TESTING AND PACKAGING OF SCIENTIFIC AND ENGINEERING SOFTWARE TO ACHIEVE PORTABILITY*

M. K. Butler, H. S. Edwards, L. R. Eyberger, P. A. Johnson,
M. Legan, L. L. Reed, and A. J. Strecok

National Energy Software Center
Argonne National Laboratory
Argonne, IL 60439

The National Energy Software Center (NESC) is a software exchange and information center operated by Argonne National Laboratory for the U. S. Department of Energy (DOE), with support from the Nuclear Regulatory Commission. The major objectives of the NESC program are:

- to promote the sharing of computer software among agency offices and contractors,
- to facilitate the transfer of agency-developed computer applications and technology to the information processing community, and
- to provide the focal point for DOE software exchange arrangements with other government agencies and for coordination of DOE acquisition of non-government software.

HISTORY

The Center is the successor to the Argonne Code Center, originally established by the U. S. Atomic Energy Commission in 1960 in response to a proposal by the Mathematics and Computation Division of the American Nuclear Society for a center dedicated to the collection and exchange of information on nuclear reactor codes. The Code Center operated initially as an adjunct to the Laboratory's Applications Programming activity, utilizing the major computer manufacturers' library facilities for reproduction and distribution of designated program package material, and professional society services for dissemination of abstracts describing the software collection and other information. Beginning in 1965 the Reactor Physics Branch of the AEC's Division of Reactor Development and Technology provided the direct funding to permit the development of independent Center software library and information dissemination capabilities. The Code Center program was broadened to an agency-wide software exchange and information center in 1972 with its scope expanded to cover the entire spectrum of AEC research and development activity. The Center's role was extended further with the transition from the AEC to the Energy Research and Development Administration in 1975, and later, in 1977 to the Department of Energy. In July of 1978 the Code Center was designated the National Energy Software Center to better reflect the present program and current computing terminology.

*Work performed under the auspices of the U.S. Department of Energy.

distributed in response to requests from DOE and NRC offices and contractors, other government agencies, universities, and commercial and industrial organizations. Annually, too, the Center reviews, tests, and packages over one hundred software contributions from agency offices and contractors. These contributions may represent new programs to be added to the library, additional versions of existing library programs implemented on other machines or computer systems, revisions or modifications of library packages, or complete replacements and new editions of library packages. The NESC library is made up primarily of scientific and engineering applications written for the most part in the FORTRAN language for operation on large CDC, IBM, and UNIVAC computer systems.

RECOMMENDED PACKAGE ELEMENTS

The software package is defined by NESC as the aggregate of all elements required for use of the software by another organization or its implementation in a different computing environment. Based on two decades of experience in software exchange the NESC has identified those package elements which provide the means for achieving portability. These are listed in the NESC checklist of recommended program package contents. Associated with each element are guidelines for preparing that element so as to ensure its transferability.

Package elements are generally available in one of two forms — either as traditional printed material, or as computer-media material which is distributed on cards, magnetic tape, or via communications network facilities. The first element on the NESC checklist of recommended package contents, documentation, is usually available as printed material.

Printed Material

Documentation may take the form of a single, comprehensive report or a number of manuals and reports addressing specific program features (e.g., use of the program to solve a particular class of problems, associated graphics capabilities), or different audiences (e.g., users, programmers). As a minimum, portability considerations require that the software documentation include a description of the program, application and programming information, and computing environment data. The narrative *program description* conveys the function of the software, or the nature of the problems it was designed to solve and identifies methods and computational algorithms employed.

Application information, frequently presented as a user's manual, guides the user in making effective use of the software, in preparing the input data required to solve local problems and in interpreting the results obtained as program output. Input and the associated output for sample problems or test cases serve to illustrate the proper format of this information and enhance the user's understanding. A discussion of the program operation, covering the options available to the user and any limitations imposed on program parameters by the author's choice of methods and algorithms, and advice on use of restart and recovery procedures and the meaning of error messages which may be encountered with suggestions for corrective action, should be included.

care of initialization of variables and error monitoring (e.g., treatment of underflow) when required. Although a listing of the program's source statements is frequently found in the program documentation, this practice is not endorsed by NESC. Often source statement changes are required to correct errors discovered after the publication has gone to press, resulting in discrepancies between the listing in the printed material and the source statements distributed as a package element. This can be confusing to recipients of the program. It is extremely simple to produce listings from the magnetic tape containing the source statements, and elimination of the listing from the printed material can considerably reduce the size of the documentation.

