

Fantasia/3000

Manual Supplement

March 2003

The last major revision of Fantasia/3000 manuals was done early in 1997. This document provides an update for all the Fantasia manuals with information on features added to the software since January 1997.

Questions, comments, suggestions, and requests for additional functionality should be addressed to:

LARC Computing
339 South San Antonio Road
Los Altos, CA 94022
650/941-9310
650/949-0750 Fax
Support@LARC-Inc.com

Our telephone line is covered live from 9AM to 1PM Pacific Time. An answering machine will take your message outside of those hours and we will return the call.

Open Seas is the owner of Fantasia/3000. From the United States and Canada you can reach them through LARC Computing. Their home office address is: Open Seas, Ltd. The Old School Hall, 6 Perrin Street, Headington, Oxford, OX3 7AS, England; e-mail: support@openseas.co.uk; phone 44-1865-744-688.

Contents:

- 1. Updating the Software**
Installing Fantasia/3000
Documentation
- 2. Listing of New Features**
- 3. Partial List of Problem Fixes**
- 4. Additional Information on Selected Features**
- 5. General Introduction to Fantasia/3000**
- 6. FSPOOL Efficiency Note**

Updating the Software

If you are running Fantasia Release F.03.14 or a later version the software is updated by just loading the new files provided. You need to ensure that no one is using the software when the files are loaded. If you use FSPOOL stop that job. `RESTORE *T;@.@.LARC`

Every effort has been made to ensure compatibility between releases. All of the files that printed correctly with the old version of Fantasia should work with the new. However it would be prudent to back up the LARC account before loading the new version.

The Editor is now compiled as a native mode program. A number of Editor commands that were no longer functional and/or not easily converted have been dropped. None of them were likely to be used in a file called by FSPOOL or another batch process; But caution is urged. The old version of the Editor is also on the release as `OLDEDIT.EDIT.LARC`.

Be sure that the FSPOOL job is not running when you perform the back-up and when you load the new files. Watch for messages from the STORE and RESTORE commands to ensure that all files on the disc in the LARC account have been stored prior to the update and that all files on the tape have been restored.

The update provides no files in the PUB group of the LARC account.

Installing Fantasia/3000

`RESTORE @.@.LARC;CREATE` loads the files you need. Three additional steps need to be taken after the restore:

1. Alter the access and capability parameters for the LARC account and the groups in that account.
2. Create files, including the FSPOOL database, required in the PUB group of the LARC account.
3. Establish appropriate parameters for the printers to be used as well as a default LDEV for output.

The JOBFile `INSTJOB.LASER.LARC` handles the first two tasks. `STREAM INSTJOB.LASER.LARC` to make that happen.

Documentation

`DOCUMENT.LASER.LARC` is a list of files provided with the Fantasia software.

`INSTDOC` provides additional instructions for installation and updating of the Fantasia software.

Four manuals come with the system; they can be printed with Fantasia/3000; some documentation (including this manual) are provided in a PDF version that can be used with Adobe's Acrobat on a PC.

The manuals are: `CFANTGUD.LASER.LARC` is a guide for the new user. It provides a good overview of the way the package works. `LASERDOC.LASER.LARC` is the Technical Reference Manual; `COMSDOC.LASER.LARC` has detailed information on the LASER program organized by command; and `EDITDOC.EDIT.LARC` documents the LARC Editor. Online help is available with `LINFO.LASER.LARC` for tables of useful information and `HELPME.LASER.LARC` with information on commands in LASER and the LARC Editor. This document is in the file `NEWDOC.LASER.LARC`.

Version

Changes to this document have been made for version H.02.02 of the software.

2. Listing of New Features

The installation and update procedures have been simplified. Updating is accomplished by loading new files from a tape. See the previous section of this document for more information.

Fantasia can produce PDF output with use of the `OPENPDF` program available from OpenSeas. PDF is the format used by Adobe Acrobat on PC's; it is widely used for documentation, transfer of documents between PC's, and in fax or e-mail applications. If the software is installed, the `LASER` program will call it automatically when `LASER` is run with `PARM=97`. PDF files can be generated automatically from `FSPool`. Provisions exist to allow the user to specify the output file name. See Section 4 of this document for further information.

Fantasia/3000 now can use forms drawn by (almost) any software on the PC. A new program `CLEANUP` takes care of the file format after the file has been moved to the `HP3000`. See Section 4 of this document for further information.

A pseudo device can be established with `LINEPARM` to have `LASER` write either PCL or PDF output to a disc file. This option also works when `LASER` is run through `FSPool`. See Section 4 of this document for further information.

A new program `SHOWPARM` displays the parameters as set by `LINEPARM`. From the colon prompt `RUN SHOWPARM.LASER.LARC`. It should be self explanatory. No files are changed by this program.

New options in `EDITOR` take further advantage of strings stored by the program. It is now possible to copy a string; to set it to a string in the workfile; and to test string values with an `@IF` statement. See Section 4 of this document for further information.

The `FSPool` configuration program `FSPCONF` offers an option to back up the data base at the end of the run. See Section 4 of this document for further information.

