Inter-Office Memorandum

To	BCPL Users	Date	August 20, 1973
From	James Curry	Location	Palo Alto
Subject	Loading BCPL Programs	Organization	PARC/CSL

XEROX

The BCPL loader consists of four files, normally called BLDR.SV, BLDR.YU, BLDR.YI, and BLDR.YD. The .Y* files are copies of files that the loader needs for initialization of the save file which it creates. The .Y* files must have the same name as the loader; so if you rename BLDR.SV, you must rename the .Y* files as well.

A typical command to BLDR looks like

```
    BLDR/D/L/V PROG IO1 IO2
```

This would create the file PROG.SV, an executable Nova save file, from the BCPL relocatable binary files PROG.BR, IO1.BR, and IO2.BR. The /D switch causes the Nova debugger to be loaded into the save file. The /L/V switches create a symbol table file named PROG.BS, containing information about where things will be in core when the program runs. The loader prints

```
    BLDR 2.Ø -- PROG.SV, PROG.BS
```

at the beginning of the loading process, and, when it is done,

```
    PROG.SV -- 14162  (6256)  WORDS
```

The numbers give the size of the program in octal (decimal). A typical .BS file listing is attached.

Errors

Errors in the command line to BLDR are fatal; the loader immediately aborts. Most such errors will result in a message like

```
        BAD SWITCH L IN PROG/L/S
```

Undefined file names, and other DOS-detected errors (such as an attempt to create a save file which exists as a permanent file on the disk), will result in something like

```
        CANNOT OPEN PROG.BR
```

Fatal error messages are always printed on the terminal.

The loader detects two types of external name conflicts. If an external name is defined (by 'static [name = ...]' or by 'let name (...) be ...') in more than one relocatable binary file, the loader generates a message like

```
PROG2.BR
THE EXTERNAL NAME name WAS ALSO DEFINED IN PROG1.BR
```

for each such conflict detected in PROG2. If an external name is declared to be a common (page zero) variable in some files (by 'external [@name; ...]') but not in the first file in which the name appears, the loader genrates a message like

```
PROG2.BR
THE COMMON NAME name WAS NOT DECLARED COMMON IN PROG1.BR
```

These messages appear in the .BS file if one is being created; the message

```
n ERRORS DURING LOADING
```

is printed on the terminal if any name conflicts are detected. You must recompile the offending files and reload before attempting to run the program.

External names which have been used but not defined result in the message

```
n UNDEFINED EXTERNALS
```

being printed on the terminal. The names are listed in the .BS file if one is being created; or on the terminal otherwise.

The loader also generates "warnings" if it detects space allocation conflicts in the save file being created. The most common of these are

```
NOT ENOUGH COMMON SPACE
```

if too many common (page zero) variables have been declared, and

```
NOT ENOUGH STATIC SPACE WAS RESERVED
```

if too many non-page-zero statics have been used. The available page zero space cannot be increased; you must redefine some common variables to be ordinary statics. The space reserved for statics can be specified with the local /W switch; see below for this and for other space allocation controls.

Global switches

/D Load the Nova debugger into the save file. This switch is legal only if no assembly language file is specified with the /I switch; if you load assembly language programs, you should include the debugger when you load them with RLDR.

/U Convert the names of all external symbols to upper case. This is needed, for example, if you are loading the IO package (IO1, IO2) with programs written in upper case; the IO procedure names in your files are upper case, but in IO1 and IO2 they are defined in lower case. Without /U, the upper case externals in your programs would be undefined. (Alternatively, you could recompile the IO package source files with BCPL/U.)

/W Do not print warning messages. Normally the loader will tell you if you do something suspicious, like loading a program on top of something else. If you know what you are doing, and if the warning messages bother you, you can turn them off with /W.

/L/V/N Generate lists of static variable names. /L prints procedure and label names, sorted by the location of the procedure or label in the code; the /L listing is, in effect, a core map. /V prints non-procedure names (variables). /N prints all static names, sorted by address. The most useful combination is /L/V; it lists all statics, separating procedure names from variable names. (The /R listing is no longer available; it is replaced by /L/V.) The listings go to the file "savefilename.BS" unless the /T switch is used.

