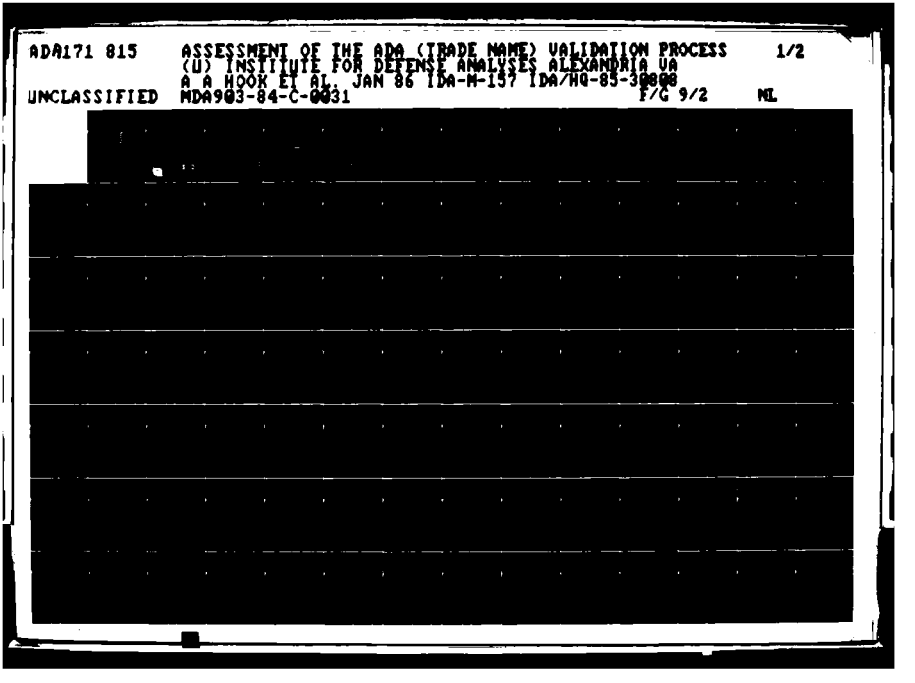
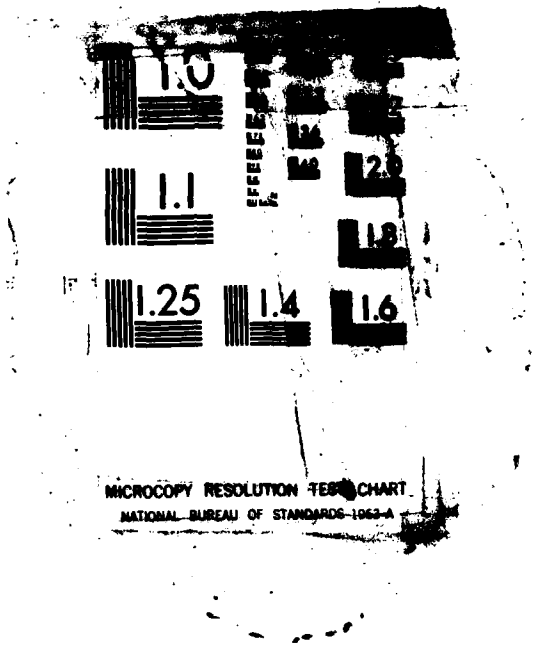
AD-A171 815 ASSESSMENT OF THE ADA (TRADE NAME) VALIDATION PROCESS (U) INSTITUTE FOR DEFENSE ANALYSES ALEXANDRIA VA A A HOOK ET AL. JAN 86 IDA-M-157 IDA/HQ-85-30808 1/2

UNCLASSIFIED MDA903-84-C-0031 F/G 9/2 NL




1.0
1.1
1.25
1.4
1.6
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

IDA MEMORANDUM REPORT M-157

ASSESSMENT OF THE Ada* VALIDATION PROCESS

Audrey A. Hook
R. Danford Lehman

December 1985

Prepared for
Ada Joint Program Office (AJPO)
Office of the Under Secretary of Defense for Research and Engineering

INSTITUTE FOR DEFENSE ANALYSES
1801 N. Beauregard Street, Alexandria, Virginia 22311

*Ada is a registered trademark of the U.S. Government, Ada Joint Program Office.

IDA Log No. HQ 85-30808

The work reported in this document was conducted under contract MDA 903 84 C 0031 for the Department of Defense. The publication of this IDA Memorandum Report does not indicate endorsement by the Department of Defense, nor should the contents be construed as reflecting the official position of that agency.

SECURITY CLASSIFICATION OF THIS PAGE AD-A171815

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS Public release; distribution unlimited.
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) M-157	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Institute for Defense Analyses 6b. OFFICE SYMBOL (If applicable) IDA	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State, and ZIP Code) 1801 N. Beauregard St. Alexandria, VA 22311	7b. ADDRESS (City, State, and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Ada Joint Program Office 8b. OFFICE SYMBOL (If applicable) AJPO	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code) 1211 Fern St., Room C-107 Arlington, VA 22202	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. T-5-304 / WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)
Assessment of the Ada Validation Process (U)

12. PERSONAL AUTHOR(S)
Audrey A. Hook, R. Danford Lehman

13a. TYPE OF REPORT	13b. TIME COVERED FROM ____ TO ____	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
		1985 Dec.	166

16. SUPPLEMENTARY NOTATION

17. COSATI CODES		
FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Ada, compiler validation, Ada Compiler Validation Capability (ACVC), Army Language System (ALS), verification, evaluation & validation, Ada Validation Office (AVO)

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This memorandum report is based on analyses of working papers, contact reports, ARPANET messages, and other records maintained by IDA as required to perform the assessment of the validation process as required by the Ada Joint Program Office (AJPO). The time frame of the assessment was from November 1984 through December 1985. The role of IDA is to adjudicate procedural and technical issues, interpret existing policies, and to retain an objective viewpoint concerning the issues and alternatives for their resolution. Recommendations are made which can improve the validation process.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION	
☐ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS		
22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE (Include Area Code)	22c. OFFICE SYMBOL

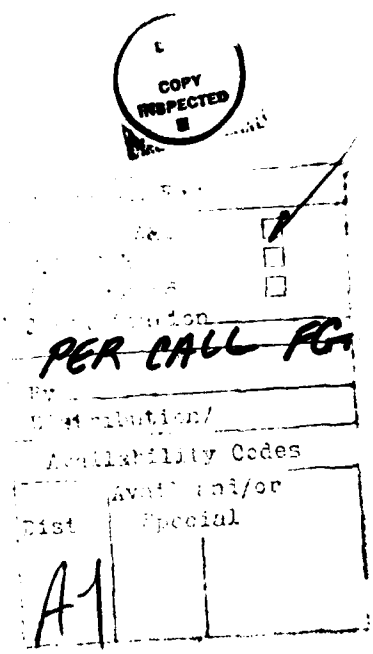
DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

IDA MEMORANDUM REPORT M-157

ASSESSMENT OF THE Ada* VALIDATION PROCESS

Audrey A. Hook
R. Danford Lehman



December 1985

INSTITUTE FOR DEFENSE ANALYSES

Contract MDA 903 84 C 0031
Task T-5-304

TABLE OF CONTENTS

ACRONYMS

ACVC	Ada Compiler Validation Capability
AJPO	Ada Joint Program Office
ALS	Army Language System
AMO	ACVC Maintenance Organization
AVF	Ada Verification Facility
AVO	Ada Verification Office
COTR	Contract Technical Representative
E&V	Evaluation & Validation
ECS	Embedded Computer Systems
FRT	Fast Reaction Team
FSU	Florida State University
ISA	Instructions Set Architecture
ISO	International Standards Organization
LAN	Local Area Network
LMC	Language Maintenance Committee
MCCS	Mission Critical Computer Systems
VHSIC	Very High Speed Integrated Circuit
VSR	Validation Summary Report

LIST OF TABLES

1.0 INTRODUCTION

This memorandum report was prepared as one of the deliverables under IDA Task T-5-304, Ada Validation, to provide the Director, Ada Joint Program Office (AJPO) with an assessment of the validation process as it has been implemented from November 1984-December 1985.

Although IDA participated in the validation process this past year, acting as the Ada Validation Office (AVO), its role has been to adjudicate procedural and technical issues, interpret existing policies, and to retain an objective viewpoint concerning the issues and alternatives for their resolution. Therefore, the purpose of this report is to provide the Director, AJPO, with an analysis of the procedural and technical issues encountered during the past 14 months and to recommend actions which can improve the validation process.

2.0 SCOPE

This report is based upon the analysis of working papers, contact reports, ARPANET messages, and other records maintained by IDA as required to perform the assessment of the validation process within the technical scope of task order T-5-304. This report addresses sub-tasks a, b, d, and e of task order T-5-304. These sub-tasks are as follows:

a. "Technical assistance in maintaining an Ada Validation Office (AVO) which will be manned by DoD personnel, transferring operations to it and improving the operation of the AVO and Satellite facilities during the transfer. Independent technical assessment of the performance of the Satellite facilities to ensure that consistent validation practices are enforced and that a certificate issued by one is equivalent to a certificate issued by another."

b. "Technical assistance in developing policies, procedures and tools to validate truly embedded computer systems and technical support to the ECS Ada Validation Working Group."

(c. Policy revision - subject of other deliverables.)

d. "Technical assistance to the AVO operation of the Ada validation fast reaction team, and participation in and documentation of the results of their deliberations."

e. "An analysis of the feasibility of testing implementation dependent features that are outside

of the standard and the possibility of an automated tool that will generate test cases for a particular compiler, given implementation dependencies as input."[TO2 1985]

3.0 BACKGROUND

The Ada compiler validation process is nearly three years old. By the end of this (the third) year, 45 validations have been completed---2 in the first year, 16 in the second, and 27 during the third. Nine vendors have re-validated compilers. By the end of December 1985, 13 vendors, including academic, government and commercial sources will have implemented compilers for 17 different hardware product lines including three compilers targetting chips intended for Mission Critical Computer Systems (MCCS) applications.

3.1 Summary of Problems. In this assessment of the validation process, the following problems have been addressed:

a. Clarification of steps in the validation process for both vendors and the government

b. The need for documented procedures defining Ada Validation Facilities (AVF) roles and responsibilities

c. Validation of a compiler for all possible configurations within a given time frame and particular Ada Compiler Validation Capability (ACVC) version

d. Impediments to the use of conforming Ada compilers for MCCS applications

e. Constraints imposed by Ada Compiler Validation Capability (ACVC) tests when executed on embedded targets

f. Inaccessibility of schedules for correcting disputed tests and incorporating them in a particular ACVC version

g. Non-standarization of the content of VSR's (redundancy, information omitted)

h. Maintenance activities of the ACVC maintenace organization (AMO) which hamper baselining the number of tests for each ACVC

i. Workload problems of AVF's

3.2 Historic Perspective of Validation Problems. During the first two years of the validation process, IDA performed validations and assisted the AJPO in establishing and training "Satellite" validation organizations which are now called AVF's. By December 1984, there were four new AVF's which were embarked on providing services for vendors of Ada compilers using a set of procedural memoranda issued by IDA during 1984. However, the AVF managers needed a procedural document defining organizational roles and responsibilities and AVF practices that are consistent with the AJPO policy and with the International Standards Organization (ISO) software certification system.[ACF 1984]

In November 1984, a scheduling difficulty for validation of the Army Language System (ALS) was encountered because of the December expiration of the current version of the ACVC. IDA convened a planning meeting with the AVF, the contractor, and Army representatives to resolve disputed tests and develop a plan of action.[ALS 1984] During the resolution of the ALS problem, it was evident that the steps in the validation process and the time required to complete them were not understood by the government and contractors. Additionally, IDA had received many telephone inquiries from vendors and government project managers who asked for the steps in the validation process.

During 1984, the problem of validating a compiler for several similar but not identical host/target pairs became an issue that required resolution. A vendor desired a certificate for a compiler which could be ported to several different hardware configurations but was unable to assemble these configurations for on-site testing before expiration of the ACVC version that had been used for pre-validation testing. Special consideration was given to this vendor based upon conditions established by the Director, AJPO, which resulted in a series of actions that could have significantly changed validation procedures had they been recognized as being appropriate for all vendors.[LIE 1985] However, the issue of costs for full ACVC testing on all possible combinations of hardware configurations and the requirement to complete validation prior to expiration of a particular version of the ACVC became part of a public debate on the interpretation of AJPO validation policy.[ADA 1985]

A group of DoD software contractors and compiler vendors joined together as an informal ECS working group to discuss the validation practices that could impede the use of conforming Ada compilers for MCCS applications. The requirement for annual revalidation during software development and maintenance, the perceived requirements for formal re-validation whenever a compiler has been re-hosted/re-targetted to a different hardware configuration, and validation of the real target machine were the primary

issues addressed by this group. IDA provided an AJPO interface with the ECS working group for the purpose of assisting them develop constructive policy recommendations for the Director, AJPO, after they had given adequate consideration to technical and procedural problems that their policy recommendation might entail.[ECS 1985]

The technical problems encountered when ACVC tests are executed on embedded targets (simulator or an instantiation of a real target processor) were issues that required cooperative resolution by IDA, AVF's and compiler implementors. The requirement for being able to process ACVC tests for input and output was challenged by an implementor whose target machine, a Z8002, had only 48K bytes of random access memory and did not support external memory devices. An implementor who intended to implement an Ada compiler targeting Intel Very High Speed Integrated Circuit (VHSIC) processors with optional run-time support environments, was concerned that validation could not be performed because of the implementation differences in these run-time environments.[SOF1985] Another implementor found that the reliability and implementation differences in 1750A hardware greatly increased the complexity of ACVC executions and was forced to delay on-site testing.

During the use of ACVC versions 1.5 and 1.6, there were 117 tests disputed by vendors who challenged their correctness. Most of these disputes were filed with IDA after the vendor had set a target date for on-site testing. Expeditious resolution of these disputes was necessary so that the vendor could proceed with validation. Since the tests were disputed over a 12-month period, those vendors who validated early in the period passed tests that were withdrawn from the test suite pending Language Maintenance Committee (LMC) action later in the year. The schedule for correcting these tests and incorporating them in a particular ACVC version has not been published. Therefore, vendors who validated before these tests were withdrawn are uncertain as to when they must change their compilers to comply with LMC interpretations.

As the AVF's began preparing Validation Summary Reports (VSR's) documenting the results of ACVC testing, they often copied sections of past reports that did not reflect the results of the validation being documented. The content of the VSR was repetitious in test listings, presenting multiple opportunities for errors in transcribing numbers; it also lacked the implementation dependent information required of the vendor by Appendix F of the Ada Programming Language, ANSI/MIL-STD 1815A. Errors were made in accounting for the number of tests in the ACVC suite and the disposition of these tests (withdrawn, inapplicable, passed/failed).[WIC 1985] Information describing the testing environment was omitted from some VSR's.[ICH 1985]

The ACVC maintenance contractor may remove tests from the ACVC test suite whenever an error (logic or typo) has been discovered by an implementor or by internal quality control procedures. Even though the ACVC has a pre-release period of six months (public review period), removal of tests with errors usually occurs during the first to third month after the suite has been released for use in the validation process. However, the AMO can remove a test at any time during the release period as part of their maintenance activity.[KNO 1985] This practice makes it difficult to establish a baseline for the number of tests each ACVC version contains when first distributed and, then, to account for the changes in the number of tests in that version while it is being used for validations.[MGR 1985]

The workload for AVF's is increasing due to the the increasing numbers of vendors who are offering Ada compilers for validation and the number of different hardware systems and operating systems being used by these vendors. The pre-validation and on-site analysis has been mostly a manual process. AVF managers need more automated aids to reduce the labor intensiveness of the testing method since the staff to do this work is a limited resource. Several AVF's have had difficulty recruiting and retaining qualified people for validation work; however, if these people can be found in greater numbers, the AVF managers do not want to increase their charges to vendors based upon more people engaged in a manual process. AVF managers need validation process improvements.[MGR 1985]

The findings, conclusions, and recommendations that follow address these problem areas.

4.0 FINDINGS AND CONCLUSIONS

During the past 14 months, some problems encountered have been policy issues while others have been procedural and technical issues. Many of the procedural and technical issues have been resolved while others have not been adequately addressed because the AJPO policy for validation has been evolving while validations were taking place.

4.1 Summary of Findings. Based on the present assessment of the validation process, we found that:

a. Formalization of the draft AVF procedures document is needed.

b. Challenges (successful and unsuccessful) were issued to the requirements for assembling all equivalent configurations for on-site testing.

c. Attempts were made to resolve the issue of how to validate a compiler for a family of architecturally-related processors through the evolution of the family architecture test method.

d. Issues regarding the validation of an embedded target were raised by industry and dealt with on an individual, negotiated basis.

e. The successful application of the Fast Reaction Team (FRT) method encourages the establishment of documenting procedures for providing test corrections and generating new tests.

f. Issues have been raised in the past year regarding the quality, content, and timeliness of the VSR. An action list has been generated.

g. The problems of heavy cyclical workloads for AVF's and re-validation costs to vendors could be minimized with automated test analysis and documentation tools. Changes to the official ACVC release schedule could more evenly distribute validations.

h. Design of automated tools for detecting and testing for features that are outside a formal standard are in the early research phase.

Discussion and conclusions relating to these findings follow.

4.2 Formal Documentation of the Validation Procedures. Up to 1985, the written procedures distributed to implementors defined the validation process as consisting of three steps (i.e., Scheduling, Certification, Validation).[IDA 1984] Vendors, government project managers, and AVF managers soon learned that completing these three steps involved other actions and events that required planning and scheduling. As part of the first ALS planning meeting, IDA produced a PERT chart that identified all the actions required by the implementor, the AVF, the AVO, and the AJPO. Based upon this analysis, IDA defined a ten-step validation process which was incorporated in a draft AVF procedures document. Although these procedures have not been issued as a final document during 1985, they have been the "working" document used by European AVF's to produce their internal procedures, obtain national certification, and publish brochures for customers. In the U.S., this "working" document has been used by the Federal Software Testing Facility (now the Federal Software Management Support Center)for negotiating schedules, and by IDA in responding to frequent telephone calls for this information. Appendix A provides these steps as they are written for the draft procedures document.

4.2.1 Content of Procedures. The draft procedures (IDA Paper P-1900) for conducting the validation process defines organizational roles and responsibilities consistent with ISO concepts for software certification and identifies responsibility for executing the steps in the validation process. These procedures clarified the AVF's role as the vendor's interface to the validation process so that vendors do not need to contact the AJPO or the AVO to obtain information or to resolve procedural issues. These procedures also contain samples of a contract, vendor letter of intent and declaration of conformity, on-site testing procedures, and a VSR.

4.2.2 Conclusion. Since the AVF managers, Ada Validation Committee, and the AJPO have reviewed and commented on these procedures, we conclude that they meet the need expressed by AVF managers at the beginning of 1985. Additionally, this document forms the basis for consistent practices among AVF's.

4.3 Testing Procedures. During 1984, three vendors (Data General, Digital and Honeywell) validated their compilers for more than one configuration within their product line by compiling and executing the full ACVC test suite on all host/target pairs. For on-site testing, Data General provided six configurations and Digital provided five configurations. The testing for each vendor required five days. Certificates were issued for six and five configurations respectively. Honeywell provided four configurations and testing required three days. A certificate was issued for the four configurations provided on-site.[VSR's 1984] Several other vendors were unable to obtain configurations for on-site testing. Therefore they either scheduled and paid for another validation when the configuration could be provided (Verdix) or else omitted on-site testing for a configuration they certified as being equivalent (TeleSoft). Beginning in 1985, vendors challenged the requirement for assembling all equivalent configurations for on-site testing.

4.3.1 Vendor Certification. A vendor has been prohibited, by current policy interpretation, from marketing the compiler as validated if that configuration does not appear on the certificate issued by the AJPO. Wording of the certificate implies that the compiler is authorized for use only on those configurations enumerated on the certificate. Moreover, policy had been strictly interpreted to discount any pre-validation testing done by the vendor on equivalent configurations and to list on the certificate only the configurations on which the AVF had compiled and executed the full ACVC suite. However, in 1983 an exception was made for Data General by listing the MV8000-II on the certificate based upon vendor certification that the configuration was architecturally and instruction set identical with the MV8000.[IDA 1983] A requirement for providing that

configuration for AVF testing when the compiler was due for re-validation was attached to this exception.

Another exception was made for TeleSoft in 1984 which involved an AVF in the analysis of vendor conducted testing on equivalent configurations and issuance of a certificate by the AJPO based upon the AVF's off-site analysis of the vendor's test results. As in the 1983 exception case, the vendor would be required to provide all configurations added to the list for re-validation by an AVF before expiration of the original certificate.[W-P 1984] This agreement created a major problem in that the vendor used it to expand his market base by offering ACVC testing services himself with assurances to his client that a certificate would follow. This precedent for modification of testing procedures so that on-site testing by an AVF team is not required before a certificate is issued by the AJPO was brought to an end in 1985.[KRA 1984]

4.3.2 Test Methods for Family Architectures. The issue of how to validate a compiler for a family of architecturally related processors was addressed jointly by the AJPO, IDA, and the U. S. AVF managers. The existing requirement for validating a compiler for host H targetting T was to compile the entire ACVC on H, and to execute the compiled ACVC on T. But when a compiler was developed for a series of hosts H1-Hx targetting a series T1-Ty, such a requirement was seen to be too severe: it would require that the vendor run the ACVC on all possible combinations of host/target pairs, and to have all of the series H and T present for on-site testing. The following indicates how this issue was addressed through the evolution of the family architecture test method.

4.3.2.1 Verdix. The first family validation was performed by the AVF at Wright-Patterson AFB for the Verdix Corporation. Verdix requested validation of a compiler that could be hosted on and target any of four VAX models operating with either of two operating systems---eight possible host-target pairs. Ultimately, Verdix was unable to obtain one member of the series, so their validation covered six possible host-target configurations. The AVF analyzed pre-validation runs from one member using both operating systems, and the other two members using different operating systems. In this way, it was confirmed that the compiler was indifferent to which of the series's members or operating systems was used. On-site testing was performed, at different times, with two members of the series, each using a different operating system. However, the VSR documentation did not reflect the agreement between the AVF and implementor as to the acceptability of pre-validation analysis for establishing an on-site test basis which could be extended to cover all configurations. Consequently, the implementor received two certificates, one for each configuration tested on-site.

This case illustrates the problem of a software vendor who is not able to assemble all configurations at one time for pre-validation testing and, again, for on-site testing. The AVF manager attempted to assist this vendor by accepting pre-validation results as evidence that a limited number of configurations could be used for on-site testing.

Two problems were encountered with this approach. First, the vendor was unable to gain access to all configurations required to establish an on-site test basis in a reasonable time frame. Therefore, the vendor scheduled on-site testing for one configuration as soon as possible to have a marketable validated compiler for that configuration. The second problem involved the lack of documentation of the test method in the subsequent VSR issued after the vendor had submitted agreed upon pre-validation results for equivalent configurations and had assembled the second test basis configuration for on-site testing. This lack of documentation on the acceptability of pre-validation testing to establish equivalence of configurations other than the one that could be tested by the AVF resulted in the vendor receiving certificates without the equivalent configurations being listed.[M115 1985]

4.3.2.2 Data General. The second family architecture validation was performed by the GSA's Federal Software Testing Center (now the Federal Software Management Support Center) for Data General Corporation. Data General requested validation for a compiler that could be hosted on and target any of eight Data General models. Targets could use either of two operating systems. Thus there were 128 possible host-target configurations to be validated. Pre-validation testing showed that only four of the models could use one of the operating systems, so the actual validation covered 96 host-target configutations.

