WG14 N399/X3J11 94-084
December 7, 1994
Rex Jaeschke
rex@aussie.com

Guidelines for making a submission to the C9X revision

A. Introduction

Work is about to begin on a revision of the C Standard. Most of this work will be done by a group of about 30-40 people, all of whose time and expenses are funded by their employers. More than a few of these people represent small companies or are self-employed. This group will also meet only three times per year, five days per meeting. In order to make the best use of their time and to meet the revision schedule, it is necessary to streamline the proposal submission process. Specifically, EACH submission MUST have a cover sheet attached that summarizes that submission. Submissions WITHOUT this cover sheet WILL BE RETURNED to the submittor for completion. The cover sheet is intended not only to provide a summary for the committee members reviewing submissions but also to get the submittor focussed.

Each distinct topic MUST be submitted as a separate proposal with its own cover sheet. A submission containing multiple topics that are NOT closely related WILL BE RETURNED to the submittor for further processing.

B. Cover Sheet

Before filling out the following cover sheet, read the instructions in the next section.

```
====================== Cover sheet starts here ============

          Document Number:  WG14 N___/X3J11 __-___

                      C9X Revision Proposal
                      =====================

Title: ______________________________________________________
Author: _____________________________________________________
Author Affiliation: _________________________________________
Postal Address: _____________________________________________
E-mail Address: _____________________________________________
Telephone Number: ___________________________________________
Fax Number: _________________________________________________
Sponsor: ____________________________________________________
Date: _______________________________________________________
Proposal Category:
   __ Editorial change/non-normative contribution
   __ Correction
   __ New feature
   __ Addition to obsolescent feature list
   __ Addition to Future Directions
   __ Other (please specify)  ______________________________
Area of Standard Affected:
   __ Environment
   __ Language
   __ Preprocessor
   __ Library
      __ Macro/typedef/tag name
      __ Function
      __ Header
   __ Other (please specify)  ______________________________
Prior Art: __________________________________________________
Target Audience: ____________________________________________
_____________________________________________________________
Related Documents (if any): _________________________________
_____________________________________________________________
_____________________________________________________________
Proposal Attached: __ Yes __ No, but what's your interest?
Abstract: ___________________________________________________
_____________________________________________________________
...

Proposal: ___________________________________________________
_____________________________________________________________
...

====================== Cover sheet ends here =============
```

C. Filling Out the Cover Sheet

Here are instructions for filling out the cover sheet:

Document Number -- This number will be assigned once a submission is determined to be complete, from a packaging point of view. That is, it has a complete cover sheet and the proposal details are clearly stated. Once a submission is accepted for consideration, the submittor will be so informed and will be given the corresponding document number.

Title -- Make it short (less than one line of text) and to the point. For example ''Add more E* macros to errno.h'' or ''Stricter type checking for enumerated types.''

Author -- State your name and title in the way in which you wish to be addressed in future correspondence.

Author Affiliation -- Who are you officially representing? Your employer? A user organization? Yourself?

Postal Address -- Self explanatory. Please include country.

E-mail Address -- Please specify in the form user@xxx.yyy

Telephone Number -- Please specify your telephone number using the ITU International Format. This format is: +country-code area-code number. For example, a number in England looks like +44 1923 813541 while one in the U.S. looks like +1 703 8600091.

Fax Number -- See Telephone Number above.

Sponsor -- For a submission to be accepted for debate, it MUST have a sponsor. That sponsor can either be the convener of WG14, the WG14 committee itself, or a national member body of WG14 (such as the ANSI C committee X3J11 in the US, JSA in Japan, BSI in the UK, DS in Denmark). If this field is left empty, a sponsor will be assigned provided a proposal is accepted for review.

Date -- Date of submission in the format YYYY-MM-DD. (Dates like 4/7/95 are ambiguous; is it April or July?)

Proposal Category -- Check one of the following categories; 'Editorial change/non-normative contribution,' 'Correction,' 'New feature,' 'Addition to obsolescent feature list,' 'Addition to Future Directions,' and 'Other (please specify).' Since these categories are mutually exclusive, you must check only one per submission.

