

SCIENCE AND ENGINEERING RESEARCH COUNCIL  
RUTHERFORD APPLETON LABORATORY

COMPUTING DIVISION

DISTRIBUTED INTERACTIVE COMPUTING NOTE 927

PROGRAM DOCUMENTATION AIDS

Issued by  
D R Gibson

22 November 1983

DISTRIBUTION:

R W Witty  
K Robinson  
R E Thomas  
D A Duce  
C P Wadsworth  
E V C Fielding  
A J Kinroy  
D R Gibson  
G P Jones  
H K F Yeung  
C Prosser  
A S Williams  
J C Malone  
K G Dancey  
C J Webb  
P J Hemmings  
A P Ferryman  
K Lewis  
T Watson  
P Tempest-Mitchell  
M M Martin  
K J Fermor  
P J Smith  
J J C Hutchinson  
CBP/Internal User Notes File

There now exist three tools to aid program documentaion. *calltree* is the most general, giving information about the procedural structure of a program file, or suite of files, in several formats. *callsby* gives details of which procedures are called by those given, within the files specified. *whocalls* gives details of which procedures call those given, albeit indirectly, in the files specified.

The parsers are based on the Portable C and the York Pascal compilers. Dialects which have additional or different constructs may cause the parser to fail.

**NAME**

**callsby** - documentation aid for procedure referencing

**SYNOPSIS**

**callsby** [ -p ] [ -u ] *name*... -f *file*...

**DESCRIPTION**

*callsby* is a tool for parsing syntactically correct C and Pascal files and giving tree structured output showing which other functions are called by the argument procedure *name* (providing that the declaration and function body lies in the argument *file*.) The -f flag must always occur since it separates the *name* list from the *file* list. The -u flag will cause all procedures called by *name* to be given, including 'library' functions and those declared implicitly. The -p flag will give output which can be input to another program (not yet implemented) which will produce the FR80 instructions for drawing the tree.

**FILES**

/usr/tmp/drg\*

**SEE ALSO**

prof(1), calltree(1), whocalls(1)

**DIAGNOSTICS**

The syntax of the C parser is based on that of the Portable C compiler, and that of the Pascal parser on the York Pascal compiler. Only syntactically correct C or Pascal files may be processed. Giving a set of unrelated files, rather than a suite, may give spurious output if there is a duplication of procedure names within the files.

**NAME**

calltree - documentation aid for procedure referencing

**SYNOPSIS**

calltree [ -prst ] [ -u ] file...

**DESCRIPTION**

*calltree* is a tool for parsing syntactically correct C and Pascal programs and giving information about the procedural make up of the *file* in different ways, depending on the options set. The options are:

- p This causes the tree structure of the -t option to be output in a form which a subsequent program (not yet implemented) will read to produce the FR80 instructions for drawing the tree.
- r This gives a stripped down version of the program, consisting only of procedure declarations and calls.
- s This is similar to -r except that repeated procedure calls are reduced to a single call.
- t (default) This gives a 'tree' structured representation of the procedure hierarchy, i.e. 'who calls who'. There is a limited amount of back tracking allowed for a particular procedure, with an asterisk showing that further calls may be made by the given procedure, but will not be shown.
- u This gives the calls of implicitly declared procedures, i.e. 'library' functions, or functions that are formally declared in a file not given in the argument list. The default only recognises functions which are explicitly declared in the file(s) given and not those specified by a directive (e.g. extern, forward etc.) or by call.

**FILES**

/usr/tmp/drg\*

**SEE ALSO**

prof(1), whocalls(1), callsby(1)

**DIAGNOSTICS**

The syntax of the C parser is based on that of the Portable C compiler, and that of the Pascal parser on the York Pascal Compiler. *calltree* assumes that the files given will compile correctly. It is inadvisable to use the -u option on a 'large' program suite, it is better to use that option on the individual files and then use the less verbose option for the suite as a whole. Note that *calltree* is intended for use on a program suite: a sequence of unrelated files, having duplicate procedure names may give spurious output.

**BUGS**

*calltree* is limited by the available memory space as to how many references to procedures it can process.

**NAME**

**whocalls** - documentation aid for procedure referencing

**SYNOPSIS**

**whocalls** [ **-p** ] *name*... **-f** *file*...

**DESCRIPTION**

*whocalls* is a tool for parsing syntactically correct C and Pascal files and giving information on which procedures call the argument procedure *name*. The full path(s) to *name*, within the specified *file(s)*, are given by recursion on the caller, i.e. *whocalls* the caller. The **-p** flag will give output which can be used to generate the FR80 instructions for drawing the tree (via a program not yet implemented).

**FILES**

/usr/tmp/drg\*

**SEE ALSO**

calltree(1), prof(1), callsby(1)

**DIAGNOSTICS**

The syntax of the C parser is based on that of the Portable C compiler, and that of the Pascal on the York Pascal compiler. Only syntactically correct C or Pascal files may be processed. Giving a set of unrelated files, rather than a suite of files, may give spurious output if procedure names are duplicated.