IEEE P1003.2a Draft 8 – December 1991

Copyright © 1991 by the
Institute of Electrical and Electronics Engineers, Inc.
345 East 47th Street
New York, NY 10017, USA
All rights reserved as an unpublished work.

This is an unapproved and unpublished IEEE Standards Draft, subject to change. The publication, distribution, or copying of this draft, as well as all derivative works based on this draft, is expressly prohibited except as set forth below.

Permission is hereby granted for IEEE Standards Committee participants to reproduce this document for purposes of IEEE standardization activities only, and subject to the restrictions contained herein.

Permission is hereby also granted for member bodies and technical committees of ISO and IEC to reproduce this document for purposes of developing a national position, subject to the restrictions contained herein.

Permission is hereby also granted to the preceding entities to make limited copies of this document in an electronic form only for the stated activities.

The following restrictions apply to reproducing or transmitting the document in any form: 1) all copies or portions thereof must identify the document's IEEE project number and draft number, and must be accompanied by this entire notice in a prominent location; 2) no portion of this document may be redistributed in any modified or abridged form without the prior approval of the IEEE Standards Department.

Other entities seeking permission to reproduce this document, or any portion thereof, for standardization or other activities, must contact the IEEE Standards Department for the appropriate license.

Use of information contained in this unapproved draft is at your own risk.

IEEE Standards Department
Copyright and Permissions
445 Hoes Lane, P.O. Box 1331
Piscataway, NJ 08855-1331, USA
+1 (908) 562-3800
+1 (908) 562-1571 [FAX]

7540 If the −u option is used with −T, the idle time shall be added to the end of the pre-
7541 vious format in an unspecified format.

7542 #### 5.36.6.2 Standard Error

7543 Used only for diagnostic messages.

7544 #### 5.36.6.3 Output Files

7545 None.

7546 ### 5.36.7 Extended Description

7547 None.

7548 ### 5.36.8 Exit Status

7549 The who utility shall exit with one of the following values:

7550 0 Successful completion.

7551 >0 An error occurred.

7552 ### 5.36.9 Consequences of Errors

7553 Default.

7554 ### 5.36.10 Rationale. *(This subclause is not a part of P1003.2a)*

7555 **Usage, Examples**

7556 None.

7557 **History of Decisions Made**

7558 Due to collisions with historical implementations, the options provided were a
7559 compromise to allow users to get at that functionality. The working group also
7560 weighed the idea of removing all the options, but felt these options offered users
7561 some real value.

7562 It is recognized that the who command may be of limited usefulness, especially in
7563 a multilevel secure environment. It was felt, however, that having some standard
7564 method of determining the ‘‘accessibility’’ of other users would aid user
7565 portability.

7566 The ‘‘domain of accessibility’’ was considered to be a sufficiently broad concept to
7567 permit interpretation either on a very secure basis or even to allow a network-
7568 wide ‘rwho-like’’ implementation.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

The System V implementation of `who` is overloaded with some unrelated functionality (such as displaying the time of the last system reboot). The group decided that the prime function of `who` was to determine which users could be contacted on a system, by `write`, `talk`, etc., and therefore did not consider it proper to burden `who` with unrelated functionality.

The historical `who am I` command, while being one of the more intuitively obvious UNIX system commands, had to succumb to the tide of internationalization. It is replaced by the somewhat less charming `-m` option.

No format was specified for the default `who` output so that existing implementations should not need to be changed, except for adding the new options. In such an obviously user-oriented command, designed only for human consumption, this was not considered to be a deficiency.

The format of the terminal name is unspecified, but the descriptions of `ps`, `talk`, and `write` require that they use the same format.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

5.37 `write` — Write to another user

5.37.1 Synopsis

`write` *user_name* [*terminal*]

5.37.2 Description

The `write` utility shall read lines from the user's standard input and write them to the terminal of another user. When first invoked, it shall write the message:

```
Message from
sender-login-id
(sending-terminal )
[date ]...
```

to *user_name* . When it has successfully completed the connection, the sender's terminal shall be alerted twice to indicate that what the sender is typing is being written to the recipient's terminal.