The `Editor` (`EDITOR.EDIT.LARC`) has been compiled as a native mode program. In the process a number of Editor commands have been dropped. (The new version is less convenient for program development as the `compile/prep/run` commands are among those eliminated.) The old editor is also included with the Fantasia update as `OLDEDIT.EDIT.LARC`. See the note in the Installation instructions above.

The `EDITOR` command `FILEDATE` followed by a file set specification shows create, modify, and access dates for each file in the set.

The `EDITOR` deals with variable record length files. If work file came from a file with variable record length it will be kept the same way. The option word `SHORT` used with `KEEP` creates a variable record length file; `LONG` creates a fixed record length file.

The Fantasia MainMenu program has been reorganized to use three screens rather than four. Programs used less frequently have been dropped. Users can customize a fourth screen for their own use and/or the dropped programs can be accessed from the MPE prompt or with UDC's.

The `CODES` program displays, and interprets, PCL codes in a file; the current version will also provide a list of valid PCL commands. When the program is used to list PCL codes in a file it also gives a summary of the codes found. Try running the program; it should be self explanatory.

`FFCONV` is the program that reformats forms created by JetForm Design or Fantasia/PC. It is called by `FORMCONV`. It is now possible to run `FFCONV` stand-alone. See Section 4 of this document for further information. The documentation below explains the use and features of `FORMCONV` as it describes the new capability.

FSPPOOL can call the LARC Editor instead of LASER. In this case the "environment" file in the mapping entry is an Editor USEFile. This feature allows editing of the spoolfile and/or conditional printing. See Section 4 of this document for further information.

A new file, FILESET.SPOOL.LARC reestablishes the data files used by FSPPOOL when it is STREAMED. This is helpful when recovering from a situation in which FSPSYNCH (or one of the other files) got corrupted.

Entries from here to the end of this section refer to features of the LASER program in the Fantasia package:

Provision has been made to print the Euro currency symbol. See Section 4 of this document for further information.

Mail/Merge applications tend to reopen the same INFile repeatedly. A parameter called ECONOMY is set with the command \ECONOMY to have the program leave an INFile open after it is used.

The one-at-a-time option is now governed with LISTSPF. Previously a file was maintained with flags; if the program was aborted, the flag stayed on locking that device for Fantasia.

The commands for shading can take advantage of user-defined patterns. Pattern number 7-20 are used for that purpose. See Section 4 of this document for further information.

The command \PCL sends a string to the output. It works like PRINT except: any caret ("^") in the string is converted to an escape (octal 33); and no carriage return or linefeed follows the string to the printer.

A new command DATASPEC allows manipulation of data within a CCTL file. See Section 4 of this document for further information.

The command SYMBOLSET has been added to allow LASER to use a symbolset other than that specified in the FONTINX file. The escape command ^ac__^ changes the symbol set immediately; the change goes away when another font call is encountered. The SYMBOLSET retains the desired symbol set on a change of font. However, symbol sets can only be changed in this way for internal printer fonts. The command is ignored for downloaded softfonts.

If SPSAVE is included in the INFO string, the save flag is turned on for the output spoolfile written by LASER.

The VMARK command allows a second number to set the mark to a vertical location in inches (or centimeters). Without the number the mark is set to the current cursor location.

The CCTL codes %302-%307 now space to VMARK2 through VMARK7. Usage corresponds to "skip to channel."

IF LAST can be tested in a footing. The result is true if the footing is being printed on the last page of the document.

The OUTFILE name can be specified in the INFO string as four (or fewer) alpha-numeric characters followed by a number sign ("#"). LASER will create a unique output file name. See Section 4 of this document for further information.

Two new INFO string commands ECONON and ECONOFF set the economy mode on the printer for the specific job.

FEEDNEXT MACRO x where "x" is any letter uses the string stored as macro "x" at the end of the current page. That string needs to trigger a page throw by the printer. If the first character of the macro is a caret ("^") it is changed to an escape.

The `FEED` and `FEEDNEXT` have new options to specify paper types for those printers that recognize the strings. See Section 4 of this document for further information.

The `AT` command is enhanced to work on a range. `\AT 1 to 2 DO MA` calls macro A at each line between inches one and two on the page.

The HP Variable `FANTFILENAME` is blank if `LASER` did not write a disc file; otherwise it contains the name of the output disc file written.

Running `LASER` with `PARM=20` generates an output file of type `VB`. It is a binary file with variable length records.

`ALTFONT` followed by a font number specifies the font to call when a shift out (octal 16) is found in the data. The font in use when the *shift out* was found is restored on a *shift in* (octal 17).

The new escape command `^de^` deletes characters to the end of the record in which it appears. This can be used to insert comments at the end of a text line as the `"*"` does for command lines.

The keyword `ERRJCW` followed by a legitimate `JCW` name in the `INFO` string causes `LASER` to set that `JCW` with the number of errors in the run.

Some printers allow paper source to be specified with an alpha name such as "punched", "envelope" etc. The set of recognized paper types is included in the `MESSAGES` file, records 1249-1268. The `FEED` and `FEEDNEXT` commands can be used with those alpha strings. If there is an alpha string after `FEED` (or `FEEDNEXT`) the string is compared to the records in `MESSAGES` with a non-case-sensitive compare for the number of letters entered. The first match found is used. Example: `FEEDNEXT punch`.

