed values are: <ol style="list-style-type: none"> <li>0 Simplex, duplex off</li> <li>1 Duplex on, long edge binding</li> <li>2 Duplex on, short edge binding</li> </ol>
<i>topreg</i>	An integer defining the top offset registration in decipoints (1/720 inch). This parameter adjusts the text position on the page to allow additional room for page binding. This may be a positive or negative value. The default is 0. <p><b>Note:</b> Data loss may result if the registration offset causes the logical page to be moved outside the printable area.</p>
<i>leftreg</i>	An integer defining the left offset registration in decipoints (1/720 inch). This parameter adjusts the text position on the page to allow additional room for the page binding. This may be a positive or negative value. The default is 0. <p><b>Note:</b> Data loss may result if the registration offset causes the logical page to be moved outside the printable area.</p>

## PENS SAVE

Instruct the LaserJet to stop saving data. PENS SAVE is used in conjunction with the PSAVEDATA and PPRINTSAVED intrinsics.

### Syntax

```
LA  
PENS SAVE(comArea)  
in/out
```

### Parameter

*comArea*            The Formation communication area.

### Description

The PSAVEDATA, PENS SAVE, and PPRINTSAVED intrinsics are used mostly for multi-part forms on LaserJet printers to quickly repeat a set of commands. The PSAVEDATA intrinsic tells Formation to begin recording the commands that follow. A PENS SAVE intrinsic is issued to tell Formation to stop recording commands. A PPRINTSAVED intrinsic can then be issued to perform the commands that have been saved. This is useful if you have to repeat a sequence of steps more than once when printing a form. Enter the correct number of PPRINTSAVED intrinsics, and the steps are performed the required number of times.

## PERRMSG

Return the error or warning message associated with the error number received when a call to an intrinsic is unsuccessful. This intrinsic provides readable error messages whenever a Formation error or warning occurs.

### Syntax

	LA	BA	I	I
PERRMSG(	<i>comArea</i> ,	<i>errmsg</i> ,	<i>arrayInth</i> ,	<i>msgInth</i> )
	in/out	out	in	out

### Parameters

<i>comArea</i>	The Formation communication area. The <i>comArea</i> should be left exactly as it was after the unsuccessful intrinsic call.
<i>errmsg</i>	A string used to return the error message. The error message is padded with blanks to the end of the array (given by <i>arrayInth</i> ). This array must be at least 80 bytes long.
<i>arrayInth</i>	An integer containing the number of bytes available in <i>errmsg</i> . In BASIC, this parameter is ignored, and Formation uses the length supplied by BASIC.
<i>msgInth</i>	An integer to which Formation returns the number of characters in the error message. This value will not include any padding. For BASIC callers, Formation will set the length of the string and will also supply this value, but most programs will not need to use the value returned in <i>msgInth</i> .

### Description

PERRMSG should be used by your application after every intrinsic, to provide a readable error message whenever a Formation error or warning occurs. You should not disturb any of the values in the *comArea* between the failed intrinsic call and the call to PERRMSG.

After retrieving the error message, PERRMSG sets the first word of the *comArea* to 0. This means that a second call to PERRMSG will fail, because the *comArea* shows that no error has occurred.

PERRMSG depends on having a message catalog available. If you do not have either the IFS/3000 catalog (PMSGCAT.PUB.SYS) or the Formation catalog (PSPMCnnn.FORM.TYM, where *nnn* is the language number you are using), the intrinsic will return a generic message in English containing the values of the error number and error parameter.

## PFIELDINFO

Return information about a field. Information available with this intrinsic includes where the field is located on the form, its size, and subfield information.

### Syntax

```

                LA      BA      IA
PFIELDINFO(comArea, fldname, fldinfo)
                in/out   in      out
    
```

### Parameters

- comArea*            The Formation communication area.
- fldname*            A character string specifying the name of a smart field previously defined on the current Formation Design form.
- fldinfo*            An integer array containing the following information about the specified field:

Table 6-14: PFIELDINFO array

Word	Type	Content
0	char	Field name (18 chars)
9	word	X coordinate top left (dots)
10	word	Y coordinate top left (dots)
11	word	Width in dots
12	word	Height in dots
13	word	Count of subfields
14	word	0=vertical, 1=horizontal subfields

## PFLASHRASTER

Print a raster image. The COBOL version, which uses character strings for the real-type parameters, is PFLASHRASTERA.

### Syntax

	LA	BA	R	R	I16	I16
PFLASHRASTER[A]	( <i>comArea</i> ,	<i>rfile</i> ,	<i>x</i> ,	<i>y</i> ,	<i>units</i> ,	<i>mode</i> )
	in/out	in	in	in	in	in

### Parameters

<i>comArea</i>	The Formation communication area.
<i>rfile</i>	The name of a partitioned raster file created with HP's PCONVERTFIGURE intrinsic or Formation's PPRINTFIGURE intrinsic. The name may be up to 35 characters long and must be terminated by a blank or EOS (except in BASIC).
<i>x</i>	A real variable containing the horizontal coordinate of the left edge of the figure. If the <i>mode</i> parameter specifies absolute positioning (i.e., has a value of 1), this distance is measured from the left edge of the logical page. With relative positioning, a positive value moves the image to the right; a negative value moves it left.  The value is specified in the units given by the <i>units</i> parameter.
<i>y</i>	A real variable containing the vertical coordinate of the top edge of the image. If the <i>mode</i> parameter specifies absolute positioning (i.e., has a value of 1), this distance is measured from the top edge of the logical page. For relative positioning, a positive value moves the image down; a negative value moves the image up. Note that this is the opposite of a normal mathematical coordinate system.  The value is specified in the units given by the <i>units</i> parameter.
<i>units</i>	An integer defining the units in which <i>x</i> and <i>y</i> are specified. Possible values are:  0: Dots 1: Inches 2: Centimeters 3: Millimeters
<i>mode</i>	An integer variable used to specify whether the <i>x</i> - and <i>y</i> -positions are relative to the current pen position or relative to the edge of the logical page.  0 Indicates positioning relative to the current pen position. 1 Indicates positioning relative to the upper left corner of the logical page.

### Description

For LaserJet printers, PFLASHRASTER prints a raster image without first storing it as an "addressable" image. This saves the memory required to store the image and costs only the amount of memory required to print it. Using this intrinsic, the LaserJet can print any number of images on the page, limited only by available memory.

The HP2680 and HP2688 have a limitation of 32 images that can be printed on one page using this intrinsic. (Another 32 images can be printed using PLOADRASTER/PDELETERASTER combinations.)

When Formation opens the raster file name, it searches the MPE temporary file domain first, followed by the permanent file domain.

You can completely control the positioning of the image on the logical page using *x* and *y*. Formation checks the position that you supply to insure that it is within the bounds of the current logical page. It does not check the position of the lower-right corner. If the image is too large for the logical page, the printer prints it anyway, clipping it only at the physical page boundary.

On the LaserJet, the pen is left at the upper left corner of the image. On the H2680 and HP2688, the pen is unaffected by this intrinsic.



## PFONTINFO

Return information about a font. Information returned includes font size, character and line spacing, and whether the font is proportional or fixed.

### Syntax

```

                LA      I16      BA      LA
PFONTINFO(comArea, fontnum, fontname, fontinfo)
                in/out   in      out      out
    
```

### Parameters

- comArea*            The Formation communication area.
- fontnum*            The number of the font for which information is to be retrieved. This must be an integer between 0 and 31 inclusive. If the font number is 31 and no font number 31 exists in the environment, information about the default font is returned. If the number specified is not an existing number, an error is returned in the *comArea*.
- fontname*           A string to which the name of the font, as supplied in the environment file, is returned. This array must be at least 16 characters long.
- fontinfo*           An array to which the font information is returned. The array must be at least 30 words long. Its contents are detailed in the following table.

Table 6-15: PFONTINFO array

Word	Contents
0	Rotation (0, 90, 180, 270)
1	Height in dots
2	Cell width in dots (width of largest cell for proportional fonts)
3	Baseline (dots) Bottom row is row 1
4	Extra character spacing (dots)
5	Extra line spacing
6	Proportional spacing flag (0 = monospaced; 1 = proportional)
7	Font height in decipoints

### Description

This intrinsic can be used in word processing and other applications that require knowledge of user-supplied fonts. PFONTINFO returns the name associated with a given font number. The name is returned as a full 16-character string padded with blanks. (In BASIC, if the font has no name, the intrinsic sets the length of the passed string to 0.)

If you need the font number associated with a name, your program can call PFONTNUM. For more information about the PFONTNUM intrinsic, see "PFONTNUM" on page 6-46.

## PFONTNUM

Retrieve the font number associated with a font name.

### Syntax

	LA	BA	I16
PFONTNUM( <i>comArea</i> , <i>fontname</i> , <i>fontnum</i> )			
	in/out	in	out

### Parameters

<i>comArea</i>	The Formation communication area.
<i>fontname</i>	A string, up to 16 characters in length, terminated by a blank, an EOS or the 16th character. In BASIC, the length of the string is determined by the string itself.
<i>fontnum</i>	An integer variable to which the font number will be returned. If the font name is not found, this variable will remain unchanged.

### Description

Use the PFONTNUM intrinsic to return the font number

- Required by PFONTINFO, PUSEFONT, and PSTRINGWIDTH intrinsics.
- Of the current primary font.

If the supplied font name is blank, consists only of an EOS, or (in BASIC) has a length of zero (0), PFONTNUM returns the current primary font number.

If you know a font number, and need the font name, use the FONTINFO intrinsic.

---

## PFRONTPAGE

Similar to PNEWPHYSPAGE except that it ensures the new page will be the front side when duplex printing.

### Syntax

```
LA
PFRONTPAGE(comArea)
in/out
```

### Parameter

*comArea*            The Formation communication area.

### Description

If this intrinsic is used when duplex printing is not requested, or with a printer which does not support duplexing, PFRONTPAGE behaves as PNEWPHYSPAGE. For information on the PNEWPHYSPAGE intrinsic refer to page 6-60.

## PINITDEVICE

Initialize the Formation comArea for use with other intrinsics.

### Syntax

```

                LA      I16      I16      I16      BA
PINITDEVICE(comArea, areaInth, language, filenum, outputdev)
                in/out  in       in       in       in
    
```

### Parameters

- comArea*                    The Formation communication area.
- areaInth*                 Length of the comArea array in words.
- language*                 An integer indicating the language from which the intrinsic is being called. The valid values are:

Value	Language
0	COBOL or COBOL II when not using the CALL INTRINSIC statement.
1	BASIC
2	FORTRAN when not using the SYSTEM INTRINSIC statement
3	SPL, COBOL II when using the CALL INTRINSIC statement, or FORTRAN when using the SYSTEM INTRINSIC statement
5	Pascal, C

- filenum*                    An integer containing the MPE file number of the output file. This value can be obtained in various ways depending on the language you are using:

Language	File Number
COBOL	Formal file designator from the SELECT clause
BASIC	SFN ( <i>file number</i> ) (This value must be assigned to or converted to an integer)
FORTRAN	FNUM ( <i>logical unit number</i> )
SPL	Value returned by FOPEN
Pascal	FNUM ( <i>file identifier</i> )
C	Value returned by call to fopen or _fopen

- outputdev*                 Character string containing the name of the output device. The string may be up to 8 characters long and terminated by a blank, the eighth character, or an EOS (except in BASIC). The following table lists the allowable device types are:

Device Type	Description	Duplexing?
2680A	HP2680 printer	
2688A	HP2688A printer	
2686A	Classic LaserJet, LaserJet Plus or LaserJet Series II printer	
LJPlus	LaserJet Plus printer	
2686D	LaserJet 500 Plus printer	
LJ500	LaserJet 500 Plus printer	
2684A	Laser Jet III or LaserJet Series IID, IID, or 2000	
2684D	LaserJet 2000, IID or III D printers	Enabled
LJ2000	LaserJet 2000 printer	
LJII or LJSII	LaserJet Series II printer	
LJIID	LaserJet II D printer	Enabled
LJIIP	LaserJet II P printer	
LJIII or LJSIII	LaserJet III printer	
LJIIID or LJSIIID	LaserJet IIID printer	Enabled
LJIIP	LaserJet IIIP printer	
LJIIIS	LaserJet IIISi printer	Enabled
LJIIISD	LaserJet IIISi printer	Enabled
LJ4	LaserJet 4 printer	
LJ4SI	LaserJet 4 printer	Enabled
F100	HP 5000 F100 printer	

### Description

PINITDEVICE (and PINITIZE) begin the process of using Formation. They both supply the other intrinsics with the language, the communication area length, and the file number of the output file. PINITDEVICE also allows you to specify an output device when using a LaserJet.

**Note:** If you use a device in PINITDEVICE with duplexing enabled, your output is automatically requested in duplex mode. Instead, use PINITDEVICE for a device without duplexing enabled and control any desired duplexing with the PDUPLEX intrinsic.

Either PINITDEVICE or PINITIZE must be called before any other Formation intrinsic. The PINITDEVICE and PINITIZE intrinsics can use a significant amount of stack space during their execution. Both use at least 1800 words while initializing the comArea. This varies with the MPE version under which your program is running. In addition to this basic requirement, if your output is being routed to a LaserJet, and you are not using a dual environment file (i.e., one prepared with the Environment Manager), an additional 5000 words are used during execution of the intrinsic.

In both cases, this space is returned to your program when the intrinsic finishes. If an insufficient amount of space is available when the intrinsic starts, you could receive either an IFS error or a stack overflow when you call PINITDEVICE or PINITIZE.

## PINITIALIZE

Initialize the Formation comArea in preparation for calling other Formation intrinsics.

### Syntax

```

                LA      I16      I16      I16
PINITIALIZE(comArea, areaInth, language, filenum)
                in/out    in      in      in
    
```

### Parameters

- comArea*            The Formation communication area.
- areaInth*          An integer containing the length of the comArea array, in 16-bit words.
- language*          An integer variable specifying the programming language used.

Value	Language
0	COBOL or COBOL II when not using the CALL INTRINSIC statement.
1	BASIC
2	FORTRAN when not using the SYSTEM INTRINSIC statement
3	SPL, COBOL II when using the CALL INTRINSIC statement, or FORTRAN when using the SYSTEM INTRINSIC statement
5	Pascal, C

- filenum*            An integer containing the MPE file number of the printer spool file. This value can be obtained in various ways, depending on the language and calling method you are using:

Language	File Number
COBOL	Formal file designator from the SELECT clause
BASIC	SFN ( <i>file number</i> ) Assign or convert this value to an integer
FORTRAN	FNUM ( <i>logical unit number</i> )
SPL	Value returned by FOPEN
Pascal	FNUM ( <i>file identifier</i> )
C	Value returned by call to fopen or _fopen

### Description

Call PINITIALIZE or PINITDEVICE before using any other Formation intrinsic. If you are using PINITIALIZE with the LaserJet, set the FMFORCELJ JCW to the value appropriate to the printer you are using. The exception to this is if you are using a dual environment file and the device class associated with the open spool file is not associated with a 2680 or 2688. In this case, Formation assumes that you are using a LaserJet of the type specified in the environment file. See the PINITDEVICE description for additional information.

---

## PJOBSEPARATE

Job separation provides a means of identifying one print job from another in the printer's output tray. This is normally achieved by offsetting the output of one print job from the next.

### Syntax

```
LA
PJOBSEPARATE(comArea)
in/out
```

### Parameter

*comArea*            The Formation communication area.

### Description

This function is only available for some printer types, including the following:

- LaserJet 2000s (2684A and 2684D)
- LaserJet III Si
- LaserJet 4 Si

If this intrinsic is used in conjunction with printers which do not support job separation, it is ignored.

## PLOADRASTER

Load a raster file into laser printer memory for later printing.

### Syntax

	LA	BA	I16
PLOADRASTER(	<i>comArea</i> ,	<i>filename</i> ,	<i>imagenum</i> )
	in/out	in	in

### Parameters

<i>comArea</i>	The Formation communication area.
<i>filename</i>	A string containing the name of the file that contains the image to be loaded. The name may be up to 35 characters long, and must be terminated by a blank, an EOS, or the 35th character (except in BASIC).
<i>imagenum</i>	An integer containing a number between 0 and 31 inclusive. This will be the image number by which the loaded image can be referred to when printed with the PPRINTRASTER intrinsic or deleted with the PDELETERASTER intrinsic.

### Description

A maximum of 32 raster images can be loaded into the printer's memory at any one time, subject to memory limitations. Stored images are printed with the PPRINTRASTER intrinsic, and may be deleted with the PDELETERASTER intrinsic. All raster images are deleted at the end of the print job.

When printing on a LaserJet printer, raster images take as much memory to store as they do to print. For example, storing a raster image that is 30kb in size takes approximately 30kb of printer memory. Printing that image from storage takes another 30kb. If the printer indicates Error 21 when you print a stored raster image, you have tried to print a raster image that is too large. You can use PFLASHRASTER or PPRINTFIGURE to eliminate the problem.



## PLOGPAGEINFO

Return information about a particular logical page, or about the current logical page.

### Syntax

	LA	I16	LA
PLOGPAGEINFO(	<i>comArea</i> ,	<i>pagenum</i> ,	<i>pageinfo</i> )
	in/out	in	out

### Parameters

- comArea*            The Formation communication area.
- pagenum*            An integer variable containing the page number for which information is to be returned. If a value of -1 is specified, information about the currently-active logical page is returned.
- pageinfo*            A logical array at least 30 words in length to which Formation will return information about the specified logical page. The contents of the array are described in the following table.

Word	Contents
1	Orientation, in degrees (0, 90, 180, or 270)
2	Logical page height in dots
3	Logical page width in dots
4	Distance from left edge of paper to left edge of logical page, in dots
5	Distance from the top of the physical page to the top of the logical page, in dots
6	Left margin in dots
7	Currently active? (0=no; -1=yes)
8	-1 if this page has a VFC associated with it, 0 if not
9	Horizontal character spacing in dots
10	Vertical line spacing in dots
11	Baseline of base character set in dots
12	Top margin (if no VFC)
13	Fractional character spacing in quarter-dots
14	Fractional line spacing in quarter-dots
15	Logical height in decipoints
16	Logical width in decipoints
17	Vertical line spacing in decipoints
18-30	Reserved for system use

### **Description**

The vertical line spacing specifications that you enter on the IFS/3000 Logical Page menu may result in line spacing that is not an even multiple of one dot. For these situations, the printer provides a way of specifying 1/4-dot spacing, and adjusts spacing line by line to maintain an average spacing that adheres to your specification as closely as possible. The same applies to fractional character spacing, and again, the printer adjusts spacing between characters so that the average is as close as possible to your specification.

## PMOVEPENABS

Move the laser printer "pen" to an absolute location on the current logical page.

### Syntax

```
LA      I16 I16
PMOVEPENABS(comArea, x, y)
         in/out in  in
```

### Parameters

- comArea*      The Formation communication area.
- x*             An integer variable which supplies the x (horizontal) coordinate of the new location of the pen. This value is expressed in dots, and must be greater than or equal to zero.
- y*             An integer variable which supplies the y (vertical) coordinate of the new location of the pen. The origin is at the upper left corner of the logical page. This value is expressed in dots, and must be greater than or equal to zero.

### Description

The laser printer's pen defines the location of the next character to be printed. The left end of the character's baseline will be placed at the current pen position.

If you specify a pen location that would be off the logical page, the printer's behavior depends on the printer type. On the HP2680 and HP2688, the pen is left at its current location, and an error trailer is printed. On the LaserJet, the pen is moved to the appropriate logical page boundary, and no error occurs.

This intrinsic requires its measurement in dots, and is based on the HP 2680's resolution of 180 dots per inch. For example, an *x* value of 180 would move the pen 1 inch to the right of the left edge of the logical page. For device-independent movement, specify *x* and *y* variables in HP2680 terms (180 dots per inch). To run the program on a LaserJet, set the FMFORCELJ JCW to the appropriate value over 10000. Pen movement is automatically translated to accommodate the LaserJet's resolution of 300 dots per inch.

**Note:** If your environment file is only compiled for a LaserJet, the *x* and *y* specifications should be relative to 300 dots per inch.

## PMOVEPENREL

Move the laser printer pen relative to its current position.

### Syntax

```
LA      I16 I16
PMOVEPENREL(comArea, x, y)
         in/out in in
```

### Parameters

<i>comArea</i>	The Formation communication area.
<i>x</i>	An integer variable containing the number of dots by which the pen is to be moved in the horizontal direction. A positive value moves the pen to the right; a negative value moves it to the left.
<i>y</i>	An integer variable containing the number of dots by which the pen is to be moved in the vertical direction. A positive value moves the pen down; a negative value moves the pen up. Note that this is the opposite of a normal mathematical coordinate system.

### Description

If you specify coordinates that would move the pen off the logical page, the effect depends on the printer that you are using. If you are printing on an HP2680 or HP2688, the pen is left in its current position and an error is generated. The LaserJet moves the pen to the appropriate page boundary, and does not return an error.

This intrinsic requires its measurement in dots, and is based on the HP 2680's resolution of 180 dots per inch. For example, an *x* value of 180 would move the pen 1 inch to the right of its current position. For device-independent movement, specify *x* and *y* variables in HP2680 terms (180 dots per inch). To run the program on a LaserJet, set the FMFORCELJ JCW to the appropriate value over 10000. Pen movement is automatically translated to accommodate the LaserJet's resolution of 300 dots per inch.

## PNEWFORM

Select the form to which you want to write. Subsequent data is merged with the named form.

### Syntax

```
LA      BA
PNEWFORM(comArea, formname)
in/out  in
```

### Parameters

*comArea*            The Formation communication area.

*formname*           A string containing the name of the form to which you wish to write. The form must be on the current logical page. *Formname* may be up to 16 characters long, and must be terminated by a blank, an EOS, or the 16th character (except in BASIC). The form name is that assigned to it in *IDSForm*.

### Description

You need to call this intrinsic only when there is more than one form on the logical page, and if you have not called *PNEWPAGE* or *PNEWPHYSPAGE*.

You can also use the special values "1" and "2" as the form name. This selects the first or second form on the logical page, respectively. The first form is that which is listed first on the IFS/3000 Form menu.

## PNEWPAGE

Advance to the next active logical page.

### Syntax

```

                LA
PNEWPAGE(comArea)
        in/out

```

### Parameter

*comArea*            The Formation communication area.

### Description

The PNEWPAGE intrinsic is used to move to the next active logical page. When you call PNEWPAGE, the first (or only) form on that logical page is made the active form. You do not need to call PNEWFORM after most calls to PNEWPAGE.

The practical results of using the PNEWPAGE intrinsic are governed by the following factors:

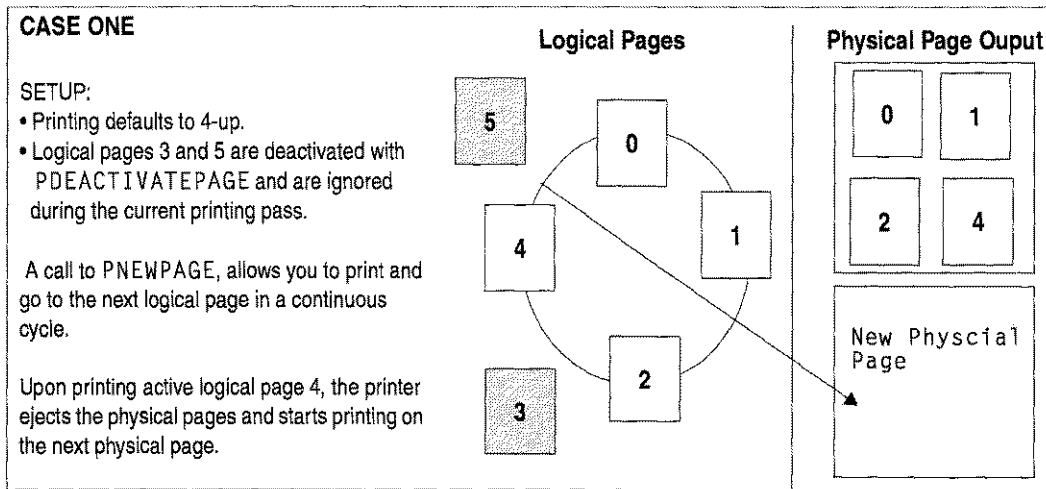
- Number of logical pages specified in the environment file.
  - If the last active logical page number is a higher number and the next logical page is lower, the printer will eject the physical page and start a new blank physical page.
  - The first-to-last order is determined by the page numbers as assigned by the Environment Manager.
- Existence of inactive pages specified with the PDEACTIVATEPAGE intrinsic.
  - If a logical page is inactive, PNEWPAGE skips over it.
- The flow of control used in the program.

You can use the capabilities of the PNEWPAGE intrinsic to start a continuous print cycle that rolls through and prints all active logical pages in order. With PNEWPAGE, you do not have to know the number of logical pages to print.

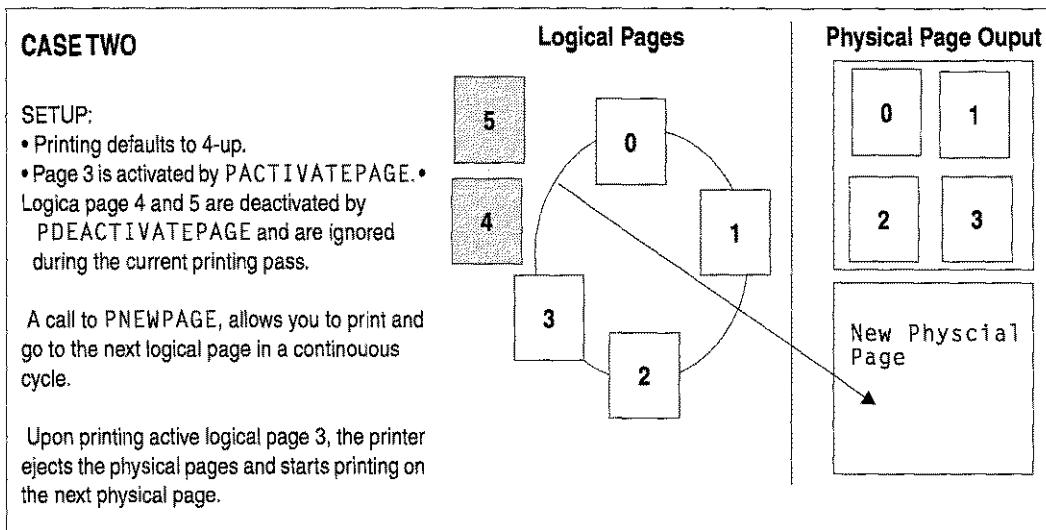
Many times in printing forms, you want the ability to print different information based on a set of conditions (such as standard invoices and past due invoices). To set up this kind of conditional branching in your program, you can use PNEWPAGE in conjunction with the PACTIVATEPAGE and PDEACTIVATEPAGE intrinsics to deactivate specific logical pages in one print cycle and re-activate them in another cycle.

**Note:** When using PNEWPAGE with PACTIVATEPAGE and PDEACTIVATEPAGE, do not use PSELECTPAGE within the same sequence. When you use PSELECTPAGE, it automatically activates the selected logical page and requires a knowledge of the logical page number you are selecting.

The following diagrams illustrate a conditional forms printing cycle using PNEWPAGE with the PACTIVATEPAGE and PDEACTIVATEPAGE intrinsics.



In Case One, invoice information is printed 4-up on a single form for billing purposes. Logical page three contains 30-day past due information that prints only when that condition is met. Logical page five contains 60-day past due information that prints only when that condition is met.



In Case Two, an account that was 30 days past due was encountered by the program, which then activated logical page three and deactivated standard billing information represented by logical page four. Logical page five remains deactivated and both pages four and five are ignored in the current printing cycle.

## PNEWPHYSPAGE

Advance to the next physical page.

### Syntax

```
LA  
PNEWPHYSPAGE(comArea)  
in/out
```

### Parameter

*comArea*            The Formation communication area.

### Description

The PNEWPHYSPAGE intrinsic advances to the next physical page. It also sets the printer to the first active logical page, and makes the first (or only) form on that logical page the active form.



## PNEWSUBFORM

Select the subform to which you wish to write. This intrinsic is used for **IDSFORMS** only.

### Syntax

	LA	BA
PNEWSUBFORM( <i>comArea</i> , <i>subformname</i> )		
	in/out	in

### Parameters

*comArea*            The Formation communication area.

