IEEE P1003.2a Draft 8 – December 1991

Section 3: Revisions to Shell Command Language

⇒ **3.1 Shell Definitions.** *Modify the contents of clause 3.1, Shell Definitions, to add the following definition in the correct sorted order [disregarding the subclause number shown here].*

3.1 Shell Definitions

3.1.101 alias name: A word consisting solely of underscores, digits, and alphabetics from the portable character set (see 2.4) and any of the following characters:

`!` `%` `,` `@` 7

Implementations may allow other characters within alias names as an extension. 7

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

The characters allowed in alias names are from historical practice.

⇒ **3.3.1 Alias Substitution.** *Modify the contents of clause 3.3, Token Recognition, to add the following subclause:*

3.3.1 Alias Substitution

The processing of aliases shall be supported if the system supports the User Portability Utilities Option.

After a token has been delimited, but before applying the grammatical rules in 3.10, a resulting word that is identified to be the command name word of a simple command shall be examined to determine if it is an unquoted, valid alias name. However, reserved words in correct grammatical context shall not be candidates for alias substitution. A valid alias name (see 3.1.101) shall be one that has been defined by the `alias` utility (see 5.1) and not subsequently undefined using `unalias` (see 5.31). Implementations also may provide predefined valid aliases that are in effect when the shell is invoked. To prevent infinite loops in recursive aliasing, if the shell is not currently processing an alias of the same name, the word shall be replaced by the value of the alias; otherwise, it shall not be replaced.

If the value of the alias replacing the word ends in a `<blank>`, the shell shall check the next command word for alias substitution; this process shall continue until a word is found that is not a valid alias or an alias value does not end in a `<blank>`.

When used as specified by this standard, alias definitions shall not be inherited by separate invocations of the shell or by the utility execution environments invoked by the shell; see 3.12.

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

The alias capability was added in the UPE because it is widely used in historical implementations by interactive users. It was omitted from the base standard because it is not commonly used in application scripts, particularly since aliases are not passed to child shells.

The definition of *alias name* precludes an alias name containing a slash character. Since the text applies to the command words of simple commands, reserved words (in their proper places) cannot be confused with aliases.

The placement of alias substitution in Token Recognition implies that it precedes all of the word expansion steps.

An example concerning trailing `<blank>`s and reserved words follows. If the user types:

```
$ alias foo="/bin/ls "
$ alias while="/"
```

The effect of executing

```
$ while true
> do
> echo "Hello, World"
> done
```

is a never-ending sequence of `Hello, World` strings to the screen. However, if the user types

```
$ foo while
```

the result will be an `ls` listing of `/`. Since the alias substitution for `foo` ends in a `<space>`, the next word is checked for alias substitution. The next word, `while`, has also been aliased, so it is substituted as well. Since it is not in the proper position as a command word, it is not recognized as a reserved word.

If the user types

```
$ foo; while
```

`while` retains its normal reserved-word properties.

See the rationale for `alias` for a discussion of exportable aliases.

3.5.3 Variables

⇒ **3.5.3 Variables.** *Modify the contents of subclause 3.5.3, Variables, to add the following environment variables in the correct sorted order.*

ENV This variable, when the shell is invoked, shall be subjected to parameter expansion (see 3.6.2) by the shell and the resulting value shall be used as a pathname of a file containing shell commands to execute in the current environment. The file need not be executable. If the expanded value of **ENV** is not an absolute pathname, the results are unspecified. **ENV** shall be ignored if the user's real and effective user IDs or real and effective group IDs are different. This standard specifies the effects of this variable only for systems supporting the User Portability Utilities Option.

LINENO This variable shall be set by the shell to a decimal number representing the current sequential line number (numbered starting with 1) within a script or function before it executes each command. If the user unsets or resets **LINENO** , the variable may lose its special meaning for the life of the shell. If the shell is not currently executing a script or function, the value of **LINENO** is unspecified. This standard specifies the effects of this variable only for systems supporting the User Portability Utilities Option.

PPID This variable shall be set by the shell to the decimal process ID of the process that invoked this shell. In a subshell (see 3.12), **PPID** shall be set to the same value as that of parent of the current shell. For example, `echo $PPID` and `(echo $PPID)` would produce the same value. This standard specifies the effects of this variable only for systems supporting the User Portability Utilities Option.

PS 1 Each time an interactive shell is ready to read a command, the value of this variable shall be subjected to parameter expansion and written to standard error. The default value shall be `"$ "`. For users who have specific additional implementation-defined privileges (see 2.2.2.6) the default may be another, implementation-defined, value. The shell shall replace each instance of the character `!` in **PS 1** with the history file number of the next command to be typed. (See 5.12.) Escaping the `!` with another `!` (i.e., `!!`) shall place the literal character `!` in the prompt. This standard specifies the effects of this variable only for systems supporting the User Portability Utilities Option.