Hermann Schmitt
Dipl.Volksw. Dipl.Math.

Tel.049 02204 69078
Siegenstr.35
5060 Bergisch Gladbach 1
Germany
05.06.95

User's Guide to the PostScript Forms System.

General Description

The purpose of the system is to create forms of any kind. All features of standard graphics software can be used in the forms. The forms are created with a standard graphics program.

The use of standard graphics software is advantageous, because there are efficient tools available, and all graphics elements , which are offered with the graphics software, can be used directly. In addition there is no special knowledge necessary to create the forms, because it can be assumed, that the graphics software is already used for other purposes. Specific in creating forms is only, that parts of the graphic are interpreted as variable fields, into which text strings are put when forms are printed. The fields contain special characters by which they are discernible and they contain names by which the text strings can be assigned to the fields. The attributes of the variable text, e.g. type and size of the fonts, are fixed, while the form is designed with the graphics software, only strings of ASCII characters have to be put into the forms afterwards.

Probably most important is the insertion of text strings from files. Those files are called varfiles in the following description. It is assumed that the a varfile contains a number of instances of the same kind and a form is to printed for each of the instances. Examples of those instances are data concerning persons or data concerning products.
The varfiles can be structered in three ways:
- Each line of the varfile contains the data for one instance. The fields have fixed positions in the line.
- Each line of the varfile contains the data for one instance, the fields in the line are separated by a separator character, which is the same for all the file.
- Each line of the varfile contains one field of an instance. At the beginning of the lines containing the data for one instance there is a separator line.

Besides from varfiles text strings to be inserted into the forms can come from other sources:
- Values constant for all instances can be inserted.
- Values constant for one run of the form print program can be inserted. An example is the date of invoices.
- A powerful but relatively difficult possiblity is the insertion of text strings obtained by the invocation of functions. All functions written in the programming language "C" are usable. It could be standard functions available or functions specifically written for a form. For instance the date of invoices could also be fetched from the computer. The functions can also use fields of the varfile as parameters. For instance with quantity and price fetched form the varfile the amount could be calculated or fields of the varfile in unprintable format (e.g. binary format) could be transformed into text strings.

The form print programs are created by a general generation program. Input to the general generation program is the form, which is created by printing the graphic as PostScript data stream, and a print specification file, which specifies , how the values from the varfile and values from other sources are to be inserted into the form. These specifications are called assignment specifications. The connecting link between the text string to be inserted and its position in the form is the assignment specification which contains the name of the variable field in the form. The print specifications additionally contain directives, which set, how variable fields in the from are to be identified, which separator character is used if applicable and which fill character is to be applied, if the text string to be inserted into the form has less positions than the field in the form. These directives are called print directives. There exist default specifications for the print directives.

The general generation program creates a source in the "C" programming language, which is compiled and linked.
The generated form print program reads the lines of the varfile and prints the completed forms into a file.

Hermann Schmitt
Dipl.Volksw. Dipl.Math.

Tel.049 02204 69078
Siegenstr.35
5060 Bergisch Gladbach 1
Germany
05.06.95

Detailed Description of the Application of the PostScript Forms System.

Description of the Specifications for the PostScript Forms System.

The Print Declaratives.

FieldCharacters = "BFE"
B: Begin Character
F: Fill Character
E: End Character

Each varable field in the graphic begins with the begin character, then the variable name follows and it ends with the end character. The variable field in the graphic must have the same number of positions as the positons planned in the form. Therefore fill characters may have to be inserted between the name and the end character.
The default field characters are: "{ }".
The field characters have to be chosen in such a way, that they do not appear elsewhere either in the fixed text to be printed or in the variable fields except as field characters.

ShortNames="..","."....
Variable fields, which two positions or one cannot be structured as described under "FieldCharacters". In these cases one or a pair of characters have to be selected, which identify the field and are also the name of the field. The short names have to be chosen in such a way, that they do not appear elsewhere either in the fixed text to be printed or in the variable fields.
These short names have to be specified in the declarative described here.

SeparatorCharacter = "y".
For type 6 of input specifications a character is used to separate the fields in a line. This character can be set in this declarative.
The default separator character is ";".

