This package contains:
1.   Minutes of the review
2.   Original Call for review
3.   Comment list

Minutes of ALMA Software Documentation and Coding Standards Review
2001-01-10

Invited reviewers:

 Giorgio Filippi (ESO)
 Brian Glendenning (NRAO)
 Birger Gustafsson (ESO)
 Koh-Ichiro Morita (NRO)
 Martin Pokorny (NRAO)
 Gianni Raffi (ESO - Meeting Chair)
 Fritz Stauffer (NRAO)
 Gie Han Tan (ESO)

Principal author:
 Alan Bridger (ATC)

The following Draft documents were reviewed. They are available under:
http://www.alma.nrao.edu/development/computing/docs/memos/index.html

The ALMA Documentation Standard
[A Document template, including a MS Word style file]
 Alan Bridger

C Coding Standards
 Alan Bridger, Mick Brooks, Jim Pisano

C++ Coding Standards
 Alan Bridger, Jim Pisano

Conclusion:

The Review Panel approves the content of the three reviewed documents. These
should therefore become official documents approved by the Computing Group,
once the changes agreed at reply and review time will have been incorporated.

These standards shall be applicable to all new ALMA software activities.
Existing code and documents should also be aligned, taking as a goal the next
scheduled major revision.

Agenda:

- The 1.5 hour telecon time was subdivided in three 30 min units for the final
  decisions. Three list of comments and replies had been circulated in advance
  by A. Bridger to the reviewers, so that there was no need to discuss on
  comments already accepted.
- A final list of comments will be circulated by A.Bridger (see point 2 in the
  Actions below). Here a few notes are given, as a reminder of decisions taken
  at the meeting.

1) ALMA Documentation Standard

- References: style as from Birger's comment + author. Reeference to specific pages given instead in the text. References at the end of a document.
- Software Group to decide on document categories for software. TBD in document.
- Owner. Replace box and concept of Owner with Review box giving Author(s), Approved by and Release by, as proposed by Gie Han. In case there are several authors the e-mail of one should be given.
- pdf for distribution. Yes, but it must be possible to have exceptions (html). Sources will not be distributed, but should be under configuration control (accessible in a controlled way).
- keywords. Use of consistent keywords.

2) C Coding Standards

- Variables with pointers consistent with C++
- Casting should be explicit for "unusual" kind of conversions.
- Composite words separated byupper case characters, with underscore restricted to a few clear cases where it is needed.
- need to look into macros by Alan.

3) C++ Coding Standards

- rephrasing of 2.2 on constants needed
- .cpp extension chosen
- classes to coincide with filenames is a guideline
- Name Spaces for Unix software, but not for VxWorks (not supported yet).

Actions:

1. G.Raffi to issue these Minutes
2. A.Bridger to circulate a final Draft list of comments/replies that takes the review decisions into account to reviewers first.
3. A.Bridger to edit documents according to comments and submit reviewed version for publishing on the Web.
4. B.Glendenning to explain to ALMA_sw_workers the use of GPL to insert copyright disclaimers into source code.

```
===============================================================================
            SOFTWARE DOCUMENTATION REVIEW PROCEDURE - REVIEW CALL
===============================================================================
Date      : 2000-12-15

Document : ALMA Software Documentation and Coding Standards
Principal Author: Alan Bridger (ATC)

Invited reviewers:

 Giorgio Filippi (ESO)
 Brian Glendenning (NRAO)
 Birger Gustafsson (ESO)
 Koh-Ichiro Morita (NRO)
 Fritz Stauffer (NRAO)
 Martin Pokorny (NRAO)
 Gianni Raffi (ESO - Meeting Chair)
 Gie Han Tan (ESO)


===============================================================================
```

The documents to be reviewed are available under the DRAFT documents at:
http://www.alma.nrao.edu/development/computing/docs/memos/index.html

C Coding Standards
 Alan Bridger, Mick Brooks, Jim Pisano

C++ Coding Standards
 Alan Bridger, Jim Pisano

The ALMA Documentation Standard
[A Document template, including a MS Word style file]
 Alan Bridger

Please send your comments directly to the principal author, Alan Bridger
(ab@roe.ac.uk) Written comments deadline: Jan.5, 2001

Review meeting: By phone, proposed Jan.10, 2001 (Wed) (or the Jan.11-
        Thursday) Reviewers please confirm your attendance at the Telecon
        review on the foreseen day to Gianni (graffi@eso.org). In case you
        cannot attend on that day please indicat e if the alternative date in
        brackets is better.

Review Time (4.30 pm CET, 8.30 MST). Foreseen time for review: 1.5
        hour. The Telecon phone number to call will be provided on Tuesday
        Jan.9, 2001.

Responses to the comments on The ALMA Documentation Standard"

The comments have been extracted to be in the page order of the document and the comments from each individual preceeded by that individual's forename in angle brackets, e.g. <Birger>.

Comments were recieved from:

Birger   Birger Gustafsson <bgustafs@eso.org>
Gie Han   Gie Han Tan <ghtan@eso.org>
Gianni   Gianni Raffi <graffi@eso.org>
Martin   Martin Pokorny <mpokorny@nrao.edu>
Brian   Brian Glendenning <bglenden@nrao.edu>
Koh-Ichiro Koh-Ichiro Morita <morita@nro.nao.ac.jp>

I have attempted to group together comments on the same section and give just one response. Responses are preceeded by RESPONSE:.

Author:
  Alan Bridger
Modification history:
  09-Jan-2001 - Original
  16-Jan-2001 - Modified after review comments. Further response indicated by
                REVIEW:

Page 1

<Gianni>
p1 Title. Should it not be "The ALMA Software Documentation Standard"?
ALMA as a whole can join this, but we do not know it yet.