The `MAXICODE` bar code is implemented in `LASER`. Use of this bar code on UPS labels requires coordination with UPS. Contact LARC Computing for details.

`PDFMIN` and `SHOWPDF` are two new parameters for `LASER` that apply only with `PARM=97` (or `98`). The latter displays the `INFO` string passed to `OPENPDF` by `LASER`. The former eliminates any settings from `LASER` for that string other than the file names. If `PDFMIN` is used `LASER` will pass information from record 1269 of the `MESSAGES` file (See Section 4 of this document for further information.) and the file names, nothing else.

`SHOWFONT` is a new command that displays a list of all fonts specified by `FONTSET` commands to that point in the file.

3. Partial List of Problem Fixes

To the best of our knowledge all problems reported to LARC Computing have been fixed in the current release. The list below is not comprehensive. This is meant to call attention to problems that might have gone unnoticed at some sites. If you find a problem, or think of an enhancement that make make your use of Fantasia/3000 more convenient, you are encouraged to contact us. (Contact information is on the first page of this document.)

LASER has the capability to use TableHeadings when a TABLE extends over more than one page. The program had several problems. The last of them are thought to be fixed in the current version.

When LASER finds a token that is too large to fit the current column width it reports "Error 110 Line break forced at an unnatural place!" The program used to force a line break and split the token. It now prints on the current line allowing the output to extend past the right margin. Under some circumstances the previous solution caused LASER to loop; the new handling solves that problem.

BILDINX handles the index of fonts. It has a facility to edit that did not work. Modified records were not written back to the file. That has been fixed.

BORDERS2 is a font containing characters to use in fancy borders. It contained data for several characters with zero weight and no dots; those caused errors when the font was loaded to a D640 printer. The font has been fixed.

DBCHART did not run under MPE 5.5 or later. That has been fixed.

FSPCONV allows a search for Mapping Entries by Report name. When using that capability the report name is now upshifted and the "Next Entry" function key will cycle through all the mapping entries using the specified report name.

FSPPOOL retains the number of copies when using the PASSTHRU logic. (Previously it forced a single copy for this case.) It also retains the copy type (collated or uncollated) when the number of copies in the spoolfile was greater than the number configured. (Previously it used collated for that case.)

FSPPOOL when there are several mapping entries for one file and on of them, other than the first, used an "*" to specify the output device the output was sent to the device used by the prior mapping entry. That has been fixed.

When FSPPOOL began it looked at all spoolfiles on the disc at the History priority. If there were a large number of files, it could take a very long time and, under some circumstances get the program into a loop. FSPPOOL no longer checks spoolfiles at the History priority.

Whenever a softfont whose file name begins with the string MICR is called, the HMI is set to eight characters per inch. Previously that was only done if it was thought that output was going to specific LaserJet models.

If the BARCODE command is used with an empty string nothing is sent to the printer.

In LASER the setting of NOLEADZERO is ignored for bartype 9. That barcode scheme requires an even number of digits; the leading zero may be needed!

When selected pages are printed far down a long document, the printer can reset before the output arrived. That is avoided by sending a few nulls at the end of each not-printed page.

4. Additional Information on Selected Features

PDF Output

The `OPENPDF` program is available from Open Seas. It converts PCL files to PDF. PCL is the printer control language used on the LaserJet family of printers. PDF is the formatting language used by Adobe applications on the PC. `OPENPDF` is available through LARC Computing acting as agent for Open Seas in the United States.

The `LASER` program can be run with `PARM=97` to create a disc file, then call `OPENPDF` to automatically convert the output file to PDF format. The `OPENPDF` program must be loaded to the `OPENPDF.LARC` group for `LASER` to find it automatically.

If `LASER` is run with `PARM=98` the result is a bytestream file; the program `MGTOBYTE` is called to convert the output.

An output file name can be specified in the `INFO` string with the `OUTFILE` command. If no file name is specified, the output file name will be `PDFddd` where "ddd" is a three digit number calculated to provide a unique file name.

Record #1269 of `MESSAGES.LASER.LARC` contains parameters that are passed to `OPENPDF` by `LASER`. Consult the documentation for `OPENPDF` for interpretation of those parameters. Users can modify the record to tailor to their own circumstances. A dollar sign ("\$\$") must be the last non-blank character on the line; it marks the end of the string that is passed to `OPENPDF`.

Backing up the FSPPOOL DataBase

It is important to have a back-up copy of the data base used by `FSPPOOL`. Some sites do not bring down `FSPPOOL` before doing a regular back-up and they have no recent copy. `FSPCONF`, the program used to modify the data base, now offers an option to create a back-up tape at the end of the program.

`FSPCONF` asks: "Want to back-up the `FSPPOOL` data base to tape?" after closing the data base and the `VIEW` screens. Respond with "Y" (or "y") to create a `STORE` tape with the data base files. Note: A positive response will require an operator response and use of a magnetic tape.

Forms Drawn on a PC

Forms drawn on a PC must meet certain restrictions in order to work with Fantasia/3000. Until recently that required use of specific software on the PC. A new program in the Fantasia package now can reformat most forms designed with most PC programs.

