

Update on Standards, Fortran Specialist Group meeting, May 13, 1999

This has been a productive year for ISO Fortran Standards. Three new standards have been published recently. These are on Exception Handling, Allocatable components in derived types and Conditional Compilation. In addition progress is being made with the development of Fortran 2000 and with the revision of the Varying Length String module. This short presentation will give an overview.

The thinking behind publishing ISO Technical Reports on specialized areas with Fortran is that, since they can be easier and quicker to produce than full language standards, they can give an indication of the direction of the language and give both compiler writers and language users an opportunity to test a new feature before it is fixed in the following standard. The intention is that all the features defined in these reports be incorporated in Fortran 2000 unless any major disadvantages be found in the mean time.

Exception handling

There have been various attempts to add exception handling to the Fortran standard since the early 1980s but the diversity of underlying hardware and operating systems has proved a major hindrance. Now, increasing use of the IEEE standard for floating point arithmetic has allowed development of a Technical Report on floating point exception handling in Fortran (ISO/IEC TR 15580:1998 Fortran Floating-point exception handling) which has been approved by ISO.

The aim of this report is to provide support for handling the five conditions: overflow, invalid, divide-by-zero, underflow, and inexact. The proposal involves three standard modules:

- IEEE_EXCEPTIONS contains a derived type, some named constants of this type, and some simple procedures. They allow the flags to be tested, cleared, set, saved, or restored.
- IEEE_ARITHMETIC behaves as if it contained a USE statement for all of IEEE_EXCEPTIONS and provides support for other IEEE features through further derived types, named constants, and simple procedures.
- IEEE_FEATURES contains some named constants that permit the user to indicate which IEEE features are essential in the application. Some processors may execute more slowly when certain features are requested.

To facilitate maximum performance, each of the proposed functions does very little processing of arguments. In many cases, a processor may generate only a few inline machine code instructions rather than library calls.

In order to allow for the maximum number of processors to provide the maximum value to users, we do not require IEEE conformance. A vendor with no IEEE hardware need not provide these modules and any request by the user for any of them with a USE statement will give a compile-time diagnostic. A vendor whose hardware does not fully conform with the IEEE standard may be unable to provide certain features. In this case, a request for such a feature will give a compile-time diagnostic. Another possibility is that not all flags are supported or that the extent of support varies according to the kind type parameter. The user must utilize an inquiry function to determine if he or she can count on a specific feature of the IEEE standard.