RESPONSE:
AGREED. If it later becomes the project standard then the title can
change back.

<Brian>
p.1 I think it should be ALMA, not Alma

RESPONSE:
ACCEPTED. I just like turning acronyms into words...

Page 2

<Brian>
p.2 The columns in the Change Record seem to be too narrow to conveniently
display the information (e.g., in revision 2, even the Section/Page affected
wrapped). I wonder if a format where the remarks are under the "header"
might be better?

RESPONSE:
That's a good idea. I'll try it, or something like it.
REVIEW:
What I think Brian means here (correct me if I'm wrong) is something like:

```
------------------------------------------------------
|Revision  |  Date  |  Author  |  Sections Affected  |
|          ------------------------------------------
|          |  Remarks                               |
------------------------------------------------------
```

This can be done, though it seems to be a bit fiddly in Word. I will do it
and then actually using it should be straightforward for authors starting
with the template.

Page 3

<Brian>
p.3 Are appendices numbered I, II, III, IV etc? Would A, B, C be better? (I
wasn't sure if it was Appendix "eye" or Appendix "one").

RESPONSE:
I guess this is a matter of personal preference. I'm happy to go with
alphabetic if others agree.
REVIEW: Alphabetic was agreed.

Page 4

<Gianni>
p4 1 Summary. 2nd phrase:... ask questions to reviewers
After review I would substitute the second phrase with a statement
about the fact that this document has been approved via a formal review
process by the Computing and Software group.

RESPONSE:
ACCEPTED.

<Gie Han>
p.4 2.2 I would suggest that a limited set of acronyms and glossary
essential for understanding the document are included in the document
because the reader doesn't always have access to the on-line project
glossary.

<Gianni>
p4 2.2 Glossary. I would prefer upgrading the general glossary, referring
to it as the valid refernce and anyhow giving a subset in the documents
for reasons of readibility. Admittidly not a very strong reason.

RESPONSE:
It seems that a short list of relevant acronyms/glossary items would
be a good idea so I will include it in the standard. However, the general
online glossary should be the principal source.

<Gianni>
p4 2.3 References
I am used to have both Applicable and Referenced documents. The first ones
are the (contractually) applicable ones only.

6

RESPONSE:
I don't think there will be a consensus on this. I suggest referenced
documents only. If there are applicable documents with no obvious
citation point then I suggest making an explicit reference to them in
the text. This will give each document at most one reference list.

<Birger>
p4 section 2.3 References. I don't like any of your proposals. In
documents we would probably mostly reference other documents and
therefore they should be idetified unambigously. A style which is
already used in ALMA sw memos and we also use at ESO is: doc. number,
issue, date(not mandatory), title
seems more appropriate. Example

[1] ALMA Computing Memo #7, 2000-04-18, ALMA Monitor and Control Bus

Books and articles you could reference as:title, author, publisher,
year.

[2] CAN System Engineering, Wolfgang Lawrence, Springer-Verlag, 1997

<Gie Han>
p.4 2.3 I prefer the first style for references as well above the second
style you present. The reference used as an example for the second style
is a bad one. References to "private communication" should not be
allowed because they are not well traceable for the reader.

<Gianni>
p4 2.3 Style of references
I find the second style, while the first is criptic at a first view.

<Martin>
I prefer the coded reference style, as I tend to find it to be more
informative, and easier to use when reading a document. However, my
preference on this is rather weak.

<Brian>
p.4 Does word offer any built-in referencing support? That might determine
the most convenient referencing style

<Koh-Ichiro>
2.3 References
I prefer the first style.

RESPONSE:
There is clearly not a consensus on reference styles. Word97 has a
"Table of Authorities" which might serve (Brian's comment) but I
haven't investigated it. I did think it might be overkill for our
needs (it cross-reference pages etc.) and all its examples are from
the legal world. And in any case I think you can still customise it.
I don't know about Word2000.

I went for my first proposal, though I also quite like Birger's.

Gie Han's comment about "private communication" not being a good reference is

a good point and accepted.

REVIEW: Agreed to use Birger's style, with the addition of author, at
the end of the reference, when refering to documents and memos. It was also
agreed that references should go at the end of the document. Where it is
necessary to reference specific pages this should be inserted in the text.

Page 5

<Birger>
p5 3.3 I would certainly wish we had a standard list of document types.
It could also be part of the document number. If system engineering does
not propose document types, I propose that sw group should propose the
types it needs.

<Gie Han>
p.5 3.3 Use of document styles should be mandatory. A/o it helps the
reader in understanding what the purpose of the document is.
Furthermore, depending on the type of document different approval
procedures must be followed. Please indicate which types of documents
are needed.

RESPONSE:
I feel that a list of document types is for the systems eng. group to resolve.
Perhaps this will come out of the IPT structure? However, this document should
certainly include that list. In the meantime the software group could certainly
decide its own catagories.
REVIEW:
This document should mandate use of document types, but is not responsible
for determining what they are. The text shall say that they are TBD, and
the software group is to propose a list.

<Gie Han>
p.5 3.5 Proposed document label is fine for this moment. I hope that in
the near future we will have a single project numbering scheme, in which
your proposal should be easy to integrate.

<Brian>
p.5 This will interact with the ALMA wide numbering system. Until we have on
in place, I suggest:
    ALMA-SW-NNNN to make it clear at least that it's a computing document.

RESPONSE:
I'm happy to add the "-SW" suggested by Brian if there is consensus.
REVIEW:
It was agreed that the "-SW" would be added, but that the text should state
that this is a temporary scheme until a full project numbering scheme is
adopted.