If the recipient wants to reply, this can be accomplished by typing

`write` *sender-login-id* [*sending-terminal*]

upon receipt of the initial message. Whenever a line of input as delimited by a NL, EOF, or EOL special character (see POSIX.1 {8} section 7.1.1.9) is accumulated while in canonical input mode, the accumulated data shall be written on the other user's terminal. Characters shall be processed as follows:

— Typing the `<alert>` character shall write the `<alert>` character to the recipient's terminal.

— Typing the erase and kill characters shall affect the sender's terminal in the manner described by the POSIX.1 {8} *termios* interface.

— Typing the interrupt or end-of-file characters shall cause `write` to write an appropriate message (`"EOT\n"` in the POSIX Locale) to the recipient's terminal and exit.

— Typing characters from LC_CTYPE classifications `print` or `space` shall cause those characters to be sent to the recipient's terminal.

— When and only when the `stty` *iexten* local mode (see 4.59.4.4) is enabled, additional special control characters and multibyte or single-byte functions shall be processed in an implementation-defined manner. (See POSIX.1 {8} section 7.1.1.9.)

— Typing other nonprintable characters shall cause implementation-defined sequences of printable characters to be written to the recipient's terminal.

To write to a user who is logged in more than once, the *terminal* argument can be used to indicate which terminal to write to; otherwise, the recipient's terminal is selected in an implementation-defined manner and an informational message

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

shall be written to the sender's standard output, indicating which terminal was chosen.

Permission to be a recipient of a `write` message can be denied or granted by use of the `mesg` utility (see 5.17). However, a user's privilege may further constrain the domain of accessibility of other users' terminals. The `write` utility shall fail when the user lacks the appropriate privileges to perform the requested action.

5.37.3 Options

None.

5.37.4 Operands

The following operands shall be supported by the implementation:

user_name Login name of the person to whom the message shall be written. This operand shall be of the form returned by the `who` utility (see 5.36).

terminal Terminal identification in the same format provided by the `who` utility.

5.37.5 External Influences

5.37.5.1 Standard Input

Lines to be copied to the recipient's terminal shall be read from standard input.

5.37.5.2 Input Files

None.

5.37.5.3 Environment Variables

The following environment variables shall affect the execution of `write`:

LANG This variable shall determine the locale to use for the locale categories when both **LC_ALL** and the corresponding environment variable (beginning with **LC_**) do not specify a locale. See 2.6.

LC_ALL This variable shall determine the locale to be used to override any values for locale categories specified by the settings of **LANG** or any environment variables beginning with **LC_**.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

LC_CTYPE This variable shall determine the interpretation of sequences of bytes of text data as characters (e.g., single- versus multibyte characters in arguments and input files).

LC_MESSAGES This variable shall determine the language in which diagnostic messages should be written.

5.37.5.4 Asynchronous Events

If an interrupt signal is received, `write` shall write an appropriate message on the recipient's terminal and exit with a status of zero. It shall take the standard action for all other signals; see 2.11.5.4.

5.37.6 External Effects

5.37.6.1 Standard Output

An informational message shall be written to standard output if a recipient is logged in more than once.

5.37.6.2 Standard Error

Used only for diagnostic messages.

5.37.6.3 Output Files

The recipient's terminal shall be used for output.

5.37.7 Extended Description

None.

5.37.8 Exit Status

The `write` utility shall exit with one of the following values:

0 Successful completion.

>0 The addressed user is not logged on, or the addressed user denies permission.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

5.37.9 Consequences of Errors

Default.

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

Usage, Examples

None.

History of Decisions Made

The `write` utility was included in the standard since it can be implemented on all terminal types. The working group considered the `talk` utility, which cannot be implemented on certain terminals, to be a ''better'' communications interface. Both of these programs are in widespread use on historical implementations. Therefore, the working group decided that both utilities should be specified.

The format of the terminal name is unspecified, but the descriptions of `ps`, `talk`, `who`, and `write` require that they all use or accept the same format.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

Annex E
(inform ative)

Rationale and Notes

This annex summarizes the deliberations of the IEEE P1003.2 working group, the committee charged by the IEEE Computer Society's Technical Committee on Operating Systems and Application Environments with devising an interface standard for a shell and related utilities to support and extend POSIX.1 {8}.