*subformname*        A string containing the name of the subform to which you wish to write. The string may be up to 16 characters long, and must be terminated by a blank, an EOS, or the 16th character (except in BASIC).

### Description

If you are writing data to a form (using **PWRITEFIELD**) that contains fields that have the same name, you must use **PNEWSUBFORM** to tell Formation which subform to search for the field named in your **PWRITEFIELD** call.

For forms with many subforms, **PNEWSUBFORM** can result in a performance improvement by reducing the number of places that Formation has to look in order to find a named form. An exception to this is when you tell Formation to search all fields in the current form by specifying a blank or EOS as the subform name.

Fields are always defined within subforms. You can have duplicate field names in your form as long as the fields are in different subforms. If you do not call **PNEWSUBFORM** when you write to a field whose name occurs in two or more places, the field to which your data will be written is unpredictable.

**Note:** This intrinsic is only for IDS use. Currently, Formation Design does not support the use of subforms. You will not need to use this intrinsic when accessing forms designed with Formation Design.

## PPRINTFIGURE

Convert a figure in a figure file to a partitioned raster image, optionally save it in a file, then print it on the laser printer. The COBOL version, called with ASCII strings for the real parameters, is PPRINTFIGUREA.

### Syntax

	LA	BA	BA	BA	R	I16	I16	R	R	I16	I16
PPRINTFIGURE[A]	( <i>comArea</i> ,	<i>ffile</i> ,	<i>fig</i> ,	<i>rfile</i> ,	<i>ht</i> ,	<i>units</i> ,	<i>rot</i> ,	<i>x</i> ,	<i>y</i> ,	<i>mode</i> ,	<i>type</i>
	in/out	in	in	in	in	in	in	in	in	in	in

### Parameters

- comArea*            The Formation communication area.
- ffile*                A character string containing the name of the figure file in which the figure you are printing resides. The name may be up to 35 characters long, and must be terminated by a blank, an EOS, or the 35th character (except in BASIC). The MPE temporary file domain is searched first, followed by the permanent file domain.
- fig*                  A character string containing the name of the figure you wish to print. The name may be up to 16 characters long, and must be terminated by a blank, an EOS, or the 16th character (except in BASIC).
- rfile*                A character string containing the name of the raster file that will be created by this intrinsic. It may be up to 35 characters in length, and must be terminated by a blank, an EOS, or the 35th character (except in BASIC). You need not supply a raster file name if you do not want to preserve the raster image. If you choose this option, Formation uses a temporary file called OUT2680A or OUT2688A. Specify a blank or EOS as the filename (or a null string in BASIC).
- ht*                    A real number that supplies the height of the converted image, in units specified by the *units* parameter. The height is specified in the direction of its orientation. (The width of the figure is calculated from its height.)
- The intrinsic adds one dot to the height you specify, so a height of 300 dots creates an image 301 dots high. You need only take this into account when working with extremely precise dot measurements.
- If you are programming in COBOL, specify this value as an ASCII string.
- units*                An integer which specifies the units in which *ht*, *x* and *y* are given. The possible values are:
- 0: Dots
  - 1: Inches
  - 2: Centimeters
  - 3: Millimeters

---

<i>rot</i>	An integer which indicates the rotation of the image relative to the logical page. The value is specified in degrees, and must be 0, 90, 180, or 270.
<i>x</i>	A real variable used to specify the x-coordinate of the upper-left corner of the image on the logical page. The value is specified in the units given by the <i>units</i> parameter, and is either relative or absolute depending on the value of <i>mode</i> . If <i>mode</i> indicates that you want relative positioning, a positive value moves the image to the left; a negative value moves the image to the right. If you are programming in COBOL, you should specify this parameter as an ASCII string.
<i>y</i>	A real variable giving the vertical position of the upper left corner of the image. The value is specified in the units given by the <i>units</i> parameter, and may be relative or absolute depending on the value of the <i>mode</i> parameter. If <i>mode</i> specifies relative positioning, a positive value moves the image down, while a negative value moves the image up. Note that this is the opposite of a normal mathematical coordinate system. If you are programming in COBOL, you should specify this value as an ASCII string.
<i>mode</i>	An integer that specifies whether <i>x</i> and <i>y</i> are absolute or relative. The possible values are:  0: Relative 1: Absolute
<i>type</i>	An integer variable used to specify whether the image is permanent or temporary:  0: Temporary 1: Permanent

### Description

The PPRINTFIGURE intrinsic (and its COBOL version, PPRINTFIGUREA) allows you to specify a filename to hold the partitioned raster image created during the process of printing the figure. Since the conversion can take considerable time, it is a good idea to save raster images whenever a picture will remain unchanged between uses.

PPRINTFIGURE takes care of updating an out-of-date raster image. If the figure you specify has changed since the raster file was created, or if you specify a different size or rotation, the intrinsic automatically reconverts the figure. Otherwise, it uses the data from the raster file.

## PPINTRASTER

Print a partitioned raster image that has been stored in laser printer memory with PLOADRASTER.

### Syntax

	LA	BA	R	R	I16	I16
PPINTRASTER[A]	( <i>comArea</i> ,	<i>imagenum</i> ,	<i>x</i> ,	<i>y</i> ,	<i>units</i> ,	<i>mode</i> )
	in/out	in	in	in	in	in

### Parameters

<i>comArea</i>	The Formation communication area.
<i>imagenum</i>	An integer containing the number of the image to be printed. The image number is the number that was given to the image when it was loaded with PLOADRASTER. The image number can be between 0 and 31 inclusive.
<i>x</i>	<p>A real number that supplies the x-coordinate of the left edge of the picture. PPRINTRASTER interprets the value in the units specified by the <i>units</i> parameter, and as relative or absolute depending on the <i>mode</i> parameter. If positioning is relative, a positive value moves the image to the right; a negative value moves the image to the left.</p> <p>If you are programming in COBOL, you should call PPRINTRASTERA and supply <i>x</i> as an ASCII string.</p>
<i>y</i>	<p>A real number that specifies the location of the top edge of the picture. PPRINTRASTER interprets the value in the units specified in the <i>units</i> parameter, and as either relative or absolute depending on the <i>mode</i> parameter.</p> <p>If you request relative positioning, the value should be positive to move the image down and negative to move the image up. The positioning is opposite of a normal mathematical coordinate system.</p> <p>If you are programming in COBOL, you should call PPRINTRASTERA, and specify this value as an ASCII string.</p>
<i>units</i>	<p>An integer that specifies the units in which <i>x</i> and <i>y</i> are specified. The allowed values are:</p> <ul style="list-style-type: none"> <li>0: Dots</li> <li>1: Inches</li> <li>2: Centimeters</li> <li>3: Millimeters</li> </ul>
<i>mode</i>	An integer value that specifies whether <i>x</i> and <i>y</i> are absolute coordinates on the logical page, or relative to the current pen position.

The permitted values are:

0: Relative

1: Absolute

### **Description**

The PPRINTRASTER intrinsic (and its COBOL counterpart, PPRINTRASTERA) print a stored raster image on the laser printer. The image must already have been loaded with PLOADRASTER.

The x and y positions are checked to make sure that they lie within the bounds of the logical page. The LaserJet printer family permits printing outside the bounds of the logical page. The image is clipped at the boundary of the physical page.

The HP2680 and HP2688 will return an error condition if you try to print outside the bounds of the current logical page.

## PPRINTSAVED

Print all saved data.

### Syntax

```
LA  
PPRINTSAVED(comArea)  
in/out
```

### Parameter

*comArea*            The Formation communication area.

### Description

This intrinsic instructs the LaserJet to print all data saved.

The PSAVE, PENDING, and PPRINTSAVED intrinsics are used mostly for multi-part forms on LaserJet printers to quickly repeat a set of commands.

1. The PSAVE intrinsic tells Formation to begin recording the commands that follow.
2. A PENDING intrinsic is issued to tell Formation to stop recording commands.
3. PPRINTSAVED can then be issued to perform the commands that have been saved.

The PPRINTSAVED intrinsic is useful if you have to repeat a sequence of steps more than once when printing a form. Enter the correct number of PPRINTSAVED intrinsics, and the steps are performed the required number of times.

---

## PRESET

This intrinsic causes a printer reset for HP LaserJet and HP 5000 printers. It is ignored by all other types of printers.

### Syntax

```
LA
PRESET(comArea)
in/out
```

### Parameter

*comArea*            The Formation communication area.

### Description

PRESET should only be used in situations where the printer will not automatically be reset by an HP 3000 Terminal Type file before the output of the next job is printed. It should not be necessary for correctly configured, serially connected LaserJet printers.

**Note:** Use this intrinsic only at the end of a print job. If used at any other time, it will clear any form and font definitions currently being used by the application from the printer.

## PSAVEDATA

Instruct the LaserJet to begin saving data.

### Syntax

```
LA  
PSAVEDATA(comArea)  
in/out
```

### Parameter

*comArea*            The Formation communication area.

### Description

This intrinsic instructs the LaserJet to begin saving data. The saved data can later be printed with PPRINTSAVED.

Application programs can use this intrinsic along with PENS SAVE and PPRINTSAVED to simulate multi-copy forms on the LaserJet. Unlike the HP2680, LaserJet printers do not have the technology to automatically print multi-copy forms. To print multiple copies of data onto different forms, the application program must use PSAVEDATA and PENS SAVE to capture the data to be printed, and PPRINTSAVED to print the data multiple times.

The PSAVEDATA intrinsic tells Formation to begin recording the commands that follow. A PENS SAVE intrinsic tells Formation to stop recording commands. A PPRINTSAVED intrinsic can then be issued to perform the commands that have been saved. This is useful if you have to repeat a sequence of steps more than once when printing a form. Enter the correct number of PPRINTSAVED intrinsics, and the steps are performed the required number of times.



## PSELECTPAGE

Activate and advance to the specified logical page. Optionally perform a physical page eject.

### Syntax

```

                LA           I16           I16
PSELECTPAGE(comArea, logicalpagenum, ejectoption)
                in/out       in           in
  
```

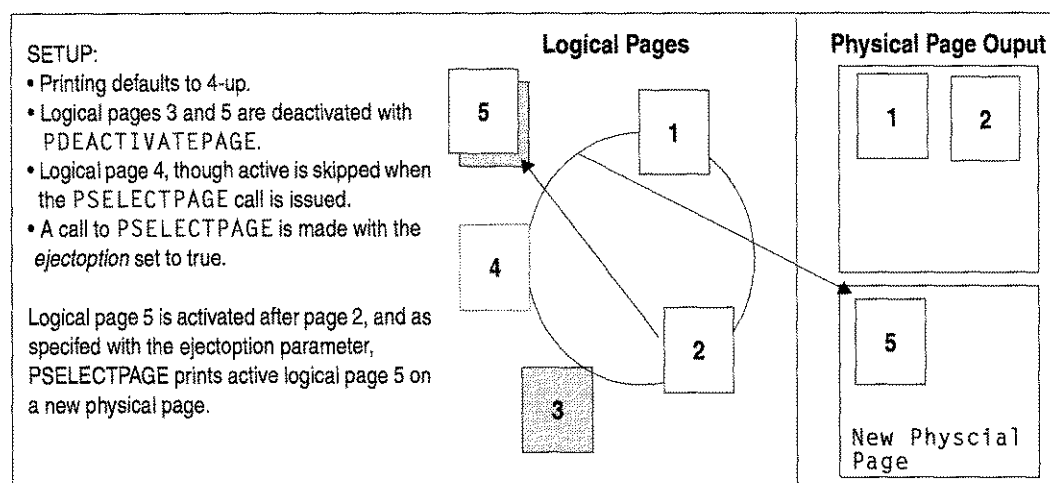
### Parameters

- comArea*            The Formation communication area.
- logicalpagenum*    The number of the logical page. *Logicalpagenum* must be a value between 0 and 31 inclusive.
- ejectoption*        An integer indicating whether you wish to eject the physical page before switching to a new logical page. A value of 0 indicates that no physical page eject should be performed. A non-zero value specifies that the page should be ejected before continuing.

### Description

Normally, the laser printer cycles through active logical pages in sequential ascending order (from low number to high number). Sometimes, you may want to change the order in which pages are printed, and insert an extra page into the sequence. The PSELECTPAGE intrinsic performs this function, the intrinsic:

- Activates the specified page, if it is not already active.
- Switches to printing that logical page.



## PSETCOPIES

Set the printer to print the specified number of copies of each page, beginning with the current page.

### Syntax

```
LA      I
PSETCOPIES(comArea, numcopies)
in/out  in
```

### Parameters

*comArea*            The Formation communication area.

*numcopies*        A 16-bit integer containing the number of copies that the printer should make of each page. The value may range from 1 to 99 for LaserJet printers, or 1 to 32767 for the HP2680 and HP2688. For dual environments, enter the HP 2680/88 value, and Formation will adjust it to 99 for LaserJet printers.

Values outside this range cause PSETCOPIES to return an error.

### Description

PSETCOPIES instructs the printer to make the specified number of copies of each page. The command takes effect with the current page, and continues until changed by another call to PSETCOPIES.

When a LaserJet printer capable of duplex printing is instructed to make multiple copies of a page, the command applies to both sides. Each physical page prints the requested number of times.

Printing using PSETCOPIES is much more efficient than using the spooler to set the number of copies; however, you must collate the output yourself.

## PSPPLUSVERSION

Print the current version of Formation intrinsics.

### Syntax

PSPPLUSVERSION
----------------

### Description

This intrinsic calls HP36580 to print the current IFS intrinsic or Formation substitute intrinsic version, then prints the current version of Formation on the \$STDLIST device. The form of the string is:

Formation x.xx (c) 1990 Tymlabs Corporation

## PSTATEINFO

Return information about the current state of the print job.

### Syntax

LA	BA	BA	LA
PSTATEINFO( <i>comArea</i> ,	<i>formname</i> ,	<i>subformname</i> ,	<i>stateinfo</i> )
in/out	out	out	out

### Parameters

- comArea*            The Formation communication area.
- formname*            A character string to which PSTATEINFO returns the name of the currently active form. The string must be at least 16 characters in length. PSTATEINFO returns a 16-character string padded with blanks if necessary. If there is no form currently active, the intrinsic fills the array with 16 blanks, except in BASIC where a zero-length string is returned.
- subformname*        A character string to which PSTATEINFO returns the name of the currently active subform. The string must be at least 16 characters in length. PSTATEINFO returns a 16-character string padded with blanks if necessary. If no form is currently active, the intrinsic fills the array with 16 blanks, except in BASIC where a zero-length string is returned.
- stateinfo*            A logical array to which PSTATEINFO returns information about the current state of the print job. Its contents are detailed in the following table.

Word	Contents
0	Current logical page number
1	Current primary character font number
2	Current secondary character font number
3	Physical page width in dots (as defined with IFS/3000 or the Environment Manager)
4	Physical page length in dots (as defined in IFS/3000 or the Environment Manager)
5-29	Reserved for future use

### Description

The physical page dimensions returned in words 3 and 4 of the *stateinfo* parameter come directly from the IFS/3000 or Environment Manager **Physical Page Control** screen. If the operator has loaded the wrong size paper, this intrinsic will not detect it.

Do not use a combination of FDEVICECONTROL and Formation calls to control the printer. If you do, the this intrinsic and other information intrinsics will return incorrect information.

## PSTRINGWIDTH

Return the width of a string in dots when printed in a specified font.

### Syntax

LA	I16	BA	I16	I16
PSTRINGWIDTH( <i>comArea</i> ,	<i>fontnum</i> ,	<i>string</i> ,	<i>stringlnth</i> ,	<i>stringwidth</i> )
in/out	in	in	in	out

### Parameters

<i>comArea</i>	The Formation communication area.
<i>fontnum</i>	An integer containing the number of the font in which the string will be printed, and for which you want to calculate the width. The font number must be between -1 and 31 inclusive. -1 indicates that the current primary font will be used for the calculation.
<i>string</i>	The character string whose width is to be calculated. It must be of the length specified by <i>stringlnth</i> , except in BASIC, where the length of the string is determined by the string itself.
<i>stringlnth</i>	An integer giving the number of characters in <i>string</i> . This may be a dummy variable when the intrinsic is called from BASIC.
<i>stringwidth</i>	An integer to which PSTRINGWIDTH returns the number of dots that will be required to print the string in the specified font.

### Description

The character font you specify must be defined in the environment file. Font number 31 is always defined. If the character font is not in the environment file, the default laser printer font is used.

PSTRINGWIDTH includes leading and embedded blanks in the calculation, but ignores trailing blanks. To determine the width of a blank, supply a *string* that consists of a single blank, and a *stringlnth* of 1.

## PSWITCHPAGE

Switch logical pages without performing a page eject.

### Syntax

```
LA  
PSWITCHPAGE(comArea)  
in/out
```

### Parameter

*comArea*            The Formation communication area.

### Description

PSWITCHPAGE and PSWITCHPHYSPAGE are used on a LaserJet to emulate the 2680/88's ability to switch logical pages without performing a physical page eject. For example, if logical page 1 is initially active in the environment, a program can call PACTIVATEPAGE/PDEACTIVATEPAGE to change to logical page 0 upon entering the program, before anything is sent to the printer.

The 2680/88 printers know that nothing has been printed, and therefore change pages without performing the physical page eject that would normally occur when switching from one logical page to a second logical page with a lower number. A LaserJet does not have this intelligence, and always assumes the normal case—that something has been printed. PSWITCHPAGE and PSWITCHPHYSPAGE let you switch logical pages without ejecting.

PSWITCHPAGE refers to the next logical page, and PSWITCHPHYSPAGE refers to the next physical page.

---

## PSWITCHPHYSPAGE

Switch physical pages without performing a page eject.

### Syntax

```
LA  
PSWITCHPHYSPAGE(comArea)  
in/out
```

### Parameter

*comArea*            The Formation communication area.

### Description

PSWITCHPHYSPAGE is used with PSWITCHPAGE on a LaserJet to emulate the 2680/88's ability to switch logical pages without performing a physical page eject. For example, if logical page 1 is initially active in the environment, a program can call PACTIVATEPAGE/PDEACTIVATEPAGE to change to logical page 0 upon entering the program, before anything is sent to the printer.

The 2680/88 printers know that nothing has been printed, and therefore change pages without performing the physical page eject that would normally occur when switching from one logical page to a second logical page with a lower number. A LaserJet does not have this intelligence, and always assumes the normal case—that something has been printed. PSWITCHPHYSPAGE and PSWITCHPAGE let you switch logical pages without ejecting.

PSWITCHPHYSPAGE refers to the next physical page and PSWITCHPAGE refers to the next logical page.

## PUSEFONT

Select primary and secondary character fonts.

### Syntax

	LA	I16	I16
PUSEFONT(	<i>comArea</i> ,	<i>primary</i> ,	<i>secondary</i> )
	in/out	in	in

### Parameters

<i>comArea</i>	The Formation communication area.
<i>primary</i>	An integer containing the number of the font that you wish to select as the new primary font. You may specify any font number in the environment, or a value of -1 to leave the primary font unchanged.
<i>secondary</i>	An integer containing the number of the font that you wish to select as the new secondary font. You may specify any font number in the environment, or a value of -1 to leave the secondary font unchanged.

### Description

The PUSEFONT intrinsic is used to specify two fonts for every intrinsic call. The following considerations apply when using this intrinsic:

- Once the primary and secondary fonts are specified, use shift-in and shift-out escape sequences to switch between the two.
- When one primary font is switched to another, the software automatically calculates and sets the ideal line spacing for that font, overriding any other line spacing you may have set. To avoid changes to the established line spacing, set the primary font to -1 and use the shift-in and shift-out to use the secondary font.
- A maximum of 31 different fonts can be used in a single environment.
- When selecting bar codes for printing, you must remember that the bar code uses printable symbolic characters to represent a blank space. The printer will print bar code characters representing blank spaces across your physical page, if you shift to a bar code font and then space over to a desired position.
  - If you want to precede a bar code segment with real blank spaces, you must shift to a font that does not print characters for blank spaces and then shift to the bar code font when you have spaced to the desired position.
  - Even in programming languages that allow you to precisely position a character, you must shift to a non-bar code font before positioning the bar code.

**Note:** Use the PFONTNUM intrinsic to determine a font number given its name.



## PWRITEFIELD

Write data to a field or subfield on a form.

### Syntax

LA	BA	I16	BA	I16
PWRITEFIELD( <i>comArea</i> , <i>fldname</i> , <i>subfld</i> , <i>data</i> , <i>dataInth</i> )				
in/out	in	in	in	in

### Parameters

<i>comArea</i>	The Formation communication area.
<i>fldname</i>	A character string that specifies the name of the field to which the data is written. The string may be up to 16 characters long, and must be terminated by a blank or EOS, or the 16th character, except in BASIC. The name must exist in the current form (and subform, if you have specified one).
<i>subfld</i>	An integer that specifies the number of the subfield to which the data will be written. Subfields are numbered starting with 1. PWRITEFIELD checks the subfield number that you supply, and returns an error if the subfield number is larger than the number of subfields in the field. A value of -1 indicates there is only one field (subfield).
<i>data</i>	A character string containing the data to be written. It must be of length <i>dataInth</i> , except in BASIC, where the length of the data is determined by the data string itself.
<i>dataInth</i>	An integer in which you supply the number of characters contained in <i>data</i> . You may supply a dummy variable for this parameter if you are programming in BASIC.

### Description

The PWRITEFIELD intrinsic writes one or more lines of data into a single subfield. Lines in the data array must be separated with a carriage return character (decimal value 13; octal %15). A carriage return is not required after the last line of data unless the last line is blank. If there are any trailing blanks at the end of each line of the data array, they are stripped before the data is written to the field. PWRITEFIELD centers data vertically in the field (unless underlines are specified in the field) and left-justifies horizontally. Since all lines of data, even blank lines, are counted when the intrinsic is centering data, you may want to include blank lines in your data to insure correct positioning within the subfield. The intrinsic adds 1/3 of the cell height as padding on the left side of the data.

If underlines are specified for any portion of the field, PWRITEFIELD does not center the data but relies on the IDForm grid for positioning. You should insure that the font you use for filling in the form matches the spacing you specified when you created the form. See the *IDForm Reference Guide* (Hewlett-Packard part number 36581-90002) for additional information.

The maximum length of a print line on the HP2680 or HP2688 is 250 characters. On the LaserJet, the maximum length is limited primarily by the physical dimensions of the page.

## PWRITEFIELDJUST

Write data to a field with right, left or centered justification.

### Syntax

	LA	BA	I16	BA	I16	I16
PWRITEFIELDJUST	( <i>comArea</i> ,	<i>fldname</i> ,	<i>subfld</i> ,	<i>data</i> ,	<i>dataInth</i> ,	<i>just</i> )
	in/out	in	in	in	in	in

### Parameters

<i>comArea</i>	The Formation communication area.
<i>fldname</i>	A character string that specifies the name of the field to which the data is written. The string may be up to 16 characters long, and must be terminated by a blank or EOS, or the 16th character (except in BASIC). This name must exist in the current form.
<i>subfld</i>	An integer that specifies the number of the subfield to which the data will be written. Subfields are numbered starting with 1. PWRITEFIELDREL checks the subfield number that you supply, and returns an error if the subfield number is larger than the number of subfields in the field. A value of -1 indicates there is only one field (subfield).
<i>data</i>	A character string containing the data to be written. It must be the length <i>dataInth</i> , except in BASIC, where the length of the data is determined by the data string itself.
<i>dataInth</i>	An integer in which you supply the number of characters contained in data. You may supply a dummy variable for this parameter if you are programming in BASIC.
<i>just</i>	An integer denoting the type of justification desired: 0 = left 1 = right 2 = centered

### Description

At this time, word wrapping is not implemented, so the data to be justified should not exceed the field length. Refer to the discussions of PWRITEFIELD on page 6-77 and PWRITEFIELDREL on page 6-79 for more information.

## PWRITEFIELDREL

Write data at a location relative to the position of a field or subfield. If provided with no data, this intrinsic leaves the pen at the requested position on the page.

The ASCII version, callable from COBOL, is denoted by an "A" after the intrinsic name: PWRITEFIELDRELA. In this case any real values must be specified as 16 character long strings, padded with blanks if necessary.

### Syntax

	LA	BA	I16	BA	I16	I16	R	R
PWRITEFIELDREL[A]	<i>comArea</i>	<i>fldname</i>	<i>subfld</i>	<i>data</i>	<i>data1nth</i>	<i>psition</i>	<i>x</i>	<i>y</i>
	in/out	in	in	in	in	in	in	in

### Parameters

<i>comArea</i>	The Formation communication area.
<i>fldname</i>	A character string that specifies the name of the field to which the data is written. The string may be up to 16 characters long, and must be terminated by a blank or EOS, or the 16th character (except in BASIC). This name must exist in the current form.
<i>subfld</i>	An integer that specifies the number of the subfield to which the data will be written. Subfields are numbered starting with 1. PWRITEFIELDREL checks the subfield number that you supply, and returns an error if the subfield number is larger than the number of subfields in the field. A value of -1 indicates there is only one field (subfield).
<i>data</i>	A character string containing the data to be written. It must be the length <i>data1nth</i> , except in BASIC, where the length of the data is determined by the data string itself.
<i>data1nth</i>	An integer in which you supply the number of characters contained in <i>data</i> . If a value of zero is specified, the pen is left at the requested position on the page. This intrinsic is essentially a field-relative equivalent of PMOVEPENABS.
<i>psition</i>	An integer variable designating relative positioning to be used. Values are: <ul style="list-style-type: none"> <li>0 Position data as PWRITEFIELD, centering first line of text within subfield. <i>x</i> and <i>y</i> parameters ignored.</li> <li>1 Dots: <i>x</i> and <i>y</i> parameters specify offset from top left corner of subfield in dots.</li> <li>2 Inches: <i>x</i> and <i>y</i> parameters specify offset from top left corner of subfield in inches.</li> <li>3 cm: <i>x</i> and <i>y</i> parameters specify offset from top left corner of subfield in centimeters.</li> <li>4 mm: <i>x</i> and <i>y</i> parameters specify offset from top left corner of subfield in millimeters.</li> </ul>

5 Points:  $x$  and  $y$  parameters specify offset from top left corner of subfield in points.

6 %:  $x$  and  $y$  parameters specify offset from top left corner as a percentage of the subfield height and width.

$x$  A real number variable specifying the offset from the top left corner of the subfield in the  $x$  dimension. Negative and fractional values may be specified.

$y$  A real number variable specifying the offset from the top left corner of the subfield in the  $y$  dimension. Negative and fractional values may be specified.

# *HP 2680 Applications Printing to LaserJets and HP 5000s*

## **Introduction**

This chapter is a Quick Start guide for HP 2680/88 users who want to use Formation to print their existing applications on LaserJets and HP 5000s, and for HP 2680/88 users who want to print multi-part forms. After reading this chapter, HP 2680/88 users should know:

- The basics of what's involved in migrating your HP 2680/88 environment files so you can print them with Formation.
- The basic steps required to convert your environment files and programs so they can be used on any HP 5000, LaserJet, or HP 2680/88. With Formation, you never have to make an abrupt change from one printer to another since all models are supported simultaneously.
- What to do when you have to make changes to a form. You have the choice of continuing to use IDSTForm or converting your forms so they can be modified with Formation Design, the PC-based WYSIWYG design tool that comes with Formation.
- How to define multi-part forms on an HP 2680 or HP 2688.

In addition, the section, "Beyond Migration" discusses some of the other things you can do with Formation such as check printing and creating intelligent forms.

## **Topics**

This chapter describes the following HP 2680/88 forms printing and migration topics:

- What is Migration?
- Updating Applications for New Printers
- Analyze Migration Requirements
- Recompile Environment Files
- Convert Programs
- Modify/Update Job Streams
- Modifying Forms
- Multi-Part Forms on HP 2680 and HP 2688 Printers
- Beyond Migration

## What is Migration?

Modifying your HP 2680/88 environment files and programs so you can print forms on HP 5000s and LaserJets is called **migration**. Migration is necessary because the software mechanisms that produce formatted output on an HP 2680 laser printer are not immediately compatible with the HP 5000 or LaserJet. These mechanisms are:

1. **Custom environment files.** Your custom environment files, created with IFS and IDForm, contain the forms, fonts, and graphics needed to print a particular job. This information is described in DCS, the internal page description language of the HP 2680. Environment files may also contain the page coordinates of named data fields to be printed on the form.
2. **Hewlett-Packard's PSP intrinsics.** This is a library of subroutines programmers can call in printing applications to implement functions like form and font selection without having to write code in the printer's internal page description language. The intrinsics can also be used to place data in named fields defined on a form.