<Martin>
Might the document label include some part that corresponds to a
document category or subject area? For the time being I prefer your
approach because this entire topic is up in the air right now, and it
could land anywhere. However, such a feature might be discussed.

RESPONSE:
My feeling on this is that if there is a suitable document management
database then the catagory/type can be handled by the db - encoding it
in the label seems unnecessary. However, in the absence of such a db
then encoding might be desirable.

<Gie Han>
p.5 3.6 I oppose to the introduction of a document owner. Mainly due to
the duration of the project, a method of using persons as a reference is
not that stable. In addition the responsibility for the contents of a
document might be transferred from one group to another one during the
project.
Instead I propose to link the documents to the products or processes to
which they belong. Using the product structure or IPT organization
diagram it is then easily traceable which person or which group is
responsible for the document and whom should be contacted. My opinion is
that this should be handled by a suitable EDM/PDM tool in the near
future.

<Koh-Ichiro>
3.6 Owner
I prefer to call "contact person".

RESPONSE:
It seems that the reason for the existence of the document owner will be
superceded by another process/tool, in which case I am happy to agree with
Gie Han.
REVIEW:
It was decided that one, and only one, author should provide an email address
and that this author should thus be the default contact. The text should
make this clear.
In addition it was decided that the "document owner" box will be replaced by
an author signing off box and a "Released by" signing off box on the front page
(added to the "Approved by" box). This may change when more formal document
handling and management procedures are in place.

<Brian>
p.5 Should there be a standard set of keywords to be selected from, or is
that too restrictive?

RESPONSE:
I think we should try to use consistent keywords, but the list will need
to be extensible and its silly to try to think of them all "up
front". I propose that I make a comment in this section that authors
should reuse from existing keywords whereever possible and try to be
consistent.
REVIEW:
The response was accepted and a suggestion that a (extensible) list of
appropriate keywords should be kept on the web-site was also accepted.

Page 6

<Gie Han>
p.6 3.8 Approval process depends on the type of the document (see also
my comment on section 3.3). Memos normally need no approval process.

9

Other documents that address formal issues that crosses IPT boundaries
are often under configuration control and need to be approved by the
ALMA configuration control board (see ALMA management plan).
Official documents that are internal to an IPT can use the following
approval procedure:

> Front Page Signatures:
> Three signatures must be included on front page, identifying:
>
> > Author and Institute
> > Approver and Institute
> > Releaser and Institute
>
> Documents shall be signed by all three signatories, otherwise
> the document shall not be considered as formally issued.
> The status of documents/drawings which are still in the
> process of preparation shall be clearly shown (e.g.
> preliminary, draft, for approval) and shall not be considered
> as released or archived.
> The following rules shall be applied:
>
> > If:
> > Author   = member of IPT
> > then
> > Approver = IPT Head
> > and
> > Releaser = Management
> >
> > or
> >
> > If:
> > Author   = IPT Head
> > then
> > Approver = Management
> > and
> > Releaser = Management

This is based on the approach normally used at ESO a/o for the VLT
project. The introduction of a "Releaser" has a legal background.

RESPONSE:
I think the standard should be modified to accomodate whatever the agreed
process is. Thus from the above I would add spaces for the author, approver and
releaser to sign. I would like to keep the same doc. standard for memos, in
which case I hope it would be acceptable to write "N/A" in these fields.

<Gianni>
p6 5.1 Overall structure
I would like to have eventually layouts or examples for various kind of doc's
like Requirements, Desing documents etc.
It does not have to be in this document though, this is later work for SE.

RESPONSE:
AGREED. I don't think this is part of this document at this stage.

<Koh-Ichiro>
5.1 Overall Structure

I understand the necessity of the hierarchical documnet
structure. But, I oppose to use Microsoft Binder.
Maybe, it is possible to make such structure
to use more simple method (using html etc.).
I personally do not like to depend on a commercial software,
too much.

RESPONSE:
Personally I sympathise with your comment very much. However, Word is
the agreed project standard. I am told that Binder is not necessary
with Word2000, but I don't have Word2000 to try this myself. Putting
together large documents from many small ones is not something I have
done with Word, though I have done it with Framemaker (Framemaker
"books") and I though the Binder did a similar thing.

<Brian>
p.6 Mention that some documents might also want a list of figures and a list
of tables.

RESPONSE:
ACCEPTED.

<Birger>
p6 5.4 headings. I think that using to many levels of headings makes the
document unreadable. Should be maximum 5.

RESPONSE:
AGREED. Will reword.

<Gie Han>
p.6 4 Modify "change control" to "change record" as is used on page 2 of
your document.

RESPONSE:
ACCEPTED.

Page 7

<Birger>
p7 5.5 Caption. I don't like having the caption in a box. It looks nicer
without the box.

<Martin>
Personally, I dislike the boxed style of figure caption, but again,
it's something I can easily deal with.

<Brian>
p.7 I prefer captions to not be in a set-off box.

RESPONSE:
ACCEPTED

<Gianni>
p7 6.1 The header.. contains ALMA and the title..

11

I cannot find the word "ALMA" prefixing the title in this document.

RESPONSE:
The "title" mentioned here is a document property as entered in the properties list (File/Properties/Summary). In this case I did not make this the same as the final document title as written on the front page. This may have been an error on my part and does illustrate one of the problems with using the file properties in this way - you must use it consistently. I should have used the TITLE properties field in the actual document title box.

<Gie Han>
p.7 6.2 Include in the footer only page count and document revision number. All other information is superfluous.

RESPONSE:
ACCEPTED - if other reviewers agree. On reflection (since writing the document) I
think I agree with Gie Han.