273

Questions about what the standard says on a particular issue are considered requests for interpretation and should NOT be submitted via this procedure. They should be submitted as Defect Reports (DRs) instead. Neither the DR process nor the revision process is intended to provide responses to questions of a tutorial nature.

Area of Standard Affected -- Check one or more of the following subcategories: 'Environment,' 'Language,' 'Preprocessor,' 'Library,' or 'Other.' Typically, these categories are mutually exclusive. If you check the subcategory 'Library,' then select any combination of the subsubcategories 'Macro/typedef/tag name,' 'Function,' and 'Header.'

Prior Art -- Indicate which implementations (if any) already provide the support being proposed. Are these commercial implementations or at least being used in the commercial marketplace? What do you estimate their installed user base to be? This is a very important issue since one of the guiding principles for the revision is ''We should accept only those concepts that have some prior art. Unless some proposed new feature addresses an evident deficiency that is actually felt by more than a few C~programmers, we should entertain no new inventions.''

If you were involved in adding the proposed extension to an implementation, indicate alternative approaches you considered and say why they were rejected.

Target Audience -- Who is the proposed change/addition aimed at? For example: Numerical programmers? Programmers using parallel or vector architectures? Those interested in interlanguage support? Developers of embedded real-time systems?

Related Documents (if any) -- If this is a revision of a previous submission, specify the number of the previous submission.

Specify the numbers/titles of any documents known to have a potential impact on this proposal or that possibly might be impacted by it. Give references to related official/de facto standards. However, keep in mind that saying something like ''This facility is provided in Ada,'' won't get you far if you don't also provide details of how it is specified in that language. We don't have expertise in all fields.

Proposal Attached -- Check 'No' if you just want to gauge the level of interest. For example, there is no point in researching in detail and submitting a 10-page document specifying the addition of indexed-sequential file support to the library if there is insufficient interest on the committee’s part to begin with.

Check ‘Yes’ if a sufficiently-complete and detailed proposal is attached.

Abstract -- In a few clearly-written paragraphs, describe the problem. If you want a specific solution, state what that is. If you can live with any reasonable solution state that. If you can suggest alternatives, do so, highlighting the pros and cons of each.

Proposal: See the next section

D. The Proposal

Keep the following in mind when writing the detailed proposal:

1) Structure the proposal in a rational and overt fashion.

2) Submissions that are complete and clearly written will very likely get more attention.

3) Having good access to a copy of the C standard and some working knowledge of using it will be advantageous in preparing your submission.

4) It is very useful to cite ISO section/clause numbers and page numbers when referencing the C standard. Please do NOT use references to the old ANSI C standard X3-159. That standard has been withdrawn and replaced by the one from ISO. They use different section numbering.

5) Ultimately, any change or addition to the standard requires changed or new words. Provide replacement/new text if you can.

6) Make code examples/fragments short and to the point. Unless the code uses completely new syntax, test your examples with some existing compiler to ensure that they are free of syntax errors, etc. Include runtime output where appropriate.

7) For new proposals, the following information is useful:

Problem Statement -- What is the problem you’re solving.

Conceptual Model -- Describe your solution in abstract terms, e.g., ``this library performs the matrix operation C = A*B’’. This information may be included in the rationale.

275

Semantics -- A technical description of what this means (or does). For example: ``An identifier denotes an object, a function, or one of the following entities that will be described later: a tag or a member of a structure, union, or enumeration; a typedef name; a label name; a macro name; or a macro parameter.''

Constraints -- When is the feature invalid and/or constrained. For example: ``In translation phases 7 and 8, an identifier shall not consist of the same sequence of characters as a keyword.''

Behavior -- Describe, if any, features that are: undefined, indeterminate, implementation-defined.

Syntax, Synopsis, Keywords, Tokens, Operators, Directives -- What is the syntax of the feature. For functions, what is the function prototype. Are there new keywords (identifiers, e.g., "complex"), tokens (punctuation, e.g., "<>"), operators (like functions but take arbitrary objects, e.g., "sizeof"), or preprocessor directives (e.g., "#pragma").