LaserJets and the HP 5000 use PCL, not DCS, as their internal language, so they cannot print custom environment files created with IFS. This alone prevents many HP 2680 applications from being printed on these printers. In addition, the PSP intrinsics do not support PCL printers. These intrinsics are found in both custom printing applications as well as packaged applications from Hewlett-Packard such as TDP and HPWord. Formation provides the tools to let you use HP 2680 applications on your PCL printers.

### Migrating with Formation

This Quick Start guide is designed help you migrate HP 2680 applications to the LaserJet or HP 5000. Here are the key steps involved:

1. **For MPE XL Users Only: Print migration analysis.** Formation can analyze the files on your system to determine exactly what must be done to migrate to the LaserJet or HP 5000. It checks each environment file on your system to see which form files, font files and graphics files are needed for recompilation. It looks for programs that call the PSP intrinsics and locates print job streams.
2. **Recompile custom environment files.** The recompiled files can include PCL, DCS, or both. These files can be printed on any HP laser printer: the HP 5000, LaserJets, and the HP 2680 or HP 2688.
3. **Convert print programs.** A simple procedure redirects calls to PSP intrinsics in both custom and packaged software to matching Formation procedures which extend the functionality of PSP to any HP laser printer: the HP 5000, LaserJets and the HP 2680 or HP 2688.
4. **Update job streams.** If you've purchased an HP 5000, or if you have MPE XL 4.0, HP has supplied you with replacements for some of the standard environment files you used with your HP 2680 or HP 2688. You should replace references to these environment files in job streams that will now print to your HP 5000. Also, in all printing job streams, you'll add a JCW that tells what printer you're using.

---

## Commonly Asked Questions About Migration

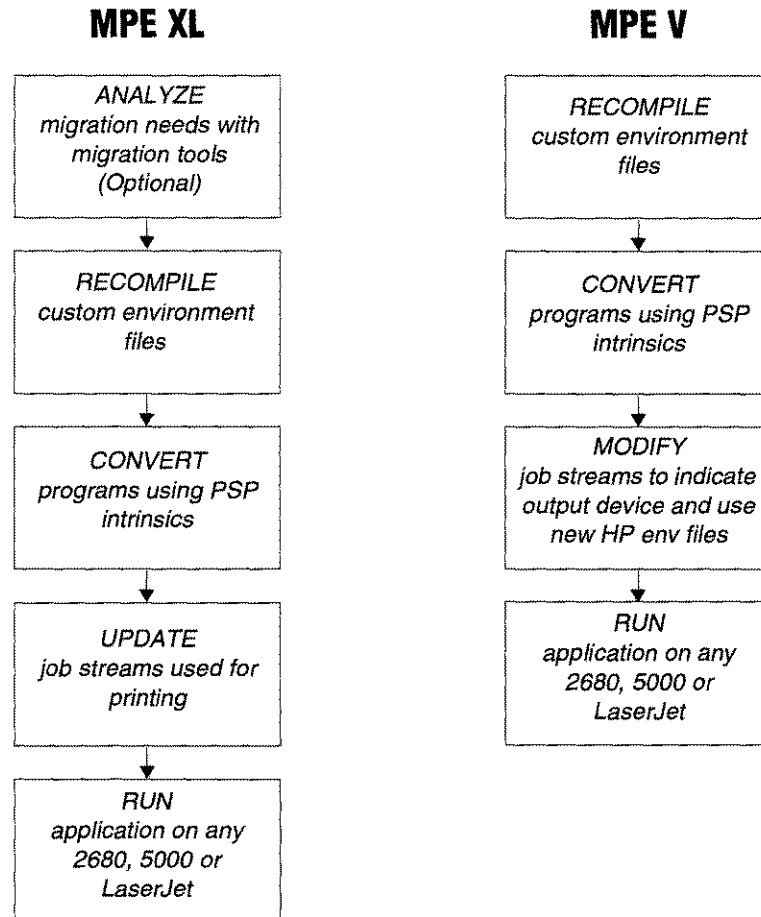
- Q:** If I use Formation to recompile an existing environment file for the LaserJet and HP 5000, can I still use this environment file for my HP 2680?
- A:** Yes. A single Formation environment file can contain both DCS and PCL versions of a form.
- Q:** Once I have installed Formation, can I still use IDSSForm to modify my current forms and to design new ones?
- A:** Yes. Just recompile with the Formation Environment Manager, and you're ready to go.
- Q:** Can I use Formation Design to create new forms for my HP 2680?
- A:** Yes, you can use either Formation Design or IDSSForm to design new forms. Formation can compile both types of forms for both types of printers.
- Q:** Will I have to change my job streams or my operating procedures to be able to print my HP 2680 jobs on the HP 5000 or LaserJet?
- A:** No. You just set a JCW to indicate which printer you want to use. To print edge-to-edge forms on the LaserJet, you'll use Formation's scaling option to reduce the form to fit the LaserJet's smaller printable area. The HP 5000 supports edge-to-edge printing.
- Q:** What PC configuration do I need to design forms with Formation?
- A:** You can run Formation Design on any PC that runs Microsoft® Windows™ 3.0 or later. We recommend at least a 386SX CPU, 2 to 4 MB of memory, 40 MB hard drive, VGA monitor and a mouse.
- Q:** After designing a form on my PC, how do I get it onto my HP 3000?
- A:** Just upload the form file using terminal emulator for Windows-based PCs.
- Q:** I have an HP 2688. Can I migrate to the newer printing technology?
- A:** The HP 2688 is essentially a slower version of the HP 2680, except that the HP 2688 prints at the same 300 dots per inch (DPI) resolution as the HP 5000 and LaserJets. With Formation, migration of HP 2688 applications to the newer printers is the same as with the HP 2680, but without the 180 DPI to 300 DPI conversion.
- Q:** I want to design an electronic form to replace a preprinted one I use with a packaged application from a third-party. How can I print on a Formation form without access to the source code?
- A:** It's easy. For a simple, one-part form, just add the ENV= parm to your printer file equation to identify the environment file containing the form you want to merge with your output. For a multi-part form, redirect your application's output to a disc file and then use LPRINT, Formation's post processor.
- Q:** IFS comes with a number of fonts. Can I use all these fonts with my HP 5000 and Formation?
- A:** Yes, Formation automatically scales the 180 DPI fonts you are currently using with your HP 2680 to the 300 DPI resolution of the HP 5000.
- Q:** Where do I get other fonts for use with Formation?

- A:** Formation includes a copy of AGFA Compugraphic's Type Director, plus several Compugraphic font families. These fonts are supplied in both 300 DPI for your HP 5000 and LaserJets and 180 DPI for your HP 2680. If you need more typefaces, you can buy additional Type Director fonts or PCL fonts from Unison Software, at your local PC retail outlet, or through the mail. Unison Software can even design custom fonts.
- Q:** Can I use bar codes with Formation?
- A:** Yes, you can use bar code fonts just like any other font, both as a part of a form and for data. Formation includes 3 of 9, 2 of 5, and the U.S. Post Office's Postnet fonts, with UPC, Code 128, and others available for a nominal fee.
- Q:** In the past, I paid HP to design forms for my HP 2680. Does anyone offer this service for Formation?
- A:** Through our Forms Express program, Unison Software offers forms design, logo and signature scanning, and customized font development services at competitive prices.
- Q:** In the past when I needed to include a logo, signature or other graphic on an IFS form, I sent it to HP. HP converted it into a special font in a PCELL file. Will these fonts work with Formation?
- A:** Yes, Formation supports such fonts just like any other PCELL file.
- Q:** How will I get graphic images into new forms I design with Formation?
- A:** Use a scanner like the HP ScanJet with HP's Scanning Gallery software or Logitech's ScanMan to produce a TIFF image. Then use Formation Design to place this image anywhere on your form. If you don't have access to a scanner, Unison Software can do this for you for a nominal charge.
- Q:** Does Unison Software offer training for Formation?
- A:** Yes, we offer standard and customized training classes for Formation, including a special one-day migration planning course for HP 2680 and HP 2688 owners who are moving to the HP 5000.
- Q:** Can I use Formation on my Classic HP 3000 with MPE V?
- A:** Yes. The HP 2680 and LaserJets are supported as directly attached devices. Even though the HP 5000 is not supported on MPE V systems, you can use Formation to create spool files which when moved to an MPE XL system can be printed on your HP 5000.
- Q:** Can I develop forms on one HP 3000 system and then print on another?
- A:** Yes. Today you can freely move forms between both MPE V and MPE XL systems.
- Q:** Do I have to buy a full copy of Formation to print on my production systems?
- A:** No, you only need a full copy for systems on which you will develop new forms. For your production systems, Unison Software offers an economical run-time version called Formation Print Services.
- Q:** What kind of support is available for Formation?
- A:** Unison Software offers a comprehensive support contract, including 24-hour phone-in consulting and automatic shipments of new releases, for an annual fee. Additional CPUs which are centrally administered may be covered under the same agreement at a reduced cost.



## Updating Applications for New Printers

This section takes you through the basic steps required to get your applications running on LaserJets and HP 5000s.



### Analyze Migration Requirements

#### **MPE XL** Users Only

The Formation migration tool analyzes the files on your system, and prints reports you can use as a road map to lead you through the conversion process. If you already know which environment files and programs you want to convert, you can skip this step and go on to the next section.

To install the migration tool:

1. Log on in FORM.TYM.
2. Add the passwords for MANAGER.SYS to line 1 of JMIG.FORM.TYM.
3. Stream the job.

```
:STREAM JMIG.FORM.TYM
```

This job creates FORMMIG.SYS, a group with PM capabilities that contains the migration analysis tool. After you are done with migration, you can purge the entire group. If you should need to use the tool again, recreate the group by streaming JMIG.

To run the migration analysis, enter the following:

```
:HELLO MANAGER.SYS, FORMMIG
:RUN MENU
```

The **Formation Migration Menu** appears:

FORMATION FROM TYMLABS -- MIGRATION MENU		
NO	TYPE	DESCRIPTION
01	(TASK)	Perform Migration Analysis
02	(TASK)	Print Environment Files with Missing Components
03	(TASK)	Print All Environment Files and Components
04	(TASK)	Print Programs Using PSP Intrinsic
05	(TASK)	Print Jobstreams Using Environment Files

E - End ? - Help  
ENTER STEP █

Type 01 and press `x`. The following fileset prompt is displayed:

```
***** PERFORM MIGRATION ANALYSIS
```

Enter the fileset you want to analyze:

```
@.@.@
```

Enter @.@.@ as your fileset. This tells Formation to examine all files on your system. During the analysis, error messages are produced on your screen if a file cannot be accessed. If, for example, a print job is running while you perform the analysis, the files involved in that job won't be examined.

When analysis is complete, the **Main Menu** is displayed. From this menu, you can print any necessary reports. As each report runs, output from the sort procedures is displayed on the screen. This output is also sent to the system line printer. The individual reports are discussed in the section that follow.

**Note:** It is best to perform migration analysis overnight, or during off-hours. This process may tie up the terminal for a while.

## Recompile Environment Files

The environment files used with HP 2680 applications contain instructions in DCS, the internal language of the HP 2680. When you recompile with Formation, you add a parallel set of instructions in PCL, the internal language of LaserJets and HP 5000s.

To recompile an environment file, you need all the components it describes. These may include:

- PFORM files, which are form files created with IDSSForm
- PCELL files, which are data fonts or form fonts supplied by HP or created with IDSSChar.

Once a particular environment file has been compiled, it can be printed without having these components on the system. However, the components must be present to recompile.

If any components are not found in the groups where the environment file expects to find them, the recompilation fails, and a list of missing components is displayed. The environment file you attempted to recompile is unusable until you find the missing components, load them on your system, and recompile again. *To avoid problems, make a duplicate copy of the environment file before you recompile.*



To obtain a list of missing components:

1. Run the Formation migration analysis as described on page 7-6.
2. When the **Formation Migration Menu** appears, enter 02 to print the **Environment Files with Missing Components** report.

The following shows an example of a missing components report.

Formation from Tynlabs Migration Analysis											
Page: 2											
Print Date: 27May92 Time: 18:48											
Acct: SYS	Environment Files with Missing Components										
Environment File Name	Printer	Access	Modify	Component			Last	Last	Modify	Access	Not Found
LP884P.HPENV.SYS	2680A	910808	920311	Data Font-	LINEPR66.CHARSETS.SYS	8 Pt	300 DPI	20 Dots	920320	920320	
				Form File-	BORDERS.HPENV.PSP	Form-	BORDERHPLP884P				*FORM FILE
				See-	BORDERS.ENVCELFO.JSTEST				920303	920311	
				See-	BORDERS.ENVFILES.FORMDEV				911206	920316	
				See-	BORDERS.ENVMGRC.LTDEV				910218	920311	
				See-	BORDERS.HPENV.SYS				910808	920311	
				See-	BORDERS.IDSZFRM.LTDEV				910826	920311	
				See-	BORDERS.PCLENV40.SYS				920115	920311	
				See-	BORDERS.PUB.LTDEV				910218	920311	
VENV80.HPENV.SYS	2680A	910115	920311	Data Font-	PARNR80S.FONTS.QUILL	180 DPI		30 Dots			*PCELL FILE
				See-	PARNR80S.ENVFILES.FORMDEV				911206	920316	
				See-	PARNR80S.HPSNSHOT.FORMCUST				920303	920316	
				Data Font-	ROMITAL.CHARSETS.SYS	10 Pt	180 DPI	25 Dots	910923	920320	
				Data Font-	ROM.CHARSETS.SYS	10 Pt	180 DPI	25 Dots	910923	920320	
				Data Font-	ELITR80G.FONTS.QUILL	180 DPI		30 Dots			*PCELL FILE
				See-	ELITR80G.ENVFILES.FORMDEV				911206	920316	
				See-	ELITR80G.HPSNSHOT.FORMCUST				920303	920316	

This report lists each environment file, what printer it was compiled for, and the dates of last modification/access to the file. This can assist you in deciding whether to locate missing components or just ignore it since the environment file is no longer used. In the

Component column, the data fonts (or PCELL files) and form files referenced in this environment file are listed. If one of these files is not found, an error is printed in the Not Found column at right. Files with the same file name in other groups and accounts are listed below the name of the missing data font or form file.

In the example, the form file BORDERS.HPENV.PSP could not be found. Other files on the system with the name BORDERS are listed. To see whether one of these is the right file, you should make a copy of the environment file LP884P.HPENV.SYS. Set a file equation for BORDERS.HPENV.PSP, equating it to the alternate BORDERS file you want to check out. Then recompile and print to see if the alternate form file is correct.

**MPE V  
Users Only**

To learn whether you are missing any components, do the following:

1. Make a duplicate copy of the environment file using another name.
2. Recompile the duplicate copy according to the instructions on page 7-9.
3. A list of any missing components is displayed.
4. Locate components as discussed in the next section, load them on your system, and compile again.

### Find Missing Components

If you are missing any forms, logos, or font files, the following suggestions may help.

If...	Then...
your environment files were created in-house,	your MIS department probably has the elements needed to recreate them.
you received environment files used with a particular application from the supplier of the application,	approach that party to get either the component PFORM and PCELL files, or a recompiled environment file ready to use with LaserJets and HP 5000s.
a form cannot be located,	it can be recreated using Formation Design. See the Formation Design User Guide.
the PCELL file containing a logo cannot be located,	<p>you must re-scan the logo. If you no longer plan to print on the HP 2680, scan it at 300 DPI to take advantage of the higher resolution of the LaserJet and HP 5000. To use the resulting TIFF file, do one of the following:</p> <ul style="list-style-type: none"> <li>- Run the CONVERT utility to convert the TIFF file into the PCELL file your environment file is looking for. Refer to "Convert Utility" on page 5-17 for more information.</li> <li>- Run IDS2FORM to convert your form to Formation Design format. Use Formation Design's logo placement tool to add the TIFF file to your form. See "Beyond Migration" on page 7-28</li> </ul>
any fonts are missing,	they can usually be supplied by Hewlett-Packard or Unison Software.

## Check for Special Features

If your environment file takes advantage of certain built-in capabilities of the HP 2680 and HP 2688, you must consider the following exceptions before recompiling and printing to a different printer.

**Rotated Data Text** If your environment file contains a rotated data font (oriented opposite to the logical page), it only prints properly on a LaserJet III, IIID, IIIP or IIISi, an HP 2680 or an HP 2688. If the rotated text is part of the actual form, it can print on any printer.

**Multi-part Forms** If your environment file relies on the HP 2680 to print multi-part forms or multiple logical pages on a single physical page, and you want to print it on a LaserJet or 5000, use the logical page control features of the LPRINT utility.

### Multiple Logical Pages

If you want to print multiple logical pages on a single physical page on a LaserJet, use LPRINT as described in Chapter 4, *Printing Options*. If you want to print multiple logical pages on a single physical page on a HP 5000, recompile the environment file with the F100LP parameter.

## Recompile for New Device

To recompile an environment file so that it can be used with LaserJets and HP 5000s as well as the HP 2680 or 2688, enter the following command:

```
:RUN ENVMGR.FORM.TYM;INFO="envfilename=device"
```

Where:

*envfilename* Specifies the name of the environment file to be compiled.  
*device* Specifies any of the devices listed in the following table.

Table 7-1: Device Types

Device	Prints to...
LJPLUS	LaserJet Plus, the 500+.
LJSII	LaserJet Series II.
LJIII	Any LaserJet III or 4.
LJ4	Any LaserJet 4.
LJET	<b>Any</b> LaserJet and the HP 5000. Because extra code is required to support older LaserJets, forms containing TIFF images print more slowly on all printers. Compile for 2684D if you don't have a 500+, Plus or II.
LJ2000	The LaserJet IID, IIP, all 2000s, all IIIs, all 4s and the HP 5000. <b>Not</b> the LaserJet 500+, Plus or II. Provides better performance than LJET.
F100	HP 5000.
F100LP	HP 5000 when printing multiple logical pages on a single physical page without using LPRINT or the intrinsics, or when recompiling any 2680/88 environment file supplied by HP for use with the F100. Environment files compiled for F100LP only print using a file equation, not with LPRINT or the intrinsics.

For example, to recompile an environment file called MYENV for use on any LaserJet or HP 5000:

```
:RUN ENVMGR.FORM.TYM;INFO="MYENV=LJET"
```

**Specify Scaling (if necessary)**

The HP 2680, HP 2688, and HP 5000 printers support printing from the top edge of the paper to the bottom, and from the left edge of the paper to the right. The print region on LaserJet printers is limited to 8" x 10.5". To print HP 2680 or 88 forms on a LaserJet without clipping, you can scale IDS forms and PCELL fonts during recompilation by 92% or 96%.

To scale environment files during recompilation, specify a scaling factor in your INFO string as shown in one of the examples below:

```
:RUN ENVMGR.FORM.TYM;INFO="envfilename=LJET S92"  
:RUN ENVMGR.FORM.TYM;INFO="envfilename=2684D S96"  
:RUN ENVMGR.FORM.TYM;INFO="envfilename=LJET S100"
```

The following table will assist you in determining which scaling factor to use.

**Table 7-2: Scaling Factors**

Scaling	Used When...
92	the original form prints edge-to-edge in either direction.
96	there is at least 1/8" margin all the way around the original form.
100	you need to return the environment file to its original unscaled form.

If you print the scaled form and realize you need either more or less scaling, recompile with the new scaling factor. For example, if you've scaled a form 96% and it's still being clipped, you can recompile it with a scaling factor of 92. Note that scaling is always done with respect to the original size of the form, rather than to any scaling you've already done.

**Note:** To print on LaserJets and HP 5000s with an identical image area on both printers, use the front control panel of the HP 5000 to set a left margin equal to the LaserJet. To match a LaserJet II, set a margin of 50 dots on the HP 5000. To match other LaserJets, set a 75-dot margin.

**Interactive Recompilation**

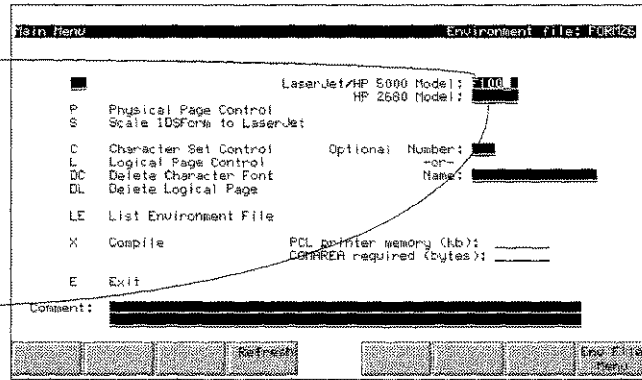
To recompile and scale an environment file interactively, do the following:

1. Run the Environment Manager.  
:RUN ENVMGR.FORM.TYM  
The **Environment File Menu** is displayed.
2. Type the environment file name and press **e**. The **Main Menu** is displayed.

3. Enter your target printer in the **LaserJet/HP 5000 Model:** field.

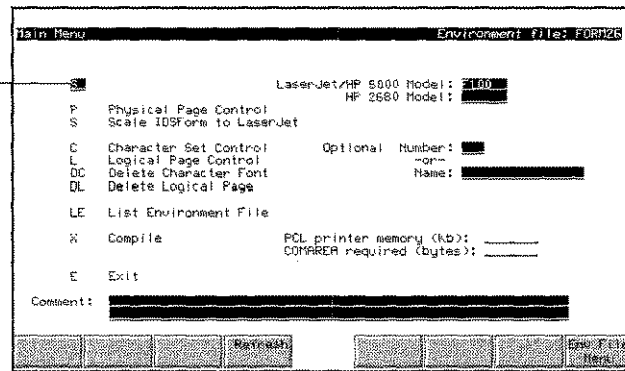
Enter LJET  
F100 or  
F100LP  
here.

Your old  
target  
printer is  
displayed  
here.



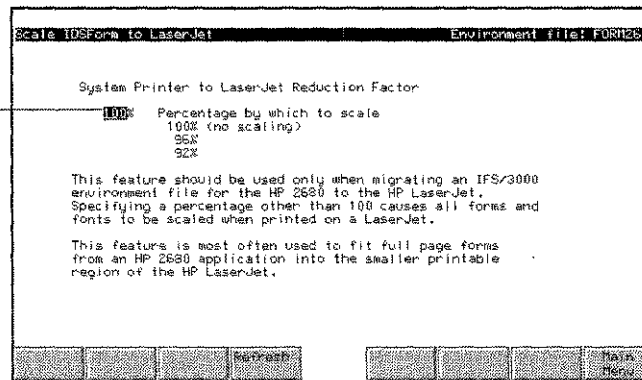
4. If scaling is required, type S to go to the **Scaling** menu.

Type S to access  
Scaling menu.



5. On the **Scaling** menu, enter the desired scaling factor.

Use chart on the  
previous page to  
choose one of the  
three values.



6. Press 8 to return to the **Main Menu**.
7. Enter X in the selection entry box on the **Main Menu**, and press e to compile the environment file.

## Convert Programs

Hewlett-Packard's PSP intrinsics are a library of subroutines that allow HP 2680 programmers to implement functions like form and font selection without having to code in DCS. To let programs containing these intrinsics run on the 5000 and the LaserJet, as well as the HP 2680, and to let you write new programs which perform these functions on any of these printers, Formation provides a set of DCS- and PCL-compatible intrinsics that replace the PSP intrinsics. To convert your existing programs to use the Formation intrinsics, follow the instructions below. Converted programs continue to run on the HP 2680 and 2688, as well as LaserJet and HP 5000 printers.

**MPE XL  
Users Only**

To print a report listing programs using the PSP intrinsics, run the Formation migration analysis as described on page 7-6. When the **Formation Migration Menu** appears, enter 04 to print the **Programs Using PSP Intrinsics** report. Excerpts from the CM and NM portions of the report follow:

Format: CM Compatibility Mode Program				Formation from Tymbabs Migration Analysis Programs Using PSP Intrinsics		Print Date: 27May92	Page: 1 Time: 18:39
Format	Program File Name	Last Modify	Last Access				
CM	FIGPRINT.JUDY.FORMDEV	09Mar92	04Apr92				
CM	LPRINT0.JUDY.FORMDEV	09Mar92	04Apr92				
CM	LPS.JUDY.FORMDEV	09Mar92	04Apr92				
CM	XINTR3.JUDY.FORMDEV	09Mar92	04Apr92				
CM	XTST2.JUDY.FORMDEV	09Mar92	04Apr92				

Format: NM Native Mode Program				Formation from Tymbabs Migration Analysis Programs Using PSP Intrinsics		Print Date: 27May92	Page: 1 Time: 18:39
Format	Program File Name	Last Modify	Last Access				
NM	MIGTOOL.MIGSRC.FORMDEV	04Apr92	04Apr92				
NM	FMTESTX.MIKEB.FORMDEV	04Mar92	04Apr92				
NM	SPRINT.SERVER.FORMDEV	28Feb92	25Mar92				

### Convert MPE V and Compatibility Mode Programs

**MPE V  
Users Only**

The JETSTART utility is used with MPE V programs and with MPE XL compatibility mode programs only. It updates references in the object program that currently point to the PSP intrinsics so that they point to the Formation intrinsics.

JETSTART can be run in batch mode or in interactive mode. The programs you are converting must not be allocated or in use at the time. Use the :SHOWALLOCATE command to check the allocation status of your program, and :DEALLOCATE to deallocate them.

To use JETSTART in batch mode, specify the fully qualified file name of the program to be converted on the INFO string of the RUN command as shown below. Wildcards can be specified.

```
:RUN JETSTART.FORM.TYM;INFO="prog.group.acct"
prog.group.acct: Converted to use Formation.
END OF PROGRAM
```

You can use the wildcards @, #, and ? in the fileset specification. For example, the following command would convert all programs in MYACCT that use the PSP intrinsics.



```

:RUN JETSTART.FORM.TYM;INFO="@.@.MYACCT"
prog1.group.MYACCT: Converted to use Formation.
prog2.group.MYACCT: Converted to use Formation.
prog3.group.MYACCT: Converted to use Formation.
...
END OF PROGRAM

```

To use JETSTART in interactive mode, run the program and supply filesets to be converted one at a time.

```

:RUN JETSTART.FORM.TYM
Formation Jetstart Utility 3.50 (c) 1990-93 Tymbabs Corp.
Fileset to convert (<cr> to end)? prog1.group.acct

```

### Convert HPWord, TDP and LPS

In addition to your custom programs, HP applications like HPWord, TDP and LPS should also be converted with JETSTART. This is shown in the example below. Note that when you convert TDP, you convert the formatter (SCRIBE or TDPFMT), not the TDP editor (TDP).

```

:RUN JETSTART.FORM.TYM
Formation Jetstart Utility 3.50 (c) 1990-93 Tymbabs Corp.
Fileset to convert (<cr> to end)? HPWORD
HPWORD.PUB.SYS: Converted to use Formation.
Fileset to convert (<cr> to end)? SCRIBE
SCRIBE.PUB.SYS: Converted to use Formation.
Fileset to convert (<cr> to end)? LPS
LPS.PUB.SYS: Converted to use Formation.
Fileset to convert (<cr> to end)? <return>
END OF PROGRAM

```

### When You Recompile

Because JETSTART modifies object code rather than source code, you need to run it again any time you recompile a source program that calls the PSP intrinsics.

**Note:** If you don't want to modify the copies of HPWord, TDP, or LPS in PUB.SYS, use the job stream JFORMX.FORM.TYM to copy HP applications into the FORMX.TYM group, and convert them to use the Formation intrinsics. This job stream also creates a group SL and places the intrinsics in it for these applications to access. You will then run these applications from FORMX.TYM with the LIB=G parameter.

### Modify Execution of Native Mode Programs

To have native mode print programs use the Formation intrinsics, add a parameter to the :RUN command that executes the program. If the program is run from a job stream, this change should be made in the job stream file.