Page 8

<Gie Han>
p.8 Appendix I I prefer to have the documents distributed via the web in PDF format only and not in HTML. PDF is a readily accepted format for distribution and there is little risk that these copies are erroneously modified.
Document originals should be kept in Word format so they can be edited.

RESPONSE:
ACCEPTED. Standard for distribution should be PDF.
REVIEW:
It was agreed that PDF distribution should be the standard, but that exceptions can be allowed.
A comment will be made in the text about the source of documents (usually Word files) being available but with a little more effort - e.g. a CVS repository. They will also be under configuration control.

<Gianni>
p8 4 .. Word 2000
Hopefully this does not mean that only Word2000 can be used.

RESPONSE:
I hope not, merely that Word2000 produces better HTML output than Word97 (at least I am told this). If HTML is not a standard web format then this will not be an issue anyway.

<Martin>
Under the "Web Document" appendix, you seem to imply that if a document originates in something other than Word, the template is not applicable. Is this true, or does it only apply to web-based documents?

RESPONSE:

I meant that the style template (the .dot file) is fairly obviously,
only applicable to Word documents. If documents originate from another
tool then meeting the standard style is not the responsibility of this author!
I will try to rephrase the paragraph.

General comments:

<Gianni>
General: I like the proposed layout and have only minor comments.

RESPONSE:
Thank you.

<Martin>
Other than that, I agree with the decisions you have made, and don't
have any further questions or comments. (Now if it all simply wasn't
so Word-centric... sigh.)

RESPONSE:
I agree!

<Koh-Ichiro>
General:
I could not open your style file.
If you attach an instruction about how to use your
style file in MS Word, it is very helpful because
MS Word is not so popular as TeX in my observatory.

RESPONSE:
Sorry about the problem. I will include some instructions on how to
use the style file.

Responses to the comments on "C Coding Standards"

The comments have been extracted to be in the page order of the document
and the comments from each individual preceeded by that individual's forename
in angle brackets, e.g. <Birger>.

Comments were recieved from:

Birger   Birger Gustafsson <bgustafs@eso.org>
Martin   Martin Pokorny <mpokorny@nrao.edu>
David   David Terrett <d.terrett@rl.ac.uk>
Brian   Brian Glendenning <bglenden@nrao.edu>
Koh-Ichiro Koh-Ichiro Morita <morita@nro.nao.ac.jp>

I have attempted to group together comments on the same section and
give just one response. Responses are preceeded by RESPONSE:.

Author:
   Alan Bridger
Modification history:
   09-Jan-2001 - Original
   16-Jan-2001 - Modified after review comments. Further response indicated by
                 REVIEW:

General comments

<David>
1) It doesn't actually say the all code should be ANSI standard or that it
should compile without any warnings. I think it should.

<Brian>
p.? Generic guideline: code should compile warning free under it's principal
compiler (at least). Warnings that can't be eliminated should be commented
in the code for the maintenance programmer. (Applies to C++ as well).

RESPONSE:
ACCEPTED. Very good points. Its not clear which section I should add them
to - perhaps there needs to be a "General" section.

<Brian>
p.? Do we want to have a TODO comment guideline for things that still need
to be done. (C++ also).

RESPONSE:
I'm happy to put in a suggested form for TODO items.

Pages 1,2,3

No comments

Page 4

<Brian>
p.4 s1.1 cf C++ comment (a)

RESPONSE:
ACCEPTED.

<Martin>
1.2: Why did you choose to include references in the C++ document but
not the C document? I read the statement in the C document, but the
argument is not entirely convincing, especially in light of the
decision made for the C++ document.

RESPONSE:
[I moved this comment to the C document where it is more relevant]
I am happy to include references. However, to try to expand on my
reasons:

1. C is more mature than C++ and most of us have much more experience
in it. The "standard" C books (the Kernigan and Ritchie and Plauger
books etc) tend to be well known. Not a really good excuse for not
listing them I agree.

2. Following on from C's greater age there are a lot of C programming
standards written down. I reviewed a few for comparison with this
document, but was not keen on referencing them in the document as some
of the standards actually differ (for various reasons) and I didn't
really think that referencing other peoples' coding standards was
appropriate - this is our coding standard. In the C++ case it is still
"our standard", but C++ is a lot younger and I felt that reference to
some other sources was a useful thing.

If references to other standards or standard C texts would be useful
in the C document then I will include them.

REVIEW:
It was agreed that I would add references to the C standards document.

<Martin>
2.1 - Standards: It might be too much to ask that the name of an
identifier make the "purpose" of the identifier "clear" in all
cases. Perhaps a rewording such as "The name of an identifier should
succinctly describe the purpose of that identifier" leaves a bit more
leeway while maintaining the intent of the statement.

RESPONSE:
I am happy with Martin's rewording and will use it unless there are objections.

Page 5

<Brian>
p.5 s2.2 guideline 1. To be consistent with C++, the names should start with
a lower case letter. Is there a reason not to do this? (Reserve starting
with an upper case character for types).

RESPONSE:
ACCEPTED.

<Birger>
p5 Function names. As we are going to use a module concept I propose to prepend
public function names with the module name. This has several advantages:

   - you have a namespace, e.g. the function name can't clash with any
function name from any other module
   - it is easy to find out to which module a fiunction belongs.
The same applies to global variables, although I think they should be
avoided as much as possible.
I particular this issue is important for VxWorks which has only a global
symbol table,
meaning that all public data and functions are seen by all modules.