An agreement in principle was reached between the AVF, the AJPO, IDA and the vendor on the testing procedures.[H00 1985] The vendor would perform fairly extensive pre-validation testing with all members of their series. On-site testing would be done on just four members.

For pre-validation, the AVF required the running of the entire ACVC on one member, with execution under both operating systems. The results from this running were analyzed, found to be correct, and thus used as the test baseline against which subsequent results could be compared to determine correctness. The load module created from this running was re-targetted to all other members and executed using either operating system. Additional testing was done on these other members with a subset of ACVC tests; this subset was compiled and executed by each member.

During on-site testing, the entire ACVC was run by the same member that ran it for pre-validation; the results were determined to be correct. These results served as the test basis. The load module from this first step was then executed on the same member under the second operating system as well as on the other three members using both versions of the operating system. All results were compared to the baseline for further testing; no differences were detected. A subset of the ACVC tests, selected by the AVF test team, was compiled and executed on two of the three other members, with one of the load modules produced also executed on the third member. This testing method, fully documented in the VSR, required five days for on-site testing with four machines, (one less than in 1984). However, it resulted in a certificate for eight configurations, two more than in 1984, and two operating systems.

Subsequently, other validations for multiple hardware and operating system configurations have been conducted by adapting the method developed for the Data General case (e.g. ALS, Digital, Honeywell, Alsys). Based upon our participation with AVF managers and vendors this year, we conclude that this testing method reduces vendor costs. However, its success depends upon obtaining a negotiated, documented agreement on pre-validation and on-site testing procedures before the vendor has completed pre-validation.

4.3.3 Conclusions. Answering the following four questions is central to the agreements:

a. What does the vendor want on the validation certificate? (e.g., enumeration of the host/target configurations and operating systems, software/hardware simulator, real embedded target that should be listed on the certificate.)

b. When does the vendor want to conduct on-site testing?

c. Does the vendor have access to all the configurations enumerated in the answer to the first question, for pre-validation testing and for on-site testing, given the desired date for on-site testing?

d. What actions can the vendor and the AVF manager take to create a technically valid testing basis for the domain of hardware configurations the vendor wants to appear on a certificate?

When a verbal agreement has been documented and agreed by all parties to the negotiation, the AVF manager and vendor have a plan of action, even though it may be revised during its execution. The VSR must document the actions completed

during pre-validation, as well as on-site; thus providing a rationale for the certificate that is being recommended.

4.4 Embedded Targets. The group of DoD software contractors and compiler vendors who met periodically with IDA and AJPO staff during 1984 and 1985 raised discussion issues concerning the requirement for full ACVC testing on all possible combinations of host/target pairs, for re-validation during a long software development project, and for ACVC testing of the embedded target processor. Appendix B provides the minutes and other related papers produced as a result of these working group meetings.

Although policy issues concerning validation of an embedded target were raised by this group, the procedural issues were not addressed in depth. However, IDA and the AVF managers were dealing with these issues on a case-by-case basis. The family architecture test method approach provided a basis for developing on-site and pre-validation procedures that were negotiated with the vendor. The following three cases illustrate the results of this approach and indicate unresolved problems.

4.4.1 Florida State University. This implementor at Florida State University (FSU) challenged the requirement for processing ANSI/MIL-STD 1815A Chapter 14 tests since the application intended for the target processor does not require I/O. IDA presented this challenge as a policy issue for consideration by the Ada Language Standard Working Group Members as follows:

> Issue: Is Chapter 14 compliance appropriate for embedded systems?
>
> Embedded applications for weapons systems and for secure systems will not require nor allow the full interface capability of Chapter 14. Current policy requires that all Chapter 14 tests be executed if only to raise exception messages for the test that cannot be supported by the hardware configuration. Moreover, implementors must successfully pass all the (dot) Ada tests. Some implementors need to use a simulator for the target system to develop the application software for a weapon system. Does extra code need to be written just to comply with all of Chapter 14?[IDA 1984]

"Loading Object Code for Execution. We loaded object code for execution by sending it across telephone lines

from the host to the target machine. This process is quite error prone. The validation process will be complicated by the presence of these errors. For example, a transmission error on one program may cause errors in the programs which follow, requiring that a whole series of programs be reloaded.

TEXT_IO. Our target does not support any external memory devices. Hence, our implementation of the input output package does not allow files to be created or opened. The Chapter 14 tests had obviously not previously been executed on such an implementation. We found many of the tests to be in error. Of the 250 tests, only some 30 had any significant execution behavior on our implementation. It would be a service to embedded system implementors to remove these tests from consideration for such implementations. The cost of compiling and executing the tests is significant, especially due to the relatively large amount of code to be loaded for text_io."[RIC 1985]

4.4.2 Westinghouse. This implementor's compiler targeted a 1750A software simulator and three or more different hardware implementations of a 1750A chip. A VAX CLUSTER® and DECNET® provided the host environment and communication links with the 1750A hardware configurations. A line printer was cable connected to the 1750A hardware configurations to pass Chapter 14 tests. A planning meeting was requested by the implementor to develop a testing method. During this meeting, the implementor indicated that ACVC testing with the 1750A hardware configurations was impacting his pre-validation schedule because of hardware downtime. Therefore, he intended to continue testing and changing his compiler until the AVF team arrived to do on-site testing.

The implementor was informed that he must freeze his compiler once he has submitted pre-validation test results to the AVF for analysis since these test results will be compared with those obtained during the on-site test. The family architecture test method was negotiated for the testing of the several target processors as follows:

For pre-validation the implementor would perform the following steps:

a. Compile and execute the entire ACVC on the host and simulator target.

VAX CLUSTER® and DECNET® are trademarks of the Digital Equipment Corporation.

b. Execute all CZ and CE ACVC tests on all available hardware targets.

c. Execute all ACVC tests not run in step 2 above, on one target used in step 2.

For on-site testing, the AVF would perform the following steps:

a. Compile and execute the entire ACVC on the host and simulator target.

b. Execute all CZ and CE ACVC tests on all available hardware targets AT THIS TIME (i.e., the actual hardware available may be a proper subset or superset of that used for pre-validation testing, or it may be another set).

c. Execute a subset of the ACVC tests (comprising approximately 60 tests) on all hardware targets used in step 2.

The Certificate will list only those target configurations tested during on-site testing; the VAX CLUSTER® will be listed as the host.[LEH 1985]

The implementor agreed that this test method would provide him with the rigorous testing he desired to ensure that the compiler is a conforming one with a simulator and hardware implementations of 1750A. Unfortunately for the implementor, he had continued target hardware problems that forced a delay in on-site testing and repetition of pre-validation testing using a later version of the ACVC.

4.3.3 SofTech. This implementor took a hardware versus software approach to the requirement for passing all applicable ACVC test. SofTech validated a version of the ALS compiler which had been modified to generate code for an Intel 86/30 chip set. The target configuration used to execute the ACVC contained the I-8087 hardware floating point and used the host I/O capability for Chapter 14 tests. After successfully passing the validation tests and obtaining a certificate from the AJPO, this compiler will be delivered to a contractor who will use it to develop an Ada application for a radio. This application will not use the floating point chip nor will it have I/O requirements.

SofTech intends to modify the ALS compiler for use by other embedded application developers who use different configurations of the Intel 8600 family. These variations will be based upon the requirements for an intended application and may include different interrupt controllers and timers, different processor frequencies, multiple

processors, network processors, etc. The goal is to "deliver a compiler in a form that allows a user to reconfigure the run-time library in much the same way that operating systems can be reconfigured through a systems generation facility."[SOF 1985]

4.4.4 Conclusion. Based upon the experience gained with the validation of compilers that target an embedded processor, we conclude that each implementor appears to have taken an application-specific approach to the implementation of a compiler. Technical decisions seem to have been based on meeting a specific user-defined application in the most cost-effective manner rather than to maximize the versatility of the compiler.

The FSU target represented an instantiation of an embedded target with limited capabilities not only for ACVC testing but for the debugging and testing of an application requiring some I/O functions or interfaces with hardware drivers. However, the usability of the compiler for debugging programs containing interfaces with I/O functions had not been a consideration in the development contract.

The Westinghouse implementor wanted to prove that the software simulator and hardware specific run-time library in the host environment were an accurate instantiation of the real target hardware. By taking this approach, he encountered technical problems that were ancillary to the problem of testing for language conformity. This implementor could have validated, first, with a software simulator, and then solved hardware and unique run-time library problems before re-validation. Presumably, company policy or a contractual specification influenced his approach.

If SofTech pursues their design goal, ACVC testing for the target with multiple run-time environments must be negotiated prior to pre-validation testing by the vendor. The ACVC does not test for the resulting correctness of a program executed when memory required by a main program is shared among optional run-time environments.[BOE 1985] Testing for exception handling may require demonstration under all optional run-time environments. Compiler vendors may chose to implement application unique run-time support libraries using pragma interface, chips, or with Ada software. The evolution of test procedures for these more complex compiler implementations will be best done by documenting the experience of doing the validation and discussions with a group such as the ECS or Ada Validation Board Committee on the technical merits of each case. It will be important that AVF managers fully understand the implementation dependent characteristics of the compiler under test. See Section 4.5 for discussion of documentation of implementation dependent characteristics.

4.5 Fast Reaction Team. The Fast Reaction Team (FRT) is a group of Ada language experts called upon to advise the AVO on the merits of vendor disputes of ACVC tests. The members are chosen on the basis of availability, interest in language interpretation, and expertise. During this past year there were seven active members on the FRT. Two additional members were added during December 1985. The current membership list is:

John Goodenough	(SofTech)
Paul Hilfinger	(University of California)
Ron Brender	(Digital Equipment)
Robert Dewar	(New York University)
Erhard Ploedereder	(Tartan Laboratories)
Robert Knapper	(IDA) (Clyde Roby, alternate)
Brian Wichman	(National Physical Laboratory)
Henry Dancy	(ALSYS) (Michael Woodger, alternate)
Stephan Heilbrunner	(IABG)

Upon receiving a vendor's list of disputed tests from an AVF, the IDA coordinator, Dan Lehman, sends the disputed test and the pertinent facts concerning the vendor's arguments to the FRT over the ARPANET. The consensus is usually clear after the initial response of all members, and a ruling can be returned to the AVF. When some FRT members do not respond to the first FRT broadcast message, the IDA coordinator polls them a second or third time, depending on the number of members who are responding. Usually, FRT members notify IDA of their intent to abstain from dispute resolution, for a period of time, when they perceive a conflict of interest due to their involvement in a particular compiler implementation. Also, many members notify IDA by ARPANET when they have vacation plans. We expect vacation absences of the U.S. members during September and of the European members during May and August. When the full team is not available, a more extended dialogue with those who are may be required.

4.5.1 FRT Performance Record. The goal is to return a decision within a two work-week period. During the past year there have been 117 tests disputed with resolution ranging from 5 to 26 days, (including weekends). The average resolution time has been 14.3 days (including weekends). The 26-day dispute resolution occurred in August and September when the vacations of some U.S. and European members coincided and ARPANET down-time exacerbated the communication problem.

4.5.2 ACVC Quality Record. Tests may be ruled correct, incorrect, dubious, or inapplicable for the particular implementation. Incorrect tests will be referred to the AMO for correcting if possible. Dubious tests will be the subject of commentaries for the LMC. These commentaries are then referred to the AMO for modification of withdrawn tests

or development of new tests. Of the 117 tests disputed by vendors, nearly 56% (65) were withdrawn or approximately 3% of the ACVC tests were in error. Of the tests withdrawn in ACVC version 1.6, there were 25 tests that had also been withdrawn from ACVC version 1.5. These test should have either been corrected or not included. Appendix B provides documentation of FRT decisions during the past year.

4.5.3 Conclusions. Based on the evaluation of the FRT, the following conclusions were reached:

a. The FRT has provided a valuable service in expediting the validation processes and dealing fairly with the technical merits of each vendor's challenge.

   Although the elapsed time for resolving a single test dispute can vary from several days to three weeks, on the average the goal of responding within two work weeks has been met. Additionally, the FRT dialogues has been conducted in a manner that is free from vendor counter-arguments so that only the technical merits of the dispute are at issue. After the AVF has provided the vendor with the resulting opinion, a vendor may enter additional arguments for FRT consideration. There has been only one instance when that has occurred this year. The FRT members agreed to refer this dispute to the LMC rather that delay the vendor's validation by causing him to make a compiler change he (as a member of the LMC) considered incorrect.

b. Although we track FRT actions and LMC actions, we have not been successful in obtaining a current baseline or schedule from the AMO concerning the reappearance of withdrawn tests in the ACVC.

   Vendors and AVF managers have requested current information concerning the number of tests contained in an "official" release of the ACVC, throughout its life cycle and the status of tests referred to the LCM. They want to know the current baseline of tests before they submit test disputes, if an LMC commentary has been approved that will change a test in the future, and when the changed test is scheduled to re-appear in the ACVC. The interest from vendors and AVF managers concerning test corrections and new tests is sufficiently great enough to warrant establishing procedures for providing this information on a recurring basis.

4.6 Validation Summary Report. During the past year, issues have been raised concerning the quality, content and timeliness of the Ada Compiler VSR. A VSR is the report

written by an AVF to document the conformity testing of a compilation system. The report contains general information about the ACVC tests and the testing procedures used for conformity testing, as well as specific information about the particular compilation system tested that was derived from, or necessary for, the testing.

4.5.1 VSR Quality. A VSR's quality is measured in terms of accuracy, comprehensiveness, and consistency with other VSRs. A sample of 20 VSR's were analyzed. The number produced by each AVF is as follows: BNI...1, IABG...2, FSTC...4, and W-P AFB...13. The quality of these VSR's does not vary significantly from one to another; the format used in all VSR's is nearly identical. However, there are inaccuracies within VSR's and inconsistencies among VSR's. For instance, there are ACVC tests detailed as "inapplicable" or "withdrawn" in the VSR body but listed as "withdrawn" or "inapplicable"--or "passed"--, respectively, in the appendix; or a test may be given as "inapplicable" in the appendix without being detailed in the body. Some VSR's do not include an appendix. Some tests were given as "withdrawn" in some VSR's and as "inapplicable" in others. Some VSR's omit the Section 4.2.5, "Performance Information," with a resulting diference in the numbers of the subsequent sections of Section 4.0.

Even the total number of tests contained in a particular ACVC version differs between VSR's. The accepted practice had been for AVF managers to submit a final VSR to IDA for signature and forwarding to the AJPO. It had also been the practice for IDA to expedite the processing of VSR's except in those cases where a major technical inaccuracy was detected, such as incorrect nomenclature for a hardware/operating systems/ACVC test. IDA changed this practice by initiating reviews of draft VSR documents; this practice is gaining acceptance.

4.6.2 Content of VSR's. Of the VSR's 35 sections, only four sections (1.1, 4.1, 4.2.6/7, & 4.2.7/8) and the appendix contain any real information applicable to the documented conformity testing. The section on withdrawn tests, for instance, may contain information produced entirely from previous conformity testing. The information that is new to a VSR is only:

a. The identification of the tested configuration (the compiler name, hardware specification, and operating system)

b. The compiler parameters (e.g., maximum integer size)

c. The information derived from the tests--usually dependency tests

d. The test method

e. The complete listing of test results contained in Appendix A

Implementation specific information has not been reported in the VSR. This information would be useful in assessing the capabilities of an Ada compilation system. This information is that which must be included in Appendix F of the compiler's Ada Language Reference Manual. One proponent of adding implementation specific information recommends that all the parameters of Appendix C and all of Appendix F be provided by the vendor as part of his declaration of conformity submitted before on-site testing. This information can be adjusted (in agreement with the vendor) if any inconsistencies arise that are detected by the validation testing or observations of the actual test machine(s), and by publishing this information in the VSR. The argument for this approach is that, although the VSR relates to just the specific hardware/software system tested, potential buyers of compiler's need to know implementation dependent features before they generalize concerning the suitability of that compiler for a similar configuration. For example, if a vendor corrects a bug in the microcode of his system to pass validation while all other delivered systems have this bug, the vendor should have declared this fact in the VSR.[WIC 1985]

The optional inclusion of timing information has created some discussion concerning the advisability of its being part of a VSR. Although the timing information was intended as a factual descriptive of the on-site testing, some AVF's have omitted it altogether, or, when it was included, did not fully describe the mode of testing or the problems encountered during testing, such as, use of a shared machine versus dedicated configuration, re-starts after system software error correction. Discussion of this problem with AVF managers led to the conclusion that the elapsed time for the testing period would be reported along with an accurate description of the testing mode and any system problems that effected the elapsed time.[IDA 1985]

The VSR is still viewed by many as the most important source of consumer information. The following view of a DoD project manager is typical of the inquiries IDA has received from individuals who are looking for buyer's information from a VSR:

> Although these summary reports provide the required information for validation of a compiler, we felt that there were some information items that could be included in such a report that could assist potential users in

decisions regarding the relative merit of a given compiler.

a. Time to generate an executable program from two or more standard modules. This time should include compile and link times and any other actions required to generate the executable file.

b. Relative compiled program size. A standard reference program should be compiled, linked, and developed to the executable file stage. The resulting file size should then be compared to a given standard; either an equivalent program compiled from another high order language such as FORTRAN or assembled from an equivalent assembly language program written for the target computer system.

c. Relative execution time. A standard reference program should be compiled, linked, and developed to the executable file stage. The resulting program should then be run and its time to execute compared to an equivalent program compiled from another high order language such as Fortran or assembled from an equivalent assembly language program written for the target computer system.

Information of this nature should be included in the report to assist management personnel select an Ada compiler for their use when required.[AFC 1985]

Since the VSR is the only documentation of independent testing conducted on a specific compiler, the public continues to use it as a buyer's reference. If the Evaluation and Validation (E&V) team were to devise a similar report for performance testing, the distinction between validation and evaluation would be apparent to the public.

4.6.3 VSR Timeliness. The VSR should be able to be issued fairly soon after completion of that testing since there is a limited amount of new information contained in a VSR. Moreover much of this new information (e.g., hardware specification, compiler name and parameters, and descriptions of withdrawn and inapplicable tests) is known well in advance of the AVF's on-site testing. Indeed, conformity testing is currently regarded as a pro forma step in the validation process that is expected merely to confirm the pre-validation analysis. To date, however, only one draft VSR has been issued 30 days after completion of on-site testing. Also, a one draft VSR was issue 80 days after on-site testing. A VSR should be issued with minimal delay, because a review of the draft VSR by IDA and the AJPO is required prior to preparing

the final VSR. Furthermore, the advertising of an Ada compiler as "validated" may not be made until the AJPO agrees that the VSR substantiates issuance of a certificate.

Table 1 provides the pertinent dates and elapsed time for validation documentation flow.

4.6.4 Conclusions. Based upon our analysis of the quality, content and timeliness of VSR's during the past year, we concluded that the following actions were required:

Action 1. A standard annotated outline for a VSR was needed. The outline should eliminate repetitious sections and should incorporate the implementation dependent information provided by the vendor in his declaration of conformity.
Status: Pat Knoop volunteered to have the AMO contractor do this. A review draft was provided to AVF managers and IDA in May. The revised standard outline was received in October and is being used to prepare the November and December VSR's. Some minor problems have been encountered during this usage period. These problems can be corrected by a revision to the current outline.

Action 2. The AVF must account for all differences between ACVC tests used by the vendor for pre-validation analysis and those used by the AVF for on-site testing.
Status: This action is being accomplished by telephone and ARPANET messages between IDA, the AVF manager, and AMO.

Action 3. The AVF must accurately describe the testing mode and any difficulties that effect the time required to complete testing. A disclaimer must be included in this section to discourage reader's from extrapolating compiler performance characteristics.
Status: IDA is ensuring that this action has been taken by its review comments on draft VSR's.

Action 4. The AVF must accurately describe the testing method used for pre-validation and for on-site testing. Care must be taken to explain these actions in terms of the domain of equivalent configurations the AVF certifies for the compiler.
Status: IDA and the AJPO are ensuring that this becomes a standard practice with review comments on draft VSR's.

Action 5. AVF's should begin preparation of the VSR prior to on-site testing using the Standard VSR outline and a word processing system. Statistical tables can be prepared from a standard format with some numbers entered, and verified, before on-site testing. Other

TABLE I
PERTINENT DATES AND ELAPSED TIME
FOR VALIDATION DOCUMENTATION FLOW

Val. Ref. # (1)	On-Site Testing Ends	# of Days Until	VSR Date (2)	# of Days Until	Total # of Days	At IDA	# of Days Until	At AJPO	# of Days Until	Val. Certif. Issued	Total Elapsed Time (In Days)
1	3 Dec	64	5 Feb	7	71	12 Feb	9	21 Feb	60	22 Apr	140
2	8 Apr	67	14 Jun	19	86	3 Jul	9	12 Jul	60	10 Sep	155
3	30 Apr	38	7 Jun	5	43	12 Jun	8	20 Jun	82	10 Sep	133
4	30 Apr	24	24 May	6	30	30 May	0	30 May	(?<14)	4	30+ (?<14)
5	3 May	15	18 May	32	47	19 Jun	1	20 Jun	82	10 Sep	130
6	10 May	14	24 May	46	60	9 Jul	3	12 Jul	60	10 Sep	123

(1) The validation numbers refer to: 1-TeleSoft VAX, 2-Verdix VAX-11/785, 3-Verdix Sun, 4-Rational, 5-Data General, 6-Rolm.

(2) The difference in times surrounding the VSR date between #s 1-4 & 5-6 is due to different practices of the 2 AVFs: Wright-Patterson AFB (#s 1-4) dates the VSR after the vendor has reviewed it, FSMSC dates (#s 5-6) the VSR prior to concurrent vendor & IDA review.

information and numbers must be extracted from the log-book after on-site testing.
Status: Has not been fully implemented due to AVF resource limitations.

Action 6. A concurrent review of the draft VSR is required. The vendor, AJPO and IDA should receive copies of the draft VSR at the same time. The AVF should set a deadline of two work weeks for the return of comments, unless unusual circumstances apply.
Status: Implemented as of mid-November 1985

4.7 AVF Workload. This section is an examination of the frequency of validations and automated support for validation procedures. The workload of an AVF is a function of the amount of testing to be performed per validation (discussed in the sections on testing methods), the amount of automated support available for use in analysis, and the frequency of validations.