Create your form with any program on the PC. Print to disk using a driver of your choice - but not a LaserJet 6, PCL6, or PostScript driver. Move the `PRN` file to the HP3000 with a binary transfer - there is no longer any need for the `FIXED` switch or concern over record size. Now process the HP3000 file with the `CLEANUP` program. Output from `CLEANUP` can be used with the `PCForm` command in `LASER`; it can also be processed by `FORMCONV` to catalog the form and create a file for use with the `FORM` command.

See the section below on Field Fill.

CLEANUP

From the colon prompt on the HP3000 enter:

```
CLEANUP.LASER.LARC; INFO = "<<filename>>"
```

where <<filename>> is the name of the file you have moved from the PC to the HP3000.

If `TESTINV` came from the PC entering the command

```
CLEANUP.LASER.LARC; INFO = "TESTINV" would cause the file TESTINV to be overwritten with the "cleaned up" file ready for use with LASER.
```

CLEANUP - more detail

The input file can be retained by passing two file names to CLEANUP:

```
CLEANUP.LASER.LARC; INFO ="infile outfile"
```

infile is the file passed from the PC; outfile is output from CLEANUP.

When PC programs print a form to disk the file includes PCL commands to reset the printer, set paper size, establish orientation, etc. If that file were used by Fantasia to overlay data on the form the data would appear on one piece of paper and the form on another. CLEANUP removes the PCL that would get in the way of normal Fantasia processing.

CLEANUP looks for a file named TABOO on the logon group/account. If the file is not there, it looks on LASER.LARC and if no file is found there CLEANUP will use its default list of PCL to be removed. CLEANUP reads TABOO until the end of the file or until it reaches a record that begins with *END. Blank lines are ignored.

The TABOO file provided with CLEANUP has a reasonably complete list of PCL. Codes before the *END record are those most reasonably removed.

Field Fill

PC forms programs that provide for fields do not pass information about the fields to the PRN file. There is an alternative way for users to locate fields on the form. Select a character to be used as a field indicator; it should be a character that does not appear elsewhere on the form. Obvious choices would be either the tilde ("~") or the caret ("^").

Using the form facility to enter text, and using a font that is built in to the printer (Courier is the best choice here) put that character in a text field at the appropriate location. This method will not work if the font is not recognized by the driver as an internal font. Each field indicator should be a separate text field on your form.

If fields are to be replicated down the form (as would be the case for line items on an invoice) only the first line of fields should be indicated on the form. It will be easier to edit the result to include additional lines.

Print the form to a disk file and move it to the HP3000.

Run CLEANUP. If input and output files are specified in the INFO string, include the field marker in quotes (single or double) in that string. For Example: INFO="PCFORM '~'" If the file names are entered in response to program prompts provide the field marker (in quotes) in response to the first prompt.

CLEANUP will ask for another file name; this file will contain the field specifications. The file will be an ASCII file that can be edited on the HP3000.

CLEANUP will also ask whether the resolution in the input file is at 300 or 600 dots per inch. Measurements in the file are in dots; measurements in the field specification file will be in inches. You need to provide the conversion factor.

Here is a sample of a field information file written by CLEANUP

```
\* Field data file from CLEANUP TUE, DEC 11, 2001, 12:11 PM
\PCFORM cely
\GOTO 0.45 0.46 ; font 03; just left; print "f001"
\GOTO 0.45 0.55 ; font 03; just left; print "f002"
\GOTO 0.45 0.63 ; font 03; just left; print "f003"
\GOTO 0.45 0.71 ; font 03; just left; print "f004"
```

CELY is the output file from CLEANUP. The field information file can be processed by LASER to print the form with fields indicated.

The field information file can be modified using an editor on the HP3000. Field start points can be changed, fonts can be changed, justification can be changed, and fields can be duplicated.

CLEANUPX will convert the field information file into a "form" with the field data that can be used with the PCFORM command in LASER.

Disc Output with a Pseudo Device

LINEPARM.LASER.LARC maintains parameters that describe specific logical devices. In previous versions of Fantasia it was possible to enter a TCP-IP address in a LINEPARM field. With that entry the logical device was a *stand in* for a printer on a LAN. In other words, that logical device was a pseudo device. That capability is still in the software.

However, with MPE/iX versions 5.5 and later HP has introduced spooled LAN printers. The HP method is similar to that used by Fantasia. An MPE table associates the TCP-IP address with a logical device number. Output directed to that LDEV goes to the LAN printer through the spooler. Details can be found in the HP manual on the spooler.

With the HP capability to handle LAN printers directly, Fantasia users are strongly urged to use the MPE facility for this purpose rather than the Fantasia construct.

An alternative use for the LAN Address field in LINEPARM exists in the latest release of Fantasia. If that field contains PARM97 or PARM17 output directed to that LDEV number by Fantasia will be written to a disc file. The string must begin in the first column of the field and the alpha characters must be upper case.

PARM17 generates a PCL output file with variable length records. The file name is of the form LASPdd where dd is a two digit number used to make a unique file name. (See the discussion below on OUTFILE Enhancements.)

PARM97 works only if the OPENPDF option is available. (See the entry above on OPENPDF.) The output file is a PDF file. The file name is of the form PDFddd where ddd is a three digit number used to make a unique file name. (See the discussion below on OUTFILE Enhancements.)