FillCharacter = "z".
The number of positions of the field in the form may be larger than the number of positions of the text string to be inserted. If this is the case the the fill character is inserted in the remaining positions.
The default fill character is blank.

The Assignment Specifications.

All assignment specificatins have the same form:

Input Specification, Output Specification

The assignment specifications differ in the input specification. The output specifications always have the same form:

The Output Specification:

name[,[{l,r,c}][,"z"]]
"name" is the a name specified in a variable field in the graphic. Names are "case sensitive".

Hermann Schmitt
Dipl.Volksw. Dipl.Math.

Tel.049 02204 69078
Siegenstr.35
5060 Bergisch Gladbach 1
Germany
05.06.95

The number of positions of the variable field may be larger than the number of positions of the text string to be inserted. {l,r,c} specifies, how the text string is to be aligned in the variable field of the form: left(l), right(r) or centered(c) and the character "z" is to be inserted into the remaining positions. The default alignment is "left", the default fill character is the fill character specified for the form in general by the print declarative "FillCharacter".

The Input Specifications.

1, position, lenght
"1" is the type of the input specification: All fields of an instance are contained in one line of the varfile at fixed positions. Position means the positon of the first character of the text string in the line, length is the length of the text string in the line.

6, index
"6" is the type of the input specification: All fields of an instance are contained in one line of the varfile and the fields are separated by a separator character (see print declarative "SeparatorCharacter").

9, index
"9" is the type of the input specification: Each field of an instance is in a separate line of the varfile. At the beginning of the lines belonging to one case there is a line beginning with "[" and ending with "]".

Hint: Only one type of the input specifications 1,6,9 can be used for one varfile.

2, "constant".
"2" is the type of the input specification: A constant is to be inserted into the form. The constant which is to be inserted follows separated by a ",".

4, name
"4" is the type of the input specification: A text string is to be inserted into the forms, which is constant for all instances of one run of the form print program.
The text strings are read in from a file at the beginning of the form print program.
This file consists of lines, in each of which there is one expression of the form:
name = "textstring".

3, function
"3" ist the type of the input specification: The function in the input specifiction is invoked and the result is inserted into the form. The result must be a text string. The "function" may consist of several functions.
In the function fields of the varfile can be parameters by using one of the following functions:
- f_pos(positon, length)
  for a varfile of type 1.
- f_index1(index)
  for a varfile of type 6
- f_index2(index)
  for a varfile of type 9.

If functions written by the user are to be introduced, they are to be defined in the following declarative:
Functions =
[
.....
.....
]

The Print Specification File.

Hermann Schmitt
Dipl.Volksw. Dipl.Math.

Tel.049 02204 69078
Siegenstr.35
5060 Bergisch Gladbach 1
Germany
05.06.95

The print directives, the assignment statements and the functions declarative are stored in one file: the print specification file. In this file the print declaratives must come first then the assignment statements and the functions declarative must be at the end.
Before, between and at the the end of the specification lines there may be comment lines.
Comment lines have the following form:
/* comment */

The Execution of the Generation Program and the Form Print Programs

The Generation Program

The following command invokes the generation program and a C-Compiler, which compiles the generated source of the form print program and links it with the object-files delivered.

PSFORM name [formfile specfile]

The generated form print program will be named automatically as: 'name'.exe If there occur any errors during the generating process, the corresponding error messages will be stored in an error file 'name'.err . The formfile has to contain the form as Postscript datastream. The print specifications (as described above) are located in the specfile.

There is no need to declare all filenames, only the first one 'name' is necessary. In this case all files must have the same name with different extensions, and must be in the same directory. The extension of the formfile has to be '.PS' and the extension of a specfile '.SPE' . Thus, if 'name' is the only parameter, the program is looking for the files 'name.ps' and 'name.spe' .

The Form Print Programs

The executable form print program NAME, is to be called it as follows.

NAME varfile printfile [constfile]

Contents of the varfile is the variable data (as described above). The Postscript stream of the form(s) filled with variable data is the printfile. This file can be printed with a PostScript printer. If LPTn is specified instead of a file, the forms can be printed on a printer immediately. "constfile" is only to be used with input specification "4".