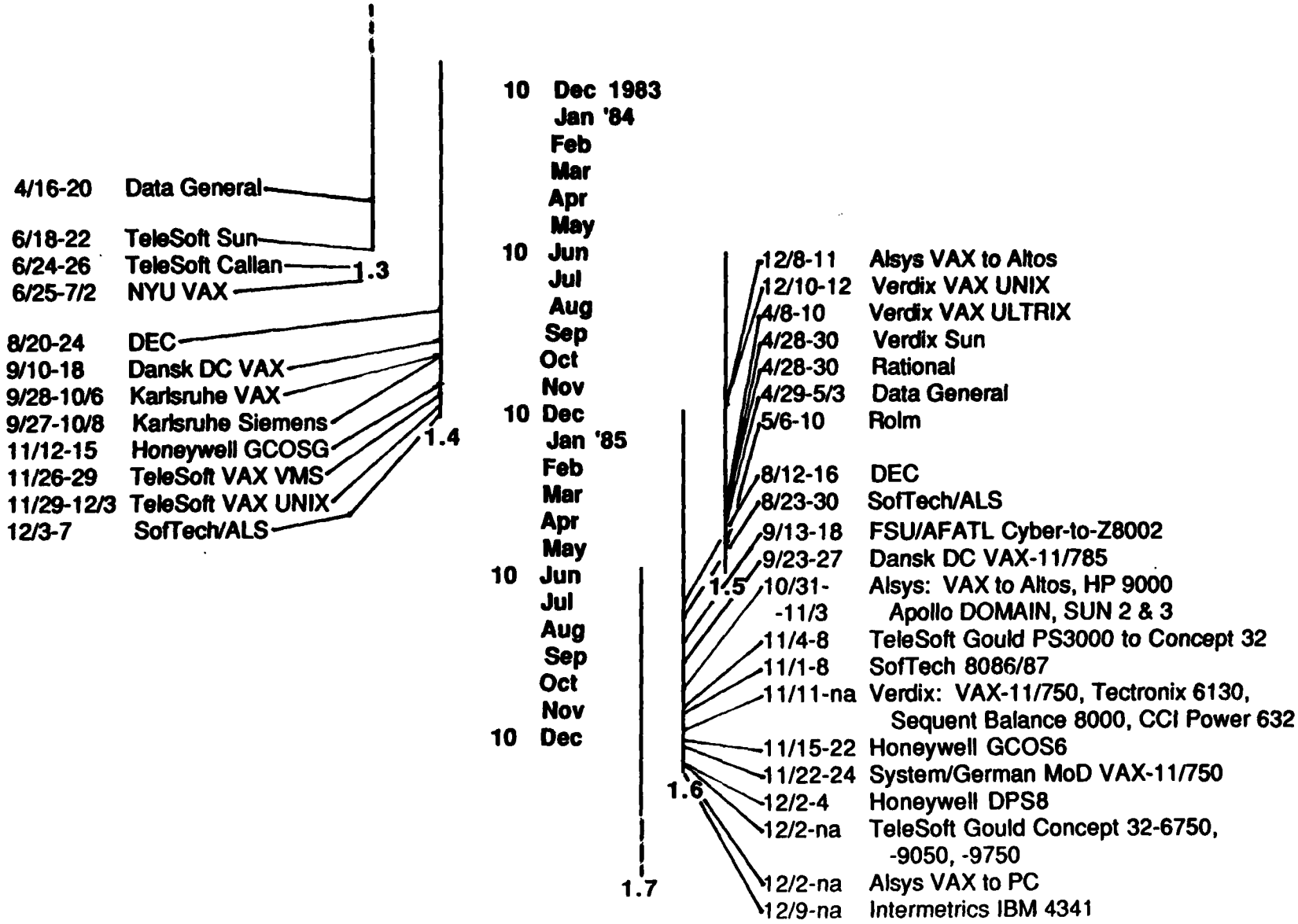
In Table 2 we have depicted the 45 validations to date by their date of occurrence and the ACVC suite used. Although the one-year life of a certificate induces a vendor to validate on an annual schedule, the choice of dates for re-validation and for first validations seems to be influenced by the expiration date of a particular ACVC version.

A new ACVC version is pre-released every six months, on the 10th of June and December. Each version is available for validation use from its seventh month to its expiration. Vendors must use a single version for both pre-validation and on-site testing. The data indicate most of the on-site testing to date has been done with an ACVC version during the last four months of its life cycle. Only three validations have been completed earlier. Thus far, no on-site testing has been done in January, February, or March; only one vendor tested in June/July; only three have tested in August.

Some vendors would like to see the ACVC cycle lengthened to 18 months. Presumably, they feel that there is insufficient time to use a particular version to develop a compiler. But one of the very arguments given for such an ACVC cycle extension---that the ACVC's interpretation of the Ada language has stabilized---is inconsistent with these vendors' claim. If the ACVC has become more stable, then it should be easier for implementers to adjust to each new release. And vendors are supposed to be implementing the Ada standard, not merely passing the ACVC tests.

4.7.1 Changes to the ACVC Life Cycle. Changes to the ACVC life cycle were discussed with AVF managers in November 1984.[AVF 1984] It was agreed, then, that the six-month, pre-release period was not necessary and that a three-month, pre-release period for field testing would provide sufficient time to discover and correct erroneous tests. The benefit derived from a shorter pre-release period is that the ACVC would be available for validation three months earlier, thus extending its period of use to nine months. This earlier availability of an ACVC version for validation would also result in two ACVC versions that could be used

TABLE 2
ACVC USE IN VALIDATION



for validations during one calendar year. Vendors have not been aware of this agreement because the AMO has continued to announce the "official" release of an ACVC version six months from its pre-release.[ICH 1985] Acknowledgement of this agreement appeared in the AMO News Letter, October 1985. However, earlier availability of the ACVC for validations was offered only as an option to the use of the "official" release cycle.[LCF 85]

4.7.2 Automation. The year-long life of a certificate necessitates annual validation for continued validation coverage. Such frequent testing is an expense that vendors have tried to minimize by the development of automated tools such as a test harness and test file scanner. Some of these tools have been provided to AVF managers on a trial basis and have been found effective in reducing the labor intensive analytical process. The original intent to produce automated tools for the ACVC has not been implemented by the AMO. The development of some test aids by the AMO could result in finding test errors before a new version of the ACVC is released. Other tools that are being routinely used by European validators for other languages could be adapted for use with Ada validation.

4.7.3 Possibility of Future Change to Re-validation Requirement. The problems of heavy cyclical workloads for AVF's and re-validation cost to vendors could be addressed favorably with a testing approach for re-validation that (1) reduces the amount of testing required to extend a certificate while also (2) providing incentive for use of the latest ACVC version. This approach would entail defining a re-validation testing method option which is to successfully pass all new ACVC tests and a subset of the of unchanged tests selected by the AVF for the vendor's pre-validation testing and another subset of the unchanged tests selected for on-site testing. Additionally, a certificate for re-validation could be issued for the number of days/weeks/months remaining in the ACVC version used to re-validate. The conditions that would apply are:

a. A re-validation must be performed with the successor ACVC version to the one used for the current certificate.

b. Re-validation may be performed no sooner than three months after validation.

c. Use of this test method will be limited to re-validation of the compiler on the same family of hardware/software initially validated.

4.7.4 Conclusions. Discussion of a new approach for re-validation is offered only for consideration at some future time. We have concluded that the two most important changes required to alleviate the cyclical AVF work load are (1) advertising the "official" release of the ACVC after a three-month pre-release period, and (2) developing automated tools for test analysis and documentation.

4.8 Tests Outside the Standard. The chairman of the LMC, Dr. John Goodenough, estimates that 800-1000 additional tests may be needed to

test the Ada language standard for those features that are manditory for conformity to the standard. An automated test generator has been developed as a European research project.[SEM 1985] However, the use of this tool has not been demonstrated as being adaptable or correct for the Ada language.

4.8.1 Research Activity. In Europe, work is beginning on the development of a test suite for Modula-2, based upon a formal definition of that language which will be used to generate a test suite with parameters added to automatically detect extensions/invalid inplementations of the language. However, testing for the effect produced by implementation dependent hardware features is thought to be beyond the state of the art even when the test suite has been based upon a formal definition of the language.[NPL 1985]

4.8.2 Conclusions. In assessing tests being developed outside the standard, the following conclusions were reached:

a. Although the development of an automatic test generator that accepts implementation dependent parameters is probably feasible, it is not practical to divert effort from the development of tests required to adequately cover the standard. See conclusion b for comments on a particular need. Additionally, the use of such a tool by AVF managers would change the current testing strategy of the test team from observing implementation dependent characteristics to detecting the presence of these characteristics.

b. Now is the time to begin to think about future ACVC tests that may be incorporated on a distributed system. Current validation procedures and tests do not address distributed targets (whether these distributed targets are the same as the host or not is irrelevant). Current thoughts on distribution of programs usually center around multiple processors which all have the same Instruction Set Architecture (ISA) and which usually communicate via shared memory. Distribution of programs within Local Area Networks (LAN's) may require communicating with similar or dissimilar ISA's and distribution may be done at link time at the compilation unit level. For example, in the Space Station program, it is currently envisioned that compilations will be done to the DIANA level. From that point, someone may invoke the linker to place a subprogram, package, etc., (basically a compilation unit) on a particular processor with a particular ISA. (Whether this "linker" actually performs the final code generation before the normal linking process is not pertinent to the discussion at hand). So, for example, at one link, SUBA may be executed on ISA-1, SUBB on ISA-2, etc. At another link, SUBA may be executed on ISA-2, SUBB on ISA-1, etc. The issues that may need to be addressed in the ACVC are as follows:

Can ACVC tests be generated in such a manner as so to not preclude the above from happening?

Can ACVC tests be generated that may have to check for proper Ada semantics over a distributed target?

Other questions may be also emerge as the subject of implementing Ada in a distributed environment is being addressed.

5.0 RECOMMENDATIONS

The following recommendations are actions that can be taken during calendar year 1986 to improve the overall efficiency and effectiveness of the Validation process. These recommendations are first listed, then discussed in greater detail.

5.1 Summary of Major Recommendations.

a. AVF Procedures. IDA Paper P-1900, Procedures for Ada Validation Facility Managers, should be approved as a documentation for consistent practices among AVF's. Supplementary recommendations are provided in Section 5.2.1.

b. Certificate. Only the configurations tested by the AVF should be enumerated on a certificate and wording on the certificate should be changed from "authorized" to "tested" to reflect the significance of these configurations. More recommendations are provided in Section 5.2.2

c. Testing Procedures. An agreement on testing procedures should be concluded between the AVF manager and vendor prior to completion of pre-validation and should be documented in the VSR. More recommendations are provided in Section 5.2.3

d. DoD Project Manager Procedures. The ECS working group should be the forum for developing a position on the merits of acquiring Ada compilers with capabilities limited to a specific MCCS application. Additional recommendations are provided in Section 5.2.4.

e. Test Resolution. The Fast Reaction Team should be maintained under its present mode of operation. An additional recommendation is provided in Section 5.2.5.

f. ACVC Ouality. ACVC configuration management and quality control practices should undergo a serious review; more recommendations regarding reporting requirements are stated in Section 5.2.6.

j. VSR Ouality. The VSR should undergo a serious review of its intent and content; more recommendations regarding the VSR are stated in Section 5.2.7.

k. Performance Information. The E&V Team should consider how compiler performance information will be made public.

l. AVF Workload. The workload of the AVF could be eased by release of the ACVC for nine months. Automation of the validation process should be a top priority in assisting the AVF. An additional recommendation is provided in Section 5.2.8.

m. Testing Outside the Standard. Continue to monitor European efforts but devote resources to research of tests for distributed systems and continued development of test that are required by the standard.

5.2 Additional Recommendations. The following recommendations are subordinate to the major recommendations as actions that are dependent upon implementation of a major recommendation.

5.2.1 Validation Procedures. The document developed by IDA for AVF managers should be reviewed annually to incorporate changes. Consideration should be given to development of supplementary procedures used only by the test team. AVF managers should meet on a regular basis to contribute to the review and preparation of these procedures.

5.2.2 Certificate. The recent practice, initiated by the AJPO, of identifying a "base configuration" on the certificate should not be continued. When the family architecture test method has been constructed properly (e.g., to demonstrate compiler conformity on all configurations), compiling and executing the full ACVC on one host/target pair does not mean that it is "more conforming" than other host/target pairs on which only the executable image of the test suite has been processed. Additionally, the vendor must demonstrate, with pre-validation test material, that the AVCV does not have to be compiled and executed on all host/target pairs. Unless the AVF is convinced, prior to on-site testing, that the risk is negligible, the full ACVC testing procedure should be used. The "base configuration" designation on a certificate can be interpreted to mean that the AJPO believes only that configuration is truly conforming and that the others have a dubious status. Indeed, certificates for COBOL and PASCAL do indicate ranges of conformity which should not be the case for Ada.

5.2.3 Testing Procedures. Use of the family architecture test method should be continued for vendors who offer cross compilers for multiple targets. The use of a test sample should be discontinued although use of the executable image created by a common host should be an option for testing multiple targets. There should be liaison between the AJPO, AVO, and AVF managers during the negotiation of an agreement with the vendor on the adaptation of this method to a specific case.

5.2.4 Embedded Targets. The ECS working group should be continued as an active sub-committee of an Ada Board Validation Committee. The AVO

and AVFs should develop case studies of embedded target validations for discussion with the ECS working group. This working group would also be a good forum for developing procurement policy recommendations for DoD Project Managers to discourage the acquisition of Ada compilers with capabilities limited to a specific application (e.g., a sub-set compiler). Participation of this group in the evolution of embedded target testing procedures will enrich the experience base of each AVF manager and can facilitate the acceptance of these procedures by industry and DoD project managers.

5.2.5 Fast Reaction Team. Some recognition from the Director of the AJPO to each member for the valuable service that has been performed is desirable. This recognition could be in the form of a certificate.

5.2.6 ACVC Configuration Information. The government Contract Technical Representative (COTR) for the contract to maintain and distribute the ACVC should distribute statistical reports for each version of the ACVC. The statistics that are useful for AVF managers and the AVO are:

a. Number of tests contained in the pre-release version

b. Number of tests found to be erroneous during the pre-release period

c. Number of erroneous tests corrected during the pre-release period

d. Final count of tests in the ACVC version when it can be used for validation

e. Enumeration of the new tests with identification of those from previous ACVC versions

f. An accounting of tests withdrawn, and the reason (i.e., error or LMC action required), throughout the release period

Additionally, a bi-annual schedule for correcting tests that have been withdrawn but not yet returned to the ACVC should be published. With this information available on a regular basis, AVF managers can respond better to vendor inquiries and requests for ACVC tape distribution. Moreover, the AVO and AJPO can use this information in performing VSR quality reviews and as ACVC management indicators.

5.2.7 Validation Summary Report. Further revision of the content of the VSR is required to reduce the repetitive presentation of information and to ensure that all pertinent testing information (including implementation dependent data) is included. The practice of concurrent reviews on the draft VSR should become normative. AVF managers should initiate procedures that ensure production of a draft VSR within 30 days after on-site testing has been completed. The draft review period should be two work weeks. When the final VSR has been submitted for signature, the AVO review should be limited to checking that all recommended corrections on the draft have been

made. The AVO signature (e.g., Director of the Computer and Software Engineering Division, IDA) should be evidence that this final VSR can be forwarded for signature by the Director, AJPO, and clearance by the DoD Information Office.

A certificate should be prepared after the AVO has signed the final VSR. The Director, AJPO, should forward the signed VSR to the AVF for presentation to the vendor. In most cases, the Director, AJPO, should sign the certificate as soon as it has been prepared and forward it to the AVF for presentation to the vendor. The VSR and certificate need not be sent at the same time. The exception to this practice may be encountered in the future when a validation has been conducted for a secure embedded system.

5.2.8 AVF Workload. A project should be organized as soon as possible to (1) develop requirements for automation of the validation process, and (2) to develop a system architecture for integrating existing tools and for developing and integrating additional tools. The development work should be assigned to individual AVF's, according to expertise and available resources. IDA should perform the system integration function.

REFERENCES

[ADA 1985] Ada Board Meeting, San Jose, CA, February 1985

[AFC 1985] Letter of 2 December 1985 from Department of the Air Force, Command and Control Systems Office (AFCC): "Ada Validation Summary Reports."

[ALS 1984] A. Hook, IDA memorandum for the record, ALS Planning Meeting of 14 November 1984

[ARP 1985] ARPANET message exchange among AVF managers and IDA

[AVF 1984] A. Hook, IDA memorandum for the record, AVF Managers Working Group meeting of 29 November 1985

[BOE 1985] Letter of 2 August 1985, Boeing Military Airplane Company and unpublished Softech paper

[ECS 1985] Embedded Computer Working Group,"Comments on AJPO Policy," October 1985

[HOO 1985] A. Hook, IDA memorandum, "Data General Validation Procedure," 1 February 1985

[Ich 1985] J. Ichbiah, ARPANET message of 25 January 1985 and Issue Paper distributed at SIGADA, February 1985

[IDA 1983] T. Probert, IDA memorandum to R. Mathis, 29 August 1983

[IDA 1984] T. Probert and R. Knapper, "Ada Implementor" letters

[IDA 1985] A. Hook and T. Probert, IDA Memorandum Report M-85, "Ada Compiler Validation Policy Workshop Proceedings and Recommendations," August 1985

[KNO 1985] P. Knoop, ARPANET message, 2 August 1985, "Withdrawn Tests, etc."

[KRA 1985] J. Kramer, IDA letter to P. Knoop, 6 December 1984

[LCF 1985] Language Control Facility Newsletter

[LEH 1985] D. Lehman, IDA memorandum for the record, "Westinghouse Validation Procedures," 5 September 1985

[LIE 1985] A. Hook, IDA letter to Dr. Edward Lieblein

[MGR 1985] CSED contact reports of meetings with AVF Managers

[M114 1985] IDA Memorandum Report M-114, September 1985: "Recommended Corrections for Verdix and Data General's Ada Compiler Validation Certificates," R. Danford Lehman

[NPL 1985] National Physical Laboratory, Private Working Papers

[RIC] G. Riccardi, "Comments on Validating an Embedded System Compiler," unpublished paper September 1985

[SEM 1984] SEMA Informatique, research grants

[SOF 1985] R. Quandrad, Softech, "Issues in the Validation of Ada Compilers for Chip Level Microprocessor Applications," unpublished paper September 1985

[TO2 1985] IDA Task Order No. MDA 903 84 C 0031: T-5-304 Amendment No. 2, 11 October 1985

[TP 1984] IDA Talking Paper for Ada Board Meeting of 26 November 1984

[VSR 1984] Validation Summary Reports 1984

[WIC 1985] B. Wichmann, ARPANET messages and private papers discussed with A. Hook, P. Knoop,and P. Hilfinger

[W-P 1984] Wright-Patterson Air Force Base, "Customer Support Agreement with TeleSoft," September 1984

APPENDIX A

STEPS IN THE VALIDATION PROCESS

STEPS IN THE CERTIFICATION PROCESS

There are ten steps in the certification process. Each step must be successfully completed for the implementor to offer a validated Ada compiler to general trade or DoD.

OBTAIN THE TEST SUITE (STEP ONE): The Ada Compiler Validation Capability (ACVC) is the only official test suite that can be used by a vendor and an AVF to conduct conformity testing; it is distributed only from AVF's. A vendor needs this test suite to compile and execute on his compiler before taking the next step in the validation process. This test suite will also be used to prepare a declaration of conformity (see Step 4).

NOTICE OF INTENT TO VALIDATE (STEP TWO): Although in-formal communication between a vendor and an AVF is encouraged, a vendor must notify the AVF in writing that he intends to become a client and desires to be scheduled for certification services. An AVF will schedule and commit resources on a first-come, first-served basis; therefore, a vendor should ask for AVF services as soon as it is possible to project when they will be needed. This written notification should include the following information:

- Target dates for submitting a declaration of conformity and for AVF on-site testing
- The ACVC version that is being used by the implementor to prepare a declaration of conformity
- The configuration(s) and compiler to be tested
- Where the on-site testing will be performed
- The point of contact for further information

When an implementor projects target dates for submitting a declaration of conformity and for starting on-site testing, he must consider the normative 90-day lead-time for an AVF to complete documentation to support a certificate. An AVF will make every effort to accommodate an vendor's desired schedule. In acknowledging receipt of the vendor's notification, the AVF will advise the vendor of known resource constraints that affect the desired schedule. When an AVF expects to have scheduling problems, the AVO should be consulted concerning possible re-distribution of workload among other AVFs.

NEGOTIATE A CUSTOMER SUPPORT AGREEMENT (STEP THREE): An AVF will require a formal agreement with a vendor, and may require payment in advance for the analytical and testing services that will be performed. Test report preparation is included in this fee. The AVF will set its fee based upon its cost-recovery scheme or other accounting method for determining costs. When

an implementor and AVF have executed a customer support agreement, the vendor becomes a client of the AVF and accepts the responsibility of compliance with the procedural practices of the AVF. A vendor who is in the process of self-testing may discover ACVC tests that incorrect or inapplicable for the compiler and configuration under test. As soon as the vendor becomes a client, the vendor should provide the AVF with a list of such tests. The sooner that each test's status can be determined, the better for the vendor.

DECLARATION OF CONFORMITY (STEP FOUR): The client shall run the ACVC, in some manner agreed upon with the AVF, upon the configurations to be certified. The results of this client self-testing shall be delivered to the AVF for analysis. This second written communication between the client and the AVF constitutes a request for resources to be available on a specific time schedule, while the first written communication to the AVF provides planning information (see Step 2).

RESOLVE TEST ISSUES (STEP FIVE): The AVF will analyze all test results thoroughly, including the list of tests considered by the client as being inapplicable for the configuration(s) under test. If the client and the AVF hold different views concerning inapplicable tests or the specific test method to be used (e.g., determination of the configurations required to test the base compiler, or the number of tests to be run on each configuration) and these issues cannot be resolved by the AVF, the issues will be referred to to the AVO for resolution. The AVO may resolve the issue with the help of the AVF and client and any other "outside" technical assistance considered advisable. The AVO decision is final, and will be binding on the AVF and the client. The client may renegotiate an appropriate course of action with the AVF as necessary to reschedule the validation or to amend or terminate the service agreement.

ON-SITE TESTING (STEP SIX): The AVF will conduct the conformity testing of the client's compiler at the location designated by the client. With the Letter of Intent, the client must provide the AVF with an estimate of the time required for testing; a final, refined estimate must be submitted two weeks before the AVF team is due to arrive at the client's site. The AVF test team will prepare, or request the AMO to prepare, the ACVC tapes or diskettes for the client's specific implementation. This customization of the ACVC for a client alters the ".TST" tests to make use of values that are implementation dependent. These customized ACVC media will exclude any test that has been withdrawn; they will have the same order as the ACVC media used by the client to prepare the declaration of conformity. The AVF test team will determine the order of test execution.

PREPARING THE REPORT (STEP SEVEN): The AVF will collect all test material generated during the conformity testing and will prepare a draft test report called a Validation Summary Report

(VSR). As a practical necessity, an AVF should use automated tools to prepare this report and should begin its preparation during Step 5 when the client's test materials are being analyzed. A draft test report should be completed as quickly as possible following on-site testing, and not longer than 30 calendar days after completing on-site testing.

REVIEW OF THE DRAFT REPORT (STEP EIGHT): The client, the AVO, and the AJPO will each recieve a copy of the draft VSR for concurrent review and comment. This review will be completed with comments provided to the AVF (including acknowledgment of "no comment") within two work weeks after receipt. The AVF will then complete a final report and recommendations for the certification body within two work weeks following receipt of comments from the client and AVO. The AVF may notify the client concerning anticipated pass/fail status prior to completing the final report, but not prior to review of the draft report by the AVO and the AJPO.

