

XEROX Inter-Office Memorandum

To	BCPL Users	Date	August 16, 1973
From	James Curry	Location	Palo Alto
Subject	Compiling BCPL Programs	Organization	PARC/CSL

The BCPL compiler consists of six files, normally called BCPL.SV, BCPL.YL, BCPL.YC, BCPL.YS, BCPL.YT, and BCPL.YG. The .SV file is the main program; the .Y* files contain the code for the five passes of the compiler. The .Y* files must have the same name as the save file and the given extensions; so to rename the compiler, you must rename the .Y* files as well as the .SV file.

Normally, to compile a source file (e.g., PROG.3), just type

```
BCPL PROG.3
```

(Only one source file may be compiled at a time.) (No extension is automatically assumed for the source file name.) The compiler will print

```
BCPL 2.0 -- PROG.BR = PROG.3
```

and begin compiling the program. (2.0 is the current version of the compiler.) If no errors are detected, the BCPL relocatable binary file PROG.BR will be created, and the compiler will print something like

```
PROG.BR -- 1426 (790) WORDS
```

The numbers are the length of the code generated in octal (decimal).

If an error is detected in the source text, the compiler will generally print each offending line and indicate the error(s) found in that line. The compiler will continue to look for further errors as long as it can do so without getting confused, and finally print the message

```
n ERRORS IN PROG.3
```

Some errors are grounds for immediate termination of compilation. The most common ones are trying to compile a source file that does not exist, or typing a command line that BCPL does not understand. Suitable messages are printed to indicate such errors. It is also possible to have a program which is "too big", in one respect or another, for BCPL to handle. This usually results in a message like "FRAME SPACE OVERFLOW" or "OUT OF FRAME

To: BCPL Users
From: James Curry
Subject: Compiling BCPL Programs
Date: August 16, 1973
Page 2

SPACE". You must split the program into separately compilable files when this happens.

The compiler normally assumes that the Nova console is a CRT terminal. Therefore, after producing 20 lines of terminal output, it rings the bell (if any), prints a colon, and waits for the user to type a carriage-return or line-feed before proceeding. Carriage-return produces 20 more lines; line-feed produces one more line; \emptyset followed by carriage-return or line-feed causes the compiler to proceed without further pauses.

Global switches

These switches can be attached to the name BCPL (or a whatever you call your compiler); e.g., "BCPL/U/A PROG.3".

/U Treat the source file as if it had been typed entirely in upper case. (See below for upper/lower case considerations.)

/P Turn off the "pause" feature described above.

/F Write error messages onto the file PROG.BT (if the source file name was PROG.3) instead of printing them on the terminal. If /F is given, the compiler prints the message

BCPL 2. \emptyset -- PROG.BR,PROG.BT = PROG.3

at the beginning of compilation.

/A Produce an assembly-language listing of the code generated. (This is useful if you want to see what kind of code BCPL generates, or if you are having a hard time debugging a particular piece of code. But the listing file is big -- it takes a long time to generate and print -- so you probably don't want to make a habit of requesting it.) The listing is written on the file PROG.BT, unless the /T switch is given; error messages still appear on the terminal, unless /F is given.

/T Causes all output (error messages and the /A listing, if requested) to appear on the terminal. The file PROG.BT is not created.

(To summarize: /F alone sends error messages to PROG.BT. /A/F sends both errors and the assembly listing to PROG.BT; /A/T sends both to the terminal. /A alone sends errors to the terminal, and the assembly listing to PROG.BT. /F/T is illegal; /T alone has no effect.)

To: BCPL Users
From: James Curry
Subject: Compiling BCPL Programs
Date: August 16, 1973
Page 3

- /D Causes the compiler to indicate when it starts a new compilation phase (LEX, CAE, SAE, TRN, and NCG), and prints debugging information with error messages.
- /H Causes the compiler to pause (be entering the Nova debugger) between compilation phases and after error messages. To resume, type (ESC)R, not (ESC)P.