The annex is being published along with the standard to assist in the process of review. It contains historical information concerning the contents of the standard and why features were included or discarded by the working group. It also contains notes of interest to application programmers on recommended programming practices, emphasizing the consequences of some aspects of the standard that may not be immediately apparent.

Just as this standard relies on the knowledge of architecture, history, and definitions from the POSIX.1 {8}, so does this annex. The reader is referred to the Rationale and Notes appendix of POSIX.1 {8} for background material and bibliographic information about UNIX systems in general and POSIX specifically, which will not be duplicated here.

Editor's Note: The text of most of the Rationale has been temporarily located in Sections 1 through 5, adjacent to the text it is explaining. The text will return to this annex after the completion of balloting.

E.4 Excluded Utilities

Many utilities were evaluated by the developers of the standard; more utilities were excluded from the standard than included. The following list contains some common UNIX system utilities that were considered by the developers, but not included as part of the UPE or POSIX.2.

Editor's Note: This list will eventually be integrated with Annex E.4 in the base standard.

`SCCS` Both SCCS and RCS were originally considered as part of a version control utilities portion of the scope. However, this aspect of the UPE was abandoned by the working group. Their was broad consensus that SCCS was the most widely implemented system and that it had some technical advantages over RCS (other than

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

its difficult user interface). Some of the implementors objected to the notion of rewriting the very complicated SCCS utilities strictly from a standard description and indicated that they would oppose any such utilities in the standard unless their original author would put the source code into the public domain. The author was unwilling to do so. The IEEE may form a working group in the future to address this issue separately from the UPE.

`compress`, `uncompress`, `zcat`

These were proposed for an earlier draft, but met with too many objections to be included. The complex algorithm in historical implementations was expressed in the draft in terms of a C-language program, but many objections were raised concerning its accuracy and exact duplication of historical output (including one common historical ''bug''). Other objections stated that a C program was insufficient as a specification and that a detailed description in the Output Files subclause was needed instead. Consideration was given to removing just the algorithm and saying that `compress` was defined only at the user interface and was not necessarily portable between systems, but straw polls indicated that that would cause other objections as well. The working group found that it could not adequately address the requirements of the balloters on these points. In order to achieve consensus on an expeditious basis, these three utilities have been removed and could be considered for a future version of the standard; however, it is more likely that they would be part of the application portability sections, and not user portability.

`emacs` The community of `emacs` editing enthusiasts was quite adamant that the full `emacs` editor not be included in POSIX. 2, because it was concerned that an attempt to standardize this very powerful environment would encourage vendors to ship versions conforming strictly to the standard, but lacking the extensibility required by the community. The author of the original `emacs` program also expressed his desire to omit the program. Furthermore, there were a number of historical UNIX systems that did not include `emacs`, or included it without supporting it, but there were very few that did not include and support `vi`. Since `emacs` is not in any of the four base documents, this finished any arguments concerning its inclusion.

`lint` This was proposed for an earlier draft, in the form of a `lint89` utility, but met with too many objections to be included. This technology is of declining importance because of the introduction of prototypes in the C Standard {7}. In particular, it is hard to specify the degree of checking for possible error conditions in programs in any compiler and specifying what `lint` would do in these cases is equally difficult.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

It is fairly easy to specify what a compiler does. It requires specifying the language, what it does with that language, and stating that the interpretation of any incorrect program is unspecified. Unfortunately, with `lint`, it is required to specify what to do with erroneous programs. Since the number of possible errors and questionable programming practices is infinite, one cannot require `lint` to detect all errors of any given class. As the text of the drafts was tightened up, it was found that fewer and fewer useful statements could be made about `lint`.

Additionally, some vendors complained that since many compilers are distributed in a binary form without a `lint` facility (because the C Standard {7} does not require one), building the UPE from scratch will be much harder. Rather than being able to plug in a standard compiler component (simply by providing `c89` as an interface), source to that compiler would most likely need to be modified to provide the `lint` functionality. This was considered a major burden on system providers for a very small gain to developers (users).