/T All printed loader output (errors, warnings, and listings) is sent to the terminal. Normally, if listings are requested, they are sent to a file. Error and warning messages, and other load map data if there are no listings, normally go to the terminal.

/F All printed output is sent to the file "savefilename.BS", except for fatal error messages, which always go to the terminal.

Note that the meaning of /T has been inverted; that neither /T nor /F is normally necessary; and that the default extension for the listing file is .BS, not .LT.

Local switches (group 1)

These switches provide global information to the loader. All occurrences of these switches must appear before any of the group 2 switches, and before the first relocatable binary file name.

name/S The name of the save file to be created. (If not specified, the name of the first relocatable binary file is used.) If 'name' has no extension, .SV is used. The 'name' will also be used for the name of the .BS file unless the local /F switch is used.

name/F All output is sent to the file 'name'. If 'name' has no extension, .BS is used.

name/I Assembly language file. The file 'name' (extension .SV if 'name' has none) is assumed to be a Nova save file. The save file created by BLDR is initialized to the contents of this file (except for locations 300-377) at the beginning of loading. If the Nova debugger is to be loaded, it must have been loaded with the /I file. If no /I file is specified, a blank save file (BLDR.YI) is used, or if the global switch /D is specified, a save file containing only the debugger (BLDR.YD) is used.

name/U BCPL runtime routines. This switch allows the user to replace the standard runtime routines (get new frame, multiply, etc.) with his own. (These normally come from BLDR.YU.) The specified file is a Nova save file, but it is special in several respects. If you want your own runtime routines, see me.

number/N Maximum number of names allowed (octal). The default is 1000 (512 decimal). BLDR must allocate a certain amount of fixed space for each name, and must also have room for the name strings themselves. If you have a large number of long names, BLDR may run out of room, and print "OUT OF NAME SPACE"; or you may have more than 512 names. In either case, you may be able to load by adjusting the number of names allowed with /N. You may also be able to get more room with /C, if none of your .BR files have as much as 5000 words of code. (The /N switch does not affect the default /W value - see below).

number/C Maximum (octal) size of code in a single .BR file. The default is 5000. The /C switch is useful either if you have an especially big .BR file, or if you need more name space (see /N). (The compiler message "PROG.BR -- 1426 (790)

WORDS" indicates the size of the code compiled, in octal and decimal).

number/Z The (octal) starting address for allocating common page-zero static variables. If not specified, common starts at ZMAX of the /I file (which is 60 if global /D is specified), 50 otherwise.

number/V The (octal) starting address for allocating static variables. If not specified, statics start just after the BCPL runtime routines (which are loaded just after the /I file).

number/W The maximum number (octal) of non-page-zero static variables. The default is 400 (256 decimal). If no /V is specified, this amount of space is reserved in the save file at the default starting address for statics; code will be loaded after this space unless /P is given. If the starting address for statics is specified with /V, it is the user's responsibility to see that enough space is left for static variables at that address; /W is then just used in checking that static and code space do not overlap.

Local switches (group 2)

These switches control the loading of BCPL code into the save file. The loader also has facilities for creating "overlay" files to allow code to be swapped in dynamically; see the section on overlays below.

name (no
switches) A BCPL relocatable binary file. If 'name' has no extension, .BR is assumed (this is the extension normally used by the compiler). The code in the file is loaded into the save file at the current PC.

number/P Set the current PC to 'number' (octal).

$number/P Add 'number' to the current PC. No spaces may appear between the '$' and the 'number'.

letter/Q
letter/X
letter/Y The 'letter' is a single character A-Z. These switches associate the current PC with the letter so that the PC can later be restored with the form of /P described below. /Q uses the value of the current PC; /X uses the larger of the current PC and the value (if any) currently associated with

the 'letter'; /Y uses the smaller of the current PC and the current value of the 'letter'.

letter/P Set the current PC to the value last assigned to the 'letter' by /Q, /X, or /Y. If no value has been assigned, an error is reported.

The final PC value, after all files have been loaded, is taken as the address of the start of frame space when the program executes. (This value can be changed with a final /P specification.) Execution will begin with the first procedure defined in the first relocatable binary file loaded. This procedure will be called with one argument, a 32 (decimal) word vector whose contents are:

word 0: The last value assigned to 'A' by /Q, /X, or /Y.

