CHAPTER

17

The EXPLODE Procedure

Overview

The EXPLODE procedure produces printed output with oversized text by expanding each letter into a matrix of characters. You can use the EXPLODE procedure to generate posters, flip charts, and header pages for computer output.

Note: PROC EXPLODE with a PARMCARDS statement cannot be included in a macro. △

Output 17.1 on page 415 shows the results of the most basic form of a PROC EXPLODE step with only one line of text. The following statements produce the output:

```
options nodate pageno=1 linesize=80
        pagesize=60;

proc explode;
   parmcards;
 TOP SECRET
;
```

Output 17.1 A Line of Expanded Text

```
                          The SAS System                          1
*****   ***   ****       ***   *****   ***   ****   *****  *****
  *    *   *  *   *     *   *  *      *   *  *   *  *        *
  *    *   *  *   *      *     *      *      *   *  *        *
  *    *   *  ****        *    ****   *      ****   ****     *
  *    *   *  *             *   *      *      * *    *        *
  *    *   *  *           *   *  *      *   *  *  *   *        *
  *     ***   *           ***   *****   ***   *   *  *****    *
```

Through options you can control spacing, the density of the text, and underlining.

Procedure Syntax

Requirements: PARMCARDS or PARMCARDS4

Message line(s)

Null statement

Reminder: You can use global statements with PROC EXPLODE. See Chapter 2, "Fundamental Concepts for Using Base SAS Procedures," for a list.

PROC EXPLODE;
PARMCARDS|**PARMCARDS4**;
message-line(s)
;|;;;;

PROC EXPLODE Statement

PROC EXPLODE;

PARMCARDS or PARMCARDS4 Statement

Signals the beginning of the message lines.

Requirement: If any part of the message contains a semicolon, you must use PARMCARDS4.

See also: "Null Statement" on page 418

Featured in: Example 1 on page 419 and Example 2 on page 420

PARMCARDS|**PARMCARDS4**;

Message Lines

Specifies the block of text (one or more lines) and any special characters that control the appearance of the text.

Featured in: Example 1 on page 419 and Example 2 on page 420

Message line(s)

<D | L>

<S*n* | P>

<spacing-control>

text

<U *character-1* <...*character-n*>>

. . . more blocks of option specifications and text lines . . .

<D | L>

<S*n* | P>

<spacing-control>

<U *character-1* <...*character-n*>>

Required Argument

text
specifies the line of printed text. It can contain only the following characters:

ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890

. – + , = * $ / _ ( )> < | & ’? ! ; # ¬ " % @ blank

The not symbol (¬) can also appear as either a hat (^)or a tilde (~) depending on your keyboard. PROC EXPLODE ignores lowercase characters.

The EXPLODE procedure reproduces horizontal spacing as it appears in the program, except for column 1, which is reserved for the *spacing-control* option.

Restriction: *text* can begin in any column except the first.

Options

To do this	Use this option
Control vertical spacing	S*n* or *spacing-control*
Control the text density	
Specify dark characters	D
Specify light characters	L
Underline text	U
Begin a new page	P

D | L
controls the density of printed characters. Specify D to produce dark characters that are formed by overprinting the characters H, T, and Q. Specify L to produce light characters that are formed of asterisks.

Default: L initially, then for each line of text the value is carried over from the previous line if you do not specify a value.

Requirement: Must appear in column 1, and must be the only character on that line.

Requirement: To produce overprinting, the SAS system option OVP must be in effect, and your printer must support overprinting.

Featured in: Example 2 on page 420

L
See D | L.

P
See S*n* | P.

S*n* | P
controls the amount of space before the next line of text.

S*n*
skips *n* lines before the next line of text.

Range: 1–9

See also: *spacing-control*

Featured in: Example 1 on page 419

P
begins a new page before the next line of text.

Featured in: Example 2 on page 420

Default: 0

Requirement: Must begin in column 1 and must be the only characters(s) on that line.

spacing-control
specifies the number of lines to skip before the next line of text.

