NAME

mkstr – create an error message file by massaging C source files

SYNOPSIS

/usr/ucb/mkstr [–] *messagefile prefix filename*. . .

AVAILABILITY

SUNWscpu

DESCRIPTION

mkstr creates files of error messages. You can use **mkstr** to make programs with large numbers of error diagnostics much smaller, and to reduce system overhead in running the program — as the error messages do not have to be constantly swapped in and out.

mkstr processes each of the specified *filename*s, placing a massaged version of the input file in a file with a name consisting of the specified *prefix* and the original source file name. A typical example of using **mkstr** would be:

mkstr pistrings processed *.c

This command would cause all the error messages from the C source files in the current directory to be placed in the file **pistrings** and processed copies of the source for these files to be placed in files whose names are prefixed with *processed*.

To process the error messages in the source to the message file, **mkstr** keys on the string '**error("**' in the input stream. Each time it occurs, the C string starting at the '"' is placed in the message file followed by a null character and a NEWLINE character; the null character terminates the message so it can be easily used when retrieved, the NEWLINE character makes it possible to sensibly **cat** the error message file to see its contents. The massaged copy of the input file then contains a **lseek** pointer into the file which can be used to retrieve the message, that is:

```
char efilname[ ] = "/usr/lib/pi_strings";
int efil = −1;

error(a1, a2, a3, a4)
{
        char
        buf[256];
        if (efil < 0) {
                efil = open(efilname, 0);
                if (efil < 0) {
oops:
                perror (efilname);
                exit (1);
```

```
                        }
                }
                if (lseek(efil, (long) a1, 0) | | read(efil, buf, 256) <= 0)
                        goto oops;
                printf(buf, a2, a3, a4);
        }
```

OPTIONS

– Place error messages at the end of the specified message file for recompiling part of a large **mkstr**ed program.

SEE ALSO

xstr(1)