APPROVAL REVIEW (STEP NINE): The AVF manager will sign the final VSR and forward it to the AVO (or a designated representative) for final review and signature. This final review by the AVO is made merely to ensure that all amendments to the draft VSR recommended in Step 8 have been made. The AVO will forward the approved VSR to the Director, AJPO for signature. When the AVO, the AVF, and the client cannot agree on the contents of the VSR or on the AVF recommendations, the Director, AJPO will decide after hearing arguments.

ISSUE CERTIFICATE (STEP TEN): A certificate of conformity with ANSI/MIL-STD-1518A will be issued by the AJPO when the Director, AJPO has signed the VSR. The certificate will designate the configuration(s) and the compiler tested by the AVF and the ACVC version used for testing. This certificate will be in force for 12 months from the date of signature. The signed VSR and certificate will be delivered to the client by the AVF.

APPENDIX B

ECS WORKING GROUP PAPERS

MINUTES FOR EMBEDDED COMPUTER WORKING GROUP MEETING - 30 January 1985

The third meeting of the Embedded Computer Working Group was hosted by IDA on 30 January 1985. Enclosure (1) provides a list of attendees. The purpose of this meeting was to begin the development of proposals for change in the validation policy so that the needs of DoD contractors and project managers may be met. These needs include:

Synchronization of compiler revalidation and project baselining schedules so that compiler validation can be done when major changes occur to either the support software or the operational software.

Definitions of: the host/target computer environments that are subject to validation by the AJPO; and the project management responsibilities.

A method of validating a compiler on a computer system that is similar but not identical to the one for which a compiler certificate was issued.

Explicit procurement guidelines for use in RFPs that sets forth the responsiblities of all parties with respect to DoD policy for using Ada(TM) to develop and maintain mission critical application software.

2. Dudrey Smith summarized the proposals developed by the working group as follows:

The framework for developing policy statements for AJPO consideration addresses the following issues:

a. Compiler validation vs. military program schedules.

Military program schedules require baselining at certain points during the software life cycle. These points are generally when there have been significant changes in the support systems and/or in the application systems. It appears that there are two possible levels of certification — one at the AJPO level and one at the project management level. The AJPO certifies that the compilation system is in conformance with ANSI/MIL-STD-1815A while the project manager must certify to his/her management that the compilation system meets the application needs. DoD project managers are likely to make cost trade-offs between the cost for obtaining an AJPO certificate and the cost for ensuring the usability of the Ada(TM) compiler.

b. Precise designation of the host/target environment to which a certificate refers vs. a generic host/target environment.

During software development/maintenance, changes will occur in the computer configuration(s) used to validate an Ada(TM) compiler. These changes may be using a new version of the host/target operating system, replacement of the host computer, or designation of a target that was unknown at the time of validation testing, and changes in the host hardware configuration. After these changes have been made, the compilation system will be re-hosted as a compilation system that was derived from the one that was certified to continue software development/maintenance. If the certificate is void when any of these changes occur, a project manager will incur testing costs that ensure that useability of the compilation system; but (s)he is not likely to want to increase those costs by formal conformity testing.

c. Full ACVC testing of all combinations of host/target configurations for which a compiler is to be certified vs. validating a "class of compiler" that is suitable for a set of host/target combinations.

The logistics problems associated with assembling all the equipment configurations for ACVC testing can be prohibitive for implementors. Project managers will experience a similar problem which will add to project costs. Is it possible to direct validation testing procedures toward defining a "domain" of configurations for which the compilation system has been certified?

d. High level policy directive for the use of Ada(TM) vs. procurement policy and project management guidelines.

There is an absence of implementing instructions that tell procurement offices what they must consider when an RFP is prepared and that tells project managers how to adjust their plans to allow for Ada(TM) validation. Because of this void, the AJPO policy has been implemented with the certificate issued for a compilation system.

Dudrey Smith acknowledged that this framework for policy proposals to the AJPO is directed toward DoD contractors and project managers and asked if it is possible to create a set of Ada(TM) policies only for that group of users. Dr. Jack Kramer responded that he believes that large corporations will experience many of the same problems and that they will also need policies that address validation.

3. John Goodenough presented as the primary objectives for Ada(TM) validation the elimination of subset and superset implementations of the language. The purpose of validation is to check for conformance to the standard; in addition, there is some checking (which cannot be exhaustive) for incorrect semantics, for bugs, and for "cheating". The AJPO should be concerned mainly with the elimination of sub- or supersets of the Ada language. General discussion of why the precise designation of computer configurations is necessary on a certificate for a compilation system led to the view that useability of the

compiler on various hardware configurations within the same family is a project management concern with which an AJPO certification should not be concerned. The issue of a definition of "validated Ada", "legal Ada", and "Ada" was raised without a resolution being provided.

4. Paul Cohen presented the concept of validation for a virtual host/target configuration that represents the physical implementations of other host/target configurations for which a compilation has been tested and certified as conforming with the standard. The virtual host/target configuration can represent the "domain" within which compilation systems have been certified by the AJPO; the focus then becomes that of establishing a candidate system's membership in this domain vs. that of (ACVC-) testing each such system for conformance. Discussion continued toward defining the boundaries of a domain in the context of the conceptual model previously developed by the working group . The first point of general agreement was that a manufacturer's instruction-set architecture as implemented in a product line of host/target CPU's represents the initial generalized layer. Discussion concerning the designation of operating systems for the host/target configurations as the second generalized layer was inconclusive. It was proposed that the domain of applicability (for the compiler) can be any generic type of operating system that is compatible with the manufacturer's instruction-set architecture. The Validation Summary Report would provide the detailed information concerning the environment under which the ACVC testing was done. It was also proposed that the operating system for the target configuration should be specified as one that is functionally

compatible or functionally equivalent to a generic type. The concensus was that additional work is needed to define the domain of applicability for a certified compilation system. Members of the working group agreed to write their views of how the concept of validation for a virtual host/target configuration could be applied to the AJPO certificate and of the policy changes that may be needed. These comments from the working group are to be forwarded to Audrey Hook, IDA for consolidation.

4. Dudrey Smith introduced the problem of validating embedded computer systems that do not have all the physical components/ charateristics of larger computers. Audrey Hook discussed the precedent cases brought before the November ADA Board in which one implementor sought to eliminate all Chapter 14 tests and another desired to use a target simulator for validation testing. In the first case, the implementor was required to run all Chapter 14 tests; they have, subsequently, discovered compiler "bugs" in the I/O features that are needed. In the other case, the implementor will validate with the simulator since the DoD project manager is accountable for ensuring that the simulator is a true representation of the target configuration.

5. Paul Hilfinger stated that the work of this group will be discussed during the ADA Board Validation Working Group meeting in San Jose on 24 February 1985. The next meeting of the Embedded Computer Working Group was not scheduled.

ATTACHMENT 1: List of Attendees

name	organization	telephone
John S. SQUIRE	Westinghouse	(301) 765-3748
Austin J. MAHER	Singer-Kearfott	(201) 785-6607
Erwin BOOK	Hughes Aircraft Co.	(243) 647-0519
Fred SIX	Singer-Kearfott	(201) 785-7107
Major Al KOPP	AJPO	(202) 694-0209
Dennis HERGERT	TeleSoft	(619) 457-2700
Mike RYER	Intermetrics	(617) 661-1840
Donna GANT	General Dynamics	(314) 851-8991
Maretta HOLDEN	Boeing	(206) 241-3381
John B. GOODENOUGH	Softech	(617) 890-6900
Paul M. COHEN	AJPO	(202) 694-0212
Paul N. HILFINGER	U.C. Berkeley	(415) 642-8401
Pat KNOOP	W-P AFB, ASD/ADOL(AVF)	(513) 255-4472
Dudrey SMITH	Lear Siegler, Inc	(616) 241-7665
Audrey A. HOOK	I.D.A.	(703) 845-2316
Jack F. KRAMER	I.D.A.	(703) 845-2263
Clyde ROBY	I.D.A.	(703) 845-2541

Minutes from October 23, 1984 meeting of the Ada Validation Working Group-Embedded Systems.

1. A working group composed of representatives from the AdaJUG validation working group, the Ada Joint Program Office (AJPO), and the Institute for Defense Analysis (IDA), met on October 23, 1984 at IDA to discuss the technical and management issues inherent in compiler validation for embedded target computer systems. Attachment 1 provides a list of attendees.

2. The group reviewed the summary of issues discussed during the previous working group meeting on September 24, 1984. Discussion continuted on several issues relating to validation and re-validation of compilers during the extended period of software development for weapons systems. Dr. Jack Kramer, IDA provided guidance on several of these issues as follows:

a. Issue: Should optimized compilers be revalidated?
Answer: Yes, if an implementor plans to sell two versions of a compiler (high/low optimization). Also, it would be DOD policy to require it. The view is that each configuration that a compiler allows should be validated for conformity to the language specification as provided by ANSI/MIL-STD 1518A.

b. Issue: What are identical compilers?
Answer: Two or more compilers may be considered identical when the binary image of executable code of the compiler or of the object code is identical. Host and target configuration changes that effect the Ada compilation set should be defined so that it is possible to determine when conformity to the language specification must be tested/re-tested.

c. Issue: Is validation associated with the compiler or with the machine it is tested on?
Answer: Validation is the processes, used by the AJPO, to test the conformity to the language specification of a compilation system operating in a host environment to generate executable code for a target environment.

d. Issue: Is it legal to deliver contracted Ada software debugged/compiled by an Ada compiler, unchanged since its last validation, which has failed a new ACVC test during its required annual revalidation? (i.e. When does Ada code cease to be Ada?)

Answer: There are no procedures for validating delivered source code. Possibly, expansion of the NYU work in semantic definition could yield something useful in this area. Current validation procedures could provide that the compiler

must be validated prior to producing software to be delivered. A program/project manager should have the right to pick a baseline for an Ada compilation system that is consistent with schedules and cost for delivery of software. The compilation system should successfully pass ACVC testing at the time of the baseline. The software delivered for a DOD weapon system should conform to the Ada language specification.

3. Dr. Maretta Holden (Boeing) and Ms. Donna Gant (General Dynamics) raised discussion topics concerning the current policy requiring annual validation of Ada compilation systems. Software development schedules do not necessarily allow for the time and expense associated with this effort; and, language used in many procurements doesnot make it clear that a contractor will be obligated to conduct annual validations in order to maintain a current certificate. Discussion concerning the ambiguity of Request for Proposal (RFP) language concluded with a concensus that more explicit language should be developed for use in RFPs so that a bidder will understand the cost and schedule associated with Ada compilation systems. It was also suggested that performance information be available to aid in the selection of a compilation system with a high degree of useability for a particular application.

4. Mr. Dudrey C. Smith (Lear Siegler) presented a conceptual model of host/target/compilation systems for continued discussion of the criteria that may be developed to determine when a compilation system becomes one that is different from the one that was validated. Discussion of this model indicated that it may be a useful approach for supporting policy recommendations to the AJPO which would permit project life cycle baselining vs annual validation of the compilation system. The group concluded that additional work would be done on the conceptual model and that a presentation of the issues being addressed by the AdaJUG working group would be made during the AdaTech meetings in late November.

5. The next working group meeting was tentatively scheduled for the end of January 1985.

LIST OF ATTENDEES

Name	Organization	Phone
Michael Ryer	Intermetrics, Inc.	(617) 661-1840
Dudrey Smith	Lear Siegler, Inc.	(616) 241-7665
Erwin Book	Hughes Aircraft Co.	(213) 647-0519
Mike Kamrad	Honeywell	(612) 378-4432
Ed McCrohan	USA CENTACS	(201) 544-2685
Lt. Col. Dick Stanley	AJPO	(202) 694-0209
George Bryant	GSA	(703) 756-6153
Maretta Holden	Boeing	(206) 655-1251
Donna Gant	General Dynamics	(314) 851-8991
Jon Squire	Westinghouse	(301) 765-3748
Paul Cohen	DCA	(703) 437-3748
Audrey Hook	IDA	(703) 845-2316
Jack Kramer	IDA	(703) 845-2263
Bob Knapper	IDA	(703) 845-2516

Attachment 1

cc: Dennis Hergert

TeleSoft
10040 Bubb Road Street San Diego, California 92121
(619) 457 2700

November 20, 1984

Dr. Robert Mathis
Director
Ada Joint Program Office
Washington, D.C.

Dear Bob,

There are many unresolved issues in the validation world which have serious impact on the way TeleSoft will do its business in the future. I think it is desirable for the two of us to discuss them in the near future, prior to your policy decisions on them. I have summarized these issues for you here.

1. First, there is a multi-faceted question on hardware configurations supported by any given validation certificate. TeleSoft has a validation certificate for an 68000 O-BUS system running the ROS operating system. The O-BUS is a collection of MC68000 Q-BUS processors. Our customers also use the single board configuration of the Q-BUS processors. But, when we asked about the Q-BUS, we were told the validation would not be documented to cover it. This has serious impact on our current customers.

   There is also the issue of our future validations on hardware which comes from a long line of minimally different configurations of the same processor. A good example is the DEC VAX line. Must implementors validate on all the current and future members of that product line? This quickly becomes a financial and logistical burden for anybody except the hardware vendors themselves.

2. Second, we face the same difficulty with guidelines on which operating systems can be covered by a certificate. The UNIX operating system has many flavors and dialects. The implementor community must have some way of deciding which flavors of

any operating system are covered under a single validation. Our discussions with DEC tell us that their ULTRIX operating system is a Berkeley 4.2 UNIX. We have a validation in-progress on the Mt. Xinu Berkeley 4.2 UNIX running on a VAX 11/780. We would like to see the validation certificate cover the ULTRIX as well.

This also relates to upgrades to operating systems for which we already hold validations. If we validate on VAX/VMS 3.4, are we validated on VAX/VMS 3.5? 4.0?

3. Third, we need a clearly stated policy on the amount of modification that can be made to a compiler before its validation is threatened. We have made corrections to our compilers whenever problems were encountered. The AVO validation teams have even made suggestions to improve the ability of the compiler to conform to the standards. We make these corrections as soon as possible after each visit. We also expect to make further improvements to the compiler when possible. This will change the results of the validation tests but not fail them.

   Our attempt to grandfather the Intellimac IN/7000 system on our already existing Labtek/ROS validation is a good example of this. The results of the validation test-runs were questioned on the basis that the compiler had been changed and the results were slightly different. The differences were those suggested by the AVO Team at the conclusion of the Labtek validation. Must we keep our compilers unchanged throughout the year of validation before making any upgrades?

4. Fourth, we need guidance on the validation policies that will apply to the exponentially more complicated cross-compiler environment. Here the issues of slight variations in host and target hardware (or underlying system software) become even more difficult to deal with, particularly if a hard line is taken that every combination must be separately validated.

   In just a few months, TeleSoft expects to begin a formal cross-compiler validation. It is urgent

that policies governing that process be agreed upon.

It will be important to keep the cross-compiler validation issues in mind even when deciding on the simpler host validation process addressed in our earlier questions, since the two sets of policies should be as consistent as possible. A policy that seems reasonable when only host variations are being considered may be entirely impractical when BOTH host and target variations are involved.

I'd like to suggest that we schedule a meeting for sometime during SigAda next week. I'll call you in advance to arrange for a specific time.

Yours Truly,

Amnon Ben-Yehuda

Amnon Ben-Yehuda
President

DRAFT SUMMARY OF THE WORK OF THE Ada COMPILER VALIDATION WORKING GROUP FOR EMBEDDED SYSTEMS

Erwin Book

Hughes Aircraft Company
Radar Systems Group
P.O. Box 92426 MS: R8/4040
Los Angeles, CA 90009

February 12th, 1985

1.0 BACKGROUND

The Ada Compiler Validation Working Group (AVWG) for embedded systems consists of a group of people from industry that has been formed to advise the Ada Joint Program Office on policy concerning the validation of Ada compilers for embedded target systems. The participants are either from defense contractors or contractors that build Ada development tools, primarily Ada compilers.

The purpose of Ada compiler validation is to ensure that an Ada compiler processes Ada, not a subset or superset. Thus, it promotes the portability of Ada programs. The term validated Ada compiler is precisely defined. It means that it has passed 100% of the tests in the current or next Ada validation test suite. It says nothing about the usability, performance or applicability of that Ada compiler to any work being performed.

For the past two years the Ada user community as represented by SIGAda (formerly AdaTec), a committee of the Association for Computing Machinery, and AdaJUG, the Ada Jovial Users Group, has been bringing up problems concerning the current Ada compiler validation policy when it is applied to the use of Ada compilers in the world of DoD embedded computer target systems. With the tacit approval of AJPO, a committee has been formed to address this issue and advise them.

The members of the AVWG are:

1. Erwin Book, Hughes Aircraft Company
2. Robert Dewar, NYU Courant Institute
3. Dr. David Fisher, Gensoft
4. Donna Gant,General Dynamics
5. Dr. Maretta Holden, Boeing
6. Mike Kamerad, Honeywell
7. Austin Maher, Singer-Kearfott
8. Mike Ryer, Intermetrics
9. Dr. Dudrey Smith, Lear Siegler, Chair
10. John Squire,"Westinghouse"Electric"Corporation

Current Ada compiler validation policy is oriented toward commercial Ada compilers for well known ground based computers manuafactured by DEC, IBM, Data General and so forth.

Attendance at these meetings are open to all interested parties. The format of the meetings are always as follows. The AVWG group members

meet on the first day and discuss their views. The second day consists of a joint meeting at the Institute for Defense Analysis (IDA). Members of the DoD and their consultants join the committee at this meeting. The additional cast of characters (DoD group) vary from meeting to meeting. The fixed DoD group consist of Jack Kramer and Audrey Hook of IDA and AJPO representatives. Pat Knoop of the Ada Validation Office at WPAFB has also been regularly present.

2.0 ISSUES ADDRESSED BY THE AVWG

The principal difficulties facing the application of current policy to embedded systems are as follows

Present policy forces the revalidation of an Ada compiler each year, whether it needs it or not. Because DoD software for embedded systems must be produced by a validated Ada compiler, revalidation of a compiler and the subsequent remcompilation of the developing software of an embedded system can cause unpredictable cost and schedule slips during system development

It has been estimated that a program such as Advanced Tactical Fighter might use 150 different variations of a basic embedded computer such as the 1750A. Validating 150 similar Ada compilers is totally impractical. Some method must be found to reduce the combinatorial explosion of validating Ada compilers for similar computers. The amount of work and the expense of validation must also be reduced both for the Ada Validation Office and the contractors for variations of embedded computers, and for commercial Ada compilers as well.

While the AJPO is preserving the integrity of the language, the various Program Offices are still responsible for the development of embedded system software and hardware. Thus the determination of the usability of a particular Ada compiler at all stages of the software system life cycle remains the responsibility of the involved Program Office. Aid to the Program Office on their involvement with their various contractors on Ada compiler validation and useability should be made available from AJPO. Deliniation of the responsibilities between AJPO and the Program Offices must be made clear.

The status of an Ada compiler that is being maintained between validations should be clarified.

The validation of Ada compilers for small memory embedded computers present unique and difficult problems. They usually do not have normal IO devices such as printers and therefore do not have sufficient environment to run the Ada Validation test suite.

3 0 SUMMARY OF THE WORK PERFORMED BY AVWG

This working group has met 3 times and has presented the results of the first 2 meetings at the SigAda AdaJUG general meeting in Washington D.C. in December. The work done to that point met with unoffical approval of the AJPO and the apparent approval of the audience.

3.1 PERIOD OF VALIDATION FOR EMBEDDED COMPUTER ADA COMPILERS

The use of Ada for mission critical embedded system development and maintenance is intended to assure that operational software has been written in Ada as defined by MIL-STD 1815A and that the delivered code has been produced by a validated Ada compiler. Often the support software must also be delivered, this can include the Ada compiler itself. It is intended that the Ada compiler, if delivered, be validated. The delivery of all software is currently covered by the baseline policy. There is no reason to deviate from this policy because of the use of Ada. In fact a uniform policy in this regard is desireable. This however requires a different policy from that used to validate commercial Ada compilers.

Baselining of software is done in preparation for delivery of systems. This can be done once or several times during the development of the system at points that are mutually agreed upon by the Program Office and the contractor. It is at least done prior to formal testing. If an Ada compiler is used for the development of the system then it will be (re)validated each time it is baselined. This is done at the mutually agreed upon time. The current rules apply to Ada compilers and the code they produce. Therefore it is possible that Ada compilers for embedded systems may retain their validation status for more than 1 year.

3.2 COVERAGE OF THE ADA VALIDATION CERTIFICATE AN ADA

compiler validation certificate should be associated with the bit representation of the executable image of the Ada Compilation system, not the agent that interprets those bits. Simulators, emulators, interpreters, and microprogrammed and hardwired machines are also interpreting agents. If a validated Ada compiler hosted on a VAX is run on an IBM mainframe it obviously won't work. But the validation status of the bit representation of that Ada compiler doesn't change; it is still validated. The question is what is included in the Ada compilation system that has been validated and to what does its validation status extend.

The certificate now names the exact environment that was involved during the running of the Validation tests. An Ada compiler system includes the Ada compiler proper and its run time support kernel. It also

includes the Ada linker/loader, the Ada program library management functions.

If any of the above items change the code produced may be affected and revalidation at some point is indicated. Portions of the host environment and target environment may also change. Which of these changes should cause revalidation due to the conditions on the Certificate becoming inapplicable is the question to be addressed.

The host environment factors are the hardware, ISA, operating system, file system, linker, loader, and the compilation system (when the compiler is not totally written in Ada).

The target environment factors are the hardware, ISA, operating system, linker, and loader.

The Ada validation certificate describes the domain of environments for which the tested Ada compiler is validated. A detailed description of the exact environment leads to an enormous number of tests for machines with the same salient characteristics, such as an Instruction Set Architecture (ISA).

The proposed policy is named validation by similarity and is as follows. The validation certificate stipulates that the compiler processes MIL-STD 1815A and only 1815A. The certificate should specify that the Ada compiler is valid for a host/target pair of ISAs. It should assume that the Ada compiler is valid for any operating system on the host. It should be valid for a specific operating system and version of the target or any other operating system declared by its manufacturer to be functionally equivalent. The certificate names the manufacturer of the Ada compiler. The certificate does not promise that the compiler is error free or that it will run in your environment.

An accompanying report describes the precise details of the environment used to run the test suite under which the compiler was validated, namely the host/target ISA, hardware, operating system, and so forth. The key point being the certificate includes a broader class of environments than the one used in the test. However the accompanying report will be used to determine whether the particular compiler would be of any use to a particular program. The implications of the above are that an Ada compiler can be validated for an instruction set simulator of an ISA which could be later be used for hardware that implements the same target ISA.

3.3 RESPONSIBILITY OF AJPO AND THE PROGRAM OFFICES

The AVO validation process is intended to determine conformance of a compiler system implementation to the Ada language standard. Determination of the acceptability and usability of a compiler system implementation is a procurement issue. Validation is a necessary but not sufficient condition. The procurement activity may require that the compiler system be revalidated after each major change during the

development cycle. The exact compiler system used to produce formally delivered operational software should be validated after its last change. Subsequent upgrade of the operational software requires use of a currently validated compiler (potential compiler upgrade).