`m4` The working group did not find that this macro processor had sufficiently wide usage for standardization. It is used in a small number of application installation tools and specialized applications, but it is rarely used by interactive users. Thus, it was omitted from this supplement as being out of the scope of work. There seems to be some interest within ISO to include `m4` in the final ISO version of POSIX.2, so it will be evaluated as part of the working group's P1003.2b revision effort.

`passwd` This utility was proposed for an earlier draft, but met with too many objections to be included. There were various reasons:

— Changing a password should not be viewed as a command, but as part of the login sequence. Changing one's password should only be done while a trusted path is in effect.

— Even though the text in previous drafts was intended to allow a variety of implementations to conform, the security policy for one site may differ from another site running with identical hardware and software. One site might use password authentication while the other did not. Vendors could not supply a `passwd` utility that would conform to the standard for all sites using their system.

— This is really a subject for the POSIX.7 system administration working group or POSIX.6 security working group.

`sdiff` This ''side-by-side diff'' utility has little portable application use and it was excluded from the UPE because it did not meet the requirement that it be part of one of the base documents.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

`shar`	Any of the numerous ‘‘shell archivers’’ were excluded from the UPE because they did not meet the requirement that they be parts of one of the base documents. Early drafts contained a `pshar` (‘POSIX shar’’) utility, but that was dropped in the face of the base document requirement (as well as considerable opposition to its complexity).

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

Annex F
(inform ative)

Balloting Instructions

This annex will not appear in the final standard. It is included in the draft to provide instructions for balloting that cannot be separated easily from the main document, as a cover letter might.

If you have received a copy of this draft before February 1992 it is important that you read this annex, whether you are an official member of the P1003.2a Balloting Group or not; comments on this draft are welcomed from all interested technical experts. **Your ballot is due to the IEEE office by 24 January 1992. This is not the date to postmark it—it is the date of receipt.**

Summary of Draft 8 Instructions

This is the third ''recirculation draft'' of P1003.2a. The recirculation procedure is described in this annex. For this recirculation, we are accepting objections against any normative changes that occurred from Draft 7 to Draft 8 and the contents of the Unresolved Objections List, provided as a separate document from the draft.

Send your ballot and/or comments to:

IEEE Standards Office
Computer Society Secretariat
ATTN: P1003.2a Ballot (Anna Kaczmarek)
P.O. Box 1331
445 Hoes Lane
Piscataway, NJ 08855-1331

It would also be very helpful if you sent us your ballot in machine-readable form. Your official ballot must be returned via mail to the IEEE office; if we receive only the e-mail or diskette version, that version will not count as an official document. However, the online version would be a great help to ballot resolution. We can accept e-mail to the following address:

`hlj@Posix.COM` or **`uunet!posix!hlj`**

or IBM PC 3.5-inch/720K diskette (plain file) or Macintosh 3.5-inch diskette (plain text file [preferred], Word, or Write) or Sun-style QIC-24 cartridge tapes to:

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

Hal Jespersen, Chair P1003.2
POSIX Software Group
447 Lakeview Way
Redwood City, CA 94062

Some degree of judgment is required in determining what actually changed in Draft 8. Use the diff marks as a guide, but they will frequently mark text that has no real normative changes. Please limit your objections to the actual changes: for example, if we change the `foo -x` option to `-y`, don't use that as an opportunity to object that we have no `-z` option. Your objection should only address why the `x` to `y` change is a problem. (We have been balloting for a long time now and it is time to tighten the consensus and finish this up.) If you find problems unrelated to changes, submit them as comments and they will be considered seriously in that category. Thanks for your cooperation on this.

We are not accepting objections about how it is difficult to synchronize the descriptions between POSIX.2 Draft 11.2 and POSIX.2a Draft 8. We've chosen this slightly more difficult procedure to accelerate the balloting process, without waiting for the nits to be rung out of Draft 11.2. Rest assured that POSIX.2a will never be finally completed until after POSIX.2 is completely done.