(/D and /H are generally useful only to compiler gurus.)

Local switches

These switches are attached to names following the compiler name in the command line; e.g., "BCPL PROG.3 PROG.LS/A":

- name (no switches) The name is taken as the source file name. No extension is assumed; you must type 'name.ext' if the source file has an extension. The source file name is used to generate the names for the relocatable binary (.BR) file and the text output (.BT) file (unless these are specified by the local switches /A, /F, /R). If a device is specified with the name (e.g., DPl:PROG.3), that device will be used for files specified in "get" directives in the source text; and for the output files (unless these are specified by the local switches /A, /F, /R). If no device is specified, the default device is used. (The device given in the last DIR command to DOS), even if the compiler is running on a different device (e.g., if you have typed "DIR DPØ; DPl:BCPL PROG...", PROG and its "get" files will come from DPØ).
- name/A Like the global /A switch, but the assembly listing is written onto "name" rather than PROG.BT. If "name" is a file name, the extension .BT will be appended to it if it has no extension; to create a file with no extension, use "name./A". If "name" is a device (e.g., MCØ:XGP.), it should be terminated with a "."; the output will be sent to the device named.
- name/F Like the global /F switch, but writes error messages onto "name" as for /A above. ("name/A/F" does the obvious thing, but you cannot send errors and the assembly listing to two different files.)
- name/R Causes the relocatable binary file to be named "name" instead of PROG.BR. The .BR extension is appended to "name" if it has no extension; to create a file with no extension, use "name./R".

To: BCPL Users
From: James Curry
Subject: Compiling BCPL Programs
Date: August 16, 1973
Page 4

name/L

name/T These switches cause the compiler to print the source text (/L) and intermediate compilation results (/T) as it proceeds through its various phases. The phases are specified by the individual characters of "name":

L for the lexical analyzer
C for the parser
S for the symbol table generator
O for the Ocode generator
1 for the code generator, pass 1
2 for the code generator, pass 2.

E.g., (C1/L" would cause the compiler to print each line of source text as it parses it, and again as it makes a first pass at generating code for the line. The output would go to the file PROG.BT unless the global /T switch were given. These switches are primarily for debugging the compiler. But they might be helpful occasionally in tracking down an obscure error, or one for which the error message does not provide enough context to locate the offending statement in the source text.

Upper case vs. Lower Case

The basic rule is as follows:

If the first word of the source program (i.e., of the file named in the command line) consists of all lower-case characters, the compiler will distinguish words on the basis of case; and reserved words must be typed in lower-case.

If the first word is not entirely lower-case, the compiler will, in effect, convert everything to upper-case on input. The global switch /U will also cause input to be converted, even if the first word is in lower-case.

This rule has implications for both compiling and loading. For compilation:

1. If your program is entirely upper-case, any "get" files specified in the program will be treated as upper-case files, even if they were prepared in lower-case. So an upper-case program can use a file of declarations (e.g., IOX for the IO package), as long as that declaration file does not depend on case to distinguish between names.

To: BCPL Users
From: James Curry
Subject: Compiling BCPL Programs
Date: August 16, 1973
Page 5

2. If your program wants to distinguish names on the basis of case, reserved words must be typed in lower case, both in your program and in any "get" files which the program needs. So in order to use a declaration file which was prepared in upper case, you must either use the /U switch (if you don't care about case) or change the declaration file's reserved words to lower-case (if you do care about case in your program).

The BCPL loader (BLDR) normally distinguishes external names on the basis of case. So if you want to load upper-case and lower-case .BR files together, you must use the /U global switch on BLDR (or, alternatively, recompile the lower-case programs with /U). In particular, you must use BLDR/U if you load the IO package (IO1.BR, IO2.BR) with upper-case programs, or recompile the source files (IO1, IO2) with BCPL/U.

Distribution

BAUDELAIRE, Patrick
BROWN, Allen
CURRY, Jim
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