The responsibility of the particular program office that will use the compiler (the customer) is to ensure that the compiler is usable, validated, and the software used in their system has been produced by a validated Ada compiler. Preparation for the delivery of mission critical software, in which Ada's use is proscribed, requires baselining of the operational and often the support software, and the Ada run time library.

3.4 THE STATUS OF DERIVED ADA COMPILERS

It is undesirable that Ada validation policy should inhibit the maintenance of validated Ada compilers. At present it does not, however the consequences of the use of newer versions of a validated Ada compiler on an embedded system development are not precisely defined.

The term derived Ada compiler is coined to describe an "improved" version of a validated compiler. It is proposed that a derived compiler can be used with the code produced remaining valid if the following procedure is followed. The vendor certifies that he has run the ACVC tests and the derived compiler has passed those tests. This implies that the vendor has described quality control procedures for his Ada compiler.

This permits the use of the derived compiler during system development until a new baseline is established. At that point, the derived compiler will be revalidated.

3.5 ADA VALIDATION FOR SMALL EMBEDDED SYSTEMS

The nature of embedded mission critical systems requires the use of MIL-STD computers in various configurations and with unusual devices. The concept of validation testing as implemented up to now demands the availability of printers for providing a human readable form of the test outputs. Many computers used for embedded systems do not have printers or even displays. Validation testing for such computers can use larger sized versions of the computers that have sufficient environments and attempt to validate by similarity. Of course, the code that is actually written to run in the very small embedded computer would not make use of text I/O, or any other features of the language not suited to the function of that computer in the overall system.

Minutes from October 23, 1984 meeting of the Ada Validation Working Group-Embedded Systems.

1. A working group composed of representatives from the AdaJUG validation working group, the Ada Joint Program Office (AJPO), and the Institute for Defense Analysis (IDA), met on October 23, 1984 at IDA to discuss the technical and management issues inherent in compiler validation for embedded target computer systems. Attachment 1 provides a list of attendees.

2. The group reviewed the summary of issues discussed during the previous working group meeting on September 24, 1984. Discussion continuted on several issues relating to validation and re-validation of compilers during the extended period of software development for weapons systems. Dr. Jack Kramer, IDA provided guidance on several of these issues as follows:

a. Issue: Should optimized compilers be revalidated?
Answer: Yes, if an implementor plans to sell two versions of a compiler (high/low optimization). Also, it would be DOD policy to require it. The view is that each configuration that a compiler allows should be validated for conformity to the language specification as provided by ANSI/MIL-STD 1518A.

b. Issue: What are identical compilers?
Answer: Two or more compilers may be considered identical when the binary image of executable code of the compiler or of the object code is identical. Host and target configuration changes that effect the Ada compilation set should be defined so that it is possible to determine when conformity to the language specification must be tested/re-tested.

c. Issue: Is validation associated with the compiler or with the machine it is tested on?
Answer: Validation is the processes, used by the AJPO, to test the conformity to the language specification of a compilation system operating in a host environment to generate executable code for a target environment.

d. Issue: Is it legal to deliver contracted Ada software debugged/compiled by an Ada compiler, unchanged since its last validation, which has failed a new ACVC test during its required annual revalidation? (i.e. When does Ada code cease to be Ada?)

Answer: There are no procedures for validating delivered source code. Possibly, expansion of the NYU work in semantic definition could yield something useful in this area. Current validation procedures could provide that the compiler

must be validated prior to producing software to he delivered. A program/project manager should have the right to pick a baseline for an Ada compilation system that is consistent with schedules and cost for delivery of software. The compilation system should successfully pass ACVC testing at the time of the baseline. The software delivered for a DOD weapon system should conform to the Ada language specification.

3. Dr. Maretta Holden (Boeing) and Ms. Donna Gant (General Dynamics) raised discussion topics concerning the current policy requiring annual validation of Ada compilation systems. Software development schedules do not necessarily allow for the time and expense associated with this effort; and, language used in many procurements doesnot make it clear that a contractor will be obligated to conduct annual validations in order to maintain a current certificate. Discussion concerning the ambiguity of Request for Proposal (RFP) language concluded with a concensus that more explicit language should be developed for use in RFPs so that a bidder will understand the cost and schedule associated with Ada compilation systems. It was also suggested that performance information be available to aid in the selection of a compilation system with a high degree of useability for a particular application.

4. Mr. Dudrey C. Smith (Lear Siegler) presented a conceptual model of host/target/compilation systems for continued discussion of the criteria that may be developed to determine when a compilation system becomes one that is different from the one that was validated. Discussion of this model indicated that it may be a useful approach for supporting policy recommendations to the AJPO which would permit project life cycle baselining vs annual validation of the compilation system. The group concluded that additional work would be done on the conceptual model and that a presentation of the issues being addressed by the AdaJUG working group would be made during the AdaTech meetings in late November.

5. The next working group meeting was tentatively scheduled for the end of January 1985.

LIST OF ATTENDEES

Name	Organization	Phone
Michael Ryer	Intermetrics, Inc.	(617) 661-1840
Dudrey Smith	Lear Siegler, Inc.	(616) 241-7665
Erwin Book	Hughes Aircraft Co.	(213) 647-0519
Mike Kamrad	Honeywell	(612) 378-4432
Ed McCrohan	USA CENTACS	(201) 544-2685
Lt. Col. Dick Stanley	AJPO	(202) 694-0209
George Bryant	GSA	(703) 756-6153
Maretta Holden	Boeing	(206) 655-1251
Donna Gant	General Dynamics	(314) 851-8991
Jon Squire	Westinghouse	(301) 765-3748
Paul Cohen	DCA	(703) 437-3748
Audrey Hook	IDA	(703) 845-2316
Jack Kramer	IDA	(703) 845-2263
Bob Knapper	IDA	(703) 845-2516

Attachment 1

APPENDIX C

FAST REACTION TEAM DELIBERATION

Message 99 2419 9 Aug 85
From: DLEHMAN@USC-ECLB
To: dlehman@USC-ECLB
Cc: ahook@USC-ECLB
Subject: DISPUTED TESTS--ACVC 1.6

C64103A

Line 66 value of LARGE is not big enough to cause Intel exception "floating overflow".

C93005A

The expression 1/ident_int(0) is of a dead assignment to I, which is never used; therefore, the only possible effect of the division is to propogate a pre-defined exception, and therefore, the division need not be evaluated, NUMERIC_ERROR need not be raised, not TASKING_ERROR propagated. [C93005B & --C were withdrawn for the above reason; odd that --A wasn't previously disputed?]

CA1011A0, --A1, --A2, --A3, --A4, --A5, --A6M

The test objective should be reversed to be consistent with AI-00199.

BA2001EOM, --E1, --E2

The LRM (10.2-5) states that the "simple names of all subunits that have the same ancestor library unit must be distinct identifiers." The test embodied in FA2001E* expects that the above condition be checked at the point of the declaration of the stub. The vendor's implementation, however, detects a duplicate subunit name under a single ancetor library unit when the subunit itself is being compiled. No program library will contain duplicate subunits since the second of the subunits will be rejected. The wording of the LRM does use the term "subunit" rather than "stub", though, admittedly, these two concepts are tightly related.

BC3009A, --B, --D

The implementation detects circular generic instantiation only on an instantiation of a "real" entity--i.e., an instantiation outside of a generic entity. Since no subprogram or package is instantiated outside of a generic unit in these tests, the circularity is not detected. In essence, since generics are treated as templates, only a "real" instantiation actually brings a copy into being; circularity withinn a template is tolerated though no instantiation of this template will be legal.

Message 101 601 9 Aug 85
From: dewar@NYU-ACF2
To: DLEHMAN@USC-ECLB, FAST-REACTION@USC-ECLB
Subject: Re: FAST-REACTION NOTICE

this set of objections requires careful looking at, whether or not all these USE_ERRORS are acceptable depends on the environment of the implementation (see AIR-325).

Message 104 2616 15 Aug 85
From: Ron Brender ‹BRENDER at DEC-MARLBORO.ARPA›
To: DLEHMAN at USC-ECLB, FAST-REACTION at USC-ECLB
Subject: Re: FAST-REACTION NOTICE
In-Reply-To: ‹[USC-ECLB] 9-Aug-85 15:16:55.DLEHMAN›
Regarding: Message from DLEHMAN@USC-ECLB of 9-Aug-85 1816-EDT

Dan,

Sorry, but I think I sent you an empty reply earlier due to some clumsy fingers. Here are some reactions...

Ron

C64103A

I don't understand the implementor claim about overflow. The test is checking whether LARGE exceeds the range of type SM_FLOAT. In order to get to line 66, it must be true that SM_FLOAT'BASE'LARGE is less than LG_FLOAT'BASE'LARGE (checked in line 64) so it should follow that LARGE (equal to LG_FLOAT'LARGE) is outside the range of SM_FLOAT, ie. greater than SM_FLOAT'LARGE.

I would require a more detailed justification than just this criptic hint! On the surface, the implementor seems to be wrong.

C93005A

The implementor is correct. (Oh, the vagories of optimizers!)

CA1011A*

Previously withdrawn.

BA2001E*

I vaguely recall this objection being raised a year or so ago, but I don't recall the outcome. The objection is questionable because it requires entering units into the library with stubs having the same name. The mere presence of these stubs in the library IMPLIES the existance of corresponding subunits.

The implementor may have a legalistic argument based on the wording of the LRM, but not a clear one. Perhaps consider "not applicable", but send to the LMC to get a clear resolution once and for all.

BC3009A, B, C [your message listed D, but that must be a typo]

This one has been raised before. The LMC considered the issue in AI-00328 in February and decided that illegalities must be detected independent of any instantiation. While currently approved only by the LMC (it goes to the Ada Board/WG9 in November), the implementor is wrong.

CE* [all the Chapter 14 objections in the second message]

Apparently the implementor wants to validate for a target that does not support any file I/O. In that case, all of these tests can be considered "not applicable".

I presume that all of the other tests do report "Not applicable". It also appears that these tests should be cleaned up a bit to handle USE_ERROR more smoothly.

Message 107 3429 16 Aug 85
From: hilfingr%ucbrenoir@Berkeley (Paul Hilfinger)
To: DLEHMAN@USC-ECLB
Cc: FAST-REACTION@USC-ECLB.ARPA, DEWAR@NYU.ARPA, BRENDER@DEC-MARLBORO.ARPA,
GOODENOUGH@USC-ISI.ARPA, HILFINGER@Berkeley,
PLOEDEREDER@TL-20B.ARPA,
KRAMER@USC-ECLB.ARPA, AHOOK@USC-ECLB.ARPA, CROBY@USC-ECLB.ARPA,
KNAPPER@USC-ECLB.ARPA, BABCOCK@USC-ECLB.ARPA,
KEVIN.PHILLIPS%RSRE@UCL-CS.ARPA
Subject: Re: [DLEHMAN@USC-ECLB: DISPUTED TESTS--ACVC 1.6]
In-Reply-To: Your message of 9 Aug 1985 13:15-PDT

> C64103A
>
> Line 66 value of LARGE is not big enough to cause Intel exception
> "floating overflow".

The test certainly seems wrong, but I don't understand the implementor's objection. Perhaps he means that even though SM_FLOAT'BASE'LARGE < LG_FLOAT'BASE'LARGE, LG_FLOAT'LARGE is still in SM_FLOAT'BASE. This is quite possible, and the test is therefore incorrect.

> C93005A
> The expression 1/ident_int(0) is of a dead assignment to I, which
> is never used; therefore, the only possible effect of the division is to
> propogate a pre-defined exception, and therefore, the division need not be
> evaluated, NUMERIC_ERROR need not be raised, not TASKING_ERROR propagated.
> [C93005B & --C were withdrawn for the above reason; odd that --A wasn't pre-
> viously disputed?]

The implementor seems to be correct.

> CA1011A0, --A1, --A2, --A3, --A4, --A5, --A6M

› The test objective should be reversed to be consistent with AI-00199.

The implementor seems to be correct.

› BA2001EOM, --E1, --E2

› The LRM (10.2-5) states that the "simple names of all subunits that have
› the same ancestor library unit must be distinct identifiers." The test
› embodied in BA2001E* expects that the above condition be checked at the
› point of the declaration of the stub. The vendor's implementation,
› however, detects a duplicate subunit name under a single ancetor library
› unit when the subunit itself is being compiled. No program library will
› contain duplicate subunits since the second of the subunits will be
› rejected. The wording of the LRM does use the term "subunit" rather than
› "stub", though, admittedly, these two concepts are tightly related.

We've discussed this somewhere. My position is that the implementor is correct.

› BC3009A, --B, --D
›
› The implementation detects circular generic instantiation only on an
› instantiation of a "real" entity--i.e., an instantiation outside of a
› generic entity. Since no subprogram or package is instantiated outside
› of a generic unit in these tests, the circularity is not detected. In
› essence, since generics are treated as templates, only a "real"
› instantiation actually brings a copy into being; circularity withinn a
› template is tolerated though no instantiation of this template will be
› legal.

We've discussed this, with the result that the implementor must be considered wrong, even though one could argue that the Standard is unclear.

Paul N. Hilfinger

Message 116 1047 21 Aug 85
From: John B. Goodenough ‹GOODENOUGH@USC-ISI.ARPA›
To: DLEHMAN@USC-ECLB.ARPA
Cc: FAST-REACTION@USC-ECLB.ARPA, DEWAR@NYU.ARPA, BRENDER@DEC-MARLBORO.ARPA,
GOODENOUGH@USC-ISI.ARPA, HILFINGER@UCB-VAX.ARPA,
PLOEDEREDER@TL-20B.ARPA,
KRAMER@USC-ECLB.ARPA, AHOOK@USC-ECLB.ARPA, CROBY@USC-ECLB.ARPA,
KNAPPER@USC-ECLB.ARPA, BABCOCK@USC-ECLB.ARPA,
KEVIN.PHILLIPS%RSRE@CS.UCL.AC.UK
Subject: Re: FAST-REACTION NOTICE
In-Reply-To: ‹[USC-ECLB] 9-Aug-85 13:15:38.DLEHMAN›

I agree with Paul's response to these tests. With respect to BC3009A, B,

and D, this issue has been discussed by the LMC in AI-00328 with the committee-approved position that the legality of a generic unit does not depend on whether or not it is ever instantiated. It seems reasonable to me that the AVO should accept the LMC position, even though it has not yet been reviewed and approved by the ADA Board.

JBG

Message 100 2110 9 Aug 85
From: DLEHMAN@USC-ECLB
To: dlehman@USC-ECLB
Subject: n

IMPLEMENTER DISPUTES:

CE2102D, --E

1) These tests incorrectly report failure when STATUS_ERROR results from attempts to RESET unopened files.

CE2103A, --B, CE3107A

2) These tests terminate when an unhandled STATUS_ERROR results from an attempt to CLOSE an unopened file.

CE2104A, --B, CE2111A, --B, --C, CE2404A, CE2405B, CE2406A, CE2408A, CE2410A
CE3108A, --B, EE3102C

3) These tests terminate when an unhandled USE_ERROR results from an attempt to CREATE a file.

CE2107A, CE3114B

4) These tests incorrectly report failure when STATUS_ERROR results from attempts to DELETE unopened files.

CE2110A, CE2201C, CE2202A, CE3114A, CE3115A

5) These tests incorrectly report failure when USE_ERROR results from attempts to CREATE files.

CE2201B, CE3305A, CE3603A, CE3706F

6) These tests incorrectly report failure when their explicitly raised INCOMPLETES are handled as OTHERS.

CE2401A, --B, --C, --D, --E, CE2402A, CE2409A

7) These tests are illegal because POSITIVE_COUNT'LAST is 1.

CE2401F

**) This test is illegal for reasons 7, 5, and 1, above.

CE3102B

8) This test reports failure when USE_ERROR always results from attempts to CREATE files.

CE3112B

9) This test incorrectly reports failure when USE_ERROR results from an attempt to OPEN a file.

==========

Message 104 2616 15 Aug 85
From: Ron Brender ‹BRENDER at DEC-MARLBORO.ARPA›
To: DLEHMAN at USC-ECLB, FAST-REACTION at USC-ECLB
Subject: Re: FAST-REACTION NOTICE
In-Reply-To: ‹[USC-ECLB] 9-Aug-85 15:16:55.DLEHMAN›
Regarding: Message from DLEHMAN@USC-ECLB of 9-Aug-85 1816-EDT

Dan,
    Sorry, but I think I sent you an empty reply earlier due to some clumsy fingers. Here are some reactions...
    Ron

C64103A

    I don't understand the implementor claim about overflow. The test is checking whether LARGE exceeds the range of type SM_FLOAT. In order to get to line 66, it must be true that SM_FLOAT'BASE'LARGE is less than LG_FLOAT'BASE'LARGE (checked in line 64) so it should follow that LARGE (equal to LG_FLOAT'LARGE) is outside the range of SM_FLOAT, ie. greater than SM_FLOAT'LARGE.
    I would require a more detailed justification than just this criptic hint! On the surface, the implementor seems to be wrong.

C93005A

    The implementor is correct. (Oh, the vagories of optimizers!)

CA1011A*

    Previously withdrawn.

BA2001E*

I vaguely recall this objection being raised a year or so ago, but I don't recall the outcome. The objection is questionable because it requires entering units into the library with stubs having the same name. The mere presence of these stubs in the library IMPLIES the existance of corresponding subunits.

The implementor may have a legalistic argument based on the wording of the LRM, but not a clear one. Perhaps consider "not applicable", but send to the LMC to get a clear resolution once and for all.

BC3009A, B, C [your message listed D, but that must be a typo]

This one has been raised before. The LMC considered the issue in AI-00328 in February and decided that illegalities must be detected independent of any instantiation. While currently approved only by the LMC (it goes to the Ada Board/WG9 in November), the implementor is wrong.

CE* [all the Chapter 14 objections in the second message]

Apparently the implementor wants to validate for a target that does not support any file I/O. In that case, all of these tests can be considered "not applicable".

I presume that all of the other tests do report "Not applicable". It also appears that these tests should be cleaned up a bit to handle USE_ERROR more smoothly.

Message 106 2597 16 Aug 85
From: hilfingr%ucbrenoir@Berkeley (Paul Hilfinger)
Subject: FRT disputes

> CE2102D, --E
>
> 1) These tests incorrectly report failure when STATUS_ERROR results
> from attempts to RESET unopened files.

> CE2103A, --B, CE3107A
>
> 2) These tests terminate when an unhandled STATUS_ERROR results from
> an attempt to CLOSE an unopened file.

> CE2107A, CE3114B
>
> 4) These tests incorrectly report failure when STATUS_ERROR results
> from attempts to DELETE unopened files.

> CE2110A, CE2201C, CE2202A, CE3114A, CE3115A
>
> 5) These tests incorrectly report failure when USE_ERROR results from
> attempts to CREATE files.

I assume the preceding attempts to OPEN failed? If so, the implementor is

correct.

> CE2104A, --B, CE2111A, --B, --C, CE2404A, CE2405B, CE2406A, CE2408A, CE2410A
> CE3108A, --B, EE3102C
>
> 3) These tests terminate when an unhandled USE_ERROR results from an
> attempt to CREATE a file.

This has come up before, hasn't it? I seem to recall that we decided to allow USE_ERROR here, although the STANDARD does say "in the absence of NAME_ERROR."

> CE2201B, CE3305A, CE3603A, CE3706F
>
> 6) These tests incorrectly report failure when their explicitly raised
> INCOMPLETES are handled as OTHERS.

The implementor appears to be right.

> CE2401A, --B, --C, --D, --E, CE2402A, CE2409A
>
> 7) These tests are illegal because POSITIVE_COUNT'LAST is 1.

The implementor appears to be right.

> CE2401F
>
> **) This test is illegal for reasons 7, 5, and 1, above.

The implementor appears to be right.

> CE3102B
>
> 8) This test reports failure when USE_ERROR always results from
> attempts to CREATE files.

> CE3112B
>
> 9) This test incorrectly reports failure when USE_ERROR results from
> an attempt to OPEN a file.

See above (ce2104a et al.).

Message 117 1278 21 Aug 85
From: John B. Goodenough ‹GOODENOUGH@USC-ISI.ARPA›
Subject: Re: FAST-REACTION NOTICE
In-Reply-To: ‹[USC-ECLB] 9-Aug-85 15:16:55.DLEHMAN›

All of these problems arise apparently because the implementation does not support file I/O. These tests should be considered inapplicable rather

than withdrawn. The problems listed should, of course, be corrected, and most of these problems have been corrected in Release 1.7 (the only tests that are currently not corrected in 1.7 are CE3114B, CE3115B, CE2401E, and CE2401F).

The issue of whether USE_ERROR can be raised when no file I/O is supported is addressed in AI-00332, which concludes that USE_ERROR is allowed (as well as NAME_ERROR). This is just a committee approved decision, which only serves to say that the vendor is okay in raising USE_ERROR.

JBG

Message 145 3929 29 Aug 85
From: DLEHMAN@USC-ECLB
Subject: WICHMANN'S REMARKS ON FRN OF 85-08-09
[DLEHMAN@USC-ECLB: WICHMANN'S BELATED REMARKS]

FRT Members:

The following message is proof that Brian is alive and well. As resolution of the subject tests will not be returned to the AVF until Monday, there is some time for comment on Brian's remarks.

The test C64103A, whose argument from the implementer was unclear, has since been dropped from contention--the implementer intends to pass the test.

I presume that the safest way to ensure that Brian receives NET notes is to use both the BWICHMANN@USC-ECLB as well as the KEVIN.PHILLPS%RSRE@UCL-CS addresses.

---Dan LEHMAN
----------*

Begin forwarded message
Received: By USC-ECLB via direct-append with Hermes; 29 Aug 85 11:43:35-PDT
Date: 29 Aug 1985 11:43-PDT
From: DLEHMAN@USC-ECLB
To: DLEHMAN@USC-ECLB
Subject: WICHMANN'S BELATED REMARKS
Message-ID: ‹[USC-ECLB]29-Aug-85 11:43:34.DLEHMAN›
Sender: DLEHMAN@USC-ECLB

FRT Members:

I've received a belated message from Brian WICHMANN today (29 August) re the Fast-Reaction Notice(s) of 09 August. I shall transcribe his telex (the NET connection for him was down).

---Dan LEHMAN

Dear Fast Reaction Team Member[s]: [some text interpretations are needed-DL]

I am now on ARPANET (again), but please minimise the messages, as the communication channel is not too convenient.
Brian Wichmann BWICHMANN at ECLB

Comments on two lists of disputed tests dated 9 August 1985 (from Dan LEHMAN).

I have inspected a sample of these tests since time does not allow more. My conclusions from these tests are as follows:

CE2102D
I agree with the challenge. All calls of the I/O routines need to be carefully checked for exception handling. Are there tools to aid this?

CE2401B [et al.]
I disagree with the challenge. To state that POSITIVE_COUNT'LAST=1 implies that COUNT'LAST=1 (see 14.2.5/1). This in turn implies that DIRECT_IO is implemented only in the syntactic sense. I do not see how one can permit [that mere syntactic implementation (Brian's actual text reads 'permitlmxrpn', which I take as so much garbage in place of "that" or "this")]. After all, the ACVC does not allow INTEGER'LAST=1. We should ask Pat KNOOP to ensure that the value of COUNT'LAST appears in VSRs. Is there an ACVC test that prints out the value of COUNT'LAST?