FFCONV Enhancements

FFCONV was part of FORMCONV; while it continues as part of FORMCONV it is now documented as a separate program. Some preliminary explanation of FORMCONV will help in understanding the new capability.

Forms can be developed on a PC and transferred to the HP3000 for use with Fantasia. When using REFLECTION to transfer the file use a BINARY transfer and specify ;FIXED after the host file name on the transfer screen. The host record size should be 256 bytes. The result on the HP3000 should be a file of type FB with 128 word records.

Compiled forms from JetForm Design are not compatible with Fantasia programs other than FORMCONV or FFCONV. Either of those programs will convert a file into a form used by LASER.

When FORMCONV processes a file from Design it converts it to the format required by the PCFORM command in LASER. FORMCONV can also write an output file and enter the form into the catalog. If this option is used the output file can be used with the FORM command in LASER.

FFCONV can be run directly. If FFCONV is run with an INFO string containing a file name it converts the file by that name into the correct format. The file name is not changed; the new file overwrites the old. If FFCONV is run with no INFO string it asks for names of the input and output files.

To restate the situation: There are three formats for forms files, call them F1, F2, and F3. Files come from the PC in format F1. An attempt to access a file of type F1 with LASER results in an error message.

FFCONV converts files from format F1 to format F2. Generally the file of the new format overwrites the original.

FORMCONV will accept files of either F1 or F2 format. If the input is in format F1, FORMCONV runs FFCONV to change the format. If FORMCONV writes an output file it is of type F3.

Use the FORM command in LASER to access a file of type F3. Use PCFORM to access a file of type F2.

FORMCONV catalogs forms and controls the generation of new forms. Those features and controls are used at many sites. Forms used with the FORM command are stored in the printer and recalled from printer memory when used multiple times within one document. Depending on the form, the printer, and the communication between HP3000 and the printer, that may be more efficient than use of the PCFORM command and format.

Calling the Editor from FSPOOL

FSPOOL generally runs LASER providing it with an environment file name, spoolfile name, and output instructions in the INFO string. This allows the spoolfiles from existing applications to be printed on forms. It is also possible to have FSPOOL use the LARC Editor to modify the file before printing it.

Using FSPCONF, create a mapping entry noting:

1. The FANTASIA Environment file field begins with an asterisk ("*") to signify that FSPOOL should call the Editor to process the spoolfile rather than using LASER.
2. The fields which normally control FSPOOL/LASER output are ignored. These include L/P, Device, Priority, Copies, Copy Type, and Banner. All control over the output is contained in the Editor USEFile.

If the spoolfile picked up by the mapping entry is O387, and the environment file specified on the mapping screen is *DOEDIT.TEST.MARTIN then FSPOOL will run EDITOR.EDIT.LARC and pass it the INFO string:

```
SET FSPOOL; T O387.OUT.HPSPOOL,unn; U DOEDIT.TEST.MARTN; exit
```

As an example assume:

- a. The spoolfile contains a letter.
- b. The LARC name is to appear in a bold font everywhere in the letter.
- c. The file called LETRHEAD contains formatting commands to put letter head on the first page, set up page headings on other pages, define fonts, etc.
- d. Output is to be printed on logical devices 106 and 51.

Then the DOEDIT file could contain these lines:

1. c 1, ', all
2. al .1,\IN LETRHEAD.PUB.LARC
3. change 'LARC','^fb^LARC^fr^', all
4. laser @ dev 106
5. laser @ dev 51

Line 1 eliminates the first character on each line (the carriage control character). Line 2 adds a line at the front of the file. Line 3 changes the string LARC to have font calls before and after it. Lines 4 and 5 use LASER to print the output on the desired printers.

This is a simple example. A real use might: move data around in the file; or insert a DATA command and field identifiers for a field fill form; or print one copy of the output, then remove data that should not appear on the second copy and print another; etc.

The USEFile must use LASER or another command to print the file. FSPOOL will not provide output. Do not KEEP the modified file.

After processing FSPOOL moves the spoolfile from the candidate queue to the History queue and retains it for the period specified. If there is an error processing with the Editor the purge time goes to NEVER.

The Editor parameter FSPOOL is set in the INFO string so the Editor will end without aborting if there is an error. It also allows an exit from the Editor without keeping the workfile.

See the paragraphs below on Editor Stored Strings for new features relevant to USEFiles.

Editor Stored Strings

The EDITOR has ten strings called Z0, Z1, . . . Z9 that could be set and retrieved. See EDITDOC.EDIT, the Editor User Manual for more information on their use.

It is now possible to set one string to the value of another and to test the values for branching within a USEFile.

SET Z1=Z2 copies the value from Z1 into Z2.

SET Z3 = 3, 5 copies the five characters in the current workfile line beginning with the third character on the line into the Z3 string.