To use the Formation intrinsics, add the parameter XL="PSPLUXL.FORM.TYM" to the MPE RUN command.

```
:RUN MYPROG;XL="PSPLUXL.FORM.TYM"
```

If there is already an XL shown, add Formation's XL after it, separated with a comma:

```
:RUN MYPROG; XL="MYXL.GROUP.ACCT,PSPLUXL.FORM.TYM"
```

If and when the program code is re-linked using the HP Link Editor, the XL can be specified on the LINK command. Once this has been done, you no longer need to specify the XL parameter on the :RUN command.

```
:LINKEDIT
```

```
HP Link Editor/XL (HP30315A.04.04)
```

```
LinkEd> link from=myobj;to=myprog;xl=pspluxl.form.tym
```

```
LinkEd> exit
```

## Modify/Update Job Streams

**MPE XL  
Users Only**

If you are using the Migration Analysis Tool, run the Formation migration analysis as described on page 7-6. When the **Formation Migration Menu** appears, enter 05 to print the **Job Streams Using Environment Files** report. The following is an excerpt from the report:

Formation from Tynlabs Migration Analysis Jobstreams Using Environment Files				Print Date- 01Jul92	Page- 1
Environment File Name	Replaced By	Referenced In Jobstream	Last Modify	Last Access	Time- 15:31
ENVINVCE.MSX.TYM		JMSPENVF.MST.TYM	Line- 23	25Jun92	26Jun92
FMSEAGRE.MSX.TYM		JNEWAGRE.MST.TYM	Line- 21	25Jun92	26Jun92

Formation from Tynlabs Migration Analysis Jobstreams Using HP Environment Files				Print Date- 01Jul92	Page- 2
Environment File Name	Replaced By	Referenced In Jobstream	Last Modify	Last Access	Time- 15:31
LP2.HPENV.SYS	LP502.HPENV.SYS	JFHPFORM.FORMMIG.TYM	Line- 36	26Jun92	26Jun92
LP4.HPENV.SYS	LP504.HPENV.SYS	HPENV.PUS.DARRIUS	Line- 2	25Jun92	25Jun92
		JDATE16.BACKUP.SYS	Line- 29	30Jun92	30Jun92
		JDATE19.BACKUP.SYS	Line- 29	30Jun92	30Jun92
		JDATEVT.BACKUP.SYS	Line- 27	30Jun92	30Jun92
		JFULL7.BACKUP.SYS	Line- 29	25Jun92	30Jun92
		JFULL7V.BACKUP.SYS			

The section of the report titled *Jobstreams Using Environment Files* lists all the job streams that use custom environment files. You have presumably recompiled the environment files as discussed earlier in this document. You need to set the FMFORCELJ JCW in these job streams in the following cases:

- The job runs a print program that calls the Formation intrinsics (i.e., programs converted in the previous migration step).
- The job uses a printer that belongs to a device class containing multiple device types.
- You use a third-party product that creates "pseudo-devices." These pseudo-devices make it impossible for programs to automatically determine what printer type they will be using.

To set the JCW, see *Set JCW for Print Device*. The second section of the report is titled *Jobstreams Using HP Environment Files*. It lists the jobs which can take advantage of new environment files from HP. See "Switch to New HP Environment Files" on page 7-16.

### Set JCW for Print Device

Depending on what type of printer you're using, you may have to set the FMFORCELJ JCW before running a program that calls the Formation intrinsics. Programs that have been converted to use the Formation intrinsics instead of the PSP intrinsics generally contain either a PINITIALIZE or PINITDEVICE statement for an HP 2680 or 2688. But once it has been converted to use Formation, such a program can send output to any printer. Instead of modifying the program code, you set FMFORCELJ at run-time (or in the job stream that runs the program) to the appropriate value, determined by whether your program expects an HP 2680 or a 2688, and by the model of your new target printer.

Table 7-3: FMFORCELJ Values

Target Printer	FMFORCELJ Value		Notes
	Program Written for 2688	Program Written for 2680	
LaserJet 500+	500	10500	
LaserJet Plus, II	1	10001	
LaserJet IID, IIP, 2000, 2000D	2000	12000	
	2001	12001	For duplexing
LaserJet III, IIID, IIP, IIISi	3000	13000	
	3001	13001	For duplexing
LaserJet 4	4000	14000	
	4001	14001	For duplexing
HP 5000	5000	15000	
HP 2680	N/A	2680	These values are used when Formation cannot automatically recognize the 2680 or 2688, either because it belongs to a device class containing multiple device types or because a third-party product which creates "pseudo-devices" is in use.
HP 2688	2688	N/A	

For example, a job stream that runs a native mode print program written for the 2688 but now printing to a LaserJet IIISi might look like this:

```
!JOB JPRINT,MANAGER.ACCOUNT
!SETJCW FMFORCELJ=3000
!FILE OUTPUT;DEV=25;ENV=MYENV
!RUN MYPROG;XL="PSPLUXL.FORM.TYM"
!EOJ
```

### Switch to New HP Environment Files

If you have an HP 5000 and/or if you installed MPE XL 4.0, you have some new environment files on your system. These files replace the old LP and ELITE environment files supplied by Hewlett-Packard with the HP 2680. The fonts in the 5000 files are designed for 300 dpi, also the correct resolution for the LaserJet.

If you continue to use the old files with your 5000 or LaserJet, the 180 dpi fonts in them will be scaled to 300. Scaled fonts do not look as professional as ones with the correct resolution. However, using the new fonts may cause the text to move on the page.

The following table lists the old and new environment file names for quick reference. The files are generally found in the HPENV.SYS group.

**Table 7-4: Environment Files**

2680 Files	5000 Files
LP	LP50
LP2	LP502
LP4	LP504
ELITE	ELITE50
ELITE2	ELITE502
ELITE4	ELITE504

The *Job Streams Using Environment Files* report illustrated on page 7-15 also provides a list of old and new environment files. For each environment file listed in column 1, the new replacement file is listed in column 2. Next, all the job streams using this file are listed, along with the line in which the file is referenced.

To update a job stream, modify the line that references the environment file with EDITOR. For example:

```
:RUN EDITOR.PUB.SYS
/T VALTEST.PUB.ACCT
/CHANGE "LP4" TO "LP504" IN 2
/KEEP
/EXIT
```

### Print Forms on any Printer

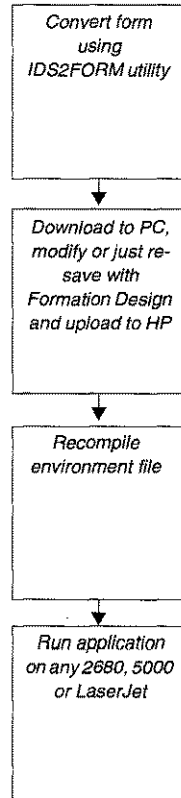
After converting your environment files, programs and job streams to use Formation, you can run most applications exactly as you did on your HP 2680 or HP 2688. Simply change your file equation to indicate the new device. For example:

```
:FILE outputfile;DEV=HP5000;ENV=envfile
:RUN application
```

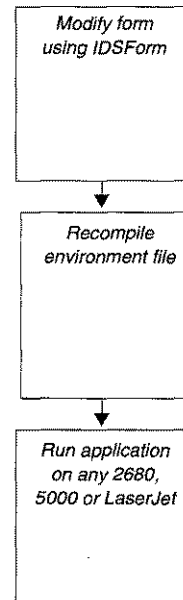
## Modifying Forms

When changes to a form layout are required, you can either use IDSForm or Formation Design. To use IDSForm, modify the form using the same procedures you always have, then recompile the environment file as described on page 7-9. Instructions for converting your form and modifying it with Formation Design are given in this section.

### Switch to Formation Design



### Or continue using IDSForm...



### Convert the Form

The IDS2FORM utility reads IDSForm files and creates corresponding Formation Design files. While IDSForm files can contain multiple forms, Formation files only contain one form. IDS2FORM provides commands that let you select an IDSForm file, list the forms it contains, set a scaling percentage, and convert those forms individually or all at once.

Table 7-5: IDIFORM Commands

Command	Purpose
CONVERT <i>formname</i> [, <i>filename</i> ]	Converts the specified form into a Formation file with the specified file name. If the file name is omitted, the name is created from the original form name. If the form name has eight characters or fewer, the file name is the same. If it is longer, the file name is composed of the first four characters and the last four characters of the original name.
CONVERT @[,@. <i>groupname</i> ]	Converts all forms in the form file into separate Formation form files. If the @ <i>groupname</i> parameter is specified, the Formation files are created in the indicated group. Individual file names are created as described above.
EXIT	Terminates execution of IDS2FORM.
FORMFILE <i>name</i>	Selects the IDSForm file that contains the forms to convert.
HELP	Lists all commands.
I2FLOG ON OFF	Enables and disables logging to the I2FLOG file, listing all the actions performed with the IDS2FORM program. This is useful for tracking old and new file names when converting a large number of forms,
LIST	Lists the names of the forms in the currently selected form file.
SCALE <i>nn.nn</i>	Used to scale forms that print edge to edge on the HP 2680 so they will fit in the printable area of the LaserJet. The scaling percentage must be between 50.00 and 100.00. Use this command before using the CONVERT command to convert forms; the percentage you set is in effect for all subsequent uses of CONVERT until you reset the scaling.  If you are changing from an 11" printable area to a 10.5" area, you need to use a scaling factor of at least 96%. You must also consider the relationship between the data font you were using on the old form and the data font you plan to use now. For example, if you were using a 10 cpi Courier, and will now be using an 11 cpi Courier, you divide 10 by 11 to get a scaling factor of 90.90%.

All commands can be abbreviated to a single letter.

**Note:** All converted forms must be downloaded, opened, saved with Formation Design, and uploaded back to the HP 3000 before they can be used. This is necessary even if no modifications to the layout are required. These steps are described further on in this section.

**Example**

The following example shows how IDS2FORM is used.

```
:ids2form.form.tym
IDS to Formation Converter 3.50 * (c) 1993 Tymlabs Corp. * Mon 31May93
IDS2FM >formfile ntiform
Form file NTIFORM.MIKEB.FORMDEV last modified MON, MAY 4, 1992.
IDS2FM >list
      Name                Last Modified                Width    Height
-----
NTICOQ                SUN, APR 5, 1992, 1:12 PM        8.5      11
NTICOQ2               WED, MAY 31, 1989, 1:04 PM       8.5      11
NTICOQ3               TUE, AUG 8, 1989, 12:26 PM       8        10.5

3 forms in file.
IDS2FM >convert @
Converting form "NTICOQ" to Formation file NTICOQ.
Converting form "NTICOQ2" to Formation file NTICOQ2.
Converting form "NTICOQ3" to Formation file NTICOQ3.

3 forms were converted.
IDS2FM >e
:END OF PROGRAM
```

**Replacing Fonts**

Fonts in your form are replaced with equivalent Formation fonts according to Table 7-6, "PCELL Fonts," on page 7-21. If you have modified any of your HP fonts with IDSCar, they are converted as if they were standard. You will probably have to modify portions of the form text where these custom fonts were used. Fortunately, because the Formation fonts have a richer character set, you may find that characters you had to add to your HP font can be typed using standard keystrokes. IDS2FORM takes care of symbol set conversion (e.g., converting ISO German to Formation Design fonts). If you:

- Use PCELL fonts not listed in Table 7-6, "PCELL Fonts," on page 7-21, they default to Univers.
- Own the Prestige Elite, Letter Gothic or Pica fonts for use with Formation, the substitution table can be modified.
- Need to add font or symbol set substitutions to the table, please contact Unison Software support for instructions.



Table 7-6: PCELL Fonts

PCELL File Name	Symbol Set	Printer Type	Formation Typeface	Style
COUR.CHARSETS.SYS	ASCII	2680	Courier	
COUR12.CHARSETS.SYS	ASCII	2680	Courier	
COURB88S.CHARSETS.SYS	ASCII	2688	Courier	Bold
COURB88X.CHARSETS.SYS	Roman-8 extension	2688	Courier	Bold
COURF.CHARSETS.SYS	ISO French	2680	Courier	
COURG.CHARSETS.SYS	ISO German	2680	Courier	
COURI88S.CHARSETS.SYS	ASCII	2688	Courier	Italic
COURI88X.CHARSETS.SYS	Roman-8 extension	2688	Courier	Italic
COURIER.CHARSETS.SYS 12	ASCII	2680	Courier	
COURIERF.CHARSETS.SYS	ISO French	2680	Courier	
COURIERG.CHARSETS.SYS	ISO German	2680	Courier	
COURIERN.CHARSETS.SYS	ISO Norwegian v1	2680	Courier	
COURIERS.CHARSETS.SYS	ISO Spanish	2680	Courier	
COURIERU.CHARSETS.SYS	ISO United Kingdom	2680	Courier	
COURIERW.CHARSETS.SYS	ISO Swedish	2680	Courier	
COURITAL.CHARSETS.SYS	ASCII	2680	Courier	
COURN.CHARSETS.SYS	ISO Norwegian v1	2680	Courier	
COURR88S.CHARSETS.SYS	ASCII	2688	Courier	
COURR88X.CHARSETS.SYS	Roman-8 extension	2688	Courier	
COURS.CHARSETS.SYS	ISO Spanish	2680	Courier	
COURU.CHARSETS.SYS	ISO United Kingdom	2680	Courier	
COURW.CHARSETS.SYS	ISO Swedish	2680	Courier	
GENAB88S.CHARSETS.SYS	ASCII	2688	Univers	Bold
GENAB88X.CHARSETS.SYS	Roman-8 extension	2688	Univers	Bold
GENAI88S.CHARSETS.SYS	ASCII	2688	Univers	Italic
GENAI88X.CHARSETS.SYS	Roman-8 extension	2688	Univers	Italic
GENAR88S.CHARSETS.SYS	ASCII	2688	Univers	
GENAR88X.CHARSETS.SYS	Roman-8 extension	2688	Univers	
GOTHB88S.CHARSETS.SYS	ASCII	2688	Courier	Bold
GOTHB88X.CHARSETS.SYS	Roman-8 extension	2688	Courier	Bold
GOTHI88S.CHARSETS.SYS	ASCII	2688	Courier	Italic

Table 7-6: PCELL Fonts

PCELL File Name	Symbol Set	Printer Type	Formation Typeface	Style
GOTHI88X.CHARSETS.SYS	Roman-8 extension	2688	Courier	Italic
GOTHR88S.CHARSETS.SYS	ASCII	2688	Courier	
GOTHR88X.CHARSETS.SYS	Roman-8 extension	2688	Courier	
HELV.CHARSETS.SYS	ASCII	2680	Univers	
HELVBOLD.CHARSETS.SYS	ASCII	2680	Univers	Bold
HELVITAL.CHARSETS.SYS	ASCII	2680	Univers	Italic
LRGCOURIER.CHARSETS.SYS	ASCII	2680	Courier	
LTITAL.CHARSETS.SYS	ASCII	2680	Courier	
PICA.CHARSETS.SYS	ASCII	2680	Courier	
PICAF.CHARSETS.SYS	ISO French	2680	Courier	
PICAG.CHARSETS.SYS	ISO German	2680	Courier	
PICAN.CHARSETS.SYS	ISO Norwegian v1	2680	Courier	
PICAR88S.CHARSETS.SYS	ASCII	2688	Courier	
PICAR88X.CHARSETS.SYS	Roman-8 extension	2688	Courier	
PICAS.CHARSETS.SYS	ISO Spanish	2680	Courier	
PICAU.CHARSETS.SYS	ISO United Kingdom	2680	Courier	
PICAW.CHARSETS.SYS	ISO Swedish	2680	Courier	
PRESB88S.CHARSETS.SYS	ASCII	2688	Courier	Bold
PRESB88X.CHARSETS.SYS	Roman-8 extension	2688	Courier	Bold
PRESI88S.CHARSETS.SYS	ASCII	2688	Courier	Italic
PRESI88X.CHARSETS.SYS	Roman-8 extension	2688	Courier	Italic
PRESR88S.CHARSETS.SYS	ASCII	2688	Courier	
PRESR88X.CHARSETS.SYS	Roman-8 extension	2688	Courier	
ROM.CHARSETS.SYS	ASCII	2680	CG Times	
ROMBOLD.CHARSETS.SYS	ASCII	2680	CG Times	Bold
ROMITAL.CHARSETS.SYS	ASCII	2680	CG Times	Italic
ROMPB88S.CHARSETS.SYS	ASCII	2688	CG Times	Bold
ROMPB88X.CHARSETS.SYS	Roman-8 extension	2688	CG Times	Bold
ROMPI88S.CHARSETS.SYS	ASCII	2688	CG Times	Italic
ROMPI88X.CHARSETS.SYS	Roman-8 extension	2688	CG Times	Italic
ROMPR88S.CHARSETS.SYS	ASCII	2688	CG Times	

Table 7-6: PCELL Fonts

PCELL File Name	Symbol Set	Printer Type	Formation Typeface	Style
ROMPR88X.CHARSETS.SYS	Roman-8 extension	2688	CG Times	
SCRIP88S.CHARSETS.SYS	ASCII	2688	Courier	
SCRIP88X.CHARSETS.SYS	Roman-8 extension	2688	Courier	
SCRIPT.CHARSETS.SYS	ASCII	2680	Courier	
TIMEB88S.CHARSETS.SYS	ASCII	2688	CG Times	Bold
TIMEB88X.CHARSETS.SYS	Roman-8 extension	2688	CG Times	Bold
TIMEI88S.CHARSETS.SYS	ASCII	2688	CG Times	Italic
TIMEI88X.CHARSETS.SYS	Roman-8 extension	2688	CG Times	Italic
TIMER88S.CHARSETS.SYS	ASCII	2688	CG Times	
TIMER88X.CHARSETS.SYS	Roman-8 extension	2688	CG Times	

### Rotated Text

If your form uses rotated text, IDS2FORM converts that text to standard horizontal orientation. Here are your choices:

1. Change the layout using Formation Design.
2. Make changes with IDSForm, recompile with Formation, and print on any printer.
3. Scan the rotated text into a TIFF file, and add it as a graphic element to print on any printer.
4. To print on the HP 2680, HP 2688, LaserJet III, and LaserJet 4 only, you can use IDSForm to remove rotated text from your form, then set up your application and your environment file to print the rotated text as part of the data. Refer to Quick Start Printing Rotated Text for more information.

### Download the Form to a PC

#### Install A Terminal Emulator

To download your form to a PC, use a terminal emulation program of your choice.

#### Modify with Formation Design

To make changes with Formation Design, follow these basic steps:

1. Turn on your PC, activate Windows and run Formation Design.
2. Choose **Open** from the File menu to open the downloaded form.
3. Use Formation Design's tools and menu items to modify your form. Refer to the *Formation Design User Guide* for more information.
4. When finished, choose **Save** and **Exit** from the **File** menu.

## Upload the Form to the HP 3000

To upload your form to a PC, use a terminal emulation program of your choice.

## Recompile with Formation

To modify your environment file to use the new form, do the following:

1. Log on to the HP 3000 and run the Environment Manager.

```
:RUN ENVMGR.FORM.TYM
```

The **Environment File Menu** is displayed.

2. Enter the name of your environment file and press **e**. The **Main Menu** is displayed.
3. Type **L** in the selection entry box and press **e** to access the **Logical Page Control** screen.
4. Use the **6** key to access the logical page to which the modified form will be assigned. Press **3** to go to the **Logical Page Form Control** screen.

Because IFS allows multiple forms in a form file, you will see both a form name and a form file name. Enter the modified form name in the **Form File 1** field, and remove any form name from the **Form 1** field.

Your IDS form name is here,

and your form file name is here.

Move the form name to the Form File field, and leave the Form 1 field blank.

5. Make any other changes you need to make. Press **e** to process the data.
6. Press **8** to return to the **Main Menu**.
7. Type **X** in the selection entry box and press **e** to compile the environment file.

## Print Forms on any Printer

You can now run your application on any printer. Set the FMFORCELJ JCW as described on page 7-15, and then enter the following commands.

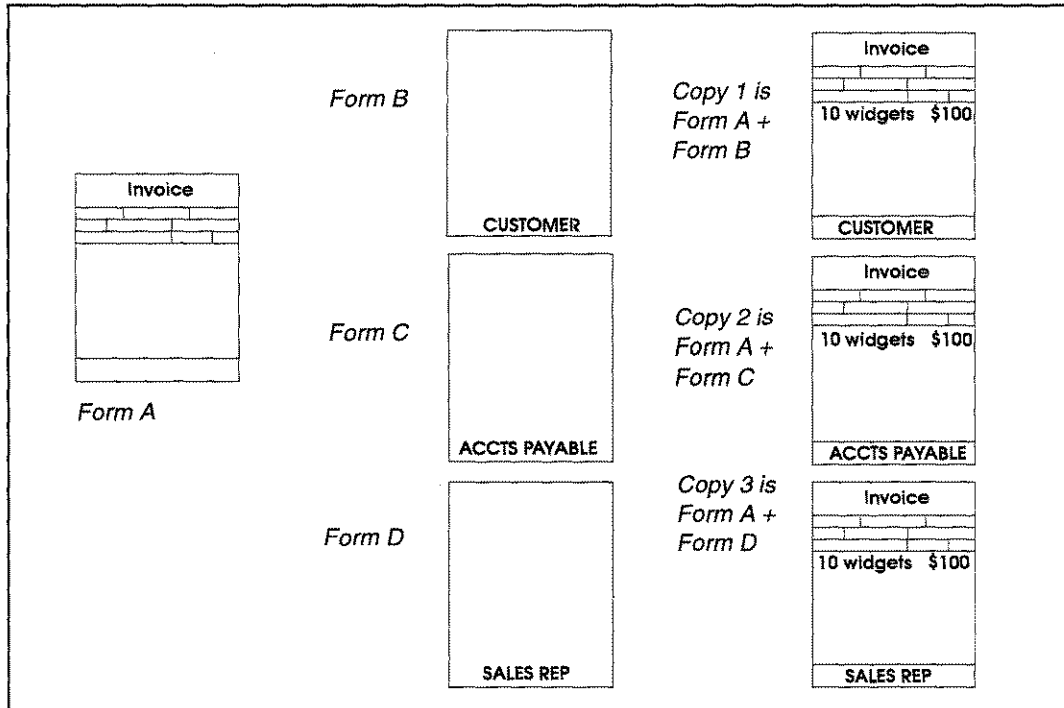
```
:FILE outputfile;DEV=device;ENV=envfile
```

```
:RUN application
```

## Multi-Part Forms on HP 2680 and HP 2688 Printers

Formation/2680 users can print multi-part forms on HP 2680 and 2688 printers, simulating the production of carbon copies in the electronic forms environment. For example, you may need to print three copies of an invoice, one for the customer, one for your accounts payable department, and one for the sales representative. Each copy is identical, except for a label area indicating who receives this copy.

To do this, create a basic form containing the elements that go on all copies (Form A below). Then create a second form for each copy, containing information specific to that copy (Forms B, C, and D below). In production, each set of data is printed on all copies of the form.

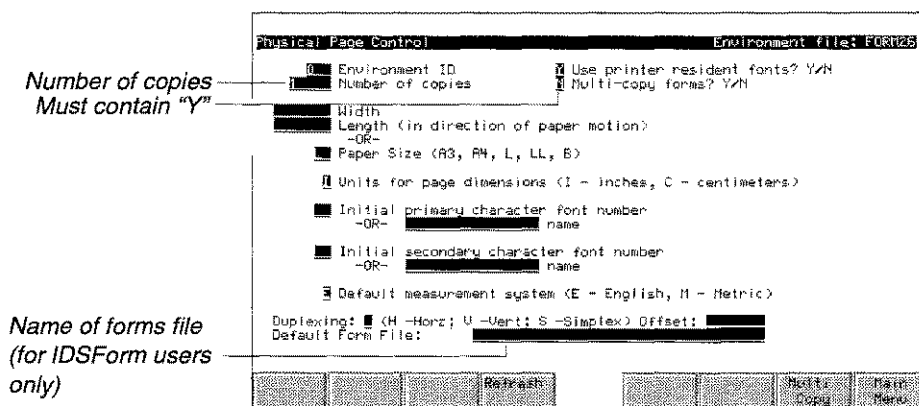


HP2680 and 2688 printers have control facilities built in to handle the production of up to eight versions of a multi-part form. The Formation Environment Manager's **Multi-Copy Forms** screen lets you set up the table these printers use to perform this function.

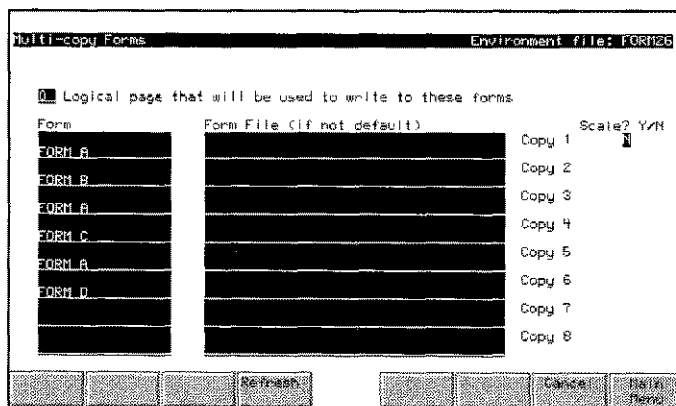
### Procedure

Create a multi-part form for use with your HP 2680/88 printer as follows:

1. Run the Environment Manager.  
:RUN ENVMGR.FORM.TYM  
The **Environment File Menu** is displayed.
2. Type the environment file name and press **e**.
3. Type **P**, enter your printer type in the **HP 2680 Model** field, and press **e** at the **Main Menu** to access the **Physical Page Control** screen.



4. Type Y in the **Multi-copy forms?** field.
5. Type the desired number of copies in the **Number of copies** field.
6. If using an IDS form, enter the name of the forms file in the **Default Form File** field.
7. Press e to process the data on this screen.
8. Press 7 to display the **Multi-Copy Forms** screen.



9. For each copy to be printed, specify up to two forms. The first of the two is usually the main form containing the elements which are identical from one copy to the next. The second form contains the variable portion of the form. Formation overlays the second form on the first, centering each form on the associated logical page. To ensure proper placement, make both forms the same size and shape.

The other fields on the menu allow you to specify a logical page to use in writing data to the form, and whether the forms are to be scaled.

## Field Descriptions

Each field on the **Multi-Copy Forms** screen is described below.

Logical page that will be used to write to these forms	<p>This optional field specifies the ID number of the logical page used to write data to the forms. The logical page defines the position where data is printed, and also controls the orientation -- the forms and printed data are rotated to match the orientation of the logical page. If you specify scaling, forms are scaled to fit the logical page.</p> <p>If you do not specify a logical page, the logical page with the lowest ID number in the environment is used. If no logical pages have been defined, the laser printer's default logical page is used.</p>
Form	<p>This field is used only if the form was created with Hewlett-Packard's IDForm software. It is used to specify the name of the form within the file indicated in the next field. Since each form created with Formation Design resides in a separate file, this field is not used if your form was created with Formation Design. You specify two forms per copy: a base form and an overlay.</p>
Form File	<p>This field specifies the name of the form file, a fully qualified MPE file name including lockword, if any. If no file is specified, the default form file specified on the <b>Physical Page Control</b> screen is assumed. The form specified here is compiled into the environment file.</p>
Scale?	<p>To scale your forms to fit the logical page specified by the ID number in the <b>Logical page</b> field, type a Y here. All forms are then rotated and scaled to fit that logical page, and all printing is restricted to the printing area it defines. To use scaling, you must specify a logical page.</p> <p>Lines and boxes are scaled so that all elements fit within the logical page defined. Fonts are scaled by selecting from the sizes available in the font defined in your form. If the smallest size is too large for the reduced form, type will extend beyond the boundaries of the logical page.</p>

## Beyond Migration

Smoothing migration to PCL printers is just one small part of what Formation does. Here are some ideas about other things you can do with the product.

### **A Programmer's Toolkit for Complex Forms**

Formation comes with a programmer's library similar to the PSP intrinsics. These intrinsics can be used to control a laser printer without writing PCL escape codes. Fonts, forms and page layouts can be selected based on variables encountered at run time (for example, printing a particular salesperson's signature on a quotation.) The routines are device-independent, so future printer architectures can be accommodated with no need to migrate applications.