I conclude b[y] agreeing with Paul HILFINGER except on CE2401A (and related tests), noted above.

From: B. A. WICHMANN
DITC
National Physical Laboratory
Teddington TW11 OLW, UK
ext.: 3976

End forwarded message

Message 146 1354 29 Aug 85
From: John B. Goodenough ‹GOODENOUGH@USC-ISI.ARPA›
Subject: Re: WICHMANN'S REMARKS ON FRN OF 85-08-09
In-Reply-To: ‹[USC-ECLB]29-Aug-85 12:14:27.DLEHMAN›

Re CE2401B

COUNT'LAST=1 presumably because the implementation does not, in fact, support

DIRECT_IO because there are no file I/O capabilities possible for the target machine. In such a case, it is acceptable for COUNT'LAST to equal 1.

Message 145 3929 29 Aug 85
From: DLEHMAN@USC-ECLB
To: FAST-REACTION@USC-ECLB, KNAPPER@USC-ECLB, AHOOK@USC-ECLB, CROBY@USC-ECLB,
    DEWAR@NYU, HILFINGER@UCB-VAX, PLOEDEREDER@TL-20B,
    KEVIN.PHILLIPS%RSRE@UCL-CS, GOODENOUGH@USC-ISI, BWICHMANN@USC-ECLB,
    BRENDER@DEC-MARLBORO, DLEHMAN@USC-ECLB
Subject: WICHMANN'S REMARKS ON FRN OF 85-08-09
         [DLEHMAN@USC-ECLB: WICHMANN'S BELATED REMARKS]

FRT Members:

The following message is proof that Brian is alive and well. As resolution of the subject tests will not be returned to the AVF until Monday, there is some time for comment on Brian's remarks.

The test C64103A, whose argument from the implementer was unclear, has since been dropped from contention--the implementer intends to pass the test.

I presume that the safest way to ensure that Brian receives NET notes is to use both the BWICHMANN@USC-ECLB as well as the KEVIN. PHILLPS%RSRE@UCL-CS addresses.

---Dan LEHMAN
--------------*

Begin forwarded message
Received: By USC-ECLB via direct-append with Hermes; 29 Aug 85 11:43:35-PDT
Date: 29 Aug 1985 11:43-PDT
From: DLEHMAN@USC-ECLB
To: DLEHMAN@USC-ECLB
Subject: WICHMANN'S BELATED REMARKS
Message-ID: ‹[USC-ECLB]29-Aug-85 11:43:34.DLEHMAN›
Sender: DLEHMAN@USC-ECLB

FRT Members:

I've received a belated message from Brian WICHMANN today (29 August) re the Fast-Reaction Notice(s) of 09 August. I shall transcribe his telex (the NET connection for him was down).

---Dan LEHMAN

Dear Fast Reaction Team Member[s]: [some text interpretations are needed-DL]

I am now on ARPANET (again), but please minimise the messages, as the communication channel is not too convenient.
Brian Wichmann BWICHMANN at ECLB

Comments on two lists of disputed tests dated 9 August 1985 (from Dan LEHMAN).

I have inspected a sample of these tests since time does not allow more. My conclusions from these tests are as follows:

CE2102D
I agree with the challenge. All calls of the I/O routines need to be carefully checked for exception handling. Are there tools to aid this?

CE2401B [et al.]
I disagree with the challenge. To state that POSITIVE_COUNT'LAST=1 implies that COUNT'LAST=1 (see 14.2.5/1). This in turn implies that DIRECT_IO is implemented only in the syntactic sense. I do not see how one can permit [that mere syntactic implementation (Brian's actual text reads 'permitlmxrpn', which I take as so much garbage in place of "that" or "this")]. After all, the ACVC does not allow INTEGER'LAST=1. We should ask Pat KNOOP to ensure that the value of COUNT'LAST appears in VSRs. Is there an ACVC test that prints out the value of COUNT'LAST?

C64103A
I agree with the challenge to version 1.6. However, the problem has been avoided in the revised coding used in version 1.7.

C93005A
The challenge applies that a dead assignment can be omitted by a compiler. I do not believe that this can be deduced from LRM 11.6. For instance, since the code of IDENT_INT is not visible to the compiler, there is nothing to stop IDENT_INT [from] raising any exception, perhaps one not visible to the main test. Hence, I think that one can argue that the test is valid. However, the coding is clearly questionable and should be rewritten to avoid the problem (which is easy to do).

I conclude b[y] agreeing with Paul HILFINGER except on CE2401A (and related tests), noted above.

From: B. A. WICHMANN
DITC
National Physical Laboratory
Teddington TW11 OLW, UK
ext.: 3976

End forwarded message

Message 146 1354 29 Aug 85
From: John B. Goodenough ‹GOODENOUGH@USC-ISI.ARPA›
To: DLEHMAN@USC-ECLB.ARPA
Cc: FAST-REACTION@USC-ECLB.ARPA, KNAPPER@USC-ECLB.ARPA, AHOOK@USC-ECLB.ARPA, CROBY@USC-ECLB.ARPA, DEWAR@NYU.ARPA, HILFINGER@UCB-VAX.ARPA,

PLOEDEREDER@TL-20B.ARPA, KEVIN.PHILLIPS%RSRE@UCL-CS.ARPA,
GOODENOUGH@USC-ISI.ARPA, BWICHMANN@USC-ECLB.ARPA,
BRENDER@DEC-MARLBORO.ARPA
Subject: Re: WICHMANN'S REMARKS ON FRN OF 85-08-09
In-Reply-To: ‹[USC-ECLB]29-Aug-85 12:14:27.DLEHMAN›

Re CE2401B

COUNT'LAST=1 presumably because the implementation does not, in fact, support DIRECT_IO because there are no file I/O capabilities possible for the target machine. In such a case, it is acceptable for COUNT'LAST to equal 1.

Re C93005A

Brian's analysis is correct as far as it goes -- the call to IDENT_INT cannot be eliminated unless the compiler knows it won't raise an exception or have any other side effects. Since the call actually doesn't raise exceptions or have side-effects, it is impossible to tell whether it is being made or not. It is clear that 11.6 allows the divide operation to be omitted, so no exception need be raised. In short, the test is indeed incorrect for an optimizing compiler.

›

[The following message (viz., 179) was forwarded to all FRT members.]

Message 179 2060 9 Sep 85
From: DLEHMAN@USC-ECLB.ARPA
To: FAST-REACTION@USC-ECLB, KNAPPER@USC-ECLB, AHOOK@USC-ECLB,
GOODENOUGH@USC-ISI, HILFINGER@Berkeley, DEWAR@NYU,
BRENDER@DEC-MARLBORO, PLOEDEREDER@TL-20B, BWICHMANN@USC-ECLB,
KEVIN.PHILLIPS%RSRE@UCL-CS, CROBY@USC-ECLB

Subject: FRN 85-09-09

FRN 85-09-09

FRT Members:

The following disputes have been raised:

---- For all tests below, the implementer argues:

These tests assume that an external file created by one I/Opackage will exist as far as another I/O package is concerned. The test tries to determine the circumstances under which the second I/O package may be used t access the external file.

In our implementation, each I/O package has associated with it a set of external files distinct from the set of extenal files associated with any other I/O package. The appropriate error when attempting to associate a file of the second package's FILE_TYPE is NAME_ERROR, since no file by that name exists, as far as the first package is concerned.

---- The disputed tests, with arguements particular to each, are given below:

CE2107B-B,
--D-B: [---- no particular arguements]

--E-B: TEMP_HAS_NAME is set FALSE if SEQ.NAME call fails; it should be set TRUE if the call succeeds. [---- for it's tested after the 6th BEGIN]

CE2108B-B,
--D-B: The DELETEs of FILE_NAME and NAMES_FILE will be executed even if the corresponding OPENs fail. These DELETEs should be moved so they will be executed only if the corresponding OPEN succeeds.

CE3112B-B: LRM 14.1/7 states that the language does not define what happens to external files after the completion of the main program. This test should be declared INAPPLICABLE if the external file written in the previous test cannot be opened. The test should only try to delete that file if the file was able to be opened.

----------end

---Dan

------- *

Message 180 1717 9 Sep 85
From: dewar@NYU-ACF2
To: AHOOK@USC-ECLB.ARPA, DLEHMAN@USC-ECLB.ARPA, FAST-REACTION@USC-ECLB.ARPA,
        KNAPPER@USC-ECLB.ARPA
Subject: Re: FRN 85-09-09

It seems to me that all these protests are subject to AI-325 (unreasonable restrictions). In the absence of further justification, I feel that these cases go beyond the AI-325 limit, i.e. they are unacceptable limitations corresponding to implementing I/O in a barely adequate manner. In particular, although of course what happens to external files after completion of the main program is undefined, an implementation which takes advantage of this to make it impossible to pass a file from one program to another cannot be considered acceptable.

On the other hand, note the phrase "in the absence of further justification" in the above discussion. If for example, this implementation had no real file storage and was simulating files in memory as a result, the described limitations are not only acceptable, but it would be clear that the implementor had gone well beyond the minimum required of an implementation in such an environment.

All questions regarding I/O tests of this nature should be phrased in terms of justifications with respect to AI-325. In other words the implementor who restricts the implementation must provide a written justification of the restrictions, following the guidelines of AI-325.

Message 183 2958 10 Sep 85
Fr . hilfingr%ucbrenoir@Berkeley (Paul Hilfinger)
To DLEHMAN@USC-ECLB
Re: [DLEHMAN@USC-ECLB.ARPA: FRN 85-09-09]
In-Reply-To: Your message of 9 Sep 1985 15:10-PDT

CE2107E-B:
> TEMP_HAS_NAME is set FALSE if SEQ.NAME call fails; it should
>   be set TRUE if the call succeeds.

Correct. Tch, tch.

> CE2108B-B,
>   --D-B: The DELETEs of FILE_NAME and NAMES_FILE will be executed even if
>..   the corresponding OPENs fail. These DELETEs should be moved so
>   they will be executed only if the corresponding OPEN succeeds.

Correct. Didn't we handle one like this just a while ago?

> CE3112B-B:
> LRM 14.1/7 states that the language does not define what happens
>   to external files after the completion of the main program. This test
>   should be declared INAPPLICABLE if the external file written

> in the previous test cannot be opened. The test should only try to
> delete that file if the file was able to be opened.

Correct, technically. However, this begins to make me uncomfortable. WHY is this implementor attempting to validate a completely useless system? Certainly a file system in which one effectively cannot create permanent external files might just as well not be implemented at all. This almost counts as a ''325 error'' worthy of being stifled, but not quite, due to the rather strong statement quoted from the Standard.

> In our implementation, each I/O package has associated with it a
> set of external files distinct from the set of extenal files
> associated with any other I/O package. The appropriate error when
> attempting to associate a file of the second package's FILE_TYPE
> is NAME_ERROR, since no file by that name exists, as far as the
> first package is concerned.

> CE2107B-B,
> --D-B: [---- no particular arguements]

Here, I draw the line. 14.1(1) says ''An external file is identified by a string (the name).'' The clear intent is that there be one file name space. The implementation described seems utterly ridiculous, so I am not inclined to bend over backwards re-interpreting the rules to make it work. It looks as if the implementor just doesn't want to support files; why didn't he just raise USE_ERROR cn everything?

Paul Hilfinger

Message 184 1619 10 Sep 85
From: John B. Goodenough ‹GOODENOUGH@USC-ISI.ARPA›
Subject: Re: FRN 85-09-09
In-Reply-To: ‹[USC-ECLB.ARPA] 9-Sep-85 15:10:41.DLEHMAN›

CE2107B-B
CE2107D-B

I certainly agree with Paul here. There certainly seems to be no reason to reject these tests based on "distinct sets of external files associated with each I/O package". I see no basis for such an interpretation in the RM.

CE3112B-B

The statement in RM 14.1/7 exists to cover the fact that after completion of a main program, someone could delete a file intentionally by operating system commands. It is certainly not the intended meaning that all external files disappear after the main program completes, else why distinguish temporary files (for which this is true) from other external files?

Now there might be some implementation dependent reason why no external file can be preserved after completion of any main program, but the implementer will have to provide more justification than has currently been provided.

CE2108B,D

I'll get back to you on these.

CE2107E-B

This test is clearly incorrect (TEMP_HAS_NAME needs to be given an initial value).

Message 188 2846 11 Sep 85
From: Ron Brender ‹BRENDER at DEC-MARLBORO.ARPA›
To: DLEHMAN at USC-ECLB.ARPA, FAST-REACTION at USC-ECLB.ARPA,
    KNAPPER at USC-ECLB.ARPA, AHOOK at USC-ECLB.ARPA
Subject: Re: FRN 85-09-09
In-Reply-To: ‹[USC-ECLB.ARPA] 9-Sep-85 15:10:41.DLEHMAN›
Regarding: Message from DLEHMAN@USC-ECLB.ARPA of 9-Sep-85 1810-EDT

FRT members:

C34002B
    I agree 100% with John on this one. AI-0002 HAS been approved as John notes and the only reason that a revised discussion has not yet been forwarded to the Ada Board/WG9 is to make sure that the discussion given can be coordinated with that to be provided for AI-00330. While the outcome regarding AI-00330 is very much in doubt, there has not been even a suggestion that the recommendation of AI-00002 should reconsidered.

CE2107B,D
    This implementation is totally unwarranted. Actually, I don't see why the implementor should even be motivated to do this. The LRM does not specify what behavior is required when a file created by sequential I/O is read by direct I/O or vice-versa. It would be legitimate to raise USE_ERROR if the implementation "tags" the file so that it knows its kind ("organization"). In the absence of tags, probably anything goes while attempting to read the file (including raising most any exception or even crashing when calling GET). But the file should be found by the OPEN.

    Hmmm, I seem to be arguing that this test is actually erroneous (but for reasons unrelated to the implementor claim). Perhaps it ought to be reclassified as an E test?

CE2107E
    The test is wrong (in agreement with Paul and John).

CE2108B,D
    The tests are wrong (and among those that need to be cleaned up).

CE2401D
    Raising NUMERIC_ERROR here is definitely questionable. While the LMC did not resolve this issue at the Sept meeting, my crystal ball says the LMC will NOT ratify such action. Since the item is, however, still officially

open before the LMC, I suppose it is appropriate to treat the test as not applicable for this implementation at this time, as John suggests.

CE3112B
This claim is silly on the face of it, and certainly more justification is needed if it is to be taken seriously. I agree with John on this one (and think Paul even concedes too much). Seems like the implementer would be better off allowing only unnamed files (and raising USE_ERROR when a non-null NAME string is specified), or even not claiming to support I/O at all (as Paul suggests).

-Ron

Message 189 1179 12 Sep 85
From: John B. Goodenough ‹GOODENOUGH@USC-ISI.ARPA›
To: DLEHMAN@USC-ECLB.ARPA
Subject: Re: FRN 85-09-09 CE2108B, D
In-Reply-To: ‹[USC-ECLB.ARPA] 9-Sep-85 15:10:41.DLEHMAN›

CE2108B, D

Well I may be missing something obvious, but I don't understand the implementer's objection. The OPEN for FILE_NAME and NAMES_FILE are enclosed in a block with a handler for USE_ERROR and OTHERS. Both handlers then raise INCOMPLETE, which exits to the end of the test, executing no DELETE calls on the way. If the innermost OPEN succeeds, it is true that the DELETE calls are executed, but this means the test failed in any case. Without further explanation from the implementer, I would have to say that this test seems to be correct.

Message 190 929 12 Sep 85
From: hilfingr%ucbrenoir@Berkeley (Paul Hilfinger)
To: GOODENOUGH@USC-ISI
Subject: Re: FRN 85-09-09 CE2108B, D
In-Reply-To: Your message of 12 Sep 1985 21:44:46EDT

John seems to be correct re: CE2108B & D. It is difficult to understand what prompted the implementor to comment.

Paul

Message 194 1362 13 Sep 85
From: KNAPPER@USC-ECLB.ARPA
Subject: Re: FRN 85-09-09...
In-Reply-To: ‹[USC-ECLB.ARPA] 9-Sep-85 15:10:41.DLEHMAN›

CE2107B.D-B

I agree with Paul and John here. I see no reason at all why an I/O package should have a distinct set of external files associated with it and no other.

CE3112B-B

I agree with John. The intent is to allow permanent files to be created by editors, compilers or any other tool written in Ada. The implementer's interpretation is too liberal unless there is an implementation dependent reason.

CE2108B-D

This test appears correct to me. The DELETEs can only be executed if the OPENs succeed.

CE2107E-B

This test is incorrect as stated.

CE2401D

I agree with everyone else, but if the test is to be revised I recommend that it be withdrawn.

Bob K

Message 199 1056 16 Sep 85
From: DLEHMAN@USC-ECLB.ARPA
To: FAST-REACTION@USC-ECLB.ARPA, KNAPPER@USC-ECLB.ARPA, AHOOK@USC-ECLB.ARPA, goodenou%wang-inst.csnet@CSNET-RELAY.ARPA, GOODENOUGH@USC-ISI.ARPA, HILFINGER@UCB-VAX.ARPA, BRENDER@DEC-MARLBORO.ARPA, DEWAR@NYU.ARPA, BWICHMANN@USC-ECLB.ARPA, PLOEDEREDER@TL-20B.ARPA, CROBY@USC-ECLB.ARPA, KEVIN.PHILLIPS%RSRE@UCL-CS.ARPA
Cc: DLEHMAN@USC-ECLB.ARPA
Subject: MORE RE FRN 85-09-09

FRT Members:

Indeed the implementation behind the FRN 85-09-09 CE* disputes is not ordinary: the targets are 1750A chips; the implementer has simulated a dissk in memory. I am sorry that I did not include this information with the issuance
of the disputes. To what extent are your responses changed? d(Obviously, those tests clearly wrong or right(execution of DELETEs, e.g.)are not affected.)

---Dan
------- *

Message 205 1176 16 Sep 85
From: dewar@NYU-ACF2
To: AHOOK@USC-ECLB.ARPA, DLEHMAN@USC-ECLB.ARPA, FAST-REACTION@USC-ECLB.ARPA,
        KNAPPER@USC-ECLB.ARPA
Subject: Re: MORE RE FRN 85-09-09

It is perfectly fine to loose all "permanent" files between jobs under these circumstances, in fact it is hard to see what else could be done (short of not implementing I/O at all). Implementing I/O using this approach is actually an instance of an implementor going much further than the minimum necessary in an attempt to fully implement the spec.

The use of two separate address spaces for sequential and random files is more peculiar, and not easily justified. On the other hand, this is a special case and I am implied to be liberal.

(couldn't the implementor have separated the name spaces of the two kind of files, thus not permitting a file to be used in both ways?)

Message 208 1064 17 Sep 85
From: KNAPPER@USC-ECLB.ARPA
Subject: Re: MORE RE FRN 85-09-09
In-Reply-To: ‹[USC-ECLB.ARPA]16-Sep-85 14:28:46.DLEHMAN›

Using the main store for a simulated disk does cause a problem. Since I can only assume that the goal for 1750A chips is not to serve as word processors or any other thing needing to keep "permanent" disk files around, I have to agree with my liberal friend that allowing the files to disappear after the completion of the main program is ok.

Bob K

Message 220 1265 17 Sep 85
From: hilfingr%ucbrenoir@Berkeley (Paul Hilfinger)
Subject: Re: MORE RE FRN 85-09-09
In-Reply-To: Your message of 16 Sep 1985 14:28-PDT

I don't think any of my responses change. The implementor's request that CE3112B be declared inapplicable looks OK (I am more comfortable with it given your note). I also see no reason, even given what you've said, for their problem with CE2107B/D. For my information, are they simulating a disk just for the sake of validation, or do they think users will want it? If the former, why did they bother?

Paul

Message 223 1409 18 Sep 85
From: John B. Goodenough ‹GOODENOUGH@USC-ISI.ARPA›

Subject: Re: MORE RE FRN 85-09-09
In-Reply-To: ‹[USC-ECLB.ARPA]16-Sep-85 14:28:46.DLEHMAN›

It seems to me that if an implementation is not going to preserve external files between executions of a main program, the implementation ought to raise USE_ERROR for any attempt to create a named external file, thereby forcing all file I/O to be done using temporary files, which are not preserved across main program invocations.

On the other hand, I suppose that if the main program executes for weeks at a time, it might be considered both useful and acceptable to have named external files that can be opened and closed during the period of the main program's execution, even though they won't be preserved if the main program should ever complete its execution, but I feel a bit uncomfortable about allowing such behavior.

JBG

Date: 19 Sep 1985 1427-EDT
From: Ron Brender ‹BRENDER at DEC-MARLBORO.ARPA›
Subject: Two FRT items

FRT members:

1) Re the 1750a with the simulated disk in memory...

   Grumble... I guess I would have to concede that it is an acceptable implementation restriction for the external files to disappear when the main program completes, although I'm not completely happy with the idea.

   I sure hope the implementer went to all of the trouble to simulate files in memory because it was perceived as important to the application domain, rather than in order to pass validation. The latter means that the AVO is not getting the right message out and ought to be a matter of concern.

2) Re BC3220B

   I think the implementer is wrong. While type SET is declared within the generic in terms of a generic formal type, in the instantiation the semantics of that type is explained in terms of a "copy" in which the occurence of the formal type name is understood to DENOTE the actual subtype ENUM, which is definitely static. Moreover, the rule regarding the use of others is interpreted in AI-00310 in terms of the staticness of the corresponding index constraint (that is, on a constraint by constraint basis rather than on the staticness of the array subtype as a whole).

Ron

Message 93 1169 26 Sep 85
From: DLEHMAN@USC-ECLB.ARPA
To: FSTC-AVF@USC-ECLB.ARPA
Cc: AHOOK@USC-ECLB.ARPA, DLEHMAN@USC-ECLB.ARPA
Subject: WESTINGHOUSE RESOLUTIONS

John STANTON:

The following disputes ahve been resolved for Westinghouse:

1) CE2107E-B is withdrawn from ACVC 1.6 for the reason given by the vendor.

2) CE2108B-B & CE2108D-B are ruled correct. The vendor's argument is NOT a correct statement of the test logic: an exception raised by either of the subject OPENs will cause INCOMPLETE to be raised in the exception handler for those OPENs; INCOMPLETE will cause control to skip the DELETEs and to go to the exception handler at the end of the test.

3) CE3112B-B is allowed to be inapplicable for this implementation.
4) AE2101A is allowed to be split in half (10 instantiations each) for this implementation.

5) CE2107B- & D-B are an unresolved issue--more on this later.

---Dan
------- *

Message 97 655 3 Oct 85
From: DLEHMAN@USC-ECLB.ARPA
To: FSTC-AVF@USC-ECLB.ARPA
Cc: DLEHMAN@USC-ECLB.ARPA, AHOOK@USC-ECLB.ARPA
Subject: WESTINGHOUSE CE2107B-B, --D-B

John STANTON:

These tests are considered correct; Westinghouse's implementation justification wherein "each I/O package has associated with it a set of external files distinct from the set of external files associated with any other I/O package" is considered unacceptable.