Rationale -- Why this feature is needed and why this solution was developed. This text will become part of the rationale. This may duplicate some of your words in the Problem Statement and Conceptual Model above.

Implementation Issues -- For vendors (compiler writers) to implement this feature, how might it affect their implementations. For example: ``it adds ... to the function prologue'', ``it adds a function to the standard C library'', ``it requires function X to be called at program startup''. Are there issues (internationalization) that affect the operation of this feature from one country to another?

Language Compatibility Issues -- If there is a feature like this in some other language, how is it the same or different. If this is a feature that is first implemented in C, are there issues that would make it easy or difficult to implement in other languages?

Subsetting -- If this feature were not adopted within the standard, but some vendors chose to implement this, how could programmers determine if this feature were included at compile-time and/or at run-time. Some common mechanisms are: ``the feature is included only if the header "xyz.h" is included'', ``the feature is present if the macro "_XYZ_H_" is defined'', ``the feature is present if the function call "xyz(...)" returns 1''.

Advice: Generally, the more important aspects of *complicated* proposals are Problem Statement, Conceptual Model, and Semantics. If your proposal is to be developed over time, resolve these issues first. Then develop wording changes. Generally, the syntax specification is the *least* important aspect of a proposal, but a *tentative* syntax is useful for discussion.

E. Where To Send Submissions

Submissions should be sent to your national member body of WG14.

In the US:

Rex Jaeschke
Chair X3J11
2051 Swans Neck Way
Reston, VA 22091

Tel: +1 703 860-0091
E-mail: rex@aussie.com

In Japan:

NODA Makoto, Chair of ITSCJ/SC22/C WG
Information Technology Standards Commission of Japan
Kikai Shinko Building 308-3
3-5-8 Shiba-Koen, Minato-ku
Tokyo 105

Tel: +81-3-3431-2808
Fax: +81-3-3431-6493
E-mail: c.wg@nec.co.jp

In the UK:

Neil B Martin
Convener BSI IST/5/-/14 C Language Panel
9 Holst Cres
Browns Wood
Milton Keynes
MK7 8DQ

Tel+Fax: +44 (0)908-640778
E-mail: bsi_neil@cix.compulink.co.uk

In Denmark:

DS S142 u22 A13
C/O Keld Simonsen, DKUUG
Fruebjergvej 3
DK-2100 Kobenhavn O

Tel: +45 3917-9944
Fax: +45 3120-8948
E-mail: u22a13@dkuug.dk

For other countries, send to any of the national member contacts above.

F. Electronic Submissions

It is preferred that submissions be made in electronic form. Having the submissions on-line enables them to be distributed via e-mail for informal committee review. And if the format is predictable, tools can be used to extract summary and key information. Therefore,

1) DON'T change the order of fields.

2) DON'T delete any fields.

3) Add more lines to the descriptive fields as needed.

4) You may eliminate extra underscores ("_") ONLY in the descriptive fields, i.e., not the check-off items. Use "Y" for yes, "N" for no, "?" for don't know, leave blank if not applicable.

The following guidelines will help us process your electronic submission.

------------------- ASCII SPECIFICATION ----------------

1. CHARACTER CODES

Please read the following carefully with regard to characters you should or should not include in your files:

0: Null. Unacceptable. No need for delays. This messes up many editors.

2: Start of Text: Unacceptable. Some systems use this to indicate the "bold" font.

8: Backspace. Not recommended since some printers and many text editors won't give you the overstrike you desire. Use "col -b -x" on UNIX (or MKS on DOS) to remove these.

9: Horizontal Tab. Not recommended because: (1) tab stops vary, (2) some systems can't do tab stops, (3) even if you think they are 8 spaces, aligned text gets messed up when exchanging E-mail (e.g., the "> " that is prepended to the lines of a response). Use "pr -e8 -t" on UNIX (or MKS on DOS) to expand these.