In combination with Formation Design, the programmer's library provides an intelligent forms capability. To create an intelligent form, a special design tool is used to draw and name each data field. The printing application that writes to this form calls a Formation procedure to write to fields by name, regardless of their location on the page. This separates form layout from application printing specifications, so that modifications to the form do not require program changes.

### **Check Printing**

Formation can print your company's checks in the format required by your bank, including the bank routing information (micr encoding). Printing checks on blank safety paper with special magnetic toner eliminates many of the hassles of using preprinted checks. You no longer need to ensure that the number of the printed check you mount is synchronized with the number maintained by your accounting software. You can also print checks in sequence at multiple sites. You can switch bank accounts without switching paper stocks.

Setting up electronic check printing with Formation is simple, and we can help you get all the supplies you need.

### **Post-processing**

HP 2680 users are accustomed to having their applications send output directly to the printer. Compared to sending output to a disc file and printing it later with a post-processor, this approach conserves both disc space and CPU resource. Formation supports this style of printing; the user simply adds the environment file name to the printer file equation in the job stream.

Sometimes it is more convenient to send data to an output file on disc, either to transfer to a dedicated print server, or to take advantage of post-processor formatting options missing from the printing application. Formation's post-processor, LPRINT, can create multi-copy forms (simulating carbon copies), forms duplexed with other form layouts or with standard terms and conditions, and reduced forms printed 2-up or 4-up on one or both sides of the paper.







## Using Formation with ASK

This chapter is a Quick Start guide for ASK users who want to use Formation to print ASK forms. Unison Software provides a set of sample forms for ASK users that are just like the pre-printed forms used now. These include a purchase order in landscape and portrait formats, an invoice, and a statement.

This chapter shows you how to customize these forms for your company, and print them with your own ASK data.

*ASK purchase order printed by Formation.*

Before you begin, install Formation. For demo purposes, you can stop after you restore the tape and stream JSETUP. You don't need to copy the intrinsics into the system SL or install the file equate facility.

### Topics

This chapter describes the following ASK forms printing with Formation topics:

- Print Sample Forms with Your Data
- Customize The Forms
- Modify the Form
- Compile and Print
- Production Printing
- Multi-Part Forms

## Print Sample Forms with Your Data

Before you modify the forms with your own logo and address, print the sample forms with your data. Decide whether you want to print purchase orders, statements, or invoices. The following table lists the sample form types and their corresponding environment files:

Table 8-16: Available Fonts

Form	Orientation	Environment File
Invoice	Landscape	ASKINVLE.SAMPLES.TYM
Purchase order	Portrait	ASKPOPE.SAMPLES.TYM
Purchase order (Two copies)	Landscape	ASKPOLE.SAMPLES.TYM
Statement	Portrait	ASKSMTPE.SAMPLES.TYM

Use your standard ASK command (such as UT912) to print a few pages of data for the form type you've selected. Specify that you want the output to go to a disk file. Record the name of the disk file.

Before you print, release the data file you just created, and then log on in the TYM account. (Once you go into production, you can eliminate this step by moving the Formation printing program into the account where you generate ASK data.)

```
:RELEASE datafile
:HELLO MANAGER.TYM, SAMPLES
```

Now follow printing instructions for your operating system. When printing is complete, you may want to purge the data file.

**MPE XL  
Users**

To print statements or purchase orders, enter the following command:

```
:ASKPRINT device datafile.grp.acct envfile
```

Where:

*device* A LaserJet or HP 5000 device number or class.

*datafile.grp.acct* A fully qualified data file name.

*envfile* One of the statement or purchase order environment files listed in the table on page 8-2.

To print invoices, use this form of the command:

```
:ASKPRINT device datafile.grp.acct ASKINVLE L-2
```

Where:

L-2 Moves the data on the form two character positions to the left. This is usually required for proper positioning on the page. If L-2 doesn't get the data exactly where it needs to be on your form (versions of ASK do vary slightly), a higher value moves it further to the left and a lower value moves it back to the right.

**MPE V  
Users**

To print statements or purchase orders, enter the following command.

```
:FILE LPIN=datafile.grp.acct
:FILE LPLIST;DEV=device;ENV=envfile
:RUN LPRINT.FORM.TYM;INFO="M";LIB=G
```

Where:

*datafile.grp.acct* A fully qualified data file name.

*device* A LaserJet or HP 5000 device number or class.

*envfile* One of the statement or purchase order environment files listed in the table on page 8-2.

To print invoices, use this form of the command.

```
:FILE LPIN=datafile.grp.acct
:FILE LPLIST;DEV=device;ENV=ASKINVLE
:RUN LPRINT.FORM.TYM;INFO="L-2";LIB=G
```

Where:

L-2 Moves the data on the form two character positions to the left. This is usually required for proper positioning. If L-2 doesn't get the data positioned correctly (versions of ASK do vary slightly), a higher value moves it further to the left and a lower value moves it back to the right.

## Customize The Forms

To add your logo and address to the forms, use Formation Design, a Windows-based WYSIWYG drawing program. Formation Design requires a system with:

- At least the capabilities of an IBM PC/AT
- At least 2 megabytes of memory available
- Microsoft® Windows™ version 3.0 or later

Refer to the *Formation Design User Guide* for more information.

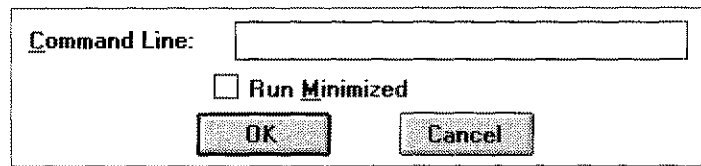
### Installation

Before installing the software, write protect the Formation Design diskettes, and make backup copies. Store the original diskettes in a safe place. Use the backup diskettes to install the software.

Install Formation Design as follows:

1. Turn on your computer.
2. Insert the **Formation Design Setup** diskette in drive A (or any available drive).
3. Load Windows, and access the **Windows Program Manager**.

4. Select the **Run** command from the **File** menu. The **Run** dialog box is displayed.



5. Enter the following in the **Command Line** entry box.

A:\INSTALL

If you used a drive other than A, replace A: with the appropriate drive designation. Click OK.

6. Follow the instructions on your screen. The installation program determines which diskettes must be installed based on your system configuration. All diskettes may not be required.

### Install the LaserJet III Printer Driver

To guarantee WYSIWYG display, install an HP LaserJet III printer driver under Windows. This must be done even if a LaserJet III is not attached to your PC. Formation Design uses the printer driver to determine the amount of space text will require when printed.

To determine if the LaserJet III printer driver is installed:

1. Open the **Windows Control Panel**.
2. Double-click the **Printers** icon.
3. If the HP LaserJet III is listed in the **Installed Printers** scroll list, it is currently installed. If it is not listed, you must install it.

To install the LaserJet III printer driver:

1. Click the **Add>>** button in the **Printers** dialog box.
2. Select the HP LaserJet III from the **List of Printers** scroll list. Click **Install**.
3. You are then prompted to insert one of your **Windows Setup** diskettes. Insert the requested diskette to install the printer driver.

It does not matter which port you assign to the printer as it will not actually be used for printing.

If the LaserJet III driver is not installed, you will receive a message telling you that it cannot locate the file HPPCL5A when you run Formation Design.

Once Formation Design and the LaserJet driver are installed, exit and restart Windows to ensure that the new screen fonts installed are made available.

## Modify the Form

The following section describes the procedure for modifying the landscape-style purchase order. This procedure can be used to modify any of the sample forms.

1. Run Formation Design by double-clicking the icon.



Designer

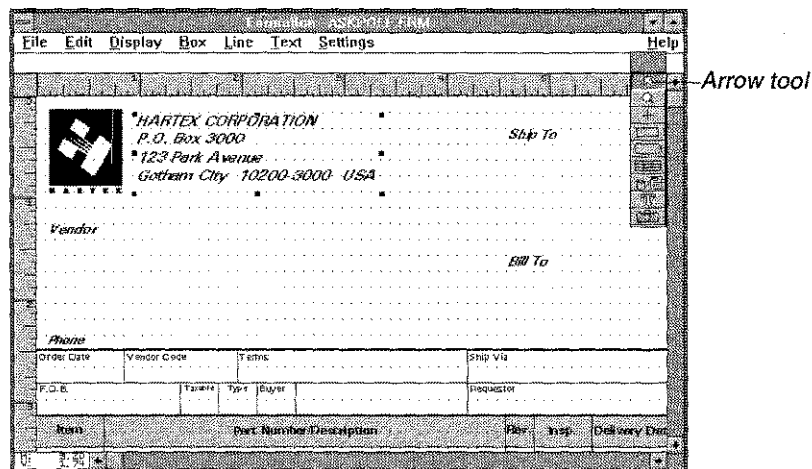
2. Choose **Open** from the **File** menu. The **Formation Form Open** dialog box appears.
3. Click on ASKPOLF in the **File Name** scroll list. Click **OK** to display the landscape purchase order form.

The following table lists all the sample form file names. To display any of the other forms, select the appropriate name from the **File Name** scroll list. Note that the form files are in C:\FORM.

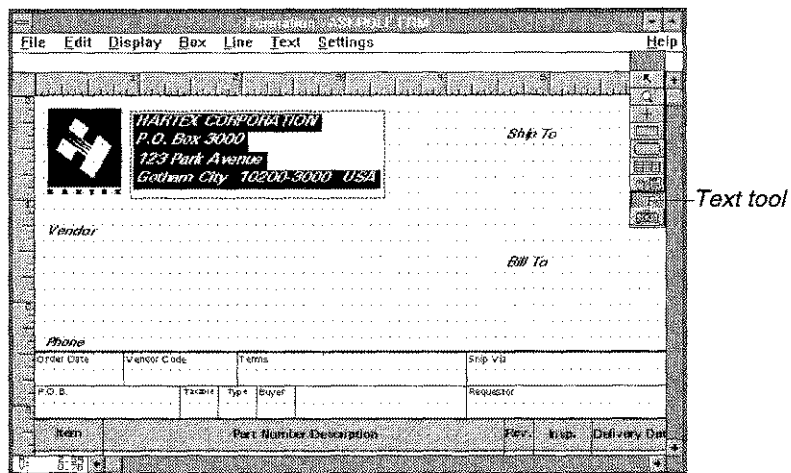
Table 8-17: Sample ASK Form Files

Form	Form File
Invoice	ASKINVLF
Portrait purchase order	ASKPOPF
Landscape purchase order	ASKPOLF
Statement	ASKSMPF

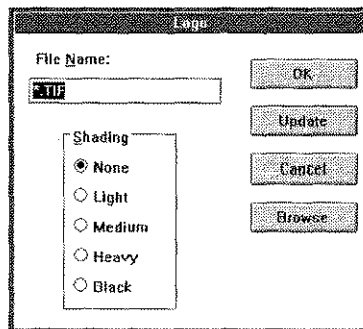
4. Select the arrow tool. Click on the address text block next to the logo. Square black handles appear around the text box.



5. Select the text tool (the "T"). Move the cursor to the beginning of the text. Click the mouse button, and drag over the existing address text so it is highlighted.



6. Type in your company name and address. Note that the existing text is automatically deleted as you begin typing.
7. To replace the existing logo with a TIFF file containing your logo, choose the arrow tool. Double-click the existing logo to display the **Logo** dialog box.



8. Enter your logo file name in the **File Name** field and click the **Update** button. Your logo replaces the existing logo.
9. Select **Save** from the **File** menu to save your form. If you do not want to overwrite the sample form, select **Save As** from the **File** menu and supply a new form name.
10. Select **Exit** from the **File** menu.

**Note:** If you don't have your logo in a TIFF file, send it to Unison Software, and we will scan it for you. If your company has a scanner, scan your logo in-house. Keep these tips in mind:

- Scan the logo at the size you want to use so no scaling is required.
- Do a line art type scan, not a gray scale scan
- Resolution should be 300 DPI
- Crop the logo to include as little white area as possible
- Clean up any diagonals using your scanning software



## Compile and Print

Depending on which form you've been working on, re-compile the appropriate environment file as described below. The available environment files are listed in the following table.

**Table 8-18: Available ASK Environment Files**

Form	Orientation	Form File	Environment File
Invoice	Landscape	ASKINVLF	ASKINVLE.SAMPLES.TYM
Purchase order	Portrait	ASKPOPF	ASKPOPE.SAMPLES.TYM
Purchase order (Two copies)	Landscape	ASKPOLF	ASKPOLE.SAMPLES.TYM
Statement	Portrait	ASKSMPF	ASKSMTPE.SAMPLES.TYM

If you've saved and uploaded your form under a different name to avoid overwriting the original file, you should now save the original file with a new name. You should then save your modified file to one of the form file names listed in the chart above. This is the form name the environment file is set up to use.

### MPE XL Users

MPE XL users can compile and print their new forms using the `ASKPRINT` command. This command file contains a line that automatically re-compiles the environment file.

To print statements or purchase orders, enter the following command:

```
:ASKPRINT device datafile.grp.acct envfile
```

Where:

*device* A LaserJet or HP 5000 device number or class.

*datafile.grp.acct* A fully qualified data file name.

*envfile* One of the statement or purchase order environment files listed in the table above.

To print invoices, use the following form of the command.

```
:ASKPRINT device datafile.grp.acct ASKINVLE L-2
```

Where:

*L-2* Moves the data on the form two character positions to the left. This is usually required for proper positioning on the page. If *L-2* doesn't get the data exactly where it needs to be on your form (versions of ASK do vary slightly), a higher value moves it further to the left and a lower value moves it back to the right.

To print the form without data, substitute `$NULL` for the *datafile.grp.acct* parameter in either command above.

**MPE V  
Users**

To print statements or purchase orders, enter following commands.

```
:RUN ENVMGR.FORM.TYM;INFO="envfile"  
:FILE LPIN=datafile.grp.acct  
:FILE LPLIST;DEV=device;ENV=envfile  
:RUN LPRINT.FORM.TYM;INFO="M";LIB=G
```

Where:

*device*                    A LaserJet or HP 5000 device number or class.  
*datafile.grp.acct*      A fully qualified data file name.  
*envfile*                   One of the statement or purchase order environment files listed  
in the table on the previous page.

To print invoices, enter the following commands.

```
:RUN ENVMGR.FORM.TYM;INFO="ASKINVLE"  
:FILE LPIN=datafile.grp.acct  
:FILE LPLIST;DEV=device;ENV=ASKINVLE  
:RUN LPRINT.FORM.TYM;INFO="L-2";LIB=G
```

Where:

L-2                        Moves the data on the form two character positions to the left.  
This is usually required for proper positioning on the page. If  
L-2 doesn't get the data exactly where it needs to be on your  
form (versions of ASK do vary slightly), a higher value moves it  
further to the left and a lower value moves it back to the right.

To print the form without data, substitute \$NULL for the data file parameter on the file  
equation for LPIN.

## Production Printing

If you are running under MPE XL, you have been using the ASKPRINT command file for printing. This file can be used as a model for setting up custom command files for MPE XL printing, or UDCs for MPE V.

The following is a listing of the file:

<p><i>If you add appropriate defaults on the PARM line (line 1), you can take out these lines, used to prompt for missing parameters. The defaults specified in line 1 will be substituted automatically.</i></p> <p><i>Take this out once you go into production; no need to recompile the environment file every time you print.</i></p>	<pre> PARM PRINTER=" ", DATAFILE=" ", ENVFILE=" ", ADDINFO=" "  IF "!PRINTER" = " "   ECHO A command line should be entered as:   ECHO   ECHO   COMMAND Ldev Datafile Envfile [ExtraLPRINTinfo]   ECHO ELSE   IF "!DATAFILE" = " "     ECHO A command line should be entered as:     ECHO     ECHO   COMMAND Ldev Datafile Envfile [ExtraLPRINTinfo]   ELSE     IF "!ENVFILE" = " "       ECHO A command line should be entered as:       ECHO   COMMAND Ldev Datafile Envfile [ExtraLPRINTinfo]     ELSE       RUN ENVMGR.FORM.TYM;INFO="!ENVFILE"       FILE LPIN=!DATAFILE       FILE LPLIST;DEV=!PRINTER;ENV=!ENVFILE       RUN LPRINT.FORM.TYM;LIB=G;INFO="M !ADDINFO"     ENDIF   ENDIF ENDIF ENDIF </pre> <p><i>Use the INFO parms required for your margin positioning, duplexing and multi-copy requirements.</i></p>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

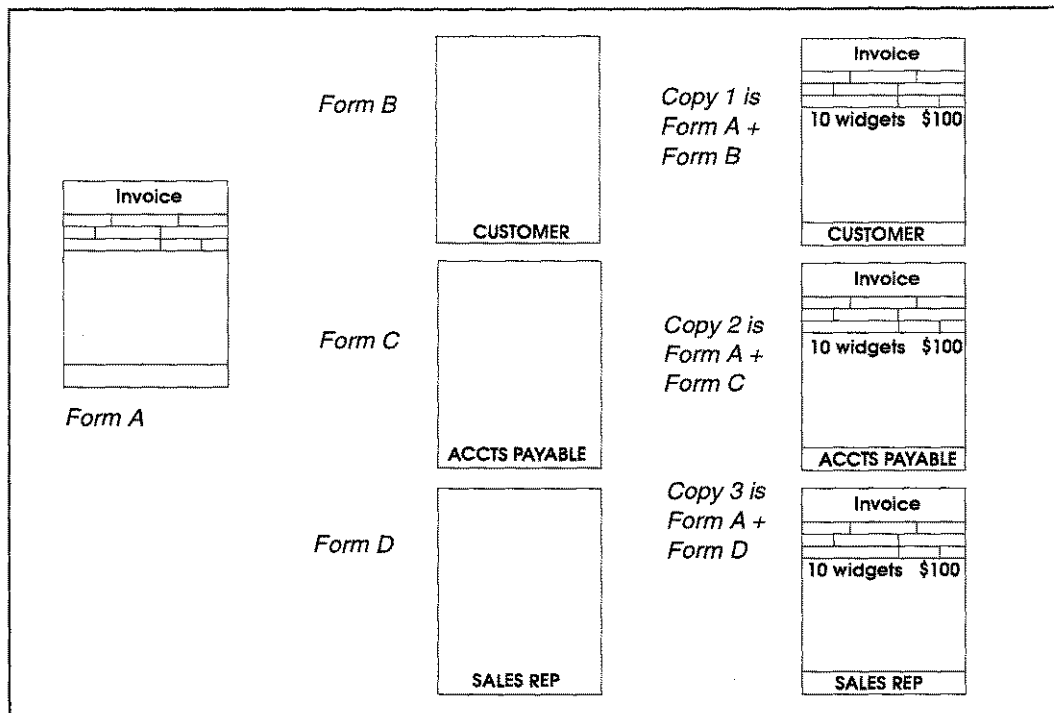
## Install Intrinsic in System SL

Once you purchase Formation and go into production, you may want to complete the installation procedure. This includes installing Formation's file equate facility and copying the intrinsic into the system SL. When you install the intrinsic in the system SL, remove the LIB=G parameter from the :RUN command that executes LPRINT in the ASKPRINT file and any other command files, job streams, or UDCs set up during your demo.

## Multi-Part Forms

Formation's multi-part forms capability simulates the function of pre-printed forms with carbon copies. For example, you may need to print three copies of each invoice: one for the customer, one for your accounts payable department, and one for the sales representative. Each copy is identical except for a label indicating who receives the current copy.

To do this, you usually have a basic form containing the elements that go on all parts (Form A below). This is usually one of the forms supplied by Unison Software. Then you create a second form for each part, containing information specific to that part (Forms B, C, and D). In production, each set of data is printed on all parts.



The steps involved in turning one of the ASK forms into a multi-part form are:

1. Use Formation Design to create a form containing the label information for each part to be printed. Upload these forms to the HP 3000.
2. Use the Environment Manager or Formation UDC's to set up a series of logical pages, each of which contains the main form and one of the labels.
3. Print with LPRINT using the INFO=M parameter. Each set of data is automatically printed on each part of the form set.

The purchase order environment file ASKPOLE is already set up to print two parts of the form set. In the following exercise, a third part of the purchase order form set is created.

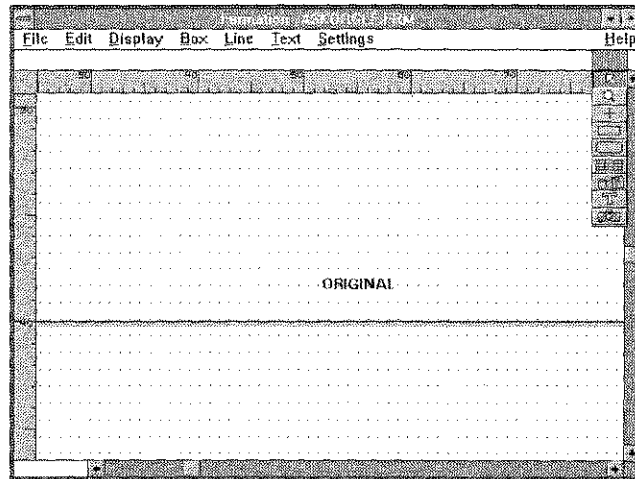
### Create Additional Forms

The Formation Design files ASKORGLF and ASKRMTLF each contain one line of text: ORIGINAL and REMITTANCE, respectively. These lines are positioned to print in the desired area on the landscape-oriented purchase order form in the file ASKPOLF. To create an additional part with the label FILE, follow the steps below.

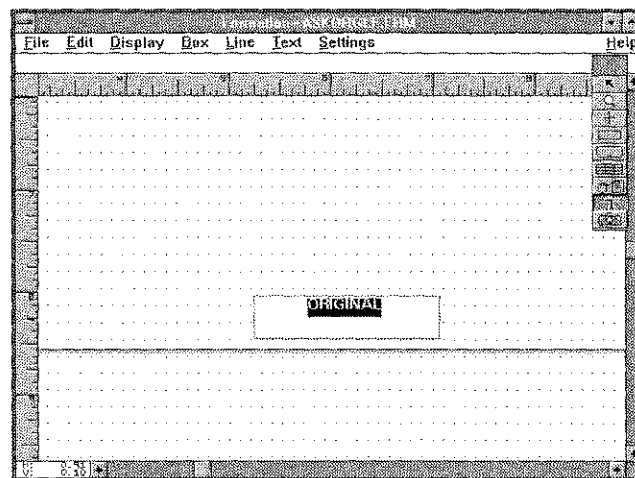
1. Run Formation Design by double-clicking the icon.
2. Select **Open** from the **File** menu, and select either ASKORGLF or ASKRMTLF from the **File Name** scroll list. Click OK to display the form.

(These files contain a line of text positioned in the appropriate place for labelling landscape forms. If you want to create labels for portrait forms, use the file ASKORGPF as your template.)

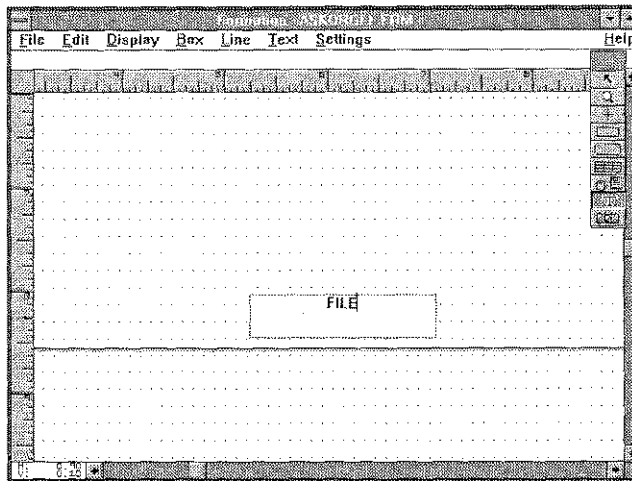
3. Once the form is open on your screen, scroll down and to the right so the text is displayed.



4. Select the arrow tool, and click the text block so handles appear.
5. Select the text tool (the "T"). Place the cursor before the O in ORIGINAL and drag over the existing text to highlight the word.



6. Type in the word FILE. The original text is automatically deleted as you begin typing.



7. Choose **Save As** from the **File** menu. Enter the name ASKFILLF in the **File Name** entry box and click OK.
8. Upload the form.

### Modify the Environment File

The environment file ASKPOLE contains descriptions of two logical pages, 0 and 1. These print the ORIGINAL and REMITTANCE copies. The next step is to define a new logical page composed of the base form ASKORGLF and the new form ASKFILLF. This can be accomplished using Formation UDCs or with the Environment Manager.

#### Defining a Logical Page Using UDCs

Enter the following commands to add the new logical page and then recompile the environment:

```
:ADDPAGE ASKPOLE, ASKPOLF, ASKFILLF, 2
:COMPILEENV ASKPOLE
```

Note if you named your form files with names other than ASKPOLF and ASKFILLF, you should insert the proper names when running the UDC.

#### Defining a Logical Page with the Environment Manager

To define a logical page using the Environment Manager:

1. Log on to the HP 3000 and run the Environment Manager.

```
:RUN ENVMGR.FORM.TYM
```

The **Environment File Menu** is displayed.

2. Type ASKPOLE.SAMPLES.TYM and press **e**. The **Main Menu** is displayed.
3. Type **L** in the selection entry box to access the **Logical Page Control** screen. The data for logical page 0 is displayed.

- Press **6** twice to go to logical page 2. Set up this logical page exactly like the others in the file. The data you need to enter is indicated in the illustration below.

For portrait forms, enter the values listed in the table below.

Field	Value (landscape forms)	Value (portrait forms)
Orientation	L	P
Width	10.5	8
Height	8	10.5
Overriding line spacing	.1569	.1591

- Press **e** to send the data, and then press **3** to go to the **Logical Page Form Control** screen.
- Enter ASKPOLF in the **Form File 1** field, and ASKFILLF in the **Form File 2** field. Press **e** to send the data.

7. Press 8 to return to the **Main Menu**.
8. Type X in the selection entry box and press e to compile the environment file.
9. Enter E in the selection entry box and press e to exit the Environment Manager.

### **Print Multi-Part Forms**

To print multi-part forms on an MPE XL CPU, enter your LaserJet/HP 5000 device number or class, fully qualified data file name, and environment file name with this command:

```
:ASKPRINT device datafile.grp.acct ASKPOLE
```

To print multi-part forms on an MPE V CPU, use these commands:

```
:FILE LPIN=datafile.grp.acct  
:FILE LPLIST;DEV=device;ENV=ASKPOLE  
:RUN LPRINT.FORM.TYM;INFO="M";LIB=G
```

One page of data is printed on each logical page in the environment file. The logical pages are printed in numerical order.







## A

# Troubleshooting

Normally, once Formation is installed on your system, and all application programs have been converted, its operation is transparent. This section contains hints and information that you can use to resolve any problems which do arise. If, after applying the remedies in this section, you are unable to resolve a problem, contact your Unison-Tym labs support representative for further assistance.

## Error Message 20

If the Environment Manager indicates that an environment file requires less than 395kb, but the printer displays error 20 (memory overflow) when you try to print, there may be several causes:

- The environment requires more dynamic memory than is available.
- The document being printed contains graphics that exceed the amount of memory available.
- The document being printed contains graphics which are first stored, then printed. Printing a stored picture takes transient memory equal to the amount of static memory required to store the picture.

There are several things you can do to avoid this situation:

- Compile the environment for a more capable printer.
- Change the application to "flash" pictures instead of storing them. Reference PFLASHRASTER in chapter 6 for more information.
- Add more memory to the LaserJet (if you have a printer which permits **memory** expansion).

## Expanding the Environment File

Depending on the original size of the environment file, and the amount of PCL created during the compilation process, the Environment Manager may run out of room in the file. If this happens, the Environment Manager displays an error message. For a batch compile, the Environment Manager terminates with an error. When this happens, you need only expand the file and retry the compilation. To do this, follow these steps:

1. Determine the size of the current environment file.

```
:listf MYENV, 2
ACCOUNT= LTDEV      GROUP= DOC
FILENAME  CODE      -----LOGICAL RECORD-----
                SIZE  TYP      EOF      LIMIT
MYENV     PENV     512W    FB      2155    2155
```

2. Enter a file equation to control the expansion. Use the LIMIT value from above, and add 100 records or half the current value, whichever is greater.

```
:file newenv;disc=computed value
```

3. Copy the old file to the new file

```
:FCOPY FROM=MYENV;TO=*NEWENV;NEW  
HP32122A.03.22 FILE COPIER (C) HEWLETT-PACKARD CO. 1986  
2155 RECORDS PROCESSED *** 0 ERRORS  
END OF SUBSYSTEM  
:
```

You can now restart the compilation process using NEWENV, or renaming it to MYENV. In rare cases, you may have to perform this expansion twice. Do not start by expanding as the environment file may have unused space even though its EOF is high.