As an example of the use of this facility, suppose the workfile contains information on several units; blocks of several hold an identification number. You need to force a new page when the identification number changes. You have positioned the cursor to the record with the identification number and want to insert \NEW ahead of this record if the number is different. The number is in columns 5-14 (ten digits). Here is a portion of the USEFile to handle that editing:

```
SET Z1 = 5, 10          << sets Z1 to the current value >>
@IF Z1 = Z2 GO TO L2   << skip the next lines if equal >>
AL *-1, \NEW          << add the \NEW one line back >>
SET Z2 = Z1           << this is now the current ID >>
@L2                   << carry on with editing >>
```

User Defined Patterns

The range allowed for the PATTERN command is from one to twenty. Patterns 1-6 refer to the HP-Defined patterns as shown in the HP Manuals. Patterns 7-20 are for user-defined patterns. User defined patterns are kept in a separate file and sent to the printer using the GRAPH command.

The first characters of that file assign a number to the pattern; the five characters <esc>*c7G set the pattern number to 7. The six characters <esc>*c17G assign 17. That is followed by the pattern.

Here is an example to download a pattern and use it to fill a rectangular area:

```
\graph shade2.laser.larc
\pattern 8
\fillboxy 2
```

SHADE2.LASER.LARC is the file identifying the pattern to be downloaded as number 8 and containing the pattern. See the LaserJet documentation for construction of that file.

PATTERN 8 uses pattern 8, a user-defined pattern, for filling with pattern. FILLBOXY 2 fills a box two inches high with the currently established pattern.

The DATASPEC Command

DATASPEC is recognized for CCTL files. It is used to provide specification for the setting of a binary flag or a macro. The flag is tested, or the macro called, in the footing or with a USE construct called with the AT. The set of constructs allow enhancement of the output.

FCCTL is "fake CCTL" and allows use of DATASPEC with a non-ctl file. A space is inserted as the "carriage control" at the front of each record and the file is treated as if it were a CCTL file.

The two major applications are to provide barcoded fields within a document and to output checks with check numbers in MICR and digitized signatures. for CCTL files. DATASPEC syntax requires that paramters be entered in the specified order. Examples are almost self-explanatory:

```
\DATASPEC LINE 26 COL 71 LEN 5 FLAG01 VALUE > "9999"
\DATASPEC LINE 23 COL 52 LEN 4 FLAG02 STRING = "VOID"
\DATASPEC LINE 26 COL 72 LEN 8 MACRO A
```

In the first example, LASER computes a value based on numeric characters beginning in column 71 of the 26th line of output on the page. If that value is greater than 9,999 FLAG01 will be TRUE; if it is less than, or equal to, 9,999 FLAG01 will be FALSE.

The line referred to in DATASPEC is the line on the output page. The MONITOR command helps to locate strings by output line. When MONITOR is on LASER reports the result of setting flags and/or macros.

The second example compares the string in columns 52-55 of output line 23 with VOID. FLAG02 is set TRUE if the string is there; it is FALSE otherwise.

The third example sets the macro MA to the string in columns 72-80 of output line 26. If "MACROX" is used in place of "MACRO" all leading and trailing blanks are eliminated; if "MARCOY" is used leading zeros replace leading blanks.

Here is an example of a complete environment file to print checks. The stub is at the top of a page with the check at the bottom. VOID is used when the stub requires two pages; in that case neither signatures nor MICR is printed at the bottom of the check. The check number is picked up from the file and reprinted in the MICR line at the bottom. If the check amount is less than \$5,000 two signatures are printed, otherwise only one signature is printed and the other must be signed manually.

```
\dataspec line 24 col 63 leng 7 flag01 value >= "0005000"  
\dataspec line 19 col 72 leng 6 macro a  
\dataspec line 25 col 2 leng 4 flag02 string = "VOID"  
\pageline 44  
\head begin  
\pcform check  
\space to 3  
\head end  
\foot begin  
\if flag02 skip 9  
\inleft 4.6  
\space to inch 5.55  
Signature One  
\space to inch 6.1  
\if flag01 skip 1  
Signature Two  
\space to inch 6.6  
\inleft 1.7  
^f46^:^ma: ;022000839; 32701 004254 4:  
\foot end  
\footsize 0  
\ctl :
```

Please contact LARC for additional examples or assistance in setting up your own use of this capability.

OUTFILE Enhancements

The OUTFILE command is used in the INFO string to name the output file written by LASER.

If output is written to a spooled print file the OUTFILE name is used as the report name associated with the spoolfile.

If OUTFILE is used with PARM=97 to generate PDF output, the file name specified is used for the file written by OPENPDF.

The string specified with the OUTFILE command must follow the MPE file naming conventions. It must begin with an alpha character, contain only alpha and numeric characters, and be no more than 8 characters long. Specification of group or group/account is not permitted.

The string can consist of four or fewer alpha characters followed by a number sign ("#"). In this case LASER builds a unique file name by appending a number to the string. Repeated runs of LASER with OUTFILE=TEST# will create files named, TEST001, TEST002, TEST003, TEST004, TEST005, etc.

The variable `FANTFILENAME` is set to the name of the output file written by `LASER` at the end of each run. If `LASER` does not write a disc file, that variable is set to blanks. The MPE command `SHOWVAR FANTFILENAME` shows the value of the variable. The MPE command `SHOWVAR FANT@` displays all the MPE variables set by Fantasia/3000.

FEED & FEEDNEXT Enhancements

Some LaserJet printers have a capability to specify the input tray by type of paper. The PCL command allows a call for "prepunched" paper, "card stock", etc. The `FEED` and `FEEDNEXT` command have been augmented to provide those options.