Background on Balloting Procedures

The Balloting Group consists of over 160 technical experts who are members of the IEEE or the IEEE Computer Society; enrollment of individuals in this group has already been closed. There are also a few ''parties of interest'' who are not members of the IEEE or the Computer Society. Members of the Balloting Group are required to return ballots within the balloting period. Other individuals who may happen to read this draft are also encouraged to submit comments concerning this draft. The only real difference between members of the Balloting Group and other individuals submitting ballots is that *affirmative* ballots are only counted from Balloting Group members who are also IEEE or Computer Society members. (There are minimum requirements for the percentages of ballots returned and for affirmative ballots out of that group.) However, objections and nonbinding comments must be resolved if received from any individual, as follows:

(1) Some objections or comments will result in changes to the standard. This will occur either by the publication of a list of changes or by the republication of an entire draft. The objections/comments are reviewed by a team from the P1003.2 working group, consisting of the Chair, Vice Chair, the Chair of the TCOS Standards Subcommittee, and one or more Technical Reviewers. The Technical Reviewers each have subject matter expertise in a particular area and are responsible for objection resolution in one or more sections.

(2) Other objections/comments will not result in changes.

   (a) Some are misunderstandings or cover portions of the document (front matter, informative annexes, rationale, editorial matters, etc.) that are not subject to balloting.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

(b) Others are so vaguely worded that it is impossible to determine what changes would satisfy the objector. These are referred to as *Unresponsive* . (The Technical Reviewers will make a reasonable effort to contact the objector to resolve this and get a newly worded objection.) Further examples of unresponsive submittals are those not marked as either *Objection* or *Comment* ; those that do not identify the portion of the document that is being objected to (each objection must be separately labeled); those that object to material in a recirculation that has not changed and do not cite an unresolved objection; those that do not provide specific or general guidance on what changes would be required to resolve the objection.

(c) Finally, others are valid technical points, but they would result in decreasing the consensus of the Balloting Group. (This judgment is made based on other ballots and on the experiences of the working group through almost five years of work and fifteen drafts preceding this one.) These are referred to as *Unresolved Objections* . Summaries of unresolved objections and their reasons for rejection are maintained throughout the balloting process, are circulated to members of the Balloting Group for their consideration, and are presented to the IEEE Standards Board when the final draft is offered for approval. Unresolved objections are only circulated to the balloting group when they are presented by members of the balloting group or by parties of interest. Unsolicited correspondence from outside these two groups may result in draft changes, but are not recirculated to the balloting group members.

Please ensure that you correctly characterize your ballot by providing one of the following:

(1) Your IEEE member number

(2) Your IEEE Computer Society affiliate number

(3) If (1) or (2) don't apply, a statement that you are a ''Party of Interest''

Ballot Resolution

The general procedure for resolving ballots is:

(1) The balloting cuts off on 24 January 1992. This is a receipt date at the IEEE , not a postmark date. (Please do not telephone or FAX on 24 January 1992 and say that your specific comments will come later; late-arriving comments will not be considered as objections.) We will accept comments after that date, including direct e-mail to the working group officers or the Technical Reviewers, but they will be treated as comments only—not objections. And we don't guarantee a written response to these late submissions.

(2) The ballots are put online and distributed to the Technical Reviewers.

(3) If a ballot contains an objection, the balloter will be contacted individually by telephone, letter, or e-mail and the corrective action to be taken

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

will be described (or negotiated). The personal contact will most likely not occur if the objection is very simple and obvious to fix or the balloter cannot be reached after a few reasonable attempts. Repeated failed attempts to elicit a response from a balloter may result in an objection being considered unresponsive, based on the judgment of the working group chair. Once all objections in a ballot have been resolved, it becomes an affirmative ballot.

(4) If any objection cannot be resolved, the entire ballot remains negative.

(5) Once more than seventy-five percent of the ballots received (that had voted either affirmative or negative) have been turned affirmative, two lists are published to the entire balloting group: the detailed list of approved changes and the list of unresolved objections, along with our reasons for rejecting them. This is known as a *recirculation* . You have minimum of ten days (after an appropriate time to ensure the mail got through) to review these two lists and take one of the following actions:

(a) Do nothing; your ballots will continue to be counted as we have classified them, based on items (3) and (4).

(b) Explicitly change your negative ballot to affirmative by agreeing to remove all of your objections from the unresolved list.