## Performance and Reduction Printing (Error 21)

Using reduction printing, you may be trying to place as many as 15,000 characters on a single page. This operation is subject to two performance limitations: the speed with which the characters can be sent to the printer and the speed with which the printer can put these on the page.

At the recommended line speed of 19,200 baud, approximately 1920 characters are received by the printer every second. The MPE spooler may stop sending for short periods of time as more data is read from your spooling disc, so the net rate is about 1850 - 1900 characters per second. This means that to fill a 4-up page, containing 15,000 characters will take about 8 seconds. This is slightly less than the rate needed to keep an 8 page/minute printer running, and much less than the rate needed to keep a 20 page/minute printer going. If you select a line speed lower than 19,200 baud, your print rate will be proportionately reduced.

Once the characters are in the printer's memory, they must be printed. This requires selecting the proper dot patterns and sending them to the laser at the right time. If the processor in the printer ever gets behind the laser, the printer aborts the remainder of the page and displays Error 21. This means that you have tried to put too much on a single page. Very dense printouts such as MPE memory dumps will usually cause this problem. Normal data processing output will rarely cause it. Duplex printing on the LaserJet 2000 does not affect the number of characters the printer can place on the page.

## Formation Error Messages

Error Type	Meaning
Printing is garbled.	If you are using the JCW method of selecting the output device, make sure that the <b>FMFORCELJ</b> JCW is set to the proper value. For more information, see Chapter 6, <i>Intrinsics</i> . If you are using a demo version of Formation, the demo may have expired.
Print is in wrong position on page, or it is in the wrong font.	Make sure that you are calling the proper (Formation) intrinsic whenever there is a proper substitute for the standard PSP intrinsic.

## Printer Problems

Error Message	Meaning
Printer shows Error 20.	If the problem occurs during an environment download, there are too many forms and/or fonts in the environment. Remove unused fonts and forms from the environment, or add additional memory to the printer and attempt to reprint.
Printer shows Error 21.	A form is too complex to be printed (the processor in the printer can't keep up with the laser). Revise the form if possible to eliminate shading from printed areas on the form.  Compile your environment file with the Environment Manager if you have not already done so.
Printer shows Error 22.	The baud rates of the printer and host do not match.  A communication problem has occurred between the host and the LaserJet such that the LaserJet is unable to send data back to the host. Check cabling and the host interface for proper insertion.  The host interface is misconfigured and is not accepting handshake signals from the LaserJet. This will not normally occur with a direct connection to the HP 3000, but may occur if the LaserJet is connected via a statistical multiplexor or other device.
Printer shows Error 40.	The baud rates of the printer and host do not match.  The cable between the printer and host is loose or defective.

## Formation Error Codes

Error Messages	Meaning	Action
101 Internal error: Expected to write record type <b>nnn</b> , not <b>nnn</b> . (PSPERR 101)	Probably a bad environment file or possibly an internal software error, not under user control.	Write down error message and contact system manager.
102 Internal error: (PSPERR 102) Assertion <b>nnn</b> failed in procedure <b>xxx</b> .	Internal software error, not under user control.	Contact system manager.
104 Internal error: bad orientation request: <b>nnn</b> . (PSPERR 104)	Internal software error, not under user control.	Contact system manager.
106 Internal error: bad low ASCII number: <b>nnn</b> . (PSPERR 106)	Internal software error, not under user control.	Contact system manager.
107 Internal error: bad high ASCII number: <b>nnn</b> . (PSPERR 107)	Internal software error, not under user control.	Contact system manager.
108 Can't write to the environment file. (FSERR <b>nnn</b> )	FWRITEDIR failed.	Refer to the file system error number for more information and possible action.
109 Form names must begin with a letter.	A non-alphabetic character begins the form name.	Check the form name.
110 This program doesn't support this terminal, since V/3000 doesn't.	Terminal is not supported by V/3000.	Consult the V/3000 Reference manual (P/N 32209-90001) for supported terminals.
111 This program doesn't support this device, since V/3000 doesn't.	Terminal is not supported by V/3000.	Consult the V/3000 Reference manual (P/N 32209-90001) for supported terminals.
201 This file is not an environment file.	Specified file exists, but has the wrong file code.	Specify an environment file (code=P2680)
203 Internal error: Bad record type: Expected <b>nnn</b> , not <b>nnn</b> . (PSPERR 203)	Probably a bad environment file, or possibly an internal software error, not under user control.	Write down error message and contact system manager.
204 Bad env file: bad pointer <b>nnn</b> . (PSPERR 204)	Error in environment file, not under user control.	Write down error message and contact system manager.
206 Bad env file: default meas. sys. bad in phys page. (PSPERR 206)	Error in environment file, not under user control.	Write down error message and contact system manager.
207 Bad env file: # recs inconsistent in free header. (PSPERR 207)	Error in environment file, not under user control.	Write down error message and contact system manager.
208 Bad env file: inconsistent log page # in LP desc. (PSPERR 208)	Error in environment file, not under user control.	Write down error message and contact system manager.
209 Bad env file: bad orientation in log page desc. (PSPERR 209)	Error in environment file, not under user control.	Write down error message and contact system manager.
210 Bad env file: bad units in log page desc. (PSPERR 210)	Error in environment file, not under user control.	Write down error message and contact system manager.
211 Bad env file: expected to find compiled phys page. (PSPERR 211)	Error in environment file, not under user control.	Write down error message and contact system manager.
212 Bad env file: cset # out of range in phys pg desc. (PSPERR 212)	Error in environment file, not under user control.	Write down error message and contact system manager.
216 Bad env file: expected compiled form descriptor. (PSPERR 216)	Error in environment file, not under user control.	Write down error message and contact system manager.

Error Messages	Meaning	Action
217 Bad env file: expected form number <i>nnn</i> , got <i>nnn</i> . (PSPERR 217)	Error in environment file, not under user control.	Write down error message and contact system manager.
218 Bad env file: wrong name in compiled form descriptor. (PSPERR 218)	Error in environment file, not under user control.	Write down error message and contact system manager.
219 Bad env file: uncompiled form found in log page desc. (PSPERR 219).	Error in environment file, not under user control.	Write down error message and contact system manager.
220 Bad env file: invalid forms part of log pg desc. (PSPERR 220)	Error in environment file, not under user control.	Write down error message and contact system manager.
221 Bad env file: expected multi-copy forms table. (PSPERR 221)	Error in environment file, not under user control.	Write down error message and contact system manager.
222 Bad env file: MC form should already be compiled. (PSPERR 222)	Error in environment file, not under user control.	Write down error message and contact system manager.
223 Bad env file: expected VFC descriptor. (PSPERR 223)	Error in environment file, not under user control.	Write down error message and contact system manager.
224 Bad env file: expected a VFC file name. (PSPERR 224)	Error in environment file, not under user control.	Write down error message and contact system manager.
225 Bad env file: VFC descriptor didn't match table. (PSPERR 225)	Error in environment file, not under user control.	Write down error message and contact system manager.
226 Bad env file: invalid VFC part of log page desc. (PSPERR 226)	Error in environment file, not under user control.	Write down error message and contact system manager.
227 Bad env file: VFC should already have been compiled. (PSPERR 227)	Error in environment file, not under user control.	Write down error message and contact system manager.
228 Bad env file: pointer to end was nil, start wasn't. (PSPERR 228)	Error in environment file, not under user control.	Write down error message and contact system manager.
229 Bad env file: bad orientation in char descriptor. (PSPERR 229)	Error in environment file, not under user control.	Write down error message and contact system manager.
230 Bad env file: expected Y or N in PS in char desc. (PSPERR 230)	Error in environment file, not under user control.	Write down error message and contact system manager.
231 Bad env file: illegal value in log page desc. (PSPERR 231)	Error in environment file, not under user control.	Write down error message and contact system manager.
237 Bad env file: incorrect version number. (PSPERR 237)	Error in environment file, not under user control.	Write down error message and contact system manager.
239 Bad env file: Bad orientation in char font desc. (PSPERR 239)	Error in environment file, not under user control.	Write down error message and contact system manager.
241 Bad env file: inconsistent char font table. (PSPERR 241)	Error in environment file, not under user control.	Write down error message and contact system manager.
242 One or more character fonts do not match for the current device.	Environment file converted from one device to another. Character references invalid for new device.	Change the font reference(s) to file(s) that contain fonts for the new device.
249 A logical page needs char font " <b>xxx</b> " which isn't in the file.	The base character font of a logical page refers to a character font which is not defined in the environment.	Check logical page definitions or define the character font.
250 A logical page needs a char font, but there are none in the file.	The base character font of a logical page refers to a character font which is not defined in the environment.	Check logical page definitions or define the character font.

Error Messages	Meaning	Action
251 A logical page needs char font # <i>nnn</i> which isn't in the file.	The base character font of a logical page refers to a character font which is not defined in the environment.	Check logical page definitions or define the character font.
256 This file can be examined, but not changed.	MPE file security violation; user does not have appropriate access capability.	Contact your system manager.
257 Bad srce env file: compiled list was inconsistent. (PSPERR 257)	Error in source environment file, not under user control.	Write down error message and contact system manager.
258 Bad srce env file: expected a form table. (PSPERR 258)	Error in source environment file, not under user control.	Write down error message and contact system manager.
259 Bad dest env file: expected a form table. (PSPERR 259)	Error in destination environment file, not under user control.	Write down error message and contact system manager.
260 Can't copy any more forms. Maximum is 32. (PSPERR 260)	Too many forms; each size and orientation of any given form counts as form.	Check forms in destination file. Recompiling destination file eliminates any deleted forms. Then try the copy again.
261 Bad srce env file: expected a VFC table. (PSPERR 261)	Error in source environment file, not under user control.	Write down error message and contact system manager.
262 Bad dest env file: expected a VFC table. (PSPERR 262)	Error in destination environment file, not under user control.	Write down error message and contact system manager.
263 Can't copy any more VFCs. Maximum is 32. (PSPERR 263)	Too many VFCs.	Recompile destination file to eliminate any deleted VFCs. Then try the copy again.
265 Can't continue compiling. (No existing compiled font to use.)	Could not access the specified character font.	Identify the problem character font, specify a valid character font, then recompile.
267 Can't continue compiling. (No existing compiled form to use.)	Could not access the specified form.	Identify the problem form, specify a valid form, then recompile.
269 Can't continue compiling. (No existing compiled VFC to use.)	Could not access the specified VFC.	Identify the problem VFC, specify a valid VFC, then recompile.
270 Warning: Required ComArea length is too big. (PSPERR 270)	Cannot use the environment with the intrinsics.	Reduce the size of the environment file. (By deleting forms, character fonts, etc.)
271 Error opening form file " <i>xxx</i> "...	First line of a 2 line message.	
272 Error accessing form file " <i>xxx</i> "...	First line of a 2 line message.	
273 Out of space in env. file - copy to a bigger file. (PSPERR 273)	File reached its physical end of file.	Create a larger environment file using the BUILD command and copy existing environment into it using FCOPY.
280 Warning: there are no active logical pages in this environment.	There are logical pages defined, but none of them are active.	Call PACTIVATEPAGE before printing any data or moving the pen.
281 Can't use primary font, since " <i>xxx</i> " (in phys. page) is undefined.	Primary font name specified in PHYSICAL PAGE menu is not defined in this environment.	Define the character font or change the primary font name.
282 Can't use primary font, since # <i>nnn</i> (in phys. page) is undefined.	Primary font number specified in PHYSICAL PAGE menu is not defined in this environment.	Define the character font or change the primary font number.



Error Messages	Meaning	Action
403 Internal error: no char. font supplied to delete. (PSPERR 403)	Internal software error, not under user control.	Contact system manager.
408 Internal error: no log page supplied to delete. (PSPERR 408)	Internal software error, not under user control.	Contact system manager.
409 IDSCHAR file " <b>xxx</b> " had wrong version #: <b>nnn</b>	Probably means the character font file has been damaged.	Contact system manager.
410 Old IDSCHAR file. Format must be converted using IDSCHAR.	IFS encountered a font/logo file made by an earlier release of IDSCHAR.	Bring the font/logo file into the new version of IDSCHAR to update the file format.
411 Error encountered while converting between devices.	An environment file couldn't be completely or accurately converted.	Manually inspect the environment and correct any incorrectly converted or unconverted values.
412 <b>nn</b> Warning(s) while converting between devices.	Non-fatal condition(s) while converting between devices.	None (This is just a warning).
413 Reduction is not available on the 2688A	2:1 and 4:1 environment initialization cannot be done for the 2688A.	Select a full size HP-supplied environment for your initialization.
414 Bad char font file: the storage fmt wasn't dot/bit. (PSPERR 414)	Probably means the character font file has been damaged.	Contact system manager.
420 Bad env file: ran out of records to free. (PSPERR 420)	Error in environment file, not under user control.	Write down error message and contact system manager.
421 Bad env file: list to free was inconsistent. (PSPERR 421)	Error in environment file, not under user control.	Write down error message and contact system manager.
428 Char font ## <b>nnn</b> in phys page has not been defined. Can't compile.	Primary or secondary character font specified in the PHYSICAL PAGE menu is not defined in the environment.	Define the character font number or change the primary or secondary font.
429 Font " <b>xxx</b> " in phys page is undefined. Can't compile.	Primary or secondary character font specified in the PHYSICAL PAGE menu is not defined in the environment.	Define the character font number or change the primary or secondary font.
431 Internal error: bad parms to ADD'TO'DELETE'LIST. (PSPERR 431)	Internal software error, not under user control.	Contact system manager.
441 Copy would result in more than 32 character fonts - no copy done.	Too many character fonts in the environment.	Recompile destination file to eliminate deleted fonts, if any.
442 <b>Nnn</b> char fonts to copy, but current env only has room for <b>nnn</b> more.	Too many character fonts in the environment.	Recompile destination file to eliminate deleted fonts, if any.
448 Copy would result in more than 32 logical pages - no copy done.	Too many logical pages in the environment.	Recompile destination file to eliminate deleted logical pages, if any.
450 <b>Nnn</b> logical pages to copy, but current env only has room for <b>nnn</b> more.	Too many logical pages in the environment.	Recompile destination file to eliminate deleted logical pages, if any.
452 Couldn't open HP defined env file ... <b>xxx</b>	First line of a 2 line message. MPE file system error follows.	Contact system manager.
453 Couldn't close HP defined env file... <b>xxx</b>	First line of a 2 line message. MPE file system error follows.	Contact system manager.
455 File code error while accessing form file.	Form file does not have a valid file code.	Specify a file with file code of PFORM.

## Formation Error Codes

Error Messages	Meaning	Action
456 Internal error: Tried to add a duplicate key to form file.	Internal software error, not under user control.	Contact system manager.
457 Internal error: # <i>nnn</i> occurred while accessing form file.	Internal software error, not under user control.	Contact system manager.
458 Internal error: Tried to open new form file with existing name.	Internal software error, not under user control.	Contact system manager.
459 Internal error: Record not found in form file.	Internal software error, not under user control.	Contact system manager.
460 Internal error: not enough space form directory in form file open.	Internal software error, not under user control.	Contact system manager.
461 This file is not an IDS/FORM file.	Form file is an incorrect type of file.	Check form file name. If valid, contact system manager.
462 The form file is at EOF.	Form file has been damaged.	Contact system manager.
463 File name supplied is not an IDIFORM file. (Wrong file code.)	Form file specified does not have valid file code.	Specify a file with file code of PFORM.
464 Form file error: bad units found in form header. (PSPERR 464)	Bad IDS/FORM file.	Contact system manager.
465 Internal error: KKA count returned didn't match request.	Internal software error, not under user control.	Contact system manager.
466 No file name available for form " <i>xxx</i> ".	Form was specified, but not the forms file name, and there was no default forms file name in PHYSICAL PAGE AND ENVIRONMENT DEFAULTS menu.	Specify the forms file name.
467 There is no default form file for this environment.	Form was specified, but not the forms file name, and there was no default forms file name in PHYSICAL PAGE AND ENVIRONMENT DEFAULTS menu.	Specify the forms file name.
470 Too many different forms in env. Can't compile. (PSPERR 470)	More than 32 forms; each size and orientation of any given form counts as a form.	Delete forms until there are only 32 forms.
478 Can't open VFC file " <i>xxx</i> ". (FSERR <i>nnn</i> )	File system error occurred.	Refer to the file system error number for more information.
479 Can't close VFC file. (FSERR <i>nnn</i> )	File system error occurred.	Refer to the file system error number for more information.
480 " <i>xxx</i> " not a VFC file: not ASCII. (PSPERR 480)	File specified as the VFC file is a binary file; VFC file must be ASCII.	Specify a valid VFC file.
481 Too many VFCs. Can't compile. (PSPERR 481)	More than 32 VFCs, not under user control.	Contact system manager.
482 Bad "VFC" line in file " <i>xxx</i> ". (PSPERR 482)	Record beginning with "VFC" is in error.	Check VFC files for valid format.
483 Can't read from VFC file " <i>xxx</i> ". (FSERR <i>nnn</i> )	FWRITEDIR failed.	Refer to the file system error number for more information.
484 " <i>xxx</i> " not a VFC file: bad 1st line. (PSPERR 484)	First record was not in VFC file format.	Check VFC file for valid format.

Error Messages	Meaning	Action
485 No "VFC" line in file "xxx". (PSPERR 485)	Missing record beginning with "VFC" in the file specified for VFC.	Check VFC file for valid format.
486 This is not IDSCHAR cell file.	Cell file specified has invalid file code, should be PCHAR.	Verify the name of the cell file specified.
490 Can't open IDSCHAR file "xxx". (FSERRR !)	FWRITEDIR failed.	Refer to the file system error number for more information.
491 File "xxx" is not an IDSCHAR file.	Cell file specified has invalid file code, should be PCHAR.	Verify the name of the cell file specified.
492 Can't read from file "xxx".	Second line of a 2 line message. First line displayed file system error.	Refer to the file system error number for more information.
493 Can't find right size in "xxx". (PSPERR 493)	Character font file has been changed since you defined it, actual size is not in the file.	Use CHARACTER FONT menu to define an actual size which exists in the file.
494 Can't read cell versions from file "xxx".	Second line of a 2 line message. First line displayed file system error.	Refer to the file system error number for more information.
495 Bad IDSCHAR file - ran out of dot/bit. (PSPERR 495)	Probably means the character font file has been damaged.	Contact system manager.
496 Can't close "xxx". (FSERR !)	FWRITEDIR failed.	Refer to the file system error number for more information.
801 Internal error: bad parameters to COMPILERFORM. (PSPERR 801)	Internal software error, not under user control.	Contact system manager.
802 Bad form file: form head missing for form "nnn". (PSPERR 802)	System is unable to process form file. Form file has been damaged and must be restored.	Contact system manager to restore file.
803 Bad form file: form head bad in form "nnn". (PSPERR 803)	System is unable to process form file. Form file has been damaged and must be restored.	Contact system manager to restore file.
804 Warning: scan line nnn. Too many triplets. (PSPERR 804)	Form is too complicated; too many graphics (line, box, shading, etc.) and characters or logos on a given line.	Simplify the form.
806 Can't do FPOINT on triplet temp file. (FSERR nnn) (PSPERR 806)	FWRITEDIR failed.	Refer to the file system error number for more information.
807 Can't sort the triplet file. SORTINITIAL failed. (PSPERR 807)	Error in SORT/3000 processing.	Contact system manager.
808 Can't sort the triplet file. SORTOUTPUT failed. (PSPERR 808)	Error in SORT/3000 processing.	Contact system manager.
810 Can't write to dot/bit temp file. (FSERR !)(PSPERR 810)	FWRITEDIR failed.	Refer to the file system error number for more information.
811 Can't read from dot/bit temp file. (FSERR !)(PSPERR 811)	FWRITEDIR failed.	Refer to the file system error number for more information.
812 Bad form file: found a diagonal line. (PSPERR 812)	System is unable to process form file. Form file has been damaged and must be restored.	Contact system manager to restore file.
813 Warning: the form had to be clipped.	Text or logo extends beyond the form boundaries.	Check placement of text or logo.

## Formation Error Codes

Error Messages	Meaning	Action
815 Bad form file: found a duplicate form head. (PSPERR 815)	System is unable to process form file. Form file has been damaged and must be restored.	Contact system manager to restore file.
816 Bad form file: found illegal element type: <i>nnn</i> . (PSPERR 816)	System is unable to process form file. Form file has been damaged and must be restored.	Contact system manager to restore file.
817 Bad form file: bad element length. (PSPERR 817)	System is unable to process form file. Form file has been damaged and must be restored.	Contact system manager to restore file.
818 Internal error: can't find form " <i>xxx</i> ". (PSPERR 818)	Internal software error, not under user control.	Contact system manager.
819 Bad form file: hit EOR before EOR element found. (PSPERR 819)	System is unable to process form file. Form file has been damaged and must be restored.	Contact system manager to restore file.
820 Can't open the triplet temp file. (FSERR <i>nnn</i> ) (PSPERR 820)	FWRITEDIR failed.	Refer to the file system error number for more information.
821 Can't open the dot/bit temp file. (FSERR <i>nnn</i> ) (PSPERR 821)	FWRITEDIR failed.	Refer to the file system error number for more information.
822 Bad form file: bad thickness value in line element.	System is unable to process form file. Form file has been damaged and must be restored.	Contact system manager to restore file.
823 Bad form file: can't find subform. (PSPERR 823)	System is unable to process form file. Form file has been damaged and must be restored.	Contact system manager to restore file.
824 Internal error: can't switch to subform. (PSPERR 824)	Internal software error, not under user control.	Contact system manager.
825 Internal error: can't switch back from subform. (PSPERR 825)	Internal software error, not under user control.	Contact system manager.
826 Bad form file: a subform contained another subform. (PSPERR 826)	System is unable to process form file. Form file has been damaged and must be restored.	Contact system manager to restore file.
827 Bad form file: bad line type <i>nnn</i> in line element. (PSPERR 827)	System is unable to process form file. Form file has been damaged and must be restored.	Contact system manager to restore file.
828 Can't write to triplets temp file. (FSERR <i>nnn</i> ) (PSPERR 828)	FWRITEDIR failed.	Refer to the file system error number for more information.
829 Bad form file: bad shading percentage <i>nnn</i> . (PSPERR 829)	System is unable to process form file. Form file has been damaged and must be restored.	Contact system manager to restore file.
830 Bad form file: corners of a box were out of order. (PSPERR 830)	System is unable to process form file. Form file has been damaged and must be restored.	Contact system manager to restore file.
831 Warning: missing character(s) or logo(s). Black rectangle used.	In a field heading or logo, tried to print a character that is not defined in the cell file.	Check the field heading text or the character font or logo file.
852 Bad form file: bad character orientation. (PSPERR 852)	System is unable to process form file. Form file has been damaged and must be restored.	Contact system manager to restore file.

Error Messages	Meaning	Action
853Bad form file: bad text direction. (PSPERR 853)	System is unable to process form file. Form file has been damaged and must be restored.	Contact system manager to restore file.
854 Couldn't close cell file "xxx". (PSPERR 854)	Second line of a 2 line message, first line displayed file system error.	Refer to MPE file system error.
855 Couldn't open cell file "xxx". (PSPERR 855)	Second line of a 2 line message, first line displayed file system error.	Refer to MPE file system error.
856 Warning: Expected logo file "xxx" not font.	Specified a logo file in the FIELD FONT menu, but file is a character font file. Form may not print as expected.	Change the specification or the file name.
857 Warning: Expected font file "xxx" not logo.	Did not specify a logo file in the FIELD FONT menu, but file is a logo file. Form may not print as expected.	Change the specification or the file name.
859 Error storing dot/bit. (PSPERR 859)	Second line of a 2 line message, first line displayed file system error.	Refer to MPE file system error.
860 No font sizes in cell file "xxx". (PSPERR 860)	Cell file is empty.	Check the specified cell file.
861 Form file containing form "xxx" is bad. (PSPERR 861)	System is unable to process form file. Form file has been damaged and must be restored.	Contact system manager to restore file.
862 Error getting XDS. GETDSEG returned <i>nnn</i> . (PSPERR 862)	GETDSEG failed.	Refer to the error number.
863 Error altering size of XDS, possible internal error. (PSPERR 863)	Internal software error, not under user control.	Contact system manager.
864 Error moving character description from XDS. (PSPERR 864)	Internal software error, not under user control.	Contact system manager.
865 Error expanding XDS. (PSPERR 865)	Internal software error, not under user control.	Contact system manager.
866 Error moving character description to XDS. (PSPERR 866)	Internal software error, not under user control.	Contact system manager.
867 Form file containing form "xxx" is bad. (PSPERR 867)	System is unable to process form file. Form file has been damaged and must be restored.	Contact system manager to restore file.
868 Bad cell file "xxx": bad units. (PSPERR 868)	Cell file has probably been damaged.	Contact system manager.
869 Couldn't free the extra data segment. (PSPERR 869)	Internal software error, not under user control.	Contact system manager.
870 Configured extra data seg size too small. (PSPERR 870)	Internal software error, not under user control.	Contact system manager.
871 Bad baseline in cell file "xxx". (PSPERR 871)	Cell file has probably been damaged.	Contact system manager.
872 Warning: a field has an unspecified character font.	Cell file name was not specified in the FIELD FONT menu.	Check the cell file name.
873 ALTDSEG error while checking what size dseg we got. (PSPERR 873)	Internal software error, not under user control.	Contact system manager.
874 Bad cell file: max P.S. bound less than min. (PSPERR 874)	Cell file has probably been damaged.	Contact system manager.

## Formation Error Codes

Error Messages	Meaning	Action
875 Error accessing cell file "xxx". (PSPERR 875)	Second line of a 2 line message, first line displayed file system error.	Refer to MPE file system error.
900 Couldn't close the environment file. (FSERR nnn)	FWRITEDIR failed.	Refer to the file system error number for more information and possible action.
901 Not enough room in the PSP ComArea. Need nnn words.	User program specified insufficient space.	Check the ComArea length parameter in PINITIALIZE and PINITDEVICE.
902 Couldn't close the environment file. (FSERR nnn)	FWRITEDIR failed.	Refer to the file system error number for more information and possible action.
903 File wasn't an environment file. Wrong file code.	Environment file associated with output print file is in error.	Check the FILE command to determine problem file.
904 File wasn't an environment file. Wrong record size.	Environment file associated with output print file is in error.	Check the FILE command to determine problem file.
905 Couldn't read the environment file's header. (FSERR nnn)	FWRITEDIR failed.	Refer to the file system error number for more information and possible action.
908 Bad environment file.	Environment file has been damaged.	Contact system manager.
909 FGETINFO failed on the environment file. (FSERR nnn)	FWRITEDIR failed.	Refer to the file system error number for more information and possible action.
910 Bad env file: file header had wrong type.	Environment file has been damaged.	Contact system manager.
911 Bad env file: illegal pointer in file header.	Environment file has been damaged.	Contact system manager.
912 Couldn't read the char font table. (FSERR nnn)	FWRITEDIR failed.	Refer to the file system error number for more information.
913 Couldn't read a char font descriptor. (FSERR nnn)	FWRITEDIR failed.	Refer to the file system error number for more information.
914 Bad env file: char font table had wrong type.	Environment file has been damaged.	Contact system manager.
915 Bad env file: a char font descriptor had wrong type.	Environment file has been damaged.	Contact system manager.
916 Couldn't read the logical page table. (FSERR nnn)	FWRITEDIR failed.	Refer to the file system error number for more information.
917 Bad env file: the logical page table had the wrong type.	Environment file has been damaged.	Contact system manager.
918 Couldn't read a logical page descriptor. (FSERR nnn)	FWRITEDIR failed.	Refer to the file system error number for more information.
919 Bad env file: a logical page descriptor had the wrong type.	Environment file has been damaged.	Contact system manager.
920 Couldn't read compiled log page rec. (FSERR nnn)	FWRITEDIR failed.	Refer to the file system error number for more information.
921 Bad env file: compiled log page rec had wrong type.	Environment file has been damaged.	Contact system manager.
922 Couldn't read the physical page descriptor. (FSERR nnn)	FWRITEDIR failed.	Refer to the file system error number for more information.