RESPONSE:
ACCEPTED. The vxWorks comment is particularly important. In fact using
a module name as a prefix has been my normal practice in the past. For C
(where there is no language namespace) I think module names should be
used whether or not CMM (which requires them) is adopted.
REVIEW:
This was agreed. It was also suggested that "obvious" module names should
be avoid (e.g "util").
It was also agreed that the standard should be clearer in stating that
names were formed from composite words, delimited by upper case letters,
with no underscores allowed (except for appending pointer identifiers etc.)
Underscores would be used as delimiters in macro names.

<Martin>
2.2 - Guidelines: It might be helpful to add an example query method
name such as "IsActive" (whose second term is not a noun).

RESPONSE:
ACCEPTED.

<Martin>
2.3 - Standards & Guidelines: A few guidelines on how to make the type
of a variable "evident within the code in which it is used" would
help.

RESPONSE:
I will try to think of some.

<Brian>
p.5 s2.3 guideline 4 (I don't think the _p adds any value as the compiler
type system will catch errors for you. If one thinks this does add value the
distinction between pointers and non-pointers is only the start, and you end
up with Hungarian notation (yuck)).

<Koh-Ichiro>
2.3 Variable Names

Pointer variables:
I prefer to put an indicater at the head of variables.
For example, p_MeaningfulName.

RESPONSE:

This is a guideline, so can be ignored or indeed Koh-Ichiro's style could be
used. Personally I like to have indications that variables are pointers - I find
that this helps me read the code. However, I certainly wouldn't want to go all
the
way to Hungarian.
REVIEW:
Appending (rather than prepending) the indicator should be the preferred style.
And there should be consistency between this and the C++ standard.

<Brian>
p.5 s2.3 Do you say anywhere that types (e.g. struct types, enums) should
have meaningful names? (Also true for C++).

RESPONSE:
No, but I will.

<Martin>
2.4 - Standards & Guidelines: The standard states that macros should
be "stylistically distinguished" from functions, but a guideline
states that macros with parameters should be defined using "the same
convention as function names". This appears to be an inconsistency,
but perhaps I've misunderstood the wording.

<Brian>
p.6 s2.4 Since even macro functions can be surprising (unless very carefully
written), I would capitalize them as well as a warning sign.

RESPONSE:
AGREED & ACCEPTED. (And see reply to Birger's related comment below)

Page 6

<David>
2) Section 3 (pg.6 bottom) mandates the use of NULL rather than 0 for an
undefined pointer; however, NULL isn't part of the ANSI standard language,
you have to include stdio.h to get it. I think this should be spelled out.
(I think that the same is not true for C++ but I'm not sure).

RESPONSE:
ACCEPTED.

<Birger>
p 6section 2.4 Names of Macros. Why do you propose a different naming
convention for macros with and without parameters. This is very
confusing.
And comparing also with the C++ coding standard, where you propose that
all macros are in upper case. I propose to have all macros in upper case
also for C.

RESPONSE:
ACCEPTED.

<Brian>
p.6 s3 bullet 1. What does this mean - that all file scope variables are
static not extern? Can we really get away with this? (I don't like global

variables, but rarely they might be required).

RESPONSE:
I'm not sure I understand all the possible repercussions of the
statement and this comment, so I'm not inclined to make a quick
response. It does to sound like the statement is a bit strong for a
standard.
REVIEW:
This was discussed much and the reviewers agreed that the standard should
state that global variables should be avoided unless really necessary - with
very good reasons given.

<Martin>
3 - Standards:
a) I don't see the harm in allowing shifts for multiplication or
division when the operands are unsigned. If we want to use these
standards with any embedded code (which is not all at certain, as I
understand it), this restriction can be too severe.

RESPONSE:
I'm not convinced. As stated elsewhere, standards can be broken for
good reasons, so the restriction can be lifted if really
necessary. Otherwise I think it should stand. Comments welcomed.
REVIEW:
The document should state that noted exceptions are allowed.

<Koh-Ichiro>
3. Variables, Operators and Expressions

" $B!& (B Avoid machine dependent ...."

I prefer to mention:
"Do not write a code depending on the byte order of machines."

RESPONSE:
I propose adding a further statement about not depending on the byte order of
machines, and also about not relying on alignments.

<Brian>
p.6 s3 bullet 3. I think when the compiler will do a cast for you (e.g. from
type* to void*), it is better to let the compiler do it.

RESPONSE:
I'm not sure I agree with this. I'd prefer explicit casting. Other input
would be appreciated.
REVIEW:
It was agreed that in general the compiler could be used to do
casting but that explicit casting is required for "unusual" type conversions.

<Martin>
b) The pointer assignment rule could cause some difficulty in dealing
with "void *" types, or are we trying to avoid their use entirely?

RESPONSE:
A good point. I'm inclined to mention (void *) as an exception, though
perhaps suggesting that it is used only with good reason.

Page 7

<Martin>
3 - Guidelines: The second guideline can be made clearer by reversing
the order of the two clauses, as in: To improve clarity, use
parentheses even when not required.

RESPONSE:
ACCEPTED.

<Martin>
4 - Standards:
a) Specifying ANSI-C style both as the "preferred style" and a
standard seems a bit wishy-washy. There may be cases where the
compiler does not support ANSI-C (not that I can come up with any
examples immediately, but some embedded microcontroller compilers can
be quirky). Just proclaim ANSI-C style as the standard, and we can
worry about problem cases with some kind of disclaimer or exculpatory
statement in the code comments (would seem to be allowed under the
standards in any case).

RESPONSE:
ACCEPTED. ANSI-C style should be the standard.

<Brian>
p.6 s4 OK, I know this is probably a lost cause, but I much prefer K&R style
formatting:

```
int foo() {
...
}
```

In any event, if it's a standard it's not "preferred." (I think it would be
better as a guideline).