(c) Explicitly change your affirmative ballot to negative based on your disapproval of either of the two lists you reviewed. If an issue is not on one of the two lists, new objections about this are not allowed. Negative ballots that come in on recirculations cannot be cumulative. They shall repeat any objections that the balloter considers unresolved from the previous recirculation. Ballots that simply say ‘‘and all the unresolved objections from last time’’ will be declared unresponsive. Ballots that are silent will be presumed to fully replace the previous ballot, and all objections not mentioned on the most current ballot will be considered as successfully resolved.

(6) The list of changes will frequently be a new draft document with the changes integrated. This is not a requirement, however, and a small number of changes may prompt merely a change list approach to recirculation.

(7) A copy of all your objections and our resolutions will be mailed to you. You can receive the full package of all resolutions from all ballots by contacting the IEEE Standards Office (who will probably charge you for the copying involved). If you don’t agree with one of our resolutions and haven’t been contacted personally before you receive this list, please accept our apologies and submit a new ballot against the new draft during the recirculation period.

(8) If at the end of the recirculation period there remain greater than seventy-five percent affirmative ballots, and no new objections have been received, a new draft is prepared that incorporates all the changes. This draft and the unresolved objections list go to the IEEE Standards Board

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

for approval. If the changes cause too many ballots to slip back into negative status, another resolution and recirculation cycle begins.

Balloting Guidelines

This section consists of guidelines on how to write and submit the most effective ballot possible. The activity of resolving balloting comments is difficult and time consuming. Poorly constructed comments can make that even worse.

We have found several things that can be done to a ballot that make our job more difficult than it needs to be, and likely will result in a less than optimal response to ballots that do not follow the form below. Thus it is to your advantage, as well as ours, for you to follow these recommendations and requirements.

If a ballot that significantly violates the guidelines described in this section comes to us, we will determine that the ballot is unresponsive, and simply ignore all the material in it.

Secondly, objections that don't contain a specification so that the correction to resolve the objection ''can be readily determined'' are also unresponsive and will be ignored.

(If we do recognize a ballot that is generally ''unresponsive,'' we will try to inform the balloter as soon as possible so he/she can correct it, but it is ultimately the balloter's responsibility to assure the ballot is responsive.)

Typesetting is not particularly useful to us. And please do not send handwritten ballots. Typewritten (or equivalent) is fine, and if some font information is lost it will be restored by the Technical Editor in any case. If you use `nroff`, you will include extraneous spacing and sometimes backspaces and overstrikes; if you really must use `nroff`, please turn off hyphenation and line adjusting:

```
.hy 0
.na
```

and run the output through `col -b` to remove all the overstrikes. (Also remember that backslashes and leading periods and apostrophes in your text will be treated impolitely by the *`roff` family). The ideal ballot is formatted as a ''flat ASCII file,'' without any attempt at reproducing the typography of the draft and without embedded control characters or overstrikes; it is then printed in Courier (or some other typewriter-like) font for paper-mailing to the IEEE Standards Office and simultaneously e-mailed to the working group Chair.

Don't quote others' ballots. Cite them if you want to refer to another's ballot. If more than one person wants to endorse the same ballot, send just the cover sheets and one copy of the comments and objections. [Note to Institutional Representatives of groups like X/Open, OSF, UI, etc.: this applies to you, too. Please don't duplicate objection text with your members.] Multiple identical copies are easy to deal with, but just increase the paper volume. Multiple almost-identical ballots are a disaster, because we can't tell if they are identical or not, and are likely to miss the subtle differences. Responses of the forms:

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

— ‘‘I agree with the item in <someone>’s ballot, but I’d like to see this done instead’’

— ‘‘I am familiar with the changes to `foo` in <someone>’s ballot and I would object if this change is [or is not] included’’

are very useful information to us. If we resolve the objection with the original balloter (the one whose ballot you are referencing), we will also consider yours to be closed, unless you specifically include some text in your objection indicating that should not be done.