---Dan
------- *

Message 86 1670 21 Aug 85
From: DLEHMAN@USC-ECLB
To: FAST-REACTION@USC-ECLB, KNAPPER@USC-ECLB, AHOOK@USC-ECLB,
BRENDER@DEC-MARLBORO, HILFINGER@UCB-VAX,
GOODENOUGH@USC-ISI, PLOEDEREDER@TL-20B, DEWAR@NYU
Cc: DLEHMAN@USC-ECLB
Subject: FAST-REACTION NOTICE

Folks:

We have received another list of disputes; they are:

C37011A

This test checks that sliding does NOT occur in record component initializations. We could not find any reason that the semantics of record component initialization should be different from that of an assignment. The LRM 3.7/5 says that the default expression must be of the type (not of the subtype) of the component. This test should be withdrawn.

[Well, it was, and then it was reinserted due to an LMC resolution (into 1.5). --allowing sliding, I presumed, since the old (1.4) disputes read "checks that sliding OCCURS ..."? Now, in 1.6 (presumably) it is ... ?]

Constraining an incomplete or private type (AI-00007)

Test E38104A, which contains discriminant constraints on private types before the full type declaration, can be rejected by the compiler. Consequently, an implementation should also be allowed to reject the following tests:

B38105B--line 29,
B74207A--lines 49 & 50,
BC3503A--line 22,
C48006B--line 133.

----Dan LEHMAN
----------*

Message 88 2284 21 Aug 85
From: hilfingr%ucbrenoir@Berkeley (Paul Hilfinger)
To: DLEHMAN@USC-ECLB
Cc: FAST-REACTION@USC-ECLB, KNAPPER@USC-ECLB, AHOOK@USC-ECLB,
BRENDER@DEC-MARLBORO, HILFINGER@Berkeley, GOODENOUGH@USC-ISI,
PLOEDEREDER@TL-20B, DEWAR@NYU
Subject: Re: FAST-REACTION NOTICE
In-Reply-To: Your message of 21 Aug 1985 11:57-PDT

> C37011A

> This test checks that sliding does NOT occur in record component
> initializations. We could not find any reason that the semantics of
> record component initialization should be different from that of an
> assignment. The LRM 3.7/5 says that the default expression must be of
> the type (not of the subtype) of the component. This test should be
> withdrawn.

See 3.2.1(16): "The initialization of an object (the declared object or one of its subcomponents) checks that the initial value belongs to the subtype of the object; for an array object DECLARED BY AN OBJECT DECLARATION, an implicit subtype conversion is first applied as for an assignment statement."

The language here is such as to indicate that sliding does not occur for subcomponents. I agree with the implementor that the restriction is not technically justifiable, but it's what the manual says.

> Constraining an incomplete or private type (AI-00007)

> Test E38104A, which contains discriminant constraints on private types
> before the full type declaration, can be rejected by the compiler.
> Conse- quently, an implementation should also be allowed to reject the
> following tests: B38105B--line 29, B74207A--lines 49 & 50, BC3503A--line
> 22, C48006B--line 133.

I haven't checked all of these, but they should certainly conform to the E-test. This comment clearly indicates that the LMC at least ought to rule whether this case is legal (as apart from where and when the checks must be performed.)

Paul

Message 118 988 30 Aug 85
From: DLEHMAN@USC-ECLB
To: FAST-REACTION@USC-ECLB, KNAPPER@USC-ECLB, AHOOK@USC-ECLB, CROBY@USC-ECLB, GOODENOUGH@USC-ISI, KEVIN.PHILLIPS%RSRE@UCL-CS, BWICHMANN@USC-ECLB, DEWAR@NYU, PLOEDEREDER@TL-20B, HILFINGER@UCB-VAX, BRENDER@DEC-MARLBORO
Cc: DLEHMAN@USC-ECLB
Subject: FRN 85-08-21 REMINDER

FRT Members:

So far, the FRN of 21 August (re tests C37011A & E38104A et al.) has received comment only from Paul (21 August). Yet the latest FRN (22 Aug) has received comments from three people. I want to ensure that the subject notice has not been overlooked in the (admittedly) bunch of confusingly unlabelled notices that have recently been issued (future notices shall be better labelled) or awaited Members upon returns from absence.

---Dan LEHMAN
-------------- *

Message 119 452 30 Aug 85
From: John B. Goodenough ‹GOODENOUGH@USC-ISI.ARPA›
To: DLEHMAN@USC-ECLB.ARPA
Cc: GOODENOUGH@USC-ISI.ARPA
Subject: Re: FRN 85-08-21 REMINDER
In-Reply-To: ‹[USC-ECLB]30-Aug-85 15:21:27.DLEHMAN›

I also responded to this FRN, which did not actually seem to dispute the E38104A test, but rather B38105B etc.

Message 122 1243 1 Sep 85
From: John B. Goodenough ‹GOODENOUGH@USC-ISI.ARPA›
ReSent-From: John B. Goodenough ‹GOODENOUGH@USC-ISI.ARPA› 1 Sep 1985 20
ReSent-To: dlehman@USC-ECLB.ARPA
To: DLEHMAN@USC-ECLB.ARPA
Cc: FAST-REACTION@USC-ECLB.ARPA, KNAPPER@USC-ECLB.ARPA, AHOOK@USC-ECLB.ARPA,
    BRENDER@DEC-MARLBORO.ARPA, HILFINGER@UCB-VAX.ARPA,
GOODENOUGH@USC-ISI.ARPA,
    PLOEDEREDER@TL-20B.ARPA, DEWAR@NYU.ARPA
Subject: Re: FAST-REACTION NOTICE
In-Reply-To: ‹[USC-ECLB]21-Aug-85 11:57:23.DLEHMAN›

C37011A I concur with Paul's response; sliding is not allowed and the test is correct.

I've checked the other tests (B38105B, B74207A, BC3503A, C48006B) and they all contain constraints imposed on incompletely declared types. The recommended interpretation in AI-00007 does not require that the cases contained in these tests be rejected as illegal, but since AI-00007 is non-binding (at present), an implementation is allowed to reject the cited lines. The tests should be revised to remove the questionable cases and should be withdrawn.

JBG

Message 123 950 3 Sep 85
From: KNAPPER@USC-ECLB
To: DLEHMAN@USC-ECLB
Cc: FAST-REACTION@USC-ECLB, KNAPPER@USC-ECLB,
    AHOOK@USC-ECLB, CROBY@USC-ECLB, GOODENOUGH@USC-ISI,
    KEVIN.PHILLIPS%RSRE@UCL-CS, BWICHMANN@USC-ECLB,
    DEWAR@NYU, PLOEDEREDER@TL-20B, HILFINGER@UCB-VAX,
    BRENDER@DEC-MARLBORO
Subject: Re: FRN 85-08-21 REMINDER
In-Reply-To: ‹[USC-ECLB]30-Aug-85 15:21:27.DLEHMAN›

I agree w/Paul on C37011A. The test is correct.

On E38104A we have a situation where the implementer wishes to have 4 more tests disputed (B38105B, B74207A, BC3503A and C48006B) if the E test is determined to be correct. If the LMC could take a look at the problem in its 4 Sept meeting we can have a clearer picture as to what to withdraw or not withdraw.

Bob K

Message 164 880 16 Sep 85
From: DLEHMAN@USC-ECLB.ARPA
To: FAST-REACTION@USC-ECLB.ARPA, KNAPPER@USC-ECLB.ARPA, AHOOK@USC-ECLB.ARPA,
goodenou%wang-inst.csnet@CSNET-RELAY.ARPA, GOODENOUGH@USC-ISI.ARPA,
HILFINGER@UCB-VAX.ARPA, PLOEDEREDER@TL-20B.ARPA, DEWAR@NYU.ARPA,
BRENDER@DEC-MARLBORO.ARPA, BWICHMANN@USC-ECLB.ARPA,
KEVIN.PHILLIPS%RSRE@UCL-CS.ARPA,
CROBY@USC-ECLB.ARPA
Cc: DLEHMAN@USC-ECLB.ARPA
Subject: RE FRN of 21 AUGUST

John, et al.:

Did the LMC resolve the issue of whether or not a compiler may reject constraints on discriminated types before the full type declaration? --affecting E38104A-related tests B38105B, B74207A, BC3503A, & C48006B.

---Dan
------- *

Message 165 1066 16 Sep 85
From: DLEHMAN@USC-ECLB.ARPA
To: HHUMMEL@USC-ECLB.ARPA, Clausen.IABG@MIT-MULTICS.ARPA
Cc: AHOOK@USC-ECLB.ARPA, DLEHMAN@USC-ECLB.ARPA
Subject: DISPUTED TESTS

Dear Helmut, Stephan,

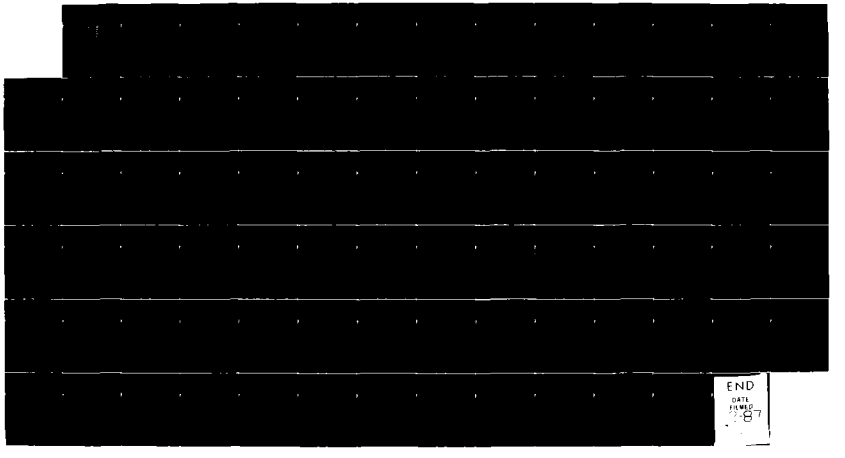
In response to your 19 August 1985 submission of disputed tests, the Fast-Reaction Team has ruled that C37011A IS CORRECT--sliding of array bounds during component initializations is NOT permitted. (One person admitted that there is no TECHNICAL justification for this rule--perhaps when the language is revised in 1988 ... !?)

The question as to whether or not a compiler may reject constraints on discriminated types was left unresolved pending the 4 Sept. Language Maintenance
Committee meeting. I have just sent a message to John GOODENOUGH et al. to see what the LMC decided. I shall return that decision sometime this week.

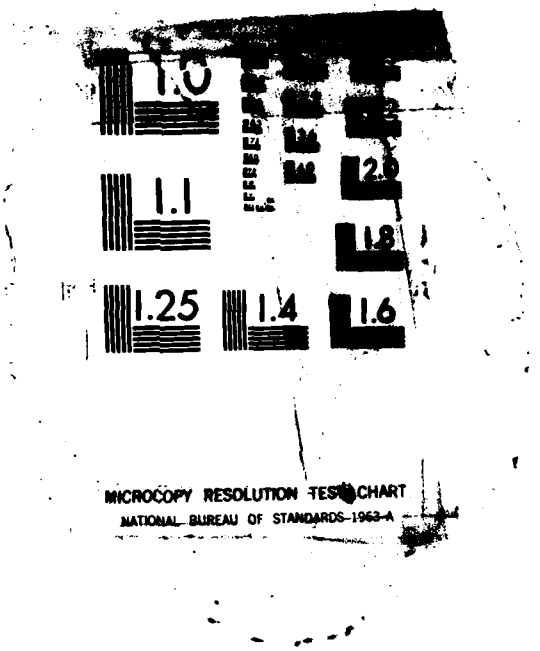
---Dan
------- *

Message 166 711 16 Sep 85
From: DLEHMAN@USC-ECLB.ARPA

AD-A171 815 ASSESSMENT OF THE ADA (TRADE NAME) VALIDATION PROCESS 2/2
(U) INSTITUTE FOR DEFENSE ANALYSES ALEXANDRIA VA
A A HOOK ET AL. JAN 86 IDA-M-157 IDA/HQ-85-30806
UNCLASSIFIED MDA903-84-C-0031 F/G 9/2 NL



END
DATE FILMED
8-87


1.0
1.1
1.25
1.4
1.6
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

To: goodenou%wang-inst.csnet@CSNET-RELAY.ARPA,
GOODENOUGH@USC-ISI.ARPA, HILFINGER@UCB-VAX.ARPA,
DEWAR@NYU.ARPA, BRENDER@DEC-MARLBORO.ARPA,
AHOOK@USC-ECLB.ARPA, KNAPPER@USC-ECLB.ARPA,
CROBY@USC-ECLB.ARPA, BWICHMANN@USC-ECLB.ARPA,
PLOEDEREDER@TL-20B.ARPA,
KEVIN.PHILLIPS%RSRE@UCL-CS.ARPA
Cc: DLEHMAN@USC-ECLB.ARPA
Subject: LMC RESULTS re FRN of 22 AUGUST

John, et al., what did the LMC decide re the two tests B74103F & CA1105B*?

---Dan
------- *

Message 171 927 16 Sep 85
From: John B. Goodenough ‹GOODENOUGH@USC-ISI.ARPA›
To: DLEHMAN@USC-ECLB.ARPA
Cc: FAST-REACTION@USC-ECLB.ARPA, KNAPPER@USC-ECLB.ARPA, AHOOK@USC-ECLB.ARPA,
goodenou%wang-inst.csnet@CSNET-RELAY.ARPA, GOODENOUGH@USC-ISI.ARPA,
HILFINGER@UCB-VAX.ARPA, PLOEDEREDER@TL-20B.ARPA, DEWAR@NYU.ARPA,
BRENDER@DEC-MARLBORO.ARPA, BWICHMANN@USC-ECLB.ARPA,
KEVIN.PHILLIPS%RSRE@UCL-CS.ARPA, CROBY@USC-ECLB.ARPA
Subject: Re: RE FRN of 21 AUGUST
In-Reply-To: ‹[USC-ECLB.ARPA]16-Sep-85 12:58:00.DLEHMAN›

This issue is not yet resolved. The E test is okay because it reflects the unsettled status of the issue. The other tests are incorrect because they require a specific interpretation of the Standard that is not fully supported by either the Standard or the LMC.

Message 173 592 16 Sep 85
From: DLEHMAN@USC-ECLB.ARPA
To: hhummel@USC-ECLB.ARPA, Clausen.IABG@MIT-MULTICS.ARPA
Cc: DLEHMAN@USC-ECLB.ARPA, AHOOK@USC-ECLB.ARPA
Subject: DISPUTED TESTS

Dear Helmut, & Stephan:

The tests B38105B, B74207A, BC3503A, & C48006B ARE WITHDRAWN from the ACVC since they allow of only one interpretation that is not fully supported by the LRM or LMC.

---Dan
------- *

›P 208,214,215
Message 208 893 23 Sep 85
From: Clausen@MIT-MULTICS.ARPA
To: DLehman@USC-ECLB.ARPA

Cc: AHook@USC-ECLB.ARPA, Clausen.IABG@MIT-MULTICS.ARPA
Subject: re: disputed test C37011A

Dan,

while I was waiting for the answer you will get from LMC, our customer is not satisfied with your reply. He wrote to us the following - and I support it.

The answer to the "19 August 1985 submission" does not give any reasonig why "sliding ... is NOT permitted". There are no rules in the LRM which explicitly (or implicitly) state this. So we cannot accept the answer without citation of any clauses of the LRM which serve as rationale for this decision.

Please give me a fast reply.

Regards, Helmut

Message 214 2155 24 Sep 85
From: DLEHMAN@USC-ECLB.ARPA
To: goodenou%wang-inst.csnet@CSNET-RELAY.ARPA, GOODENOUGH@USC-ISI.ARPA, HILFINGER@UCB-VAX.ARPA, BRENDER@DEC-MARLBORO.ARPA
Cc: DLEHMAN@USC-ECLB.ARPA, AHOOK@USC-ECLB.ARPA
Subject: RE C37011A-B(sliding bounds)

Ron, John, & Paul:

The implementer who disputed this test (FRN of 21 August) would like to have our ruling of its correctness supported by LRM references.

The rationale given for PROHIBITTING sliding by Paul, LRM 3.2.1/16,
does not seem very convincing: I suppose that 3.2.1/16 must be coupled with an assumption that, where not expressly allowed, subtype conversions are prohibitted? Though here one must note that 4.6/16 only indicates assignment statements, not initializations.

It is not clear exactly what is meant by "object" in 3.2.1/16:

1) We read in the first clause of the sentence cited by Paul "...of an object(THE DECLARED OBJECT OR ONE OF ITS SUBCOMPONENTS)..."--can we now change "object declaration" to "object or ... component declaration"?

2) But the first three paragraphs of 3.2.1 can produce the same confusion (in some of us): "An object declaration declares an object... ." "The declared object is a constant if... ." "An [declared?] object that is not a constant is called a variable ... ."--which we read on to find may be contained in "another variable that has the GIVEN VARIABLE [i.e., our "declared object"?!] as [a] subcomponent."

But wasn't this matter the subject of LMC debate? I have a NET note

from Dave SYKES where he explains the reinsertion of this test into ACVC 1.5 as being the result of the LMC ruling the test correct (though it had been objected to in 1.4 & 1.5 validations as checking that sliding DID occur! --cf the DEC VSR of 84-09-12). If it was before the LMC, is there an AI that can be given to the implementer for reference?

---Dan
------- *

Message 215 2836 24 Sep 85
From: John B. Goodenough ‹GOODENOUGH@USC-ISI.ARPA›
To: DLEHMAN@USC-ECLB.ARPA
Cc: goodenou%wang-inst.csnet@CSNET-RELAY.ARPA, GOODENOUGH@USC-ISI.ARPA, HILFINGER@UCB-VAX.ARPA, BRENDER@DEC-MARLBORO.ARPA, AHOOK@USC-ECLB.ARPA
Subject: Re: RE C37011A-B(sliding bounds)
In-Reply-To: ‹[USC-ECLB.ARPA]24-Sep-85 16:23:24.DLEHMAN›

3.2.1(16) gives the semantics for initializations in object declarations. The first clause of the sentence specifies how initial value checks are to be made for both the declared object and its subcomponents. The second phrase uses the term "object declaration" to refer to the syntax production object_ declaration to say as explicitly as can be said that a subtype conversion applies to certain object declarations as opposed to componet declarations.

(It certainly is the case that one can only apply a subtype conversion when the RM explicitly says such conversions are part of the semantics of a construct. Also, although the RM semantics might seem inconsistent, as I recall, the rationale for the rule forbidding sliding for component initialization was that sliding was not allowed for array aggregates given as the value of a record component in a record aggregate; record component initialization was viewed as being similar to creating a value with an aggregate, e.g., consider the case where every component of a record has an initial value and one writes two object declarations. In the first no initialization expression is given for the object; the default values are used. In the second, an aggregate is given as the initialization expression, and the aggregate has exactly the same expressions as are given for the default initial values. Wouldn't it seem strange for CONSTRAINT_ERROR to be raised for one declaration and not the other? It is the recollection of this argument plus the clear wording in the RM that makes me so sure that the test in its current form is correct. The incorrect version of the test, as I recall, had been developed before the Standard was finalized on this issue, and the test anticipated a resolution that would have allowed sliding. When the resolution went the other way, no one remembered that the test existed.)

As for 3.2.1(3), it is true that objects can be declared other than by object declarations, and that a subcomponent of a variable is a variable (as it should be if we want to be able to assign to subcomponents of variables).

I guess I don't see a counter-argument based on the text of the RM that the test is incorrect, or that the RM is ambiguous.

I can't find any commentary by the LMC on this point. Perhaps I raised the issue informally.

Message 193 4984 18 Sep 85
From: sykesd@wpafb-jalcf
To: AHOOK@USC-ECLB
Cc: DLEHMAN@USC-ECLB, SYKESD@WPAFB-JALCF
Subject: DISPUTED TESTS

Audrey,

We have some disputes that need to be looked at by the FRT that have come up in the pre-validation of the TeleSoft Gould compiler. The validation is scheduled for 4 NOV 85.

1.6 ACVC Suite Disputes for Gould Validation

Explanations of tests disputed in ACVC 1.6:

BA1101C -
Our dispute with this test relates to the dispute of test CA1003B(withdrawn test). Because the compilation file BA1101C4 contains an illegal compilation unit, our implementation will reject the entire compilation and not update the library even though the file contains a legal body for package BA1101C3. As a result, when the subunit BA1101C5 is compiled our compiler will issue an error since the subunit's parent body (BA1101C3) has not been successfully compiled into the library. Apart from this extra message, our implementation reports all required errors in the test.

C35904A -
The elaboration of the subtype declarations for SFX3 and SFX4 in this test raise NUMERIC_ERROR in our implementation. This is because the bounds given in the range constraints exceed the range of the fixed-point base type chosen by our implementation for FIX (the base type is chosen to have bounds -1.0 .. 1.0). The exception is raised on the conversion of the real literals 2.0 and 5.0 to the base type of FIX.

C52008B -
This test declares a record type REC2 with four discriminants of type integer. Because a default is provided for the discriminants, the type may be used in the declaration of unconstrained objects. However, the size of any such unconstrained objects would be very large (due to the string components of the type) and exceeds the maximum object size of our implementation. Our compiler computes the maximum size of the unconstrained type at type elaboration time and this raises the exception NUMERIC_ERROR. As it happens, the test itself does not declare any unconstrained objects, and hence does not require the discriminants of REC2 to be defaulted. It should be noted that all other tests declaring discriminated record types containing potentially large discriminant-dependent arrays are careful either to omit defaults for the discriminants or to provide discriminant subtypes that are constrained to a small range. The point is not that the test is in error, but that declaring a discriminated type with defaults is tantamount to expressing the intent to allow unconstrained objects of the

type to be declared. In the case of type REC2, many implementations are likely to at least raise an exception on daclaring unconstrained objects of that type. We do not consider it onerous to raise an exception on the declaration of the type since it lets the user know that he has declared an expensive construct that would be better formulated by omitting the default or constraining the subtypes of the discriminants. It should be noted that there is precedent for the legitimacy of raising an exception on the elaboration of certain types and subtypes in tests C52103X, C52104X, and C52104Y. These tests declare array subtypes with ranges exceeding INTEGER'LAST components and allow implementations to raise NUMERIC_ERROR during the elaboration of the subtypes.

C94004A, C94004B, C94004C -
These tests are disputed because they seem to make assumptions that library tasks will terminate normally after the main program has completed its execution. As stated in the notes of section 9.4 of the Ada reference manual (in reference to tasks that depend on library packages):"the language does not define whether such tasks are required to terminate." In our implementation, all such tasks are destroyed after the main program has completed its execution. The consequence for these three tests is that REPORT.RESULT will not get called. Perhaps it is the intent that this is allowable behavior for these tests. In that case, it would be helpful if it were stated more explicitly that it is not necessary for a "PASSED" message to be written out in order to pass the tests.

CE3605A -
This test is disputed because it performs character output that exceeds the line length limit of the underlying I/O subsystem on Gould's MPX operating system. MPX limits output text lines to 250 characters in length and the test attempts to write a line of 360 characters. Our implementation of TEXT_IO for MPX causes the exception USE_ERROR to be raised on an attempt to 'PUT' more than 250 characters on a single line. We recommend that this test be reclassified as a dependency test or be parameterized according to implementation limits on line lengths.

