NAME

varsub – variable substitution filter

SYNOPSIS

varsub [**-db** *mapfile*] [**-s** *var=value*] [**-k** *keychars*] [**-b** *blankspec*] [**-af** *argfile*] [**-if** *infile*] [**-of** *outfile*] [**-env**] [**-V**] [**-D**] [**-?**]
[file(s)]

DESCRIPTION

This program takes as input one or more source files and converts symbolic filter references found in them to their associated target values as defined in a *filter-map* file. Source (input) files can be specified in a variety of ways, but are most commonly just listed as invocation arguments.

Key Options

Variable substitutions can be specified directly with the program invocation command using the **-s** *var=value* argument sequence. The *var* string specified the variable name to be substituted for and the *value* string is the value to substitute. This option can be repeated as desired to build up a set of key-value pairs for substitution. All substitution keys specified in this way take precedence over substitution keys found in the database filter-map file.

The **-db** *mapfile* option introduces a filter-map file to be used by the program. The file-file is a file that contains key-value pairs. Keys are specified in the first column and the associated value is specified in the second column. Keys cannot have whitespace in them. If the value has any whitespace in it, it should be quoted with double quotes. If this option is not specified, then the environment variable **VARSUB_MAP** is accessed for the name of the filter configuration file. If that environment variable is not present in the environment, then the program will behave similar to the `cat(1)` program except that variable keys in the source file may be substituted for blanks (the default situation).

Two types of characters can delimit keys in the source files. These are parenthesis and squiggle brackets. The **-k** *keychars* argument can be given to turn ON or OFF the recognition on either one of these characters. By default, both characters are recognized as key delimiters in the source text.

Blank handling is specified using the **-b** option. The option alone will turn ON blank substitution (the default). If it is specified as: `-b=n` then blank substitution is turned off and an error will result when a key is encountered in the source text and there is no associated map value for that key specified on either the invocation command or a filter-map file.

If the **-env** option is specified, then the program environment is used for value substitutions when an otherwise unknown key (one which was not given at program invocation or in a file-map file) is encountered.

The **-V** option causes the program to write out to standard error the version of the program. The program will continue to execute according to the other files or options specified. If the user just wants the program's version without actually performing any filter operation, the user should use the **-?** option as above in conjunction with this option.

The **-?** option is causes the program to terminate with a message printed to standard error explaining the usage of the program. This is useful when the user forgets the exact syntax of the program's arguments and does not want to consult the manual page.

The **-D** option enables the printing of some debugging information to the standard error. Information such as whether a new freeze file was created is supplied. All severe errors of any kind are printed to standard error regardless of the setting of this option.

Filter Map File

The filter configuration file contains the mapping of symbolic names to target string definitions. Comments are introduced into the filter configuration file by the pound sign character. Blank lines are ignored.

Two fields are entered per line with white space separating them. The first field is the symbolic name field. The second field is the target substitution string. The target substitution string is that which is substituted in place of a symbolic name when one is encountered in a source SATtool script. Quote characters are not allowed within either field. An example of a filter configuration file is given below.

Input File

Input source files are generally specified as arguments at program invocation. Optionally, an **- if** *infile* argument can be specified to give the name of an additional input file to be used in the desired conversion. Input shource files can also be specified with: **- af** *argfile* In this latter case, each line of the file *argfile* should contain one source- text file to be filtered. If no explicit input source files are specified then the standard input is used.

Output

The default filtered output goes to STDOUT. An alternate output file can be specified with the argument invocation specification of: `-of outfile.`

EXAMPLES

☞ To convert a source SATtool script file into the object script

```
varsub -db /usr/auto/etc/f.v5d6 source.ap > object.ap
```

☞ To convert an object SATtool script file into a source SATtool script file:

```
varsub raw.ap -db /usr/adm/etc/f.v5d9 > introduced.ap
```

☞ Get the program version only without performing any filtering function:

```
varsub -V
```

☞ Perform a macro expansion operation with debugging turned on:

```
varsub infile.ap -D -db /usr/adm/etc/f.v5d9 > outfile.ap
```

☞ Perform a macro expansion operation writing to a specified output file but using standard input for the program input file:

```
varsub - -db /usr/adm/etc/f.v5d9 > outfile.ap
```

☞ Perform a macro introduction operation using the environment variable **CARSUBMAP** to specify the filter configuration file instead of specifying one on the command line:

```
VARSUB_MAP=/usr/adm/etc/f.v5d9 ; export VARSUB_MAP
varsub - > outfile.ap
```

CAVEATS

Symbolic names with embedded quote characters cannot be handled since quote characters are not allowed within the field definitions as given in the filter configuration file. White space must always separate command line option key letters from their associated parameter.

ENVIRONMENT VARIABLES

The following environment variables affect the program operation.

VARSUB_MAP

This environment variable specifies the variable substitution mapping file used to map variable references to their values.

TMPDIR

This environment variable specifies a temporary directory to use to create temporary files which may be needed by the program. When this is not specified, the directory `/tmp` is used by default.

FILES

There are no predefined variable mapping files.

MAINTAINER

David A.D. Morano

AUTHOR

David A.D. Morano