Be very careful of ‘‘Oh, by the way, this applies <here> too’’ items, particularly if they are in different sections of the document that are likely to be seen by different reviewers. They are probably going to be missed! Note the problem in the appropriate section, and cite the detailed description if it’s too much trouble to copy it. The reviewers don’t have time to read the whole ballot, and only read the parts that appear to apply to them. Particularly where definitions are involved, even if the change really belongs in one section but the relevant content is in another, an extra cross-reference would be indicated. If you wish to endorse someone else’s ballot, either in whole or part, be specific about whether you will be automatically satisfied if they are satisfied. If you will not necessarily be satisfied if they are, your ballot could be deemed unresponsive because it does not give achievable conditions under which your ballot could be converted to affirmative. You then must give the conditions under which you would be satisfied as well. If you would be satisfied in some areas and not in others, it is best to specifically point to each specific objection in the ballot you point to, giving the conditions for each.

Please consider this a new ballot that should stand on its own. Please do not make backward references to your ballots for previous drafts—include all the text you want considered here, because the Technical Reviewer may not have your old ballot. And, the old section and line numbers won’t match up anyway. If one of your objections was not accepted exactly as you wanted, it will not be useful to send in the exact text you sent before; read the nearby Rationale section and come up with a more compelling (or clearly-stated) justification for the change.

Please be very wary about global statements, such as ‘‘all of the arithmetic functions need to be defined more clearly.’’ Unless you are prepared to cite specific instances of where you want changes made, with reasonably precise replacement language, your ballot will be considered unresponsive.

Ballot Form

The following form is recommended. We would greatly appreciate it if you sent the ballot in electronic form in addition to the required paper copy. Our policy is to handle all ballots online, so if you don’t send it to us that way, we have to type it in manually. For the last POSIX.2 ballot, only one or two balloters could not accommodate us on this and thus we had very little typing to do. See the first page of this Annex for the addresses and media. As you’ll see from the following, formatting a ballot that’s sent to us online is much simpler than a paper-only ballot.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

The ballot should be page-numbered, and contain the name, e-mail address, and phone number(s) of the objector(s). (If you send us only a paper copy, make sure this information appears on every page; electronic ballots just need it once, in the beginning.) The lines before the first dashed line are a page header, and should only appear once on each page. Please leave adequate (at least one inch) margins on both sides. Each objection/comment/editorial comment should be sequentially numbered, not in individual ranges [i.e., not Objection #1, Comment #1]

Since we deal with the ballots online, there is no longer any requirement to put only one objection or section per page.

Don't format the ballot as a letter or document with its *own* section numbers. These are simply confusing. As shown below, it is best if you cause each objection and comment to have a sequential number that we can refer to amongst ourselves and to you over the phone. Number sequentially from 1 and count objections, comments, and editorial comments the same; don't number each in its own range. If you don't do this, we'll number them ourselves, but you won't know what numbers we're using.

Please precede each objection/comment with a little code line (if you don't, we'll have to do it ourselves):

@ <*section*> . <*clause*> <*code*> <*seqno*>

where:

@ At-sign in column 1 (which means no @'sin any other column 1's).

<*section*> The major section (chapter or annex) number or letter in column 3. Use zero for Global or for something, like the frontmatter, that has no section or annex number.

<*clause*> The clause number (second-level header). Please do not go deeper than these two levels. In the text of your objection or comment, go as deep as you can in describing the location, but this code line uses two levels only.

<*code*> One of the following lowercase letters, preceded and followed by spaces:

- o Objection.
- c Comment or Editorial Comment.

<*seqno*> A sequence number, counting all objections and comments in a single range.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

Objection:

```
Balloter Name              (202)555-1212       page  x of nn.
E-Mail Address             FAX:   Fax Number
Balloter2 Name                    (303)555-1213
E-Mail Address2            FAX:   Fax Number2
-----------------------------------------------------------------
@ x.y o seq#
<Seq#> Sect x.y OBJECTION. page xxx, line zzz:
```

Problem:

A clear statement of the problem that is observed, sufficient for others to understand the nature of the problem. (Note that you should identify problems by section, page, and line numbers. This may seem redundant, but if you transpose a digit pair, we may get totally lost without a cross-check like this. Use the line number where the problem starts, not just where the section itself starts; we sometimes attempt to sort objections by line numbers to make editing more accurate. If you are referring to a range of lines, please don't say ''lines 1000ff;'' use a real range so we can tell where to stop looking. We will really love you if you can manage to include enough context information in the problem statement (such as the name of the utility) so we can understand it without having the draft in our laps at the time. (It also helps you when we e-mail it back to you.) If you are objecting to an action in the Unresolved Objections List, use the section/page/line number reference for the appropriate place in the standard; don't refer to the UOL except to cite its number and for clarification of your points.