Key words are stored in the `MESSAGES` file in records 1249-1268 inclusive. That allows for 20 options; not that many are used yet. The entries in `MESSAGES` can be changed to suit conditions at specific installations. Be sure the string that the printer expects is exactly correct in the file; the keyword must begin in the first column and must be terminated with a dollar sign ("\$"). File entries are case sensitive since that is sent to the printer as it appears in the file.

Command syntax is: `FEEDNEXT word` where "word" indicates the paper type. You can abbreviate the paper type. `LASER` checks your entry against the file and selects the first record for which the input matches the entry in the file for the lessor of the number of characters in the record and the number entered. The command entry is not case sensitive. `LASER` matches "PLA" to "Plain."

Examples:

```
\FEEDNEXT plain
\feed PLA
\feednext color
```

The paper types specified in the `MESSAGES` file as provided with files delivered after October, 2000 are:

Plain	Bond	Card Stock	Prepunched
Color	Labels	Preprinted	Transparency
Recycled	Letterhead	Custom	

Printing the Euro Symbol

The methodology for handling fonts and symbols on LaserJet printers is remarkably robust and flexible. One problem, not anticipated, that has now arisen is a new symbol needing to be used with old fonts and printers. Using the "Euro" monetary symbol with current laser printers and software is not convenient.

`LASER` has three solutions. Any site can use any or all of them for various needs.

A scalable softfont is included on the Fantasia update tape. It contains the Euro character in place of the lower case "e". The escape sequence `^eu^` calls that font in the appropriate point size. The font is downloaded to the printer on the first use of the sequence. This does not work with printers that do not deal with scalable fonts. Example: `e1,234`

Newer Hewlett-Packard printers include a new symbol set. The new symbol set is called HP Roman-9. The Euro symbol is at character position 186 (octal 272, hex BA). You can switch to that character set using the `^ac19^` escape sequence. The command `\SYMBOLSET dd` sets the symbol set to be used on all softfonts overriding the symbol set defined in `FONTINX`. (Some HP compatible printers also have that symbol set included. If you are using a non-HP printer, try it.) If this is printed on a printer with HP Roman-9 the Euro shows up here: €45.

Three non-scalable softfonts are included on current Fantasia update tapes: `EURO08`, `EURO10`, and `EURO12`. Each has the Euro symbol as "e"; any or all of these can be installed in your font index with `BILDINX`. You can then call the font to print the Euro character.

The fonts are supplied by Hewlett-Packard without charge. An HP document on printing the Euro symbol can be found on the web at: www.hp.com/cposupport/printers/support_doc/npl01069.html

5. General Introduction to Fantasia/3000

This is an overview of the Fantasia/3000 software. At those sites where Fantasia has been working well for a long time those who set up the system may have left and/or forgotten what they did. This will be a reminder for some and an introduction to Fantasia for others. This should help you understand the process, play a more active role in fixing site-related problems and make changes when necessary.

All Fantasia/3000 programs and documentation reside in the LARC account. DOCUMENT.LASER.LARC lists files in the account with a brief descriptions of each. The PUB group holds files particular to your site. The LASER, EDIT, and SPOOL groups hold files provided as part of the software.

There are two major programs in Fantasia. LASER is the formatting program. It reads one or more input files and writes a file for a LaserJet printer. Output from LASER can go directly to a "hot" printer on the HP3000, to a spooled printer, or to a disc file.

FSPPOOL is the second major program. It is not used by every Fantasia/3000 site. FSPPOOL runs continuously in a JOB on the HP3000. FSPPOOL is the program that formats and prints output from application programs. Some set-up and customization is required for FSPPOOL to function at your site. The FSPPOOL set-up will be different for every site. FSPPOOL is the primary focus of this document.

FSPPOOL works from tables that identify which spoolfiles it should process. When it finds an appropriate spoolfile the mapping information specifies an environment file with formatting instructions. FSPPOOL then runs LASER with an INFO string containing file names, output destination, etc.

Every HP3000 job has a STDLIST output file. The STDLIST file is critical if there is any problem with FSPPOOL; it contains error messages and a complete record of the work done or attempted by FSPPOOL.

The FSPPOOL job should be stopped for a system backup on a regular basis. The IMAGE database with configuration information will not be backed up if the job is running.

FSPPOOL must be stopped in an orderly manner. Do not abort the job. Run FSPCTRL.SPOOL.LARC;INFO="STOP" to stop the job. If FSPPOOL is aborted files used for signaling and messaging between processes can get fouled up. If that happens, STREAM FILESET.SPOOL.LARC to clear the relevant files.

Find the FSPPOOL job with SHOWJOB JOB=@J. That gives a list of all running jobs. One entry in the list should look like:

```
#J3      EXEC      10S LP          MON 10:32A  JFSPPOOL,OPERATOR.SYS
```

The first token on that line is the job number. In the example above it is job number 3.