Default: 0

Range: 1–9

Requirement: Must appear in column 1.

Restriction: Spacing control does not work at the top of the page.

See also: S*n* option

<U *character-1* <...*character-n*>>
underlines the *text* on the previous line with asterisks. The *character* values can be anything. The nonblank characters determine where the underline appears. PROC EXPLODE skips two lines before printing the underline.

Featured in: Example 2 on page 420

Null Statement

Ends the PROC EXPLODE step.

Requirement: The Null statement must begin in the first column. If any part of the message contains a semicolon, use four semicolons instead of one.

See also: "PARMCARDS or PARMCARDS4 Statement" on page 416

;|;;;;

Examples

Example 1: Controlling Spacing

Procedure features: PARMSCARDS statement

Message lines options: S

spacing-control

This example

- controls horizontal spacing in the output by shifting the starting point of the text lines in the program
- controls vertical spacing with an initial gap of two lines and another gap of two lines before the second line of text.

Program

```
options nodate pageno=1 linesize=88 pagesize=60;
```

PARMCARDS= specifies the file reference, EXTFILE, of the file, PARMFILE, to which PROC EXPLODE writes the text in the message lines.

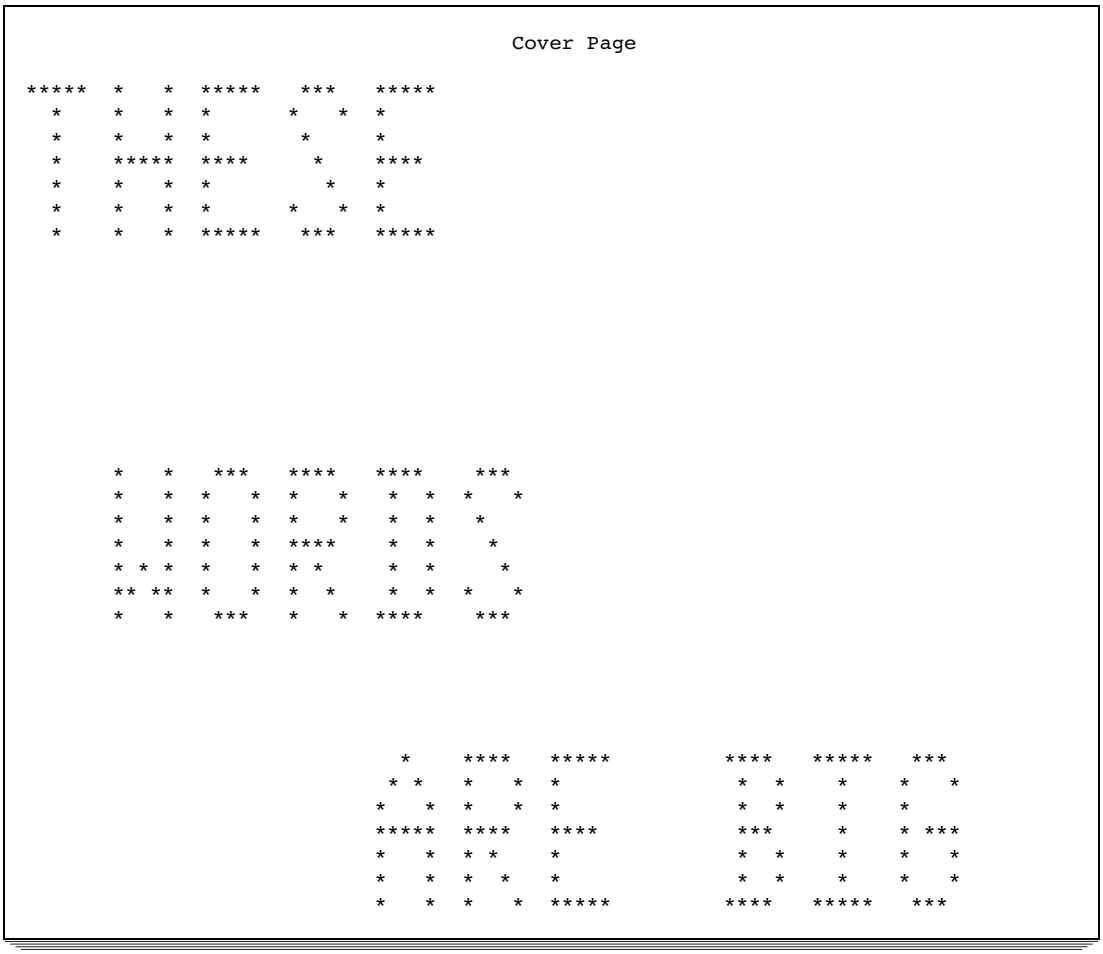
```
options parmcards=extfile;
filename extfile 'parmfile';


proc explode;
   title 'Cover Page';
```