<b>Error Messages</b>	<b>Meaning</b>	<b>Action</b>
923 Bad env file: the physical page descriptor had the wrong type.	Environment file has been damaged.	Contact system manager.
924 Couldn't read the compiled physical page. (FSERR <i>nnn</i> )	FWRITEDIR failed.	Refer to the file system error number for more information.
925 Bad env file: the compiled physical page had the wrong type.	Environment file has been damaged.	Contact system manager.
926 Bad env file: bad pointer in phys page desc.	Environment file has been damaged.	Contact system manager.
927 Bad env file: nil pointer to compiled phys page in PP desc.	Environment file has been damaged.	Contact system manager.
928 Couldn't read the multi-copy fo ms descriptor. (FSERR <i>nnn</i> )	FWRITEDIR failed.	Refer to the file system error number for more information.
929 Bad env files: MC forms descriptor had wrong type.	Environment file has been damaged.	Contact system manager.
930 Couldn't read a single copy forms descriptor. (FSERR <i>nnn</i> )	FWRITEDIR failed.	Refer to the file system error number for more information.
931 Bad env file: Single copy forms descriptor had wrong type.	Environment file has been damaged.	Contact system manager.
932 Ran out of space while building the PSP ComArea.	Internal software error, not under user control.	Contact system manager.
933 Couldn't read the form table. (FSERR <i>nnn</i> )	FWRITEDIR failed.	Refer to the file system error number for more information.
934 Bad env file: the form table had the wrong type.	Environment file has been damaged.	Contact system manager.
935 Couldn't read a compiled form descriptor. (FSERR <i>nnn</i> )	FWRITEDIR failed.	Refer to the file system error number for more information.
936 Bad env file: a compiled form descriptor had the wrong type.	Environment file has been damaged.	Contact system manager.
937 Bad env file: bad symbol table.	Environment file has been damaged.	Contact system manager.
938 Can't read symbol table record. (FSERR <i>nnn</i> )	FWRITEDIR failed.	Refer to the file system error number for more information.
939 Bad env file: a symbol table record had the wrong type.	Environment file has been damaged.	Contact system manager.
940 Logical page orientation in ComArea was bad.	ComArea has been damaged.	Check your program.
941 No current form - can't write to fields.	Called PWRITEFIELD on a logical page that has no form.	Check your program.
942 The current logical page has no entry in the PSP ComArea.	ComArea has been damaged.	Check your program.
943 The logical page entry for the current LP is bad.	ComArea has been damaged.	Check your program.
944 Can't move the pen. (FDEVICECONTROL error <i>nnn</i> )	FDEVICECONTROL failed.	Refer to FDEVICECONTROL error number.
945 Can't write to field. (FSERR <i>nnn</i> )	FWRITEDIR failed.	Refer to the file system error number for more information.

Formation Error Codes

Error Messages	Meaning	Action
946 Form "xxx" is not on the current logical page.	PNEWFORM was called using a form that is not on the logical page. (The current logical page is not the MULTICOPY FORMS logical page.)	Check your program.
947 Form "xxx" isn't on the current log page or the multi-copy table.	PNEWFORM was called using a form that is neither on the logical page nor in the MULTICOPY FORMS TABLE.	Check your program.
948 Bad character font entry.	ComArea has been damaged.	Check your program.
949 Couldn't find field "xxx".	Called PWRITEFIELD using a field which is not in the current subform. If no current subform then field is not in any subform.	Check your program.
950 There are no active logical pages.	All pages have been deactivated or no initially active pages were defined in the environment.	Activate a logical page.
951 Can't go to the next page. (FSERR <i>nnn</i> )	FWRITEDIR failed.	Refer to the file system error number for more information.
952 <i>Nnn</i> is an illegal logical page number. Should be 0 to 31 or -1.	An intrinsic was called using an invalid logical page number.	Check your program.
953 Can't activate logical page. (FDEVICECONTROL error <i>nnn</i> )	FDEVICECONTROL failed.	Refer to FDEVICECONTROL error number.
954 Can't deactivate logical page. (FDEVICECONTROL error <i>nnn</i> )	FDEVICECONTROL failed.	Refer to FDEVICECONTROL error number.
955 Can't select character fonts. (FDEVICECONTROL error <i>nnn</i> )	FDEVICECONTROL failed.	Refer to FDEVICECONTROL error number.
956 A font number of <i>nnn</i> is illegal. Must be 0- 31 or -1.	An intrinsic was called using an invalid character font number.	Check your program.
957 The specified primary font ( <i>nnn</i> ) is not in the environment.	An intrinsic was called using a primary font number which is not defined.	Check your program.
958 The specified secondary font ( <i>nnn</i> ) is not in the environment.	An intrinsic was called using a secondary character number font which is not defined.	Check your program.
959 Warning: line too long -truncated.	A given line of data exceeded PWRITEFIELD's buffer size (250 characters).	Check your program.
960 <i>Nnn</i> isn't a legal subfield number. Should be greater than 0, or -1.	An intrinsic was called using an invalid subfield number.	Check your program.
961 Subform "xxx" is not in the current form.	PNEWSUBFORM was called using a subform name that is not defined in the current form.	Check your program.
962 The string is too long - its length would be > 32767 dots.	PSTRINGWIDTH intrinsic was called using too large a value in the string length parameter.	Check your program.
963 Bad current character font number in the ComArea.	ComArea has been damaged.	Check your program.
964 Character font number <i>nnn</i> is not in the environment.	An intrinsic was called using an undefined character font number.	Check your program.



Error Messages	Meaning	Action
965 Character font "xxx" is not in the environment.	An intrinsic was called using an undefined character font.	Check your program.
966 The primary character font number is not in the environment.	An intrinsic was called using a primary font number which is not defined.	Check your program.
967 The secondary character font number is not in the environment.	An intrinsic was called using a secondary font number which is not defined.	Check your program.
968 Bad current logical page number in the ComArea.	ComArea has been damaged.	Check your program.
969 Logical page number <i>nnn</i> is not in the environment.	An intrinsic was called using an undefined logical page number.	Check your program.
970 Invalid language parameter.	PINITIALIZE or PINITDEVICE was called with an invalid value in the language parameter.	Check your program.
971 <i>Nnn</i> isn't a legal subfield for this field. Should be 1 or -1.	PWRITEFIELD was called using an invalid subfield number.	Check your program.
972 Can't get environment file name. (FSERR <i>nnn</i> )	FWRITEDIR failed.	Refer to the file system error number for more information and possible action.
973 The file supplied apparently had no environment file.	FFILEINFO indicated there was no environment file associated with output print file.	Check file equation for FOPEN.
974 PERRMSG was called when error had occurred.		Check your program.
975 PNEWFORM was called with a blank form name.		Check your program.
976 Bad form name "xxx". (Must begin with a letter, or be "1" or "2".)	PNEWFORM was called using an invalid form name.	Check your program.
977 A -1 was used as the subfield number in a field with <i>nnn</i> subfields.	Value of -1 for subfield number is valid only when there is one subfield.	Check your program.
978 Can't go to the next physical page. (FSERR <i>nnn</i> )	FWRITEDIR failed.	Refer to the file system error number for more information.
979 <i>Nnn</i> isn't a legal subfield for this field. Should be 1 to <i>nnn</i> , or -1.	PWRITEFIELD was called using an invalid number in the subfield parameter.	Check your program.
980 Warning: logical page <i>nnn</i> was already active.	PACTIVATEPAGE was called using the page number of an active page.	Check your program.
981 Warning: logical page <i>nnn</i> was already inactive.	PDEACTIVATEPAGE was called using the page number of an inactive page.	Check your program.
982 Warning: there are no initially active pages in the environment.	No logical pages were defined as initially active.	Activate a logical page.
983 There is no currently active logical page to go to.	PNEWPAGE or PNEWPHYSPAGE was called when there were no active logical pages.	Activate a logical page.
984 There is no current logical page.	No logical pages were defined as initially active in the environment.	Activate a logical page.

Formation Error Codes

Error Messages	Meaning	Action
985 -1 was used for a logical page #, but there is no current page.	The value -1 used in the logical page number parameter in PLOGPAGEINFO means to use the current page, but there is none.	Check your program.
986 Deactivating the default logical page is not allowed.	PDEACTIVATEPAGE was called using the default page number.	Check your program.
987 Warning: all logical pages have been deactivated.	There must always be one active logical page. Activate a logical page before moving off the last deactivated logical page.	Check your program.
988 Not enough space to even begin building the PSP ComArea.	Need at least 158 words in the ComArea to determine the actual space needed for ComArea. User program specified insufficient space.	Check the ComArea length parameter in PINITIALIZE or PINITDEVICE.
989 Not enough space to even begin building the PSP ComArea.	Need at least 158 words in the ComArea to determine the actual space needed for ComArea. User program specified insufficient space.	Check the ComArea length parameter in PINITIALIZE or PINITDEVICE.
990 Not enough space to even begin building the PSP ComArea.	Need at least 158 words in the ComArea to determine the actual space needed for ComArea. User program specified insufficient space.	Check the ComArea length parameter in PINITIALIZE or PINITDEVICE.
991 Not enough space to even begin building the PSP ComArea.	Need at least 158 words in the ComArea to determine the actual space needed for ComArea. User program specified insufficient space.	Check the ComArea length parameter in PINITIALIZE or PINITDEVICE.
992 Illegal character font name "xxx". (Must begin with a letter)	PFONTNUM was called using an invalid character font name.	Check your program.
993 Bad ComArea - probably hasn't been initialized yet.	PINITIALIZE or PINITDEVICE must be called once before calling any other intrinsic.	Check your program.
994 (Nnn,nnn) is illegal. Can't have negative absolute coordinates.	PMOVEPENABS was called using a negative number in either the x or y coordinate parameters or both.	Check your program.
995 X coordinate nnn is too big. Logical page width is nnn.	PMOVEPENABS or PMOVEPENREL was called using a value in the x-coordinate parameter which exceeds the logical page width.	Check your program.
996 Y coordinate nnn is too big. Logical page height is nnn.	PMOVEPENABS or PMOVEPENREL was called using a value in the y-coordinate parameter which exceeds the logical page height.	Check your program.
997 This intrinsic is not applicable to the initialized device xxx.	The intrinsic being called cannot be used with the output device specified.	Check your program.

Error Messages	Meaning	Action
998 This intrinsic is not useable without an opened spoolfile.	Only the PCONVERTFIGURE and PFIGUREINFO intrinsics can be used without an open spoolfile.	Check your program. Spoolfile should be set to -1 in a call to PINITDEVICE.
1000 Image number <i>nn</i> is out of range; must be between 0 and 31 inclusive.	The raster image number must be between 0 and 31 inclusive.	Check your program.
1001 Can't open raster file <i>xxx</i> (FSERR <i>nnn</i> ).	Unable to access specified file.	Refer to the file system error number for more information.
1002 FGETINFO failed; can't read the file label (FSERR <i>nnn</i> ).	Internal software error; not under user control.	Note the file system error number and contact system manager.
1003 The file is not a raster file; invalid file code <i>nnn</i> .	The specified file has a code that is not a raster file code.	Check your file. The raster file code is 1114 or RASTR.
1004 The file is not a raster file; invalid record size.	The specified file has the wrong record size to be a raster file.	Check your file. Raster file record size is 512 words.
1005 No raster file name was specified.	The raster file name parameter is all blanks.	Check your program.
1006 Can't close raster file <i>xxx</i> (FSERR <i>nnn</i> ).	File system error occurred in attempt to close raster file.	Refer to the file system error number for more information.
1007 An error occurred while reading the file header (FSERR <i>nnn</i> ).	Internal software error; not under user control.	Note file system error number and contact system manager.
1008 Reached EOF while trying to read the file header (FSERR <i>nnn</i> ).	Error in raster file; probably bad number of file header records.	Recreate the raster file.
1009 DMOVIN failed; bounds violation error.	Internal software error; not under user control.	Contact system manager.
1010 DMOVOUT failed; bounds violation error	Internal software error; not under user control.	Contact system manager.
1011 XDS access failed; invalid Figure Load Table index.	Internal software error; not under user control.	Contact system manager.
1012 The raster file was converted for a device other than that initialized.	Output device specified for initialization must be the same as that specified for conversion.	Check your program.
1013 The partitioned raster file contains no raster data.	Error in raster file.	Create a raster file that contains data.
1014 An error occurred while trying to read the raster file (FSERR <i>nnn</i> ).	Internal software error; not under user control.	Note file system error number and contact system manager.
1015 Encountered FDEVICECONTROL error <i>nnn</i> while trying to load image <i>nn</i> .	Internal software error; not under user control.	Contact system manager.
1016 <i>xxx</i> is not a device supported by IFS/3000.	The specified output device cannot be used with the IFS/3000 intrinsics.	Check your program or contact system manager.
1017 The header record count does not match the number found.	Error in raster file.	Recreate the raster file or create a new raster file.
1018 Internal problem; downloaded image count <i>nnn</i> is invalid.	Internal software error or corrupted ComArea.	Check your ComArea or contact system manager.
1019 Warning: new image replaces previously loaded image <i>nn</i> .	Image <i>nn</i> replaces the image previously assigned number <i>nn</i> .	None - warning only.
1020 Encountered FDEVICECONTROL error <i>nn</i> while trying to delete image <i>nn</i> .	Internal software error; not under user control.	Contact system manager.

## Formation Error Codes

Error Messages	Meaning	Action
1021 Image <b>nn</b> has not been loaded.	An image cannot be printed or deleted before it has been loaded.	Check your program.
1022 Bad data word count <b>nn</b> found in raster file.	Error in raster file.	Recreate the raster file.
1023 Invalid units parameter <b>n</b> ; must be 0, 1, 2 or 3.	Units must be 0 for dots, 1 for inches, 2 for cm, or 3 for mm.	Check your program.
1024 Invalid position mode parameter <b>n</b> ; must be 0 or 1.	Position mode must be 0 for relative or 1 for absolute positioning.	Check your program.
1025 Encountered FDEVICECONTROL error <b>nn</b> while trying to print image <b>nn</b> .	Internal software error; not under user control.	Contact system manager.
1026 Relative position ( <b>n, n</b> in dots) would be off the logical page.	X or Y position is too large or negative and too large.	Check your program and logical page dimensions.
1027 Absolute position ( <b>n, n</b> in dots) would be off the logical page.	X or Y position is too large.	Check your program and logical page dimensions.
1028 X-position does not reduce to a value in the range -32768..32767.	X position is too large or too small.	Check your program.
1029 Y-position does not reduce to a value in the range -32768..32767.	Y position is too large or too small.	Check your program.
1030 Can't specify a negative absolute X-position ( <b>nnn</b> in dots).	Only positive values are allowed with absolute positioning.	Check your program.
1031 Can't specify a negative absolute Y-position ( <b>nnn</b> in dots).	Only positive values can be used with absolute positioning.	Check your program.
1032 Invalid device <b>xxx</b> for figure file conversion.	Output device specified cannot accept a figure file for conversion.	Check your program.
1033 Invalid raster image rotation <b>n</b> ; must be 0, 90, 180 or 270.	Only 0, 90, 180, and 270 are valid raster image rotations.	Check your program.
1034 AGLERR - the device <b>xxx</b> is not supported.	Internal problem; not under user control.	Contact system manager.
1035 Not enough space to create AGL ComArea; requested <b>nnn</b> .	Internal software error; not under user control.	Contact system manager.
1039 Can't open temporary file <b>xxx</b> (FSERR <b>nnn</b> ).	Specified file has not been opened.	Refer to the file system error number for more information.
1040 Can't rename temporary file <b>xxx</b> (FSERR <b>nnn</b> ).	Specified file has not been renamed.	Refer to the file system error number for more information.
1041 Can't close temporary file <b>xxx</b> (FSERR <b>nnn</b> ).	Specified file has not been closed.	Refer to the file system error number for more information.
1042 Can't FCLOSE renamed temporary file; failed on re-rename (FSERR <b>nnn</b> ).	File has not been renamed; attempt to rename back to original name also failed.	Refer to the file system error number for more information.
1043 Can't FCLOSE renamed temporary file (FSERR <b>nnn</b> ).	File has been renamed; but attempt to close failed.	Refer to the file system error number for more information.
1044 Internal problem; no Device Driver matches <b>xxx</b> in case selection.	Internal software error; not under user control.	Contact system manager.
1045 Internal problem; no Units matches <b>xxx</b> in case selection.	Internal software error; not under user control.	Contact system manager.

Error Messages	Meaning	Action
1046 Encountered FDEVICECONTROL error <i>nnn</i> while printing temporary image.	Internal software error; not under user control.	Contact system manager.
1047 Image height ( <i>nnn</i> in MM) is larger than the device will allow.	Image has not been printed.	Change image height parameter to a smaller value.
1048 Image height ( <i>nnn</i> in MM) cannot be 0 or negative.	Image has not been printed.	Change image height parameter to a larger positive value.
1049 The file header crash word is set to true.	Error in raster file.	Recreate raster file.
1050 Image width ( <i>nnn</i> in MM) larger than the device will allow.	Image height is too large, resulting in width also being too large.	Change image height parameter to a smaller value.
1051 Figure height read from figure file is 0; no image to convert.	Error in figure file.	Recreate figure file.
1052 Figure height read from figure file is <0; no image to convert.	Error in figure file.	Recreate figure file.
1053 Figure width read from figure file is <0; no image to convert.	Error in figure file.	Recreate figure file.
1054 Invalid raster image type <i>n</i> ; must be 0 or 1.	Raster image type must be 0 for temporary or 1 for permanent.	Check your program.
1055 Bad Image Height; can't reduce to a value in the range -32768..32767.	Image height is too large.	Check your program.
1056 Internal error; PFIGUREINFO item <i>nn</i> not applicable to raster file.	Internal software error; not under user control.	Contact system manager.
1057 GETDSEG error. Returned index of <i>n</i> ; invalid length.	Internal software error; not under user control.	Contact system manager.
1058 GETDSEG error. Returned index of <i>n</i> ; too many XDSs this process.	Internal software error; not under user control.	Contact system manager.
1059 GETDSEG error. Returned index of <i>nnn</i> ; not enough storage for XDS.	Internal software error; not under user control.	Contact system manager.
1060 GETDSEG error. Returned index of <i>nnn</i> ; stack frozen - can't expand.	Internal software error; not under user control.	Contact system manager.
1061 GETDSEG error. GETDSEG returned index of <i>nnn</i> .	Internal software error; not under user control.	Contact system manager.
1062 Can't rename temporary file to <i>xxx</i> (FSERR <i>nnn</i> ).	Specified file has not been renamed.	Refer to the file system error number for more information.
1063 <i>xxx</i> already exists as a permanent file.	Specified file has already been created.	Purge file or use another file name.
1064 <i>xxx</i> already exists as a temporary file.	Specified file has already been created.	Purge temporary file, save and rename to permanent file, or use another file name.
1065 Invalid type <i>n</i> ; must be -1, 0 or 1.	File type must be -1 for unknown, 0 for raster, 1 for figure.	Check your program.
1066 Figure width read from figure file is 0. No image to covert.	Error in figure file.	Recreate figure file.
1067 Real number in <i>xxx</i> too small to represent.	ASCII string contains real number that is too small.	Check your program.

## Formation Error Codes

Error Messages	Meaning	Action
1068 Invalid char and real number too small to represent in <b>xxx</b> .	ASCII string contains invalid character and real number that is too small.	Check your program.
1069 Real number in <b>xxx</b> too large to represent.	ASCII string contains real number that is too large.	Check your program.
1070 Invalid char and real number too large to represent in <b>xxx</b> .	ASCII string contains invalid character and real number that is too large.	Check your program.
1071 Invalid character in <b>xxx</b> to convert to a real number.	ASCII string contains invalid character.	Check your program.
1072 No number found in <b>xxx</b> to convert to a real number.	ASCII string must contain a real number.	Check your program.
1073 Error while attempting to convert <b>xxx</b> to a real number.	Internal software error; not under user control.	Contact system manager.
1074 Cannot convert real number to a string; result returned is <b>xxx</b> .	Real number to be converted is probably too large.	Check your program.
1100 AGLERR 501. XMin >=XMax or YMin >=YMax.	Internal software error; not under user control.	Contact system manager.
1101 AGLERR 508. One or more viewport corners outside the NDC space.	Internal software error; not under user control.	Contact system manager.
1102 AGLERR 510. NDC position is outside the current viewport.	Internal software error; not under user control.	Contact system manager.
1103 AGLERR 512. WC position is outside the current viewport.	Internal software error; not under user control.	Contact system manager.
1104 AGLERR 705. The view surface is already initialized.	Internal software error; not under user control.	Contact system manager.
1105 AGLERR 706. No output device is associated with the view surface.	Internal software error; not under user control.	Contact system manager.
1106 AGLERR 708. The specified surface has not been initialized.	Internal software error; not under user control.	Contact system manager.
1107 AGLERR 717. The graphics package has not been initialized	Internal software error; not under user control.	Contact system manager.
1108 AGLERR 718. The function is not supported by the device.	Internal software error; not under user control.	Contact system manager.
1109 AGLERR 1001. The GSET array is not large enough.	Internal software error; not under user control.	Contact system manager.
1110 AGLERR 1004. The string length is <0.	Internal software error; not under user control.	Contact system manager.
1111 AGLERR 1101. Mapping value is not 0 or 1. Map remains unchanged.	Internal software error; not under user control.	Contact system manager.
1112 AGLERR 1201. FOPEN failure (FSERR <b>nnn</b> ).	Internal software error; not under user control.	Note file system error number and contact system manager.
1113 AGLERR 1202. FCLOSE failure (FSERR <b>nnn</b> ).	Internal software error; not under user control.	Note file system error number and contact system manager.
1114 AGLERR 1203. Error writing to device (FSERR <b>nnn</b> ).	Internal software error; not under user control.	Note file system error number and contact system manager.
1115 AGLERR 1204. Error reading from device (FSERR <b>nnn</b> ).	Internal software error; not under user control.	Note file system error number and contact system manager.

Error Messages	Meaning	Action
1116 AGLERR 1210. Error occurred when calling FCHECK.	Internal software error; not under user control.	Contact system manager.
1117 AGLERR 1215. FFILEINFO failure (FSERR <i>nnn</i> ).	Internal software error; not under user control.	Note file system error number and contact system manager.
1118 AGLERR 1216. DMOVOUT failure.	Internal software error; not under user control.	Contact system manager.
1119 AGLERR 1217. DMOVIN failure.	Internal software error; not under user control.	Contact system manager.
1120 AGLERR 1218. GETDSEG failure.	Internal software error; not under user control.	Contact system manager.
1121 AGLERR 1219. FREEDSEG failure.	Internal software error; not under user control.	Contact system manager.
1122 AGL error 1301. Internal error. A call to BINARY failed.	Internal software error; not under user control.	Contact system manager.
1123 AGLERR 1302. Internal error. Floating point result out of range.	Internal software error; not under user control.	Contact system manager.
1124 AGLERR 1502. No figure of that name was found.	The specified figure does not exist.	Check the figure name.
1125 AGLERR 1504. Specified figure file is currently in use.	Figure file is being written to at this time.	Try again later.
1126 AGLERR 1505. No figure file of that name can be found.	The specified figure file does not exist.	Check the figure file name.
1127 AGLERR 1506. Rotation angle out of range.	Internal software error; not under user control.	Contact system manager.
1128 AGLERR 1507. Scale factor out of range.	Internal software error; not under user control.	Contact system manager.
1129 AGLERR 1509. Attempt to open a figure file failed (FSERR <i>nnn</i> ).	Internal software error; not under user control.	Note file system error number and contact system manager.
1130 AGLERR 1510. Attempt to close a figure file failed (FSERR <i>nnn</i> ).	Internal software error; not under user control.	Note file system error number and contact system manager.
1131 AGLERR 1515. Figure file is opened for both input and output.	Internal software error; not under user control.	Contact system manager.
1132 AGLERR 1516. System crash flag set to true in figure.	Error in figure file.	Recreate figure file.
1133 AGLERR 1518. Invalid figure name.	Incorrectly specified figure name.	Check the figure name.
1134 AGLERR 1800. File system error (FSERR <i>nnn</i> ) figure not converted.	Internal software error; not under user control.	Note file system error number and contact system manager.
1135 AGLERR 1801. IFS/3000 error trying to print image. (PSPERR <i>nnn</i> )	Internal software error; not under user control.	Refer to the IFS error number in this appendix for more information.
1136 AGLERR 1802. OUT2680A error (FSERR <i>nnn</i> ); figure not converted.	PCONVERTFIGURE intrinsic caused error in OUT2680A temporary file.	Refer to the file system error number for more information.
1137 AGLERR 1803. CREATEPROCESS returned error <i>nnn</i> ; figure not converted.	Figure has not been converted or printed.	Refer to CREATEPROCESS error number in the MPE Ininsics Reference Manual.
1138 AGLERR 1804. SORT/3000 returned error <i>nnn</i> ; figure not converted.	Figure has not been converted or printed.	Refer to the error number in the SORT/3000 Reference Manual.

## Formation Error Codes

Error Messages	Meaning	Action
1139 AGLERR 1805. Conversion process failed; figure not converted.	Internal software error; not under user control.	Contact system manager.
1140 AGLERR 1806. Partitioning process failed; figure not converted.	Internal software error; not under user control.	Contact system manager.
1141 AGLERR 1807. Figure not converted.	Internal software error; not under user control.	Contact system manager.
1142 Warning: AGL warning 1851. Figure may contain errors.	Internal conversion error; not under user control.	Check printed output for errors.
1143 Warning: AGL warning 1852. Figure may contain errors.	Internal partitioning error; not under user control.	Check printed output for errors.
1144 AGLERR 1808. Image would be positioned off page; image not printed.	Internal software error; not under user control.	Contact system manager.
1145 AGLERR 709. Unable to load AGL '2680DDD module; (LOAD ERR <i>nnn</i> ).	If LOAD ERR number is 65, there are not enough system CST entries available.	Note LOAD error number and contact system manager.
1198 AGLERR <i>nnn</i> (FS parameter is <i>nnn</i> ).	Internal software error; not under user control.	Contact system manager.
1199 Warning: AGL warning <i>nnn</i> . (FS parameter is <i>n</i> ).	Internal software error; not under user control.	Contact system manager.

If an error occurs when using intrinsics, the intrinsics set a status word in the comarea. A second word is set to the number of the error that has occurred. You should check error status words after each call to the intrinsics. The following table lists the error numbers and their meanings.

Error Number	Meaning	Action
1300	Not enough memory for run-time conversion of environment file from DCS to PCL.	Recompile environment file, including the PCL printer to which you want to print. If impossible, rerun the application with a larger comarea specified.
1301	PCL download list is corrupt.	Recompile environment file. If still a problem, recreate the environment file.
1302	Bad logical record in PCL download list.	Recompile environment file. If still a problem, recreate the environment file.
1303	Bad record type in PCL download list.	Recompile environment file. If still a problem, recreate the environment file.
1304	Bad status returned when environment file loaded.	Recompile environment file. If still a problem, recreate the environment file.
1400	Incorrect width or height specified in call to <b>PBOX</b> .	Width or height cannot be < 0. Correct and rerun program.



<b>Error Number</b>	<b>Meaning</b>	<b>Action</b>
1401	Incorrect line width specified in call to <b>PBOX</b> .	Line width cannot exceed 32 points or 150 dots. Correct and rerun program.
1402	Incorrect line style specified in call to <b>PBOX</b> .	Must be 0 for solid lines. Correct and rerun program.
1403	Incorrect shading specified in call to <b>PBOX</b> .	Must be an integer between (and including) 0 and 100 which represents a percentage of shading. Correct and rerun program.
1500	Incorrect justification specified in call to <b>PWRITEFIELDJUST</b> .	Justification must be one of the following: 0=left, 1=right, or 2=centered. Correct and rerun program.
2000	Input string to <b>PBAR2OF5STRING</b> is too large.	Correct parameter and rerun program.
2001	Output string for <b>PBAR2OF5STRING</b> is too short for encoded output.	Correct parameter and rerun program.
2002	Input string to <b>PBAR2OF5STRING</b> contains a non-numeric character.	Correct parameter and rerun program.
2010	For <b>PWRITEFIELDREL</b> , the positioning parameter cannot be 6 while the data length parameter is 0.	Correct parameter and rerun program.
6000	Call to <b>HPSWITCHTOCM</b> from the Formation Intrinsic Switch Stubs has failed.	Attempt to determine last intrinsic called and correct any incorrect parameters. Call Unison-Tymlabs for help.
6001	Call to <b>HPFP_CONVERT</b> from the Formation Intrinsic Switch Stubs has failed.	Attempt to determine last intrinsic called and correct any incorrect parameters. Call Unison-Tymlabs for help.