.
.
.

word 25: The last value assigned to 'Z' by /Q, /X, or /Y.

word 26: The address at which statics were loaded.

word 27: The address of the last static variable.

word 28: The address of the first procedure loaded.

word 29: The address (+1) of the last word of BCPL code loaded.

word 30: The final value of PC (frame space start).

word 31: The highest memory address available.

The save file produced by BLDR looks just like an ordinary Nova save file. The core image it produces is organized as follows:

0...15
(Not part of a save file. Nova save files start with location 16; DOS considers locations 0-15 sacred. The addressess listed below are core addresses; subtract 16 (octal) if you are looking at the save file itself (e.g., with OEDIT).

16...277
An image of these words from the /I file. Common variables will normally be allocated starting at ZMAX, the first page zero (.ZREL) location not used by the /I file; this can be changed by the /Z switch to BLDR.

300...377
Reserved part of page zero (used by the BCPL runtime routines). You should refrain from clobbering these locations, unless you know what you are doing. Locations 340-377 are relocated by BLDR to point at various runtime routines.

400...777
An image of these words from the /I file. DOS depends heavily on this page being correct, so users should not clobber it. BLDR fixes a few words in this page to make the save file look as if it was created by the Nova loader.

1000...NMAX-1
An image of the rest of the /I file. NMAX is the first unused word of the /I file. If there is no /I file, NMAX will be approximately 4300 if /D was used (the debugger is about 3300 words long), 1000 otherwise.

NMAX...UMAX-1
The BCPL runtime routines. These currently are about 700 words long (UMAX = NMAX + approx. 700).

UMAX...VMAX-1 (if /V was not used)
Space for static variables, unless the starting address for statics was explicitly specified by /V. The size of the space reserved (VMAX - UMAX) is 400, unless changed with /W.

VMAX... (if /V was not used)
UMAX... (if /V was used)
The default starting address for loading BCPL code. If the group 1 switch specifications are followed by just a list of file names, the BCPL code will be loaded sequentially starting here, unless the PC is changed with /P.

<u>Overlays</u>

All occurrences of these switches must appear after all .BR file names which are to be loaded into the save file have been specified.

name/A Create the file 'name' (extension .BB if 'name' has no extension) and load the following relocatable binary files sequentially into that file. The code is intended to be read into core and run at the current value of PC; ✓ procedures and labels defined in the files loaded into 'name' will point at this area of core. The PC should not be changed (with /P) between the .BR files. The file 'name' has the format:

word Ø: value of PC at the first .BR file loaded into 'name' word 1: length of the code loaded into 'name'

word 2: Ø (this word is 1 for a /B file - see below)

word 3: length of 'name' in words

word 4: length of 'name' in words

word 5: Ø

.
.
.

word 15: Ø

word 16: code

. .
. .

The first word of the code for each .BR file is the length of the code for that file.

name/B Similar to /A, but in addition, the file 'name' contains information about which procedure and label pointers must be fixed when the code is read into core. /B is used when the place at which the code will be executed is not known at load-time. The format of 'name' is:

word Ø: value of PC at the first .BR file

word 1: length of code

word 2: 1 (to distinguish between /A and /B files)

word 3: L, the word in the file at which the relocation table starts

word 4: length of 'name' in words

word 5: Ø

.
.
.

word 15: Ø

word 16: code

. .
. .
. .

word L: number of relocation pairs N

```
word L+1: static address

word L+2: relative PC
     .
     .
     .
word L+N*2-1;  static address

word L+N*2:    relative PC
```

When the code is read in at location P, each "static address" must be set to P+ "relative PC", so that the procedures and labels which reference the code will point to the correct places.

<u>Distribution</u> / <u>BCPL</u>

BAUDELAIRE, Patrick
BROWN, Allen
CURRY, Jim
DEUTSCH, Peter
DUVALL, Bill
GUIBAS, Leo
HAUGELAND, Willie Sue
LAMPSON, Butler
MCCREIGHT, Ed
MCDANIEL, Gene
PAYNE, Adrienne
SHOUP, Dick
SPROULL, Bob
STURGIS, Howard
WEYER, Steve

JEC:amp