

NAME

auex - execute command on source

SYNOPSIS

auex [-acou] file command-line

DESCRIPTION

Auex is a sophisticated program for sophisticated minds. In its simplest form it can be thought of as a program which reads pg, gu, and au files and executes a given command-line consisting of a unix command and arguments. The pg, gu and au files read can be specified as arguments via the use of special meaning keywords. The keywords which may be used and their special meaning are:

\$pr	current pr number
\$pg	current pg file name
\$ve	version of the current gu file (assumes 2 charcaters)
\$gu	current gu file name
\$au	current au file name
\$pr	current pr number

Auex, when given a pg file as the file argument, will remember the pg file name as the current pg file. The program then reads the pg file for the gu files listed therein. Only the #PR section is examined. Each gu file listed is processed one by one and is remembered as the current gu file as it is processed. The pr number of the gu file is also remembered as the current pr number. Each gu file may be read for the au files listed within it. Again, each au file, in turn, is remembered as the current au file name. Finally, each au file may be read and each source file therein remembered as the current source file name.

After all necessary readings are performed and the required special keywords evaluated, auex will execute the command-line. The actual execution is performed by executing the shell, sh (I), as a child and sending the command-line to the shell via a pipe.

Variations of the above theme follows. If the command-line contains no \$src keyword, then no source file is processed. This actually avoids the need to read the au file and as a result, the program is faster because it does less work. Likewise, if no \$src and no \$au keywords are present in the command-line then the gu files are not read, saving more time. In short, auex reads only those source control files (pg, gu, and au) which are necessary to perform the keyword substitutions prior to executing the command-line.

Auex may be used to process a gu file (and all au and source files referenced by it) or an au file (and all source files referenced by it) by specifying the gu or au file as the file argument. In such a case undefined keywords such as \$pg would have

a null value.

To process a pg file for the file argument, auex will chdir to the current pr directory before executing the command-line. As such, the program should be initially executed by a user whose current directory contains all the required pr directories (eg., /gensrc/src). To process a gu (or au) file, the user should reside in the directory with the au files and source files referenced by the gu file (or au file).

The flags available to auex:

- v inhibit verbose output. Auex will execute the shell with the verbose option unless the -v option is specified.
- u do not sort files. Auex will sort the au files listed in a gu file and the source files listed in an au file before processing them unless the -u option is specified.
- c concatenate the source files listed in an au file into a single long string; each file name is separated by a blank. Backslash and newline characters are inserted between every fourth file name to make the string more readable by breaking it up into multi lines. The resulting string is used as the value for \$src. If the keyword \$src is not specified and the -c option is on, then the au files listed in a gu file are concatenated into one string and used for the value of \$au.
- a execute the command-line for all au files. Auex will normally not execute the command-line after processing an au file if 1) the command-line contains \$src and 2) the au file has no source files listed in the #PROGRAM UNITS and #DATA UNITS sections. The -a option forces auex to execute the command-line for all au files, ignoring the above exception.

The quantity of arguments in the executed command-line may be large enough to exceed the shell's internal maximum number of arguments. This problem is most probable when the -c option is specified. Hence, a mechanism is included in auex to execute a command-line of at most 43 arguments when the -c option is specified. Thus, for a single au file several executions of the command-line may be required if the quantity of source files listed in the au file is large. In the following example, the au file, mine.au, has 50 source files:

```
auex -c mine.au 'ls $src'
```

and the ls command would be executed twice, the first time with 42 source files and the second time with 8 source files as the value of \$src.

Auex must remember a lot and unfortunately it may run out of internal "memory". The only true solution is to recompile the program with more memory.

The character \$ has special meaning to auex. Besides interpreting \$pg, \$pr, \$gu, \$au, \$src as special keywords, \$\$ is mapped to \$ and \$<anything else> is mapped to <anything else>.

FILES**SEE ALSO**

au(5L), gu(5L), pg(5L)

DIAGNOSTICS

Error messages are to file descriptor two.

BUGS

Auex should dynamically allocate memory

A line read from a pg/gu/au file of greater than 200 characters may cause auex to drop a core