**B***Glossary*

<b>Baseline</b>	An imaginary line along which the printer aligns character cells. The baseline runs along the bottom of most characters in a font, but is above the lowest part of letters with descenders (g, j, p, y). When characters are underlined, the underline generally lies over the baseline.
<b>CPI</b>	Characters per inch A measure of the spacing of adjacent characters on a page. Also called pitch.
<b>Character cell</b>	A rectangle which entirely encloses a character. Also, an array of dots large enough to hold the largest character in front.
<b>Dots or DPI</b>	Dots Per Inch (DPI). A measure of the resolution of a printed page. LaserJets print at 300 dpi and the 2680 printers print 180 dpi.
<b>Environment</b>	In this manual, the collection of fonts and page layouts with which a printer will interpret the data it receives from its host.
<b>Environment file</b>	A file containing a print environment, i.e., containing all of the fonts, forms, and page layout information that will be used in a print job.
<b>Escape character</b>	Generally, a character in a stream that tells whatever is interpreting the stream that whatever follows is to be interpreted differently. For a LaserJet, an escape character is a specific character in the ASCII character set, with decimal value 27 (octal%33) that tells the LaserJet that the following data are instructions for the formatter rather than data to be placed on the page.
<b>File system</b>	A collection of system software used to manage files and devices on a computer system.
<b>Font</b>	A family of characters that share a single typeface, weight, style and size and that generally forms a complete set of alphabetic characters, numerals, and punctuation.
<b>Formatter</b>	The part of a laser printer that interprets the print data according to the environment and directs the printer's hardware to write the output appropriately.
<b>Fuser</b>	The part of a laser printer which fuses toner powder to the paper.
<b>Gray scale</b>	The ability of a display device to show shades of gray, as opposed to color or to simple black and white.
<b>HMI</b>	Horizontal motion index. Indicates horizontal movement across a page and is equivalent to pitch in fixed pitch font measures.

---

<b>Impact printer</b>	A printer which forms an image by mechanical contact of some part of the mechanism with the paper.
<b>Italics</b>	A variation of a typeface that is usually slanted (like this), or is more ornate than the normal (Roman or upright) variation.
<b>Landscape</b>	A print direction or orientation such that printing reads across the wide dimension of the page.
<b>Leading</b>	The amount of space between printed lines (pronounced "leading"). Usually measured in points.
<b>Logical page</b>	An area of the printed page that is treated as a complete page, even though it does not cover the entire physical page.
<b>LPI</b>	Lines per inch.
<b>Monospace</b>	A characteristic of a font in which every character takes the same amount of space, like a typewriter or line printer. Also called "fixed-pitch" since the number of characters per unit never varies. Contrast with "proportional."
<b>Non-impact Printer</b>	A printer which forms an image by some means other than physical impact of the type with the paper. Laser printers, ink-jet printers, and thermal printers are examples of non-impact printers.
<b>OP</b>	system supervisor
<b>Orientation</b>	The direction of printing, either landscape or portrait.
<b>Physical page</b>	A sheet of paper.
<b>Pitch</b>	A measure of the spacing of adjacent characters on a page. Usually measured in characters per unit, e.g., characters per inch.
<b>Pixel</b>	A single displayable element of a picture. Depending on the output device, a pixel may be either black or white, or may have a single uniform color or shade of gray.
<b>Point</b>	A printer's unit of measure; approximately 1/72nd of an inch.
<b>Point Size</b>	The size of a font as measured in points. Usually, the point size of a font refers to the size of the tallest character, or to the tallest character plus normal line spacing (leading).
<b>Portrait</b>	The print direction or orientation in which printing reads across the short dimension of the page.

---

**Proportional Spacing**

A characteristic of a font in which the spacing between letters varies with the width of the characters being printed. Contrast with "monospace" or "fixed-pitch."

**Raster** An array of pixels that form a complete picture.

**Roman** An upright font, as contrasted with an italic or slanted font.

**SM** System manager

**Spooler** Usually, a part of system software (or perhaps external hardware) that stores printed output for later transfer to a printer. More generally, software which stores data at high speed for later transfer to a lower-speed device and is capable of taking data from many sources. An acronym for "Simultaneous Peripheral Operation On-Line."

**Style** The description of a font as upright, slanted, or italic.

**Terminal Type File** A file used by system software on the HP3000 to control the operation of a terminal or other asynchronous peripheral (such as a low-speed printer). The file contains information about protocol and initialization.

**Typeface** The characteristic of a font that describes its general appearance stripped of variations such as boldness, slant, and size.

**Upright** A font which is neither slanted nor italicized.

**Vector** As used in computer graphics, a line from one endpoint to another, described mathematically rather than pictorially.

**VMI** Vertical motion index. Indicates the vertical movement across a page and is equivalent to leading.

**Weight** A measure of the thickness of a font. Generally described as "bold" for a thick font and "light" for a thin font.



## Index

### A

ADDFONT UDC 3-2, 3-3 to 3-4, 5-24  
 ADDPAGE UDC 3-2, 3-5 to 3-6, 4-8  
 ASK  
   available environment files 8-7  
   compiling forms 8-7  
   defining logical pages with UDCs 8-12  
   forms printing for MPE V users 8-3  
   forms printing for MPE XL users 8-2  
   modifying environment files 8-12  
   modifying forms 8-5  
   multi-part forms use with 8-9  
   printing forms 8-7  
   sample form files 8-5  
   using Formation with 8-1  
 ASK form 8-1 to 8-14  
   custom command file 8-9  
   environment files 8-2, 8-7  
   form files 8-5  
   modifying 8-5 to 8-6  
   multi-copy 8-9 to 8-14  
     environment file 8-12  
   printing 8-2 to 8-3, 8-7 to 8-8  
     multi-copy 8-14  
   TIFF file 8-6  
   types 8-2  
 ASKFILLF 8-12  
 ASKINVLE.SAMPLES.TYM 8-7  
 ASKINVLF 8-5, 8-7  
 ASKORGLF 8-10  
 ASKPOLE 8-10  
 ASKPOLE.SAMPLES.TYM 8-7, 8-12  
 ASKPOLF 8-5, 8-7, 8-10  
 ASKPOPE.SAMPLES.TYM 8-7  
 ASKPOPF 8-5, 8-7  
 ASKPRINT command 8-2, 8-7, 8-9  
 ASKPRINT command file 8-9  
 ASKRMTLF 8-10  
 ASKSMTP.E.SAMPLES.TYM 8-7  
 ASKSMTPF 8-5, 8-7

### B

bar code fonts 5-2, 7-4  
   2 of 5 code 5-30  
   code 39 specification 5-27  
   PBAR2OF5STRING intrinsic 6-35  
   postnet 5-31  
 bitmapped fonts 5-2

### C

cartridge fonts 5-3  
 Character Font Control screen 5-24, 6-4

Character Set Control 3-28  
 CHARSETS.SYS 5-3  
 check printing 7-28  
 cheqsys fonts 5-12  
 CIERROR JCW 3-2  
 COBOL programs  
   using Intrinsic in 6-11  
 comArea, see Communication Area  
 command files 3-2  
   ASK 8-9  
 commands  
   ASK file 8-9  
   IDSFORM 7-19  
   Spool Queue Monitor 4-26  
   SQM1SESS 4-27  
   SQMKICK 4-27  
   SQMPAUSE 4-26  
   SQMSHOW 4-27  
   SQMSTART 4-26  
   SQMSTOP 4-26  
 Communication Area 6-12  
 COMPILEENV UDC 3-7  
 compiling  
   environment files 3-46  
   FDUSER.FORM.TYM 5-15  
   SQM1ENV.D into SQM1ENV.C 4-20  
 controlling form feeds 4-16  
 CONVERT utility 5-17  
   logo management screen 5-19  
   running 5-17  
   screen 5-18  
 converting  
   programs for Intrinsic use 6-7  
 CREATEENV UDC 3-2, 3-8 to 3-9, 4-8, 5-24  
 custom forms  
   ASK 8-3

### D

data  
   moving 4-13 to 4-15  
   positioning 4-3, 4-13  
 data fonts  
   assigning 5-24  
   selecting an HP 3000 5-25  
 DCS 7-2  
 Delete Character Font 3-44  
 Delete Logical Page 3-45  
 demo  
   installation 2-3  
   quick start 2-5  
   validation 2-3  
 documentation conventions iii  
 DOWNLENV Utility 3-47  
 duplex

- environment 3-8, 3-11
- printing 4-12
  - intrinsic 6-39, 6-47
  - multi-part forms 4-11
  - overriding 6-2

## E

- Environment File menu 3-15
- environment files 3-1 to 3-52
  - adding logical page 3-5
  - adding/replacing fonts 3-3
  - ASK forms 8-7, 8-12
    - listing 8-2
  - compatibility 3-50
  - compiling 3-7, 3-11, 3-46
  - compiling with COMPILEENV 3-46
  - creating 3-2, 3-8, 3-11
  - creating with Environment Manager 3-13
  - creating with UDCs 3-2
  - custom listing 7-15
  - displaying 3-10
  - downloading utility 3-47
  - form restrictions 3-2
  - HP 3-50
  - HP 2680/88 7-7 to 7-11
  - layouts 3-48
  - managing 1-3
  - Missing Components report 7-7
  - modifying 3-2
  - multi-part forms 4-8
  - naming 3-8
  - pre-defined 3-48
  - recompiling
    - for migration 7-7
    - for new devices 7-9
    - from HP 2680/88 7-24
    - missing components 7-7, 7-8
  - reduction printing 4-6
  - supplied 7-16
- Environment Manager
  - Charater Set Control 3-28
  - compiling environment files 3-46
  - creating Environment Files 3-13
  - Delete Character Font 3-44
  - Delete Logical Page 3-45
  - environment file menu 3-15
  - Logical Page Control 3-34
  - Logical Page Form Control 3-40
  - main menu 3-16
  - menus 3-14
  - multi-copy forms 3-25
  - physical page control screen 3-20
  - scale IDIFORM 3-27
  - starting 3-13

- VFC menu 3-43

- error messages A-4 to A-22
- errors
  - message 20 A-1
  - message 21 A-2
  - reduction printing A-2
  - troubleshooting A-1 to A-23

## F

- FD2000.FORM.TYM 5-13
- FD4.FORM.TYM 5-13
- FD5000.FORM.TYM 5-13
- FDCHEQ.FORM.TYM 5-12
- FDCOMP 5-4
- FDIIXIII.FORM.TYM 5-13
- FDTYM.FORM.TYM 5-4
- FDUSER.FORM.TYM 5-4, 5-6 to 5-15
  - entry format 5-6 to 5-11
- FIGPRINT utility 5-21
- File Equate facility 2-5, 4-2
- files
  - ASKORGLF 8-10
  - SQM1DEV.FORM.TYM 4-19
  - SQM1ENVD.FORM.TYM 4-20
  - SQMSAMP 4-25
  - suspect 4-26
  - terminal type 2-8
  - transferring to HP 3000 1-3
- finding missing components 7-8
- FMDUPLEX JCW 6-2
- FMFORCELJ 6-9
- FMFORCELJ JCW 7-15
- font directory 5-4 to 5-15
  - adding fonts 5-4 to 5-15
  - multiple entries 5-5
- FONTDIR 5-4
- FONTDIR.FORM.TYM 5-4
- FONTDIRK.FORM.TYM 5-4
- FONTDOWN utility 5-26
- fonts
  - about 5-1
  - accessing 5-2
  - adding
    - to environment 3-3
  - adding resident to FDUSER 5-12
  - adding to directory 5-6
  - assigning 3-8, 5-24
    - for data 5-24
  - assigning numbers 3-3
    - HP 2680/2688 3-4, 8-2
  - available on HP 3000 5-25
  - available resident 5-13
  - bar code 5-2, 5-27
  - bitmapped 5-2



- cheqsys 5-12
- considerations 5-21
- control 3-28
- converting PCEL to PCELL files 5-16
- data 5-2
- defining orientation 3-30
- deleting 3-44
- downloading PCL with FONTDOWN 5-26
- duplicate names 5-5
- for form text 5-2
- hard 5-2
- listing 5-13
  - FDTYM 5-5
- metric 5-11
- metric entries 5-11
- orientation codes 3-29
- replacing 3-3
- replacing from HP 2680/88 7-20 to 7-23
- replacing HP 2680/88 7-20
- resident 5-3, 5-4, 5-5, 5-8 to 5-10
  - adding 5-12
  - overriding 5-5
- resident in FDTYM 5-4
- resident PCL 5-3
- resident scalable 5-9
- scalable 5-2
- select HP 3000 5-25
- selecting 5-21
- selecting for data 5-1
- selection precedence 5-5
- shipped with Formation Design 5-2
- sizes 5-21
- soft 5-2, 5-3, 5-7
- soft PCELL 5-7
- Type Director 7-3
- types of 5-2
- FONT.SYS 5-3
- form feeds
  - controlling 4-16
- form files 3-2
  - ASK 8-5
- Formation
  - design program 1-3
  - system overview 1-2
  - utilities 1-5
- Formation Design
  - installation 8-3 to 8-4
- FORMMIG.SYS 7-6
- forms
  - complex 7-28
  - converting for HP 2680/88 printing 7-18
  - customizing ASK 8-3
  - modification 7-18 to 7-23
  - modifying ASK 8-5
    - modifying for HP 2680/88 printing 7-18
    - multi-part on HP 2680/88 printers 7-25
    - printing ASK forms 8-1
    - printing with File Equate 4-2
- Forms Express program 7-4
- G**
- Glossary
  - B-1 to B-3
  - printing terms 1-6
- graphics
  - converting TIFF files to PCELL or raster 5-17
  - printing figure files with FIGPRINT 5-21
  - printing raster images with intrinsics 6-4
  - TIFF 5-1
- H**
- hard fonts 5-2
- hardware requirements 1-1
- HP 2680/88 3-4, 8-2
  - applications printing 7-1
  - converting MPE V programs 7-12
  - converting programs 7-12
  - converting rotated text 7-23
  - DCS use of 7-7
  - downloading forms to PCs 7-23
  - font precedence 5-5
  - intrinsic conversion 7-12 to 7-14
  - job streams 7-15
  - migration 7-2 to 7-4
  - migration requirements 7-5 to 7-6
  - modifying forms 7-18, 7-18 to 7-23
  - modifying native mode programs 7-14
  - native mode intrinsics 7-14
  - new printer 6-9
  - printing converted applications 7-24
  - printing to LaserJets and HP 5000s 7-1
  - recompiling environment files 7-7 to 7-11, 7-24
  - scaling 7-10 to 7-11
  - switching to HP environment files 7-16
  - updating fonts 7-20 to 7-23
  - uploading forms to the HP 3000 7-24
  - vertical forms control 3-43
- HP PSP
  - report listing 7-12
- HP PSP intrinsics 6-8
- HP36580 6-32
- HP36580V 6-33
- HPENV.SYS 7-17
- HP PCL5A 8-4
- HPWord conversion 7-13
- I**
- IDS

- forms control 3-40
  - IDS2FORM
    - rotated text 7-23
    - utility 7-18
  - IDSChar 7-7
  - IDSFORM 3-26, 7-7, 7-18, 7-27
    - commands 7-19
    - file
      - converting 7-18 to 7-19
      - listing 7-18 to 7-19
      - scaling 7-18 to 7-19
  - IFS/3000 error messages A-4 to A-22
  - installation
    - ASK forms 8-9
    - demo 2-3
    - File Equate facility 2-5
    - Formation 2-2
    - Formation Design 8-3 to 8-4
    - Formation on MPE V 2-3
    - Formation on MPE XL 2-2
    - Intrinsics 2-7
    - printer driver 8-4
    - test of 2-4
    - upgrading system 2-7
  - internal fonts 5-3
  - Intrinsics 1-4, 6-1 to 6-80
    - call format 6-10
    - calling 6-10
    - capabilities 6-2
    - controlling physical page 6-2
    - converted program JCWs 6-9
    - converting programs for use with 6-7
    - data types 6-29
    - defining printer 7-15
    - graphics 6-4
    - HP 2680/88
      - conversion 7-12 to 7-14
      - native mode 7-14
      - report 7-12
    - HP36580 6-32
    - HP3658V 6-33
    - initializing 6-2
    - installing 2-7
    - multiple fonts in data 6-2
    - PACTIVATEPAGE 6-34
    - PBAR2OF5STRING 6-35
    - PBOX 6-36
    - PDEACTIVATEPAGE 6-37
    - PDELETERASTER 6-38
    - PDUPLICATE 6-39
    - PENDSAVE 6-40
    - PERRMSG 6-41
    - PFIELDINFO 6-42
    - PFLASHRASTER 6-43
    - PFONTINFO 6-45
    - PFRONTPAGE 6-47
    - PINITIALIZE 6-50
    - PJOBSEPARATE 6-51
    - PLOADRASTER 6-52
    - PLOGPAGEINFO 6-53
    - PMOVEPENABS 6-55
    - PMOVEPENREL 6-56
    - PNEWFORM 6-57
    - PNEWPAGE 6-58
    - PNEWPHYSPAGE 6-60
    - PNEWSUBFORM 6-61
    - PPRINTFIGURE 6-62
    - PPRINTRASTER 6-64
    - PPRINTSAVED 6-66
    - PRESET 6-67
    - programming requirements 6-16
    - PSAVEDATA 6-68
    - PSELECTPAGE 6-69
    - PSETCOPIES 6-70
    - PSPPLUSVERSION 6-71
    - PSTATEINFO 6-72
    - PSTRINGWIDTH 6-73
    - PSWITCHPAGE 6-74
    - PSWITCHPHYPAGE 6-75
    - PUSEFONT 6-76
    - PWRITEFIELD 6-77
    - PWRITEFIELDJUST 6-78
    - PWRITEFIELDREL 6-79
    - sample program using 6-17, 6-20
    - smart fields 6-3
    - table of 6-30
    - using in MPE XL programs 6-8
- ## J
- JCW
    - CIERROR 3-2
    - FMDUPLEX 6-2
    - FMFORCELJ 6-9, 7-15
    - setting for converted programs 6-9
    - setting for print devices 7-15
  - JETSTART utility 6-7, 7-12 to 7-13
  - JETSTOP utility 6-8
  - JFORMX.FORM.TYM 7-13
  - JMIG.FORM.TYM 7-5
  - job streams 3-2
    - including UDCs in 3-2
    - modifying and updating for migration 7-15
    - Using Environment Files report 7-15
- ## L
- LISTENV UDC 3-10
  - logical page
    - adding 3-5

- assigning first 3-8
- controlling with intrinsics 6-3
- defining sequence 4-8
- deleting 3-45
- HP 2680/2688 3-25, 7-27
- modifying 3-5, 3-6
- no data 4-11
- numbering 3-5
  - first 3-8
- orientation codes 3-35
- Logical Page Control 3-34
- Logical Page Control screen 4-8
- Logical Page Form Control 3-40
- Logical Page Forms Control 4-8
- logo, ASK form 8-6
- logos
  - conversion screen 5-19
  - converting to character formats 5-17
- LPRINT
  - 2-up printing 4-9
  - controlling form feeds 4-16
  - data positioning 4-13
  - duplexed forms 4-12
  - duplexing forms 4-12
  - moving data 4-13 to 4-15
  - multi-part forms 4-7 to 4-12
    - form limit 4-7
  - no data 4-5
    - on logical pages 4-10
  - overview 4-3
  - pre-space carriage control 4-4
  - printing multi-part forms 4-9
  - printing PC documents 4-17
  - reduction printing 4-6
  - resetting the printer 4-13
- LPRINT Utility 4-3
- LPS conversion 7-13
- M**
- MAKEENV UDC 3-2, 3-11 to 3-12, 5-24
- menus
  - environment file 3-15
  - Environment Manager main 3-16
  - Migration 7-6
  - VFC 3-43
- messages
  - SQM1SESS 4-27
- metric entries 5-11
- migration 7-2
  - analysis 7-5 to 7-6, 7-12
  - analysis report 7-7
  - common questions 7-3
  - HP 2680/88 environment files 7-2
  - install migration tool 7-5
    - updating applications for new printers 7-5
- Migration Menu 7-6
- modify ASK forms 8-5 to 8-6
- moving data 4-13 to 4-15
- multi-copy forms 3-25
- Multi-Copy Forms screen 7-26
- multi-part forms 4-7
  - ASK 8-9 to 8-14
  - duplex printing 4-11
  - HP 2680/88 7-25 to 7-27
  - page limit 4-7
  - printing ASK 8-14
  - use on HP 2680/88 printers 7-25
  - use with ASK 8-9
- O**
- outfence
  - setting for SQM 4-28
- P**
- PACTIVATEPAGE 6-34
- page control
  - logical 3-34
- paper bin codes 3-38
- paper sizes 3-23
- PBAR2OF5STRING 6-35
- PBOX 6-36
- PCELL file 7-7
  - convert from PCL 5-16
  - fonts 5-7 to 5-8
- PCL 7-2
  - file conversion 5-16
  - fonts
    - converting to PCELL files 5-16
    - resident fonts 5-3
- PCL2DCS.FORM.TYM 5-16
- PDEACTIVATEPAGE 6-37
- PDELETERASTER 6-38
- PDUPLEX 6-39
- PENDSAVE 6-40
- permanent soft fonts 5-3
- PERRMSG 6-41
- PFIELDINFO 6-42
- PFLASHRASTER 6-43
- PFONTINFO 6-45
- PFORM files 7-7
- PFRONTPAGE 6-47
- physical page control 3-20
- Physical Page Control screen 5-5
- PINITIALIZE 6-50
- PJOBSEPARATE 6-51
- PLOADRASTER 6-52
- PLOGPAGEINFO 6-53
- PMOVEPENABS 6-55

- PMOVEPENREL 6-56
- PNEWFORM 6-57
- PNEWPAGE 6-58
- PNEWPHYSPAGE 6-60
- PNEWSUBFORM 6-61
- postnet bar code 5-31
- PPRINTFIGURE 6-62
- PPRINTRASTER 6-64
- PPRINTSAVED 6-66
- PRESET 6-67
- pre-space carriage control 4-4
- printer
  - converting to new 6-9
  - driver installation 8-4
  - FMFORCELJ JCW 7-15
  - resetting 4-13, 6-67
- printer model values 3-18
- printing
  - 2-up 4-9
  - ASK forms 8-2 to 8-3, 8-7 to 8-8
    - multi-part 8-14
  - ASK production 8-9
  - checks 7-28
  - converted applications 7-24
  - duplexed forms 4-12
    - multi-part 4-11
    - with PDUPLICATE intrinsic 6-39
  - duplexed forms with LPRINT 4-12
  - forms with LPRINT 4-3
  - forms with no data 4-5
  - forms without data 4-5
  - HP 2680/88 forms 7-17
  - moving data 4-13 to 4-15
  - multi-part ASK forms 8-14
  - multi-part forms 4-7 to 4-12
    - HP 2680/88 7-25
  - multi-part forms with LPRINT 4-3, 4-7, 4-9
  - no data on logical page 4-10
  - options 1-4, 4-1
  - PC documents 4-17
  - prespace carriage control 4-4
  - reduction 4-3, 4-6
  - rotated text 6-4
  - rotated text with Intrinsic 6-4
  - sample ASK forms with data 8-2
  - simple forms with LPRINT 4-4
  - single-copy forms with LPRINT 4-4
    - with data 4-4
    - with Pre-Space Carriage Control 4-4
  - programmer's library 7-28
  - Programs Using PSP Intrinsic report 7-12
  - PSAVEDATA 6-68
  - PSELECTPAGE 6-69
  - PSETCOPIES 6-70
  - PSP intrinsic 7-2, 7-12
  - PSPLUXL.FORM.TYM 7-14
  - PSPPLUSVERSION 6-71
  - PSTATEINFO 6-72
  - PSTRINGWIDTH 6-73
  - PSWITCHPAGE 6-74
  - PSWITCHPHYSPAGE 6-75
  - PUSEFONT 6-76
  - PUSEFONT intrinsic 5-1
  - PWRITEFIELD 6-77
  - PWRITEFIELDJUST 6-78
  - PWRITEFIELDREL 6-79
- R**
  - recompiling
    - interactive 7-10
    - scaling environment files during 7-10
  - reduction printing 4-3, 4-6
    - environment files 4-6
  - resetting the printer 4-13, 6-67
  - resident fonts 5-3, 5-4, 5-5, 5-8 to 5-10
    - adding 5-12
    - overriding 5-5
  - Roman 8 fonts
    - restrictions 3-4, 8-2
  - rotating text 6-4
    - example program for 6-6
    - IDS2FORM 7-23
- S**
  - scalable fonts 5-2
  - Scale IDSFORM to LaserJet 3-27
  - scaling 7-10 to 7-11
  - scaling factors 7-10
  - simplex
    - environment 3-8, 3-11
  - simplex printing
    - overriding 6-2
  - SIZELIST utility 5-25
  - smart fields 6-3, 6-79
  - soft fonts 5-2, 5-3, 5-7 to 5-8
  - software requirements 1-1
  - spool files
    - setting priorities for SQM 4-28
  - Spool Queue Monitor 4-18 to 4-29
    - commands 4-26
    - configuring 4-18
    - jobs 4-26
    - messages 4-27
    - monitoring special devices 4-24
    - processing parameters 4-25
    - redirecting printing with 4-24
    - setting spool file priorities 4-28
    - starting/stopping 4-26

- suspect spool files 4-26
  - using 4-26
- SQM, see Spool Queue Monitor
- SQM1DEV file
  - creating 4-19
- SQM1ENV.D file
  - compiling 4-20
  - creating 4-20
- SQM1JOB 4-27
- SQM1SESS command 4-27
- SQMKICK command 4-27
- SQM\_PAUSE command 4-26
- SQMSAMP file 4-25
- SQM\_SHOW command 4-27
- SQM\_START command 4-26
- SQM\_STOP command 4-26
- STARTFEQ utility 2-6
  - starting SQM 4-26
- STOPFEQ utility 2-6
  - stopping SQM 4-26
- string search 4-3
- syntax iii

## T

- Tagged Font Metric file 1-5, 5-14
- TDP conversion 7-13
- terminal type file 2-8
- TFMTRANS utility 5-14
- TFMTRANS.EXE 5-14
- transferring files 1-3
- troubleshooting A-1
  - expanding the environment file A-1

## U

### UDCs

- ADDFONT 3-3 to 3-4
- ADDPAGE 3-5 to 3-6
- COMPILENV 3-7
- CREATEENV 3-8 to 3-9
  - creating environment files with 3-2
  - defining logical pages for ASK forms 8-12
  - including in job streams 3-2
- LISTENV 3-10
  - listing 3-2
- MAKEENV 3-11 to 3-12
  - using in job streams 3-2

upgrading 2-7

Utilities

- CONVERT 5-17
- DOWNLENV 3-47
- FIGPRINT 5-21
- FONTDOWN 5-26
- IDS2FORM 7-18
- JETSTART 6-7

- JETSTOP 6-8
- LPRINT 4-3 to 4-17
- SIZELIST 5-25
- STARTFEQ 2-6
- STOPFEQ 2-6
  - table of 1-5
- TFMTRANS 5-14

## V

- validation 2-3
- VFC menu 3-43

## W

- WYSIWYG display 8-4

# UNISON FORMATION



Corporate Headquarters  
5101 Patrick Henry Drive  
Santa Clara, CA 95054  
408/988-2800  
Fax 408/988-2236

Austin Division  
811 Barton Springs Road  
Austin, TX 78704  
512/478-0611  
Fax 512/479-0735

European Headquarters  
Mardall House, Ground Floor  
9-11 Vaughan Road, Harpenden  
Hertfordshire AL5 4HU  
United Kingdom  
+44 (0) 1582 462424  
Fax +44 (0)1582 462425