10: Line Feed. OK if this is part of a newline (whatever that is for your system). Don't use this as an "index", i.e., down one line but same column.

11: Vertical Tab. Don't use these. You can remove these with "tr -d '\013'" on UNIX (or MKS on DOS).

12: Form Feed. Not preferred. Use "pr -t" to expand the the form feeds into 66 lines/page documents. Add the "cut here" marks (see below).

13: Carriage Return. OK if this is part of a newline (e.g., on DOS systems). Don't use this for overstriking, e.g., return to the beginning of the current line and overstrike.

14: Shift Out. Only use 7-bit ASCII with only a G0 set.

15: Shift In. Only use 7-bit ASCII with only a G0 set.

26: Substitute (CTRL-Z). Unacceptable. This gets included on some DOS systems. This should be removed.

27: Escape. Unacceptable. These cause problems when rendering the document on nonconforming systems.

32: Space. OK. Preferred over tabs.

33-126: printable characters. Note: the special characters, e.g., "[]{|}`~$#@^_", should all represent ASCII (ISO 8859-1), not ISO 646 variants.

127: Delete. Unacceptable. No need for delays (or rubouts!). This may confuse some editors.

On UNIX-like environments, The following command will probably fix all problems:

```
    cat file |
        tr -d '[\001-\007]\013[\016-\037][\177-\377]' |
        col -b -x |
        pr -e8 -t >file.out
```

2. PAGE BREAKS

If you have page breaks, it is best to remove them because it makes editing the submission easier. If this is impossible, format the document as if it were going to a 66 lines/page printer. Add a "---- cut here ----" line before the first page and after the last page. The following will take the output from above and bracket it with "---- cut here ----" marks.

```
     cat file.out |
        (
            echo "-------------- cut here ------------"
            cat -
            echo "-------------- cut here ------------"
        ) >file.fmt
```

3. FORMATTING ISSUES

If possible, please use a maximum line length of 60 columns. Although most systems can print 80 columns, we may need to include our own information in the margins. Using 60 columns means much less editing for us.

Please format your documents so they are left justified and ragged right. The extra inter-word spacing makes it harder for us to edit. You may begin each line with a space (see E-mail issues, below) as a left margin.

Turn off hyphenation. Again, hyphenating words makes more work. Of course, you can use words that ordinarily have hyphens in them, e.g., "E-mail".

Regarding quotations:

``Please use this style.''

"Don't use this style."

This makes it easier to edit.

4. E-MAIL ISSUES

The committee may pass documents around via E-mail, even though you may submit you document via floppy. You should be aware of the following hazards.

(1) Many mail systems treat the word "From " at the beginning of a line specially (they may rewrite it as ">From"). Either put a leading space at the beginning of each line, OR put an "X" in front of every line.

(2) Although many systems support MIME, assume that the recipients don't have MIME. Postscript *may* be a better choice than MIME. IMPORTANT: Ask before you send.

Example 1:

```
 From may be a problem with some mailers.
 Other words are not a problem.
```

Example 2:

230

XFrom may be a problem with some mailers.
XOther words are not a problem.

Example 3 (wrong):
From may be a problem with some mailers.
Other words are not a problem.

5. FONT CONVENTIONS

If you are sending in ASCII, the following conventions should be used for indicating certain common fonts/styles:

BOLD: Put all the characters in capitals.

italic: Surround the whole phrase with asterisks.

"courier": When specifying literal text that is embedded within text, e.g., the statement "int a[10];", surround the whole phrase in double quotes (don't use `` or '' for this).

Obviously, this doesn't work for *every* case, but for 90% of the time it will (and makes it easier to edit). If this is impossible, use you own conventions and document them at the beginning of your submission.

If you are sending Postscript, please use only the following fonts: Times Roman, Times Italic, Times Bold, Times Bold Italic, Courier, Courier Bold, Courier Italic, Courier Bold Italic, Helvetica, Helvetica Italic, Helvetica Bold.

------------------- END OF ASCII SPECIFICATION--------------

==================== END OF DOCUMENT =======================