

bcc	title SPL EXECUTIVE COMMANDS	prefix/class-number.revision SPLEX/W-32	
checked <i>Sam Schuf</i>	authors L. Peter Deutsch <i>L. Peter Deutsch</i>	approval date 10/14/69	revision date
checked <i>Larry L Barnes</i>		classification Working Paper	
approved <i>Mel</i>		distribution Company Private	pages 7

ABSTRACT and CONTENTS

This document describes the current (10-7-69) state of the SPL command language in those areas not covered by CSED/W-12 or SPLDS/W-17. In particular, it lists and describes all "executive" commands not discussed in the two earlier documents.

1. The Command Processor

SPL always returns to its command processor at the end of an operation. The command processor (abbreviated CP) identifies itself by printing a herald at the left margin: the herald character depends on the current mode of the CP as follows:

:	edit mode, expert
*	edit mode, beginner
-	debug mode, expert
=	debug mode, beginner

In the edit modes, a special group of miscellaneous commands is also recognized. The CP accepts a complete line at a time, which is collected using the standard line-edit. The "old line" for the edit is the previous command line, starting with the character after the herald; this is useful for correcting errors, or executing a command several times.

The syntax of commands depends on the mode; however, all modes except expert debug mode have essentially the same syntax for the command name, to wit: the command name is the first thing on the line. If the command takes arguments, they follow the name. In the beginner modes, the command name may be written in full or abbreviated as much as desired (even down to a single letter); if anything follows the command name (modes, arguments, etc.), one or

more blanks must intervene before it, and otherwise blanks following the command name are optional. In expert edit mode, the command name is always abbreviated to a single letter; blanks following it are always optional. Expert debug mode has its own peculiar syntax, which is described in SPLDS/W-17.

2. Special Commands

In the two edit modes, the CP also recognizes some special commands beginning with a dot(.). The shortest abbreviation for these commands is a dot and a letter, rather than just a letter as for edit commands; otherwise they behave just like beginner edit commands with respect to abbreviation and use of blanks. They perform functions of an executive or global nature which logically do not fall into either edit or debug categories. The formats and actions of these commands are itemized below.

2A. COMPILE Command

.COMPILE *

Compiles all blocks.

.COMPILE <list of block names separated by blanks>

Compiles the specified blocks.

.COMPILE

Compiles just those blocks needing compilation, i.e., those which have been edited since last being compiled, caused diagnostics when last compiled, etc.

2B. FINISHED Command

.FINISHED

Returns control to 940 DDT, under which SPL runs. SPL may be resumed by transferring control to .IN.

2C. Status Commands.UNDEFINED

Lists all functions which have been called from some program which has been compiled, but which have not themselves been compiled.

.MAP

Produces a listing describing the layout of storage. The first two lines look like this:

```
USER RING: WGS 3 OF 220-3777, RSGS 14 OF 34000-37777,  
CS 40000-40135
```

This says that locations 220 through 3777 are currently assigned as WGS, of which only 3 cells are actually being used; 34000 to 37777 are assigned to RSGS, of which 14 are being used; and 40000 through 40135 are being used for CS. These two lines are followed by similar maps for the utility and monitor rings. All numbers are octal.

.STATISTICS

Print a variety of statistics about the current state of SPL. The printout looks something like this:

```
SPL 10-3.7
```

```
34567 MAPS 23456 STORES 987 READS 876 WRITES
```

```
27 IB'S, 55500+880 CELLS:
```

```
HEADERS: 500, TOKEN TABLES 2000+80, SYMBOL TABLES
```

```
53000+800
```

```
20000 (SYMS) 3000 (CONSTS) 30000 (PPT)
```

The first line identifies the assembly date (October 3)

and the patching level (7) of the version of SPL being used. The second line says that SPL has made 34567 references through its software map, of which 23456 were stores; read 987 pages from the drum; and written 876 pages. The third line says that there are 27 IBs (PROGRAMs or COMMON blocks) in the current ring, occupying 55500 useful cells and 880 cells of waste space. The last two lines break down these totals: 500 cells for headers (fixed overhead), 2000 useful and 80 waste cells for token tables, 53000 useful and 800 waste cells for symbol table and text, further broken down (useful cells only) into 20000 cells for symbol names and denotations, 3000 cells for constants, and 30000 cells for preprocessed text. All numbers are decimal.

2D. Saving and Retrieving Programs

.DUMP <file>

Dumps the entire contents of the current ring on the file. This includes source program, compiled code, global data, the status of the debugger, etc.

.RECOVER <file>

Restores the appropriate ring from the file, which must have been created with .DUMP (if this is not the case, there will be a diagnostic and no harm done).

All previous information in that ring is lost.

2E. Changing Rings.ADDRESS <digit>

Selects the user, utility, or monitor ring, according to whether the digit is 0, 1, or 2 respectively. The rings maintain an almost completely disjoint existence; source programs are permanently associated with the ring that was selected when they were read in.

Similarly, no variable or address in one ring can be used to refer to information in another ring, e.g., if F1 is a function in one ring and F2 is a function in another, F1 cannot refer to F2 by name, and SPL will assume that F1 is referring to a function F2 in the same ring as F1 if the attempt is made.

.BOUNDARY <octal number without B>

Deletes all information in the ring in which the given address falls, then sets the boundary between RSGS and CS in that ring to the given address. There will be a diagnostic with no harm done if the address is illegal. Normal values for the boundary are 400000 in the user ring, 440000 in the utility, and 640000 in the monitor.

3. Useful Knowledge

SPL is started up by RECOVERing (in the 940 exec) from a rather large file, CONTINUEing DDT, and starting up at .SPL. SPL will print its header (the first line printed by the STATISTICS command), some garbage about DUMMY COMPILING, and finally a : herald indicating it is ready for use. Leaving SPL momentarily should be accomplished with the .FINISHED command. Since SPL currently has no logic for breaking into a long unwanted printout, the following adhoc rules should be observed. There is no way to interrupt a compilation. To interrupt a long printout in edit mode, break out with control-K; put -1 into the cell QUITF; then proceed with ;P. If this gives an I>> trap, try .+1;G. To interrupt a long printout in debug mode, try control-K followed by BRR XDECOM;U. Warning: if the P-counter is below 4000B when you do either of these things, the results will probably be disastrous unless the instruction being executed was a BRS 42B or 43B.