RESPONSE:
I prefer K&R formatting too. However, this standard is meant to be
about the function declaration, and I think ANSI-style declarations
should be standard. I should reword it.

<Brian>
p.7 s4 Structures (and probably unions) are sometimes better bassed by
value. Certainly not a standard, and I don't even think it should be a
guideline.

RESPONSE:
ACCEPTED. I agree certainly not a standard. And right now I can't think of
a good reason for it to be a guideline. Unless there is a strong feeling I
will remove the statement.

<Martin>
b) The statement "Data types must not be mixed across function
boundaries" might be clarified.

<Brian>
p.7 s4 bullet 4. I don't understand this? Do you mean:

void foo(float x);
foo(4.0);

Should not be allowed? (4.0 is a literal of type double). In general, I
think that compiler supplied conversions are *preferred* over casts? Why?
Because the type of a declaration might change, but the cast would mask it.
The guideline really is: use a cast, and document it, for unusual type
conversions. (BTW, in C++ we should require new-style casts as they can be
found with a grep).

RESPONSE:
Brian's understanding is correct, and again I'm not sure I agree. This is
related to
his earlier comment too.
If the statement remains I will attempt to clarify it (Martin's comment).
REVIEW:
As per the earlier comments on casting: the standard will be that casts
should be explicit for "unusual" type conversions.

<Martin>
c) Is there any policy on empty argument lists or the use of "const"
modifier in argument lists (under C)?

RESPONSE:
I'm happy with empty argument list declarations being just that,
i.e. func ( )
I'm not sure I can think of reasons to make policy statements about
the use of "const" in argument list? There may be good reasons to use
"const". If there's a consensus for something else I'm happy to
reconsider.
REVIEW:
It was decided that the empty argument list case should be checked.
Following the review it was established that an empty argument list
should clearly be indicated using (void) even though many compilers
may well accept ( ).

Page 8

<Brian>
p.8 s5 Standards. These all seem like guidelines to me.

RESPONSE:
I disagree. Code that does not delimit the statement in an "if" (but
no "else") statement can be very hard to read and I think it is very
bad practice to modify loop counters in the for line *and* inside the
loop. I've been bitten by code that did both of these recently and
this perhaps colours my view!
I also think its sensible to always provide a default switch case.
REVIEW:
It was accepted that the third bullet (incrementing/decrementing loop
counters) would be demoted to a guideline. The others remain standards.

<Birger>

p8 section 6. Do we need copywrite notice in the header? ESO does not
require this. What about other inbstitutes involved? If not required it
should be dropped, because it just makes files longer, without adding
any new information.

<Martin>
6 - Standards:
a) A statement about variations in copyright notices may be in order
here (similar to what is written in the C++ standards doc).

RESPONSE:
Whether or not a copywrite statement should be included is, I feel, a larger
project decision. However, if one is used then I agree that it should not be
in each source file. One COPYRIGHT file per module should suffice. If copyright
statements are included then we need in some way to take into account all of the
institutes - and both documents should reflect this.
REVIEW:
It seems that copyright statements must legally be in source code in the
U.S. Brian suggested that the Gnu Public License (GPL) should be the
license of choice for ALMA and agreed to distribute a summary of its terms.

<Birger>
The standards for logs you propose are not correct if we will use cmm,
because cmm does not export the RCS logs to the user. Therefore we use
logs in the file headers with cmm. Pls. take this into account.

<Martin>
b) The standard requiring that headers should not contain a change log
would appear to conflict with the ESO/CMM style of doing things.

<Brian>
p.8 s6 bullet 2. While I personally agree with this, it contravenes the ESO
CMM practice.

RESPONSE:
Yes. This standard should be changed if CMM is adopted. The document
is largely from the NRAO standards, and there seemed to be little
point in changing it to meet the CMM standards until that decision is
made.
REVIEW:
It was decided that the standard should state that a change log must be
maintained, in a manner appropriate to the development environment. This
statement could be made more explicit when the development environment
is confirmed.

<Koh-Ichiro>
6. Code Commenting

The standard header information should contain
the purpose of the module and the input/output information.

RESPONSE:
ACCEPTED. I'll make this a bit clearer.

<Martin>

5 - Standards & Guidelines:
a) There is no standard for the use of braces in while loops.

RESPONSE:
ACCEPTED. Will include.

<Martin>
b) The guideline on "goto" statements may as well be a standard, since
standards may be violated with justification anyway.

RESPONSE:
ACCEPTED.

Page 9

<Brian>
p.9 s6 cf C++ comment (b).

RESPONSE:
Fine with me. (Same as for C++)

Page 10

<Martin>
6.1 - Standards: It is stated that comments ought to document "what
the function is going to do to the parameters and to the object
itself". To what "object" does this refer?

RESPONSE:
Ooops. You spotted an historical relationship between this and the C++
document. The "object" here should be removed.

<David>
4) I prefix all header files that declare public procedures with:

```
#ifdef _cplusplus
extern "C" {
#endif
```

and append

```
#ifdef _cplusplus
}
#endif
```

so that the header can be used by C++ code. I would put this in the standard
(unless the C++ experts disagree or have a better way of achieving the same
thing).

RESPONSE:
AGREED. Again, also unless the experts have another suggestion.

<Martin>

6.2 - Standards: A mention about style consistency, like that in the
C++ standard, would be a good addition here.

RESPONSE:
AGREED.

<Martin>
6.2 - Guidelines: The comment on "sufficient white space" does not
quite agree with the "four space" standard in the previous section.

RESPONSE:
I'm not sure they are in disagreement, but I do agree that there shouldn't be
two
very similar statements like this and that I should clarify it.