On the other hand, the inclusion of the *sample problem* input data along with printed output and graphs in the documentation is encouraged. The input data should be available in computer-media form also as a convenience to the package recipient who is expected to execute the test cases to verify the successful transfer of the package from source to target system. If a large quantity of output is generated by the sample problem, selected portions of the output listings together with a complete copy on microfiche, or recorded on magnetic tape, is suggested as a package element.

The *operating system control language* records used in the successful compilation or assembly of the software and execution of the sample problems on the source system make up the third package element of computer-media material. This element defines the device assignments, storage allocation, and the loading directives and overlay structure pertinent to the source system. With this knowledge the recipients can determine the appropriate substitutions for their local computing environment. Control information may be supplied alternatively as printed material since only in rare instances would this information be directly applicable at another installation.

System-dependent *object decks or load modules* prepared preliminary to program execution are sometimes included as a package element. The element is redundant when accompanied by the equivalent source statements. However, if source statements cannot be distributed for some reason, these run decks can be an adequate substitute to accomplish limited interchange between users of like systems.

External data files or *data libraries* (e.g., cross-section libraries, material properties tables) required for program operation may also appear as a package element, or a separate package. To accommodate data on computer systems with different word-lengths and number representations use of the decimal, or FORTRAN-formatted, form of the data is recommended. Routines for transforming the decimal data to the more efficient machine binary, or FORTRAN-unformatted, representation should be provided as auxiliary programs, the sixth computer-media package element.

The *auxiliary information* element contains supplementary programs developed for use in conjunction with the packaged software (e.g., to prepare or transform input data to meet program specifications, to process

CDC7600 facilities at Brookhaven National Laboratory and at Lawrence Berkeley Laboratory, the BCS-Richland UNIVAC1100 installation, and local IBM360, 370, 3033 and DEC equipment. For NESC testing the more popular or generally-available compilers supplied by the manufacturers are preferred over the program product or one-of-a-kind language processors in the belief that these more accurately reflect expected user experience.

During the compilation phase of testing the most frequent problems encountered are missing routines and compiler-detected errors. The missing routines are often modules automatically supplied by the system in the authors' computing environment, such as time-of-day, calendar, and accounting routines. Or they may be proprietary software, which is usually the case when the contributed program produces graphical output. Missing routines may be obtained from the authors, upon request, or replacements may be supplied by the NESC. Programs utilizing graphical output are always a challenge to exchange. The Center, and each recipient in turn, is required to insert the equivalent local plotting package before the test cases can be run. Compiler-detected errors are corrected by the staff reviewer with the authors' cooperation.

When the compilation is successful the execution phase of testing begins. In this phase the sample problem input data and the run modules, if included in the package material, are executed. The output produced is then compared with the sample problem output generated by the authors. During this phase the most common problems encountered are discrepancies between test output and the authors' sample output, unexpected error messages and abnormal termination of the test runs, and the use of uninitialized or undefined variables. Frequently, a reviewer discovers that the authors' sample problem output was not generated from the sample problem input data supplied, or changes have been made in the program since the sample output was produced and prior to submission of the other package elements. Some computing environments provide local error monitoring and error handling routines which override system error messages and initiate corrective action (e.g., set to zero upon underflow). Such local conventions can cause discrepancies which have to be resolved during testing. Operating system presetting of areas of storage to assigned values can often mask the authors' use of uninitialized or undefined variables. Reliance on such conventions should be avoided. Instead, authors should strive to write software which can stand on its own. During testing the NESC reviewer works with the authors to resolve the discrepancies detected.

Special tools have been developed at the Center to assist in the processing. They can be used to check that all routines called by the program being tested are available, to perform rudimentary checks for uninitialized or multiply-defined variables, unreferenced statement numbers, active variables in COMMON and EQUIVALENCE blocks etc., and to convert between a variety of character sets such as IBM EBCDIC, CDC Display Code, and UNIVAC Fieldata.

In addition to the package transmittal letter, the NESC supplies with each tape containing the computer-media a copy of the printed material and summary listings of the tape contents generated when the recorded copy of the computer-media was produced. A label is affixed to each reel identifying its contents, the date of recording, and indicating 7- or 9-track recording, odd or even parity, density (200, 556, 800, 1600, or 6250 bpi), and file structure including the character set (ASCII, BCD, EBCDIC, binary, etc.), and record format, length, blocking factors, and number of physical records (or blocks). The maximum record length is given for files having variable-length records.

The Center maintains records of all package transmittals so that recipients of library packages can be notified should an error be detected, or revisions or a new edition of the software become available. These statistics are used for reporting purposes also to impress our sponsors with the extent and value of software sharing and technology transfer so that the Center's budget can be increased and the backlog of software awaiting testing and packaging reduced.