May 29, 2009

Patricia Schroeder
American Nuclear Society
555 North Kensington Avenue
La Grange Park, IL 60526

Dear Ms. Schroeder:

SUBJECT: ANS-10.2 N-17 COMMITTEE BALLOT

Enclosed is my ballot for ANS-10.2-2000; R200x, “Portability of Scientific and Engineering Software.” I have voted "Approved with Comments." Please consider these comments at the next revision of the standard. Furthermore, I have reviewed the Scope of the standard, and have no objections.

If you have any questions concerning this ballot, please contact me at 301-415-1127.

Sincerely,

/RA/

Alexander Adams, Jr., Senior Project Manager
Research and Test Reactors Branch A
Division of Policy and Rulemaking
Office of Nuclear Reactor Regulation

Enclosure:
1. Ballot
2. NRC Comments

May 29, 2009

Patricia Schroeder
American Nuclear Society
555 North Kensington Avenue
La Grange Park, IL 60526

Dear Ms. Schroeder:

SUBJECT: ANS-10.2 N-17 COMMITTEE BALLOT

Enclosed is my ballot for ANS-10.2-2000; R200x, “Portability of Scientific and Engineering Software.” I have voted "Approved with Comments." Please consider these comments at the next revision of the standard. Furthermore, I have reviewed the Scope of the standard, and have no objections.

If you have any questions concerning this ballot, please contact me at 301-415-1127.

Sincerely,

/RA/

Alexander Adams, Jr., Senior Project Manager
Research and Test Reactors Branch A
Division of Policy and Rulemaking
Office of Nuclear Reactor Regulation

Enclosure:
1. Ballot
2. NRC Comments

DISTRIBUTION:
PRTA r/f
RidsNrrDprPrta
RidsNrrDprPrtb
RidsNrrDpr
CHoxie, RES

ADAMS ACCESSION NO: ML091420451 TEMPLATE #: NRR-106

OFFICE	PRTA:PM	PRTA:LA	PRTA:BC
NAME	AAdams	GLappert	KBrock
DATE	5/29/09	5/29/09	5/29/09

OFFICIAL RECORD COPY

BALLOT OF THE AMERICAN NUCLEAR SOCIETY

STANDARDS COMMITTEE

Proposed Standard's:

Numerical Designation: **ANSI/ANS-10.2-2000; R200x**
Version: **Reaffirmation of Current Standard**
Title: **Portability of Scientific and Engineering Software**

Ballot Closes: May 30th, 2009

Please indicate your vote:

Approved ☐

Approved with comments (*Comments Appended*) ☒

Not Approved (*Comments Appended*) ☐

Indicate by asterisk (*) those comments which require satisfactory resolution to effect change of your ballot, and designate those comments by number on the provided comment form.

Not Voting (*Reasons Appended*) ☐

Date May 21, 2009

Name Alexander Adams Jr.
[Please PRINT or TYPE]

Signature **/RA/**

Please return ballot and comments:

Patricia Schroeder
American Nuclear Society
555 North Kensington Avenue
La Grange Park, IL 60526
Telephone: 708-579-8269
FAX: 708/352-6464
E-Mail: pschroeder@ans.org

NRC Comments
on ANS-10.2
"Portability of Scientific and Engineering Software"

1. Section 4.2.3, Programming Language. There should be a recommendation that when developing software designed to be portable, the developer should make use of as many platforms during development and testing as possible and, more importantly, as many compilers from different vendors as possible - even on the same platform. This is because each compiler tends to have its own bugs and/or diagnostic checking that will catch (or not catch, as the case may be) programming issues that the official language standards do not address (or leave for interpretation by the compiler vendor).

   For example, portability of a Fortran program can be greatly enhanced when developing on a Windows system, by testing the software not only with the Compaq Visual Fortran compiler, but also Absoft, Lahey, Salford, Gfortran and g95.

   Along with this, something should be said about the need to develop a multi-platform build process that uses automated software tools like BuildBot which can automate the testing process across multiple platforms and create convenient dashboard displays of the testing results (even to the point of allowing users with various platform types to contribute to the testing process).

2. Section 4.2.3.5, Language Extension. Consider adding a sentence that makes the user aware that many compilers have command line flags that enforce strict checking of the program against the language standard during compilation.

3. Section 4.2.3.6, Assembly Language. Something should be said here, or in Section 4.6.2 about the important role that the build environment can play in managing and enhancing code portability when compiler extensions, assembly language, or external libraries must be used.

   For example, platform dependencies can be effectively managed in C programs through the use of C pre-processor directives and the proper command line switches in the build configuration. Another example can be to maintain several unique versions of the same source file for different compilers/platforms and let the build environment make the determination about which one to compile depending upon which platform the code is being built on. This technique is used very effectively by the F2KCLI library.

4. Section 4.4.2, Computation and Data Processing. How does this enhance or impair portability of a program? Adding these features does not, in and of itself, help to make a code more portable. Such features can, however, make it easier to identify problems that might be caused by issues related to the code's portability. Either this linkage should be spelled out, or this section should be removed.

5. Section 4.4.3, Error Detection and Diagnostics. Add a sentence that during development and testing, programs should be built with all the available compiler warning and run-time checking flags (like checking for array bounds errors, un-initialized variables, etc) turned on. The result of engaging such run-time flags will be manifested as run-time errors during testing, allowing the developer to identify many programming issues directly related to a program's ultimate portability.

ENCLOSURE 2

6. Section 4.4.4, Output and Termination Processing. Serious consideration should be given to the use of IO libraries that facilitate the creation of platform-independent binary files. That way, binaries generated by a program on one type of computer can be read in by that same program on other types of computers. A good example of such a library is the XDR library.

7. Section 4.5.1, Variable Names. Consider adding after third sentence: "Consider adopting one of the many existing naming convention standards documented in the software engineering literature."

8. Section 4.5.2, Specification Statements. Not sure how this contributes to portability.

9. Section 5, References. Consider adding a reference to the SWEBOK - Software Engineering Body of Knowledge.