Page 11

<David>
3) Section 7 suggest blanks around all binary operators except . & ->; I
agree, but several examples in the document don't follow this guideline: e.g
pg. 11 the for statement is missing the space around "<", pg.7 func(X*i,
y/i, i++) - there may be others - I haven't checked thoroughly.

<Martin>
7 - Standards: Spaces are missing around the inequality in the first
example of a "for" loop.

RESPONSE:
Examples are always badly proof-read, especially by the author(s). I'll double
check all of the examples.

<Martin>
7 - Guidelines: The guideline on indentation does not make the
relation between tabs and spaces clear. Simply changing the number of
spaces per tab does not necessarily change the indentation. This is
arguably a quibble, but it tends to be an annoyance. (You could just
leave out all mention of tabs, and simply describe indentation in
terms of spaces.)

RESPONSE:
AGREED. I will omit all mention of tabs.

Page 12

<Brian>
p.12 s7 object declarations. Personally I find it harder to read
declarations that are separated from their type by whitespace.

RESPONSE:
It is a guideline, and one that I agree should be used with a certain degree of
common sense.

<Brian>
p.12 s7 Either we should switch to K&R style (yay!), or alternatively

wouldn't it be more consistent for the struct brace to be on a new line as
well (probably indented).

RESPONSE:
Yes, it probably would.
As I said before I tend to prefer open braces on the same line as the opening
statement
with closing braces aligning with the start of the statement. However, most of
the
programmers I see around me seem to work with braces on new lines. This is also
what was in Mick's original document. I.e. it seems to be more common.
It *is* a guideline, so perhaps Brian and I can do our own thing when starting
on
new code ;-)

Page 13

No comments.

Page 14

<Brian>
p.14 cf C++ comment (c).

RESPONSE:
AGREED, it was just as an example. (Same as C++)

<Brian>
p.14 Can't DOC++ mark up C? If so, it should be shown in the example.

RESPONSE:
Yes. The simple function in the DOC++ example was meant to suggest this.

Pages 15-16

No comments.

Responses to the comments on "C++ Coding Standards"

The comments have been extracted to be in the page order of the document
and the comments from each individual preceeded by that individual's forename
in angle brackets, e.g. <Birger>.

Comments were recieved from:

Birger   Birger Gustafsson <bgustafs@eso.org>
Martin   Martin Pokorny <mpokorny@nrao.edu>
David   David Terrett <d.terrett@rl.ac.uk>
Brian   Brian Glendenning <bglenden@nrao.edu>
Koh-Ichiro Koh-Ichiro Morita <morita@nro.nao.ac.jp>

I have attempted to group together comments on the same section and
give just one response. Responses are preceeded by RESPONSE:.

Author:
   Alan Bridger
Modification history:
   09-Jan-2001 - Original
   17-Jan-2001 - Modified after review comments. Further response indicated by
                 REVIEW:

Pages 1,2,3

No comments

Page 4

<Brian>
p.4 s1 As per the document title, shouldn't the section be called "C++
Coding Standards"?

RESPONSE:
ACCEPTED.

<Brian>
p.4 s1.1 As ALMA is presently constituted, the standards apply only to the
software group, although we can try to convince other groups to adopt them.
(a)

<Martin>
1.1: I'm not sure whether firmware will be programmed according to our
group's standards. It would be desirable to have it that way, but I
doubt that many of the hardware engineers here would want to bother
with a coding standard.

RESPONSE:
I adopted the firmware comment from the previous (NRAO) version of the
document. Obviously I will change the document to reflect
reality. From what you say it seems that the firmware people won't
follow so unless I hear otherwise I will remove "firmware".

<Brian>

p.4 s1.1 "in existing code", in->on or in->with.

RESPONSE:
ACCEPTED.

<Brian>
p.4 s2 Some bullets end in ".", others don't (do I win the trivial comment
award?)

RESPONSE:
ACCEPTED. :-)

<Brian>
p.4 s2 bullet 4. I don't really understand this. Do you mean that member
data shouldn't be interspersed with member funtion declarations, or do you
mean something about grouping together similar member functions and data
(e.g. all constructors together). (a la organization guideline 2).

RESPONSE:
This is one of Jim's - I took it to mean functions and data should not be
interpersed, but there is obviously overlap with the organisational guidelines
here. I will reword it.

Page 5

<Martin>
2.2 - Standards: The case of "const" data members with respect to the
first standard is not clear.

RESPONSE:
I'm not quite sure I understand the meaning of this comment - I'm
probably missing something.
REVIEW:
The standard should be made a bit clearer. Class data members should
be private - if access to them is required (modification or viewing) from
outside the class then appropriate access functions should be provided.

<Brian>
p.5 s2.2 bullet 1. If it's a standard, then class data members must (not
should) be private. (You could comment that you can put public data in
structs, which in C++ can have member functions).

RESPONSE:
ACCEPTED, though perhaps pedantic! I will reword.

Page 6

<Martin>
3: Standards: See my comment on section 2.1 of the C document. Also, I
noticed that there is no mention of variable type here, while the
other document does make a statement about it (sec 2.3).

RESPONSE:

26

I also accept Martin's comment about the phrasing here.
I will include a guideline on variable types, similar to that in the C document.

Page 7

<Birger>
p7 section 3.2 function names: capitalization rule proposed for C++
differs from the rules for C. It would be better to have the same rules
for all languages. Otherwise you create confusion and people will apply
the wrong rules. Actually I prefer what you propose here, to what is
proposed to C. The rule with all words capitalized except the first one
is also what we use at ESO and it is also proposed by Wind River for
VxWorks code.

RESPONSE:
AGREED. If we adopt module names as a prefix in the C case then the module
name will not be capitalised, and the other words will. I am happy with this.