The numeral 6 before **WORDS** specifies the spacing control. S2 skips two lines before the next line of text.

```
   parmcards;
 THESE
6 WORDS
S2
     ARE BIG
;
```

Output



Example 2: Darkening and Underlining Text

Procedure features: PARMSCARDS4 statement
Message lines options: D
L
P
U
SAS system option: OVP

This example
- prints dark text and then returns to light text
- specifies a page break
- underlines text.

Program

OVP allows overprinted characters in the text.

```
options nodate pageno=1 linesize=88 pagesize=60 ovp;
```

PARMCARDS= specifies the file reference, EXTFILE, of the file, PARMFILE, to which PROC EXPLODE writes the text in the message lines.

```
options parmcards=extfile;
filename extfile 'parmfile';


proc explode;
   title 'Important Message';
```

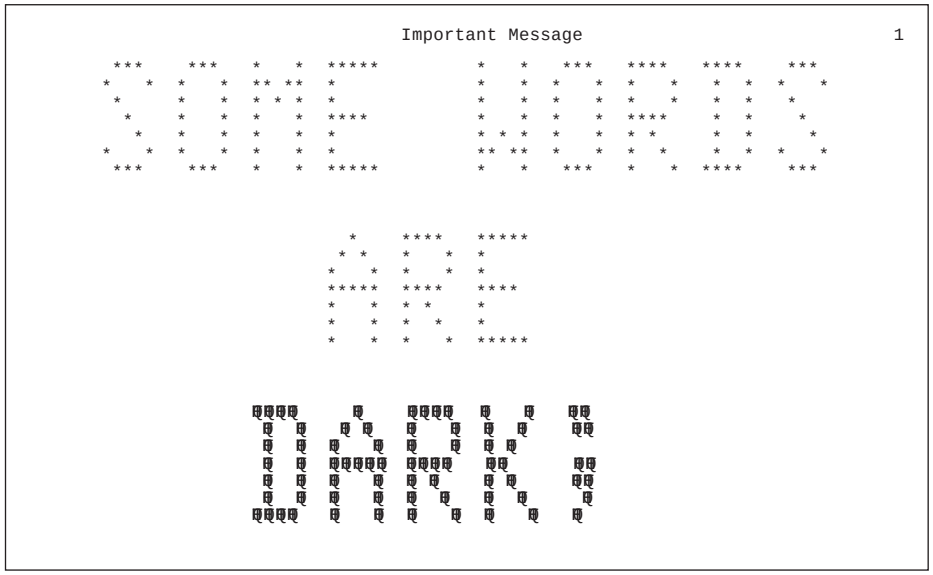
D overprints the line of text to make it darker, P begins a new page, and L returns to regular printing. U with the line of asterisks creates the underline.

```
   parmcards4;
  SOME WORDS
     ARE
D
    DARK;
P
L
   SOME ARE
     ALSO
```

The Null statement uses four semicolons because the message contains a semicolon.

```
D
  UNDERLINED
U **********
;;;;
```

Output



The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS® Procedures Guide, Version 8,* Cary, NC: SAS Institute Inc., 1999. 1729 pp.

SAS® Procedures Guide, Version 8

ISBN 1–58025–482–9

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, October 1999