Enter the command SHOWOUT JOB=#J3 to see the output files for FSPPOOL. Except: use the correct job number for FSPPOOL. One of the lines in the output from that command should look like this:

```
LP      #01043      #J3      $STDLIST OPENED      256      8      1
```

The second token on that line identifies the output file name. In the example above, the STDLIST from FSPPOOL is in the file 01043.OUT.HPSPOOL. The first character is the letter "oh"; that is followed by a number from 1 to 7 digits long. Help for an FSPPOOL problem frequently requires some, or all, of the STDLIST file. That file can be copied or texted into an editor while FSPPOOL continues to run. If the problematic run of FSPPOOL has already stopped, the STDLIST file should be retained until the problem has been solved.

Starting and Stopping

If FSPPOOL is not running, start it with: RUN FSPCTRL.SPOOL.LARC;INFO="START"

If FSPPOOL is running stop it with: RUN FSPCTRL.SPOOL.LARC;INFO="STOP"

Either command can be used at any time without causing problems. If there is an attempt to start[stop] FSPPOOL when it is [is not] running, the FSPCTRL program returns a message to that effect.

Do not stop the program by aborting it unless that is absolutely necessary and FSPPOOL will not respond to the normal stop command. It may take several minutes for the job to end after FSPCTRL has run.

More background:

Spoolfiles are output files written by a program on the HP3000 and intended for a printer. Rather than send them directly to the printer, the operating system writes them to the disc. The spooling system of the HP3000 manages the printing.

Spoolfiles have a number of characteristics. One parameter of interest to this discussion is `PRIORITY`. The priority on the spoolfile governs if and when it will be printed.

Each spooled printer on the HP3000 has an `OUTFENCE`. When you issued the `SHOWOUT` command the output included the outfence value. If there are different outfence values for different printers on your system they will be listed by logical device number.

Each spoolfile is directed to a specific printer; if the priority of the spoolfile is lower than the outfence on its intended printer the file is not printed.

`FSPOOL` picks up all the spoolfiles with priority equal to the `CANDIDATE PRIORITY`. That must be less than the outfence for the intended printer.

Suppose the outfence for your printer is set to 1 (one). Output for your invoices is sent to the printer. `FSPOOL` is suppose to put a form on the page and print the data in landscape orientation. However, with the outfence set to one, the system spooling prints the file before `FSPOOL` can handle it. Your output appears on the printer without the form and in the wrong orientation.

Now suppose the outfence is set to 12 and `FSPOOL` is set to pick up files with priority 3. `FSPOOL` prints its output at priority 9 and the application writes its output with priority 3. `FSPOOL` will pick up the output and format it but nothing will print. If you now drop the outfence to 1, the output from `FSPOOL` will print well formatted. However further output from the application will also print and that will not be formatted.

`FSPCONF.SPOOL.LARC` is the program that displays and modified the tables which govern `FSPOOL`.

6. FSPool Efficiency Note

When a large number of printers are configured as input devices for FSPool the process of selecting spoolfiles for the Candidate list may be inefficient. A site can significantly improve the use of system resources.

FSPool creates a Candidate selection command file based upon the Input Devices Configuration. That file is used as input to the MPE Command Interpreter to create the list of spoolfiles to be processed.

Improved performance may be achieved by creating a custom Candidate selection command file. When all Input Devices use the same Candidate Priority it will be particularly helpful to create your own file.

The file must be named FSPCSELQ.PUB.LARC

If this file does not exist FSPool creates it (normal processing). If the file does exist FSPool uses it.

Following is a sample FSPCSELQ file. Line numbers are used in the description that follows - the file must be unnumbered.

```
1) file SPOOFLES; acc=APPEND
2) listspf O@; seleq=[(dev=lp1) and (pri=3) and (state=READY)];
   detail > *SPOOFLES
3) listspf O@; seleq=[(dev=lp2) and (pri=3) and (state=DEFER)];
   detail > *SPOOFLES
4) listspf O@; seleq=[(dev=lp3) and (pri=3) and ((state=READY)
   or (state=DEFER))]; detail > *SPOOFLES
5) listspf O@; seleq=[(dev=LP4) and (pri>1) and (pri<8) and
   (state=ready)]; detail >* SPOOFLES
6) listspf O@; seleq=[(dev=lp5) and (pri>1) and (pri<8) and
   (state=defer)]; detail >*SPOOFLES
7) listspf O@; seleq=[(dev=lp6) and (pri>1) and (pri<8) and
   ((state=ready) or (state=defer))]; detail >*SPOOFLES
8) reset SPOOFLES
```

The sample assumes the OUTFENCE is 8 and the candidate priority is 3.

Lines (1) and (8) are required. FSPool expects the output from the Candidate selection to be in a file names SPOOFLES.

Line (2) is the "standard" command and is used when a specific Candidate Priority is configured and the "MPE/iX Defer Control" field in the Run-Time Environment is set to "X." Line (5) is the variation used when the Input Device is configured with Candidate Priority set to "@"

Line (3) is used when the "MPE/iX Defer Control" field is set to "O"; line (6) is the variation when the Candidate Priority is set to "@"

Line (4) is used when the "MPE/iX Defer Control" is set to "I"; line (7) is the variation.

If all spoolfiles at one candidate priority are to be processed by FSPool regardless of the logical device on which they reside FSPCSELQ.PUB.LARC contains one LISTSPF record (in addition to the required FILE and RESET).

```
listspf O@; seleq=[(dev=@) and (pri=3) and (state=ready)]; detail > *spoolfiles
```