Proc EXPLODE
## Contents

<table>
<thead>
<tr>
<th>Chapter 1: The EXPLODE Procedure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview: EXPLODE Procedure</td>
<td>1</td>
</tr>
<tr>
<td>Syntax: EXPLODE Procedure</td>
<td>2</td>
</tr>
<tr>
<td>Examples: EXPLODE Procedure</td>
<td>5</td>
</tr>
</tbody>
</table>
Overview: EXPLODE Procedure

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 1.1 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:

```sas
filename ft15f001 temp;
options nodate pageno=1 linesize=80
   pagesize=60;
proc explode;
   parmcards;
   TOP SECRET
;
```

Output 1.1  A Line of Expanded Text

<table>
<thead>
<tr>
<th>The SAS System</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>***** *** ****</td>
<td>*** ***** *** **** ***** *****</td>
</tr>
</tbody>
</table>
You can control spacing, the density of the text, and underlining with options.

---

**Syntax: EXPLODE Procedure**

**Requirements:** PARMCARDS or PARMCARDS4  
- Message line(s)  
- Null statement  

**Reminder:** You can use global statements with PROC EXPLODE. See the “Fundamental Concepts for Using Base SAS Procedures” chapter in *Base SAS Procedures Guide* 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, then you must use PARMCARDS4.  

**See also:** “Null Statement” on page 5  

**Featured in:** Example 1 on page 5 and Example 2 on page 6

```
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 5 and Example 2 on page 6

Message line(s)

\(<D \mid L>\)

\(<S_{n} \mid P>\)

\(<\text{spacing-control}>\)

\(<U \text{character-1 <...character-n>>}\)

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

\(<D \mid L>\)

\(<S_{n} \mid P>\)

\(<\text{spacing-control}>\)

\(<U \text{character-1 <...character-n>>}\)

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 caret (^) or a tilde (~), depending on which character set your keyboard uses. PROC EXPLODE ignores lowercase letters.

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

<table>
<thead>
<tr>
<th>To do this</th>
<th>Use this option</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control vertical spacing</td>
<td>(S_{n}) or (\text{spacing-control})</td>
</tr>
<tr>
<td>Control the text density</td>
<td>(D \mid L)</td>
</tr>
<tr>
<td>Underline text</td>
<td>(U)</td>
</tr>
<tr>
<td>Begin a new page</td>
<td>(P)</td>
</tr>
</tbody>
</table>
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 by 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 6

L
See D | L.

P
See Sn | P.

Sn | P
controls the amount of space before the next line of text.

Sn
skips n lines before the next line of text.

Range:  1–9

See also:  spacing-control

Featured in:  Example 1 on page 5

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

Featured in:  Example 2 on page 6

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:  Sn 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 6
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, then use four semicolons instead of one.

See also: “PARMCARDS or PARMCARDS4 Statement” on page 2

Examples: EXPLODE Procedure

Example 1: Controlling Spacing

Procedure features: PARMCARDS 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

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

options parmcards=extfile;
filename extfile 'parmfile';

proc explode;
   title 'Cover Page';
```
**Specify the spacing control.** 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:** PARMCARDS4 statement  
**Message lines options:** D
The EXPLODE Procedure

Program 7

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

**Program**

**Put overprinted characters in the text.** OVP allows overprinted characters in the text.

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

**Specify the file that will contain the procedure output.** 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';
```

**Customize the text in the output.** 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 **********
Your Turn

If you have comments or suggestions about *Proc EXPLODE*, please send them to us on a photocopy of this page, or send us electronic mail.

For comments about this book, please return the photocopy to
SAS Publishing
SAS Campus Drive
Cary, NC 27513
email: yourturn@sas.com

For suggestions about the software, please return the photocopy to
SAS Institute Inc.
Technical Support Division
SAS Campus Drive
Cary, NC 27513
email: suggest@sas.com