Action:

A precise statement of the actions to be taken on the document to resolve the objection above, which if taken verbatim will completely remove the objection.

If there is an acceptable range of actions, any of which will resolve the problem for you if taken exactly, please indicate all of them. If we accept any of these, your objection will be considered as resolved.

If the Action section is omitted or is vague in its solution, the objection will be reclassified as a nonbinding comment. The Technical Reviewers, being human, will give more attention to Actions that are well-described than ones that are vague or imprecise. The best ballots of all have very explicit directions to substitute, delete, or add text in a style consistent with the rest of the document, such as:

```
Delete the sentence on lines 101-102:
      "The implementation shall not ... or standard error."
On line 245, change "shall not" to "should not".
After line 103, add:
       -r    Reverse the order of bytes read from the file.
```

Some examples of poorly-constructed actions:

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

```
Remove all features of this command that are not supported by BSD.

Add -i.

Make this command more efficient and reliable.

Use some other flag that isn't so confusing.

I don't understand this section.

Specify a value--I don't care what.
```

Objection Example:

```
Hal Jespersen              (415) 364-3410            page  3 of 17.
UUCP: hlj@Posix.COM  FAX:   (415) 364-4498
---------------------------------------------------------------------
@ 2.6 o 23
23. Sect 2.6 OBJECTION. page 77, line 1217:

Problem:

The EDITOR environment variable is not used as stated
in my company.  This description would cause hundreds
of my shell scripts to break.

Action:

Change the first sentence on line 1217 to:

      The e-mail address of the editor of the user's
      favorite POSIX standard.

-----------------------
@ 3.1 o 24
24. Sect 3.1.6 OBJECTION. page 123, line 17:

Problem:

I support UO 3.01-999-6 concerning the objection to the                7
definition of "operator".                                              7
This definition would cause great hardship to the users                7
of the systems I develop.                                              7
I feel your rationale for rejection was inappropriate                  7
because you overlooked the following technical points [etc.]...        7

Action:

Change the term "operator" to "operation-symbol" in this
definition and globally throughout Section 3.
```

Comment:

```
---------------------------------------------------------------------
@ x.z c seq#
<Seq#> Sect x.z COMMENT. page xxx, line zzz:
```

A statement of a problem that you might want to be resolved by the reviewer, but which does not in any way affect whether your ballot is negative or positive. The form for objections is not required, but it increases the probability that your comment will have an effect on the final document.

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

Although there may be questions to you or responses on the topic, no changes in the drafts are required by a comment, although it will be looked at to determine whether the concern should be addressed. It is possible to abuse this rule and label all of your comments as objections, but it is a significant disservice to the individuals who are volunteering their time to address your concerns.

Remember that any issue concerning the pages preceding page 1 (the Frontmatter), Rationale text with shaded margins, Annexes, NOTES in the text, footnotes, or examples will be treated as a nonbinding comment whether you label it that way or not, but it would help us if you'd label it correctly.

Editorial Comment:

```
-------------------------------------------------------------
@ x.z c seq#
<Seq#> Sect x.z EDITORIAL COMMENT. page xxx, line zzz:
```

These are for strictly editorial issues, where the technical meaning of the document is not changed. Examples are: typos; misspellings; English syntax or usage errors; appearances of lists or tables; arrangement of sections, clauses, and subclauses (except where the location of information changes the optionality of a feature). Marking these as comments but indicating that they are editorial speeds the process.

Please be aware that after balloting concludes the document will be subjected to more sets of editors at the IEEE and ISO who are empowered to make broad editorial changes and rewording (for example, to get the text ready for translation into French.)

Thank you for your cooperation in this important balloting process.

Hal Jespersen

Copyright © 1991 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.