IEEE P1003.2a Draft 8 – December 1991

3.14.11 `set` — Set/unset options and positional parameters

⇒ **3.14.11 `set`.** *Merge the following options into the the first two synopsis lines of subclause 3.14.11 with appropriate editorial wording indicating that these options need to be supported only when the UPUO is supported:* 7 7 7

```
set [ -bm ] [ -o option ] [ argument ... ]
set [ +bm ] [ +o option ] [ argument ... ]
```
7 7

⇒ **3.14.11 `set`.** *Modify the contents of subclause 3.14.11 to add the following options in the correct sorted order.*

`−b` Cause the shell to notify the user asynchronously of background job completions. The following message shall be written to standard error:

`"[%d]%c %s%s\n"`, *<job-number>* , *<current>* , *<status>*, *<job-name>*

where the fields shall be as follows:

<current>
: The character `+` identifies the job that would be used as a default for the `fg` or `bg` utilities; this job can also be specified using the *job_id* `%+` or `%%`. The character `−` identifies the job that would become the default if the current default job were to exit; this job can also be specified using the *job_id* `%–`. For other jobs, this field is a `<space>`. At most one job can be identified with `+` and at most one job can be identified with `–`. If there is any suspended job, then the current job shall be a suspended job. If there are at least two suspended jobs, then the previous job shall also be a suspended job.

<job-number>
: A number that can be used to identify the process group to the `wait`, `fg`, `bg`, and `kill` utilities. Using these utilities, the job can be identified by prefixing the job number with `%`.

<status>
: Unspecified.

<job-name>
: Unspecified.

When the shell notifies the user a job has been completed, it may remove the job's process ID from the list of those ''known in the current shell execution environment''; see 3.9.3.1. Asynchronous notification shall not be enabled by default.

`−m` This option shall be supported if the system supports both the User Portability Utilities Option and the POSIX.1 {8} Job Control Option. All jobs shall be run in their own process groups. Immediately before the shell issues a prompt after completion of the background job, a message reporting the exit status of the background job shall be written to standard error. If a foreground job stops, the shell shall write a message to standard output to that effect, formatted as described by the `jobs` utility (see 5.15.6.1). In addition, if a job changes status other than exiting (for example, if it stops for input or output, or is stopped by a SIGSTOP signal), the shell shall write a similar message immediately prior to writing the next prompt. On systems supporting the POSIX.1 {8} Job Control Option, this option shall be enabled by default for interactive shells.

`−o` *option*

Set various options, many of which shall be equivalent to the single option letters. The following values of *option* shall be supported:

`allexport` Equivalent to `−a`.

8

`errexit` Equivalent to `−e`.

`ignoreeof` Prevent an interactive shell from exiting on end-of-file. 7 7

`monitor` This option-argument shall be supported if the system supports both the User Portability Utilities Option and the POSIX.1 {8} Job Control Option. It shall be equivalent to the `−m` option.

`noclobber` Equivalent to `−C` (uppercase C).

`noglob` Equivalent to `−f`.

`noexec` Equivalent to `−n`.

`notify` Equivalent to `−b`.

`nounset` Equivalent to `−u`.

`verbose` Equivalent to `−v`.

`vi` Allow shell command-line editing using the built-in `vi` editor. Enabling `vi` mode shall disable any other command-line editing mode provided as an implementation extension. 8 8 8

It need not be possible to set `vi` mode on for certain block-mode terminals.

`xtrace` Equivalent to `−x`.

Rationale. *(This subclause is not a part of P1003.2a)*

The `ignoreeof` setting prevents accidental logouts when the end-of-file character (typically `control-D`) is entered. A user must explicitly `exit` to leave the interactive shell.

The `-o` option was originally in the base POSIX.2 document, but was removed because it is not commonly used by scripts and all of the options required for application portability had (or were given) single-letter option names. However, users have different needs and the `-o` option was adopted from the KornShell to address these. In addition to its generally friendly interface, `-o` is needed to provide the `vi` command-line editing mode, for which historical practice yields no single-letter option name. (Although it might have possible to invent such a letter, it was recognized that other editing modes would be developed and `-o` provides ample name space for describing such extensions.)

It is not possible to use the `$-` special parameter to determine the current setting of either of these two modes. In the KornShell, the *option* option-argument can be omitted, producing a report of current option settings. Since it breaks the utility syntax guidelines, and since the output format was unspecified (it changed between KornShell versions), this usage was omitted.

The `set -m` option was added to apply only to the UPE because it applies primarily to interactive use, not shell script applications.

The new `set -o` option-arguments apply only to the UPE because they are needed to enable the line editing modes to be used in the additions to `sh` for the UPE.

The ability to do asynchronous notification became available in the 1988 version of the KornShell. To have it occur, the user had to issue the command

```
trap "jobs -n" CLD
```

The C-shell provides two different levels of an asynchronous notification capability. The environment variable **notify** is analogous to what is done in `set -b` or `set -o notify`. When set, it notifies the user immediately of background job completions. When unset, this capability is turned off.

The other notification ability comes through the built-in utility `notify`. The syntax is

```
notify [%job . . . ]
```

By issuing `notify` with no operands, it causes the C-shell to notify the user asynchronously when the state of the current job changes. If given operands, `notify` will asynchronously inform the user of changes in the states of the specified jobs.

To add asynchronous notification to the POSIX shell, neither the KornShell's extensions to `trap`, nor the C-shell's **notify** environment variable seemed appropriate (the **notify** variable is not a proper POSIX environment variable name).

The `set -b` option was selected as a compromise.

The `notify` built-in was considered to have more functionality than was required for simple asynchronous notification.

Section 4: Revisions to Execution Environment Utilities

4.12 command — Execute a simple command

⇒ **4.12.1 command Synopsis.** *Add the following lines to the* `command` *utility Synopsis:*

On systems supporting the User Portability Utilities Option, the following 7
forms shall also be supported: 7

`command [ −v | −V ] `*command_name*

⇒ **4.12.2 command Description.** *Add the following paragraph to the end of the* `command` *utility Description:*

On systems supporting the User Portability Utilities Option, the `command` utility shall also provide information concerning how a command name will be interpreted by the shell; see `−v` and `−V`.

⇒ **4.12.3 command Options.** *Add the following options in the correct sorted order:*

`−v` Write a string to standard output that indicates the pathname
or command that will be used by the shell, in the current shell 7
execution environment (see 3.12), to invoke *command_name*. 7

— Utilities, regular built-in utilities, *command_names* including a slash character, and any implementation-provided functions that are found using the **PATH** variable (as described in 3.9.1.1), shall be written as absolute pathnames.

— Shell functions, special built-in utilities, regular built-in 7
utilities not associated with a **PATH** search, and shell 7
reserved words shall be written as just their names. 7

— An alias shall be written as a command line that represents its alias definition.

— Otherwise, no output shall be written and the exit status shall reflect that the name was not found.