<Birger>
p7 3.5 File Names. Why are you proposing .cc extension for C++ source
files?
I'd prefer .cpp because that's what has been used for ACS and is also
the convention for ACE/TAO which is one of the CORBA ORBs we will use.
Unless you have strong arguments .cpp would be better. Otherwise we
would have to change ACS including also the Ljubljana team making the
ANKA software.

RESPONSE:
I have no strong arguments, but has .cc been used elsewhere in ALMA yet?
If not then I will change to .cpp.
REVIEW:
.cpp was accepted as the standard filename extension.

<Brian>
p.7 s3.5 bullet 1. "In general" means it's a guidline, not a standard.

RESPONSE:
I think this should be a standard, so I will reword. Except that I'm
not sufficiently expert in C++ to know if making this a standard
causes a problem?
REVIEW:
This was demoted to being a guideline.

REVIEW:
Also on page 7, though there was no specific comment, it is noted that
for consistency with the C standards the guideline indicator of member
variables should use an appended "_m" rather than a pre-pended one.

Page 8

<Martin>
4 - Standards:
-) Although it has nothing to do with coding standards, it is

27

important to consider that if we allow varying copyright statements
among sites, we ought to at least check that the licenses are
compatible.

<Brian>
p.8 s4 Not really a comment for the document, but perhaps for the review
meeting: can the project adopt the GPL for copyright?

RESPONSE:
Licenses and copyrights are distinct, but yes, I agree that these - if
used - need to be compatible. My understanding (disclaimer here - I'm
not a lawyer!) of European copyrights is that the author (or their
employer!) of a text (which in Europe includes software) automatically
owns the copyright, regardless of whether an actual copyright
statement is made. The statement merely clarifies. I'm not sure about
the US situation. See also similar comments/responses in the C
document. On Brian's comment: I am personally happy with GPL, and we
do have a precendent for our "masters" accepting GPL. (Not necessarily
obvious - protection, and possible exploitation - of Intellectual Property
is big in UK government at the moment...)

<Birger>
p6 4 Code commenting, for change log I have the same comment as for C
concerning the use of CMM. CMM does not allow the use of RCS changelog.

<Martin>
a) As I mentioned in my previous comments, the CMM/ESO style includes
a change log in the header.

RESPONSE:
See response to similar C comment.
REVIEW:
Again, the review comments are the same as for the similar comments in the
C standard.

<Brian>
p.8 s4 guideline. I'd shorten this to just suggesting Id for all RCS based
souce-code control tools. (b)

RESPONSE:
Fine with me.

<Birger>
p14. Remove copywrite clauses if not really required, see also my
comment to C standard.

RESPONSE:
Same response as for C case.
REVIEW:
Again, the review comments are the same as for the similar comments in the
C standard.

<Martin>
b) The detail in this section about the style of comments is
nonexistent when compared with the C document's commentary on this
subject.

RESPONSE:
Possibly I didn't want to repeat. However, I will add more, essentially copied
from the C doc. (with appropriate mods).

Page 9

<Brian>
p.9 s4.1 b) IMO in a class declaration, a variable name should always be
supplied.

RESPONSE:
I've no strong feeling on this, and so will agree unless there are many
objectors.

<Brian>
p.9 s4.1 e) What exceptions the function might throw (alternative: a throws
specification (if compilers reliably support them these days)).

RESPONSE:
ACCEPTED.

Page 10

<Brian>
p.10 s5.1 bullet 1. Perhaps namespaces are safe everywhere now (VxWorks is
no doubt the laggard, so if they work there I imagine they are safe
everywhere).

RESPONSE:
I'm not a sufficent C++ expert to comment on the status of namespaces.
REVIEW:
It was agreed that we could not be confident of namespaces working everywhere.
Therefore it was agreed that this standard should be reworded to indicate
that namespaces should be used in software intended for workstations (where
namespaces can be expected to be supported) but not in software intended for
LCUs (where support cannot be gauranteed). It was noted that common utility
software could be a problem (and thus should meet the LCU standard). Also
noted was the use of a macro definition in ACE that simulates namespaces -
this could be used (reference Birger)

Page 11

<Brian>
p.11 s5.2 last bullet. Null is not the only time you might want to pass a
pointer (e.g. you might have to reallocate storage or some such). I'd make
it a guidline, and say that references are preferred to pointers unless
pointer manipulation or a null pointer is a possibility.

RESPONSE:
ACCEPTED.

<Brian>
p.11 s5.4 bullet 1. This seems more a guidline than a standard (rarely
something tricky is required). Moreover people can differ over what's tricky
(some consider the ternary operator to be a blessing, others an

obfuscation).

and:
<Brian>
p.11 s5.4 and s5.5 All these seem like guidelines to me. If you consider 5.5
to really be a standard it should be a bullet rather than a "-" for
consistency.

RESPONSE:
I think you are quite correct, all these should be guidelines.

Page 12

<Brian>
p.12 s6 Why don't we take advantage of the review and say that DOC++ is the
code documentation tool full stop. If we find a problem, we can always
change our mind later.

RESPONSE:
ACCEPTED.

Page 14

<Brian>
p.14 alma-sw-workers is the wrong address! (We might need to create a new
one for this purpose. Maybe alma-sw-admin?). (c)

RESPONSE:
AGREED, it was just as an example.

<Brian>
p.14 Invalid (e.g., <linkto>) statements should be removed from the example.
(I think <linkto> is the same as @see?).

RESPONSE:
I agree, I left the <linkto> statements in as an example of how DOC++ will just
ignore the tags that it didn't understand that might have been used by the old
NRAO system.

Page 15-19

No comments