Mike Hill
AVF-WPAFB

Message 198 5624 20 Sep 85
From: DLEHMAN@USC-ECLB.ARPA
To: FAST-REACTION@USC-ECLB.ARPA, KNAPPER@USC-ECLB.ARPA, AHOOK@USC-ECLB.ARPA,
    PLOEDEREDER@TL-20B.ARPA, BRENDER@DEC-MARLBORO.ARPA,
    HILFINGER@UCB-VAX.ARPA, GOODENOUGH@USC-ISI.ARPA,
    goodenou%wang-inst.csnet@CSNET-RELAY.ARPA,
    BWICHMANN@USC-ECLB.ARPA, KEVIN.PHILLIPS%RSRE@UCL-CS.ARPA,
    DEWAR@NYU.ARPA, CROBY@USC-ECLB.ARPA
Cc: DLEHMAN@USC-ECLB.ARPA
Subject: FRN 85-09-20
         [DLEHMAN@USC-ECLB.ARPA: FRN 85-09-20]

FRT Members:

This message forwards implementer disputes for ACVC 1.6 .

---Dan

Begin forwarded message
Received: By USC-ECLB.ARPA via direct-append with Hermes; 20 Sep 85 08:50:38-PDT
Date: 20 Sep 1985 08:50-PDT
From: DLEHMAN@USC-ECLB.ARPA
To: DLEHMAN@USC-ECLB.ARPA
Subject: FRN 85-09-20
Message-ID: ‹[USC-ECLB.ARPA]20-Sep-85 08:50:35.DLEHMAN›
Sender: DLEHMAN@USC-ECLB.ARPA

ACVC 1.6 DISPUTES

BA1101C -

Our dispute with this test relates to the dispute of test CA1003B (withdrawn test). Because the compilation file BA1101C4 contains an illegal compilation unit, our implementation will reject the entire compilation and not update the library even though the file contains a legal body for package BA1101C3. As a result, when the subunit BA1101C5 is compiled our compiler will issue an error since the subunit's parent body (BA1101C3) has not been successfully compiled into the library. Apart from this extra message, our implementation reports all required errors in the test.

C359O4A -

The elaboration of the subtype declarations for SFX3 and SFX4 in this test raise NUMERIC_ERROR in our implementation. This is because the bounds given in the range constraints exceed the range of the fixed-point base type chosen by our implementation for FIX (the base type is chosen to have bounds -1.0 .. 1.0). The exception is raised on the conversion of the real literals 2.0 and 5.0 to the base type of FIX.

C52008B -

This test declares a record type REC2 with four discriminants of type INTEGER. Because a default is provided for the discriminants, the type may be used in the declaration of unconstrained objects. However, the size of any such unconstrained objects would be very large (due to the string components of the type) and exceeds the maximum object size of our implementation.

Our compiler computes the maximum size of the unconstrained type at type elaboration time, and this raises NUMERIC_ERROR.

As it happens, the test itself does not declare any unconstrained objects, and hence does NOT require discriminants of REC2 to be defaulted. It should be noted that all other tests declaring discriminated record types containing potentially large discriminant-dependent arrays are careful either to omit defaults for the discriminants or to provide discriminant subtypes that are constrained to a small range.

The point is not that the test is in error, but that declaring a discriminated type with defaults is tantamount to expressing the intent to allow unconstrained objects of the type to be declared. In the case of type REC2, many implementations are likely to at least raise an exception on declaring unconstrained objects of that type. We do not consider it onerous to raise an exception on the declaration of the type, since it lets the user know that he has declared an expensive construct that would be better formulated by omitting the default or constraining the subtypes of the discriminants.

It should be noted that there is precedent for the legitimacy of raising an exception on the elaboration of certain types and subtypes in tests C52103X, C52104X, and C52104Y. These tests declare array subtypes with ranges exceeding INTEGER'LAST components and allow implementations to raise NUMERIC_ERROR during the elaboration of the subtypes.

C94004A, C94004B, C94004C -

These tests are disputed because they seem to make assumptions that library tasks will terminate normally after the main program has completed its execution. As stated in the notes of LRM 9.4 (in reference to tasks that depend on library packages), "the language does not define whether such tasks are required to terminate."

In our implementation, all such tasks are destroyed after the main program has completed its execution. The consequence for these three tests is that REPORT.RESULT will not get called. Perhaps it is the intent that this is allowable behavior for these tests. In that case, it would be helpful if it were stated more explicitly that it is not necessary for a "PASSED" message to be written out in order to pass the tests.

CE3605A -

This test is disputed because it performs character output that exceeds the line-length limit of the underlying I/O subsystem on our operating system. Our limits for output text lines are 250 characters in length; the test attempts to write a line of 360 characters.

Our implementation of TEXT_IO causes the exception USE_ERROR to be raised on an attempt to 'PUT' more than 250 characters on a single line. We recommend that this test be reclassified as a dependency test or be parameterized according to implementation limits on line lengths.

---Dan

------- *

End forwarded message

Message 202 549 20 Sep 85
From: dewar@NYU-ACF2.ARPA
To: AHOOK@USC-ECLB.ARPA, DLEHMAN@USC-ECLB.ARPA, FAST-REACTION@USC-ECLB.ARPA,
        KNAPPER@USC-ECLB.ARPA
Subject: Re: FRN 85-09-20

I agree with implementor on BA1101C

I also agree with respect to C35904A
Message 203 2916 22 Sep 85
From: John B. Goodenough ‹GOODENOUGH@USC-ISI.ARPA›
To: DLEHMAN@USC-ECLB.ARPA
Cc: FAST-REACTION@USC-ECLB.ARPA, KNAPPER@USC-ECLB.ARPA, AHOOK@USC-ECLB.ARPA,
      PLOEDEREDER@TL-20B.ARPA, BRENDER@DEC-MARLBORO.ARPA,
HILFINGER@UCB-VAX.ARPA,
      GOODENOUGH@USC-ISI.ARPA, goodenou%wang-inst.csnet@CSNET-RELAY.ARPA,
      BWICHMANN@USC-ECLB.ARPA, KEVIN.PHILLIPS%RSRE@UCL-CS.ARPA,
DEWAR@NYU.ARPA,
      CROBY@USC-ECLB.ARPA
Subject: Re: FRN 85-09-20
In-Reply-To: ‹[USC-ECLB.ARPA]20-Sep-85 09:02:13.DLEHMAN›

BA1101C

Apparently the implementer is concerned because he outputs an extra error message for BA1101C5, but since this test unit only contains a single ERROR line, the usual rules for evaluating an implementer's response apply, i.e., as long as there is at least one error message, the test is passed.

This means that the implementer passes the test, and the test is not so much wrong (in the sense of failing conforming implementations) as it is ineffective in testing what was intended.

Therefore the test should be revised, but it need not be withdrawn as being incorrect.

C35904A

The implementer is right here. In fact, it's rather remarkable that no one else has caught this problem. This test needs a thorough revision.

C52008B

This test should be considered non-applicable for this implementer.

It should be revised to remove the unnecessary use of too-wide a discriminant subtype.

C94004A, 4B, 4C

These tests are correct. An implementation is not allowed to abort executing library tasks just because a main program completes! The note in 9.4(13) is being misread. It is correct to say "the language does not define whether [tasks that depend on library packages] are required to terminate." This means that such tasks must continue to execute under the normal rules for task execution. The note only says that there is no Ada rule requiring that such tasks be aborted when the main program completes, and the language places no requirements on programmers to ensure that library tasks terminate prior to execution of the main program. The note is trying to point out that unless programmers are careful, such tasks can execute forever.

The note just says that termination of library task execution can be completely independent of whether the main program terminates or not. In particular, I see no justification in the RM for aborting tasks that are executing when the main program terminates.

It is in fact very useful for the main program to be null in an embedded system -- all the work is carried out by library unit tasks, which run (conceptually) forever.

CE3605A

I guess an output line limit of 250 characters is an acceptable implementation limitation under AI-00325.

Message 210 976 24 Sep 85
From: hilfingr%ucbrenoir@Berkeley.EDU (Paul Hilfinger)
To: GOODENOUGH@usc-isi.arpa, FAST-REACTION@usc-eclb.arpa,
KNAPPER@usc-eclb.arpa, AHOOK@usc-eclb.arpa, DEWAR@nyu-acf2.arpa,
PLOEDEREDER@tl-20b.arpa, BWICHMANN@usc-eclb.arpa,
goodenou%wang-inst.csnet@csnet-relay.arpa,
BRENDER@dec-marlboro.arpa,
KEVIN.PHILLIPS%RSRE@ucl-cs.arpa, CROBY@usc-eclb.arpa,
DLEHMAN@usc-eclb.arpa
Subject: Re: [DLEHMAN@USC-ECLB.ARPA: FRN 85-09-20]

Re: BA1101C, C35904A, C52008B, C94004*, CE3605A

I agree with John Goodenough's analysis of these disputes.

Paul

Message 212 1349 24 Sep 85

From: Ron Brender ‹BRENDER at DEC-MARLBORO.ARPA›
To: DLEHMAN at USC-ECLB.ARPA, FAST-REACTION at USC-ECLB.ARPA,
KNAPPER at USC-ECLB.ARPA, AHOOK at USC-ECLB.ARPA
Subject: Re: FRN 85-09-20
In-Reply-To: ‹[USC-ECLB.ARPA]20-Sep-85 09:02:13.DLEHMAN›
Regarding: Message from DLEHMAN@USC-ECLB.ARPA of 20-Sep-85 1202-EDT

BA1101C
Seems like this is a case where it would be appropriate to "split" the test and require the implementation to pass it.

C35904A
The implementer is correct. [This is just one more case where attempting to maintain a distinction between CONSTRAINT_ERROR and NUMERIC_ERROR is elusive and counter-productive.]

C52008B
I agree with JBG.

C94004A, 4B, 4C
I agree with JBG - the tests are correct.

CE3605A
A maximum line length of 250 seems okay as an implementation limit.

Also, regarding the more recent item concerning the raising of USE_ERROR when creating a file of mode IN_FILE: This issue has been raised before with the conclusion that the test should be inapplicable to the implementation (and the test should be modified to report this).

Message 220 771 26 Sep 85
From: KNAPPER@USC-ECLB.ARPA
To: DLEHMAN@USC-ECLB.ARPA
Cc: FAST-REACTION@USC-ECLB.ARPA, KNAPPER@USC-ECLB.ARPA,
AHOOK@USC-ECLB.ARPA, PLOEDEREDER@TL-20B.ARPA,
BRENDER@DEC-MARLBORO.ARPA, HILFINGER@UCB-VAX.ARPA,
GOODENOUGH@USC-ISI.ARPA,
goodenou%wang-inst.csnet@CSNET-RELAY.ARPA,
BWICHMANN@USC-ECLB.ARPA,
KEVIN.PHILLIPS%RSRE@UCL-CS.ARPA, DEWAR@NYU.ARPA,
CROBY@USC-ECLB.ARPA
Subject: Re: FRN 85-09-20...
In-Reply-To: ‹[USC-ECLB.ARPA]20-Sep-85 09:02:13.DLEHMAN›

Before ECLB crashes again I'll quickly say that I agree w/JBG on the FRT messages of 9/20 and 9/23.

Bob K
Message 231 977 1 Oct 85
From: sykesd@wpafb-jalcf
To: AHOOK@USC-ECLB
Cc: DLEHMAN@USC-ECLB, KNOOPPA@WPAFB-JALCF, SYKESD@WPAFB-JALCF
Subject: ACVC Test Disputes

Audrey and Dan,

Any word from the FRT on the disputes that Mike Hill forwarded to you on 18 September? We have completed pre-validation analysis and need a resolution in order to complete the pre-validation report. We would appreciate a "fast" reply.

Mike also sent a request about a test withdrawn from 1.6 that has an identical counterpart in 1.7. Has that been resolved? Do we even need to consult you on such matters?

Thanks for your help. I know it's hard to get those Ada experts to work fast. especially since the questions are getting harder and harder. If you don't have a concensus yet, please let me know when you might expect one. Thanks.

Dave Sykes
Message 233 4011 2 Oct 85
From: The Mailer Daemon ‹Mailer@USC-ECLB.ARPA›
To: DLEHMAN@USC-ECLB.ARPA
Subject: Message of 2-Oct-85 17:12:22

Message failed for the following:
MHILL@WPAFB-JALCF.ARPA.#Internet: 550 Requested action not taken: mailbox unavailable.

Date: 2 Oct 1985 17:12-PDT
Sender: DLEHMAN@USC-ECLB.ARPA
Subject: GOULD'S DISPUTES RESOLVED
Subject: [John B. Goodenough ‹GOODENOUGH@USC-ISI.ARPA›: Re: FRN 85-0...]
From: DLEHMAN@USC-ECLB.ARPA
To: sykesD@WPAFB-JALCF.ARPA. PKNOOP@USC-ECLB.ARPA, MHILL@WPAFB-JALCF.ARPA
Message-ID: ‹[USC-ECLB.ARPA] 2-Oct-85 17:12:20.DLEHMAN›

Dear Dave, Pat, & Mike:

Since John's message/reply reflects the AVO decision re these tests, I am simply forwarding his response to convey our decision.

It was suggested that you might split the test BA1101C. In any case, you should report the behavior (considered correct as is) in the VSR --in the section on the B tests.

---Dan
------- *

Begin forwarded message
Received: from USC-ISI.ARPA by USC-ECLB.ARPA; Sun 22 Sep 85 21:30:48-PDT
Date: 23 Sep 1985 00:37:45 EDT
From: John B. Goodenough ‹GOODENOUGH@USC-ISI.ARPA›
To: DLEHMAN@USC-ECLB.ARPA

Cc: FAST-REACTION@USC-ECLB.ARPA, KNAPPER@USC-ECLB.ARPA, AHOOK@USC-ECLB.ARPA,
PLOEDEREDER@TL-20B.ARPA, BRENDER@DEC-MARLBORO.ARPA,
HILFINGER@UCB-VAX.ARPA,
GOODENOUGH@USC-ISI.ARPA, goodenou%wang-inst.csnet@CSNET-RELAY.ARPA,
BWICHMANN@USC-ECLB.ARPA, KEVIN.PHILLIPS%RSRE@UCL-CS.ARPA,
DEWAR@NYU.ARPA,
CROBY@USC-ECLB.ARPA
Subject: Re: FRN 85-09-20
In-Reply-To: ‹[USC-ECLB.ARPA]20-Sep-85 09:02:13.DLEHMAN›
Return-Path: ‹GOODENOUGH@USC-ISI›

BA1101C

Apparently the implementer is concerned because he outputs an extra error message for BA1101C5, but since this test unit only contains a single ERROR line, the usual rules for evaluating an implementer's response apply, i.e., as long as there is at least one error message, the test is passed.

This means that the implementer passes the test, and the test is not so much wrong (in the sense of failing conforming implementations) as it is ineffective in testing what was intended.

Therefore the test should be revised, but it need not be withdrawn as being incorrect.

C35904A

The implementer is right here. In fact, it's rather remarkable that no one else has caught this problem. This test needs a thorough revision.

C52008B

This test should be considered non-applicable for this implementer.

It should be revised to remove the unnecessary use of too-wide a discriminant subtype.

C94004A, 4B, 4C

These tests are correct. An implementation is not allowed to abort executing library tasks just because a main program completes! The note in 9.4(13) is being misread. It is correct to say "the language does not define whether [tasks that depend on library packages] are required to terminate." This means that such tasks must continue to execute under the normal rules for task execution. The note only says that there is no Ada rule requiring that such tasks be aborted when the main program completes, and the language places no requirements on programmers to ensure that library tasks terminate prior to execution of the main program. The note is trying to point out that unless programmers are careful, such tasks can execute forever.

The note just says that termination of library task execution can be completely independent of whether the main program terminates or not. In particular, I see no justification in the RM for aborting tasks that are

executing when the main program terminates.

It is in fact very useful for the main program to be null in an embedded system -- all the work is carried out by library unit tasks, which run (conceptually) forever.

CE3605A

I guess an output line limit of 250 characters is an acceptable implementation limitation under AI-00325.

End forwarded message

Message 93 1938 22 Aug 85
From: DLEHMAN@USC-ECLB
To: FAST-REACTION@USC-ECLB, KNAPPER@USC-ECLB, AHOOK@USC-ECLB, CROBY@USC-ECLB,
HILFINGER@UCB-VAX, KEVIN.PHILLIPS%RSRE@UCL-CS,
BRENDER@DEC-MARLBORO, GOODENOUGH@USC-ISI, DEWAR@NYU,
PLOEDEREDER@TL-20B
Cc: DLEHMAN@USC-ECLB
Subject: FAST-REACTION NOTICE

Folks:

Below are two new disputes.

B74103F-B.ADA

This test intends to show that generic formal array types having private elements are found illegal (similarly for generic formal access types that access a private type).

Our compiler considers that a generic formal array type declares a type; therefore, its declaration constitutes a type declaration. As such, and relying on LRM 7.4.1/4, we consider it to be LEGAL (similarly for generic formal access types).

Unless an explicit proof that a generic formal array (or access) type declaration is not a type declaration, we consider this test to be incorrect. It should be withdrawn, pending an LMC decision.

CA1105B4

In this test there is an attempt to "with" a unit (viz., CA1105B3M) that is an ancestor of the current unit (CA1105B4).

Following LRM 10.2/6, our compiler considers that the "with" clauses of the subunit CA1105B4 are appended to those of its parent unit (--B3M) to determine the visibility. Thus, in this case, there would be an attempt to "with" a unit before that unit is given.

We consider the "with" clause "with CA1105B3M;" to be illegal. This test should be withdrawn, pending an LMC decision. (Actually, the entire CA1105B* series is concerned.)

---Dan LEHMAN

Message 95 3153 22 Aug 85
From: hilfingr%ucbrenoir@Berkeley (Paul Hilfinger)
To: DLEHMAN@USC-ECLB
Cc: FAST-REACTION@USC-ECLB, KNAPPER@USC-ECLB, AHOOK@USC-ECLB, CROBY@USC-ECLB,
HILFINGER@Berkeley, KEVIN.PHILLIPS%RSRE@UCL-CS,

BRENDER@DEC-MARLBORO,
GOODENOUGH@USC-ISI, DEWAR@NYU, PLOEDEREDER@TL-20B
Subject: Re: FAST-REACTION NOTICE
In-Reply-To: Your message of 22 Aug 1985 15:42-PDT

> B74103F-B.ADA
>
> Our compiler considers that a generic formal array type declares a
>type; therefore, its declaration constitutes a type declaration.
>As such, and relying on LRM 7.4.1/4, we consider it to be LEGAL
>(similarly for generic formal access types).
>
> Unless an explicit proof that a generic formal array (or access) type
>declaration is not a type declaration, we consider this test to be incorrect.
>It should be withdrawn, pending an LMC decision.

12.1(2) does not mention type_declarations. The use of the term "type or subtype declaration" in 7.4.1(4) is intended to mean "in a type_declaration or a subtype_declaration." The thing that is declared, according to 12.1.2, is a "generic formal type". Looking at 3.3, we see that these "generic formal types" don't have the same properties as types until instantiated: there are no values of type BOl (e.g.) without reference to a specific actual type to which it is bound.

On the other hand, the implementors are (I think) correct in their implicit assumption that the restriction tested is unnecessary (other rules of the language seem to prevent harmful consequences). We might submit a note about how the restriction might be removed in the next version of the language. For now, however, the test seems to be correct.

> CA1105B4

> Following LRM 10.2/6, our compiler considers that the "with" clauses of
>the subunit CA1105B4 are appended to those of its parent unit
>(--B3M) to determine the visibility. Thus, in this case, there would be an
>attempt to "with" a unit before that unit is given.

The implementors reason that since

```
    ...
    with CA1105B3M;
    ...
    procedure CA1105B3M is ....
```

would be illegal, so must

```
    with CA1105B3M;
    separate (CA1105B3M)
    package body CA1105B4 is ....
```

be illegal. This is an interesting argument. To rephrase the question: Is it meaningful to talk about (see 10.2(6)) "the visibility that would be obtained at the place of the ... body stub [in the body of CA1105B3M] if

['with CA1105B3M'] were appended to the context clause of [the body of CA1105B3M]"? One can very well argue that since the latter would be illegal, it does not, in fact, make sense, and the implementor is correct.

I suspect, therefore, that the test should be withdrawn.

P. Hilfinger

[P.S. Is this a new European validation, or a repeat?]

Message 100 2804 23 Aug 85
From: John B. Goodenough ‹GOODENOUGH@USC-ISI.ARPA›
To: DLEHMAN@USC-ECLB.ARPA
Cc: FAST-REACTION@USC-ECLB.ARPA, KNAPPER@USC-ECLB.ARPA, AHOOK@USC-ECLB.ARPA, CROBY@USC-ECLB.ARPA, HILFINGER@UCB-VAX.ARPA, KEVIN.PHILLIPS%RSRE@CS.UCL.AC.UK, BRENDER@DEC-MARLBORO.ARPA, GOODENOUGH@USC-ISI.ARPA, DEWAR@NYU.ARPA, PLOEDEREDER@TL-20B.ARPA
Subject: Re: FAST-REACTION NOTICE
In-Reply-To: ‹[USC-ECLB]22-Aug-85 15:42:27.DLEHMAN›

B74103F-B

Paul presents a better argument in favor of the test than any I was thinking of proposing. However, I'm not sure one can really maintain that a declaration of a generic formal type does not declare a type. This position might lead to difficulties in deciding how to interpret the legality of the generic template.

The test was based on the straightforward reading that "type or subtype declaration" in 7.4.1(4) is equivalent to "type_declaration or subtype_declaration", but this equivalence is certainly a matter of interpretation rather than a clear derivation from the rules of the RM. Since 7.4.1(4) does not actually use the phrase "type declaration" (which would unequivocably mean just the syntactic production, type_declaration), it could be argued that 7.4.1(4) refers to any form of declaration of a type, including the forms used for declaring generic formal types.

Since I agree with Paul that the restriction checked by the test does not seem too sensible (at first analysis), and since the test has appeared for the first time in version 1.6, and since the wording of the RM could be considered (strictly speaking) unclear, I would support withdrawing the test until the LMC has ruled on the issue.

I'll submit a comment on this point.

CA1105B*

This issue is actually on the agenda for the next LMC meeting (Sept 4) as AI-00113. The current draft wording of the AI supports the interpretation of the test, and the comment that led to this AI asserts that it was the intent of the LDT to allow the usage checked for in CA1105B.

From the AVO point of view, I think it is relevant to note that this test is

one of the oldest in the suite and has never before been challenged. It would not do the cause of Ada stability any good to withdraw the test at this point (or for the LMC to reverse the interpretation given in AI-00113).

So my recommendation is that the AVO declare the test valid (even though the implementer has raised a worthy point). Alternatively, call the test inapplicable for this implementer but leave it in the suite rather than withdraw it now and then find out two weeks from now that it should not be withdrawn after all.

JBG

Message 107 1435 27 Aug 85
From: KNAPPER@USC-ECLB
To: DLEHMAN@USC-ECLB
Cc: FAST-REACTION@USC-ECLB, KNAPPER@USC-ECLB,
AHOOK@USC-ECLB, CROBY@USC-ECLB, HILFINGER@UCB-VAX,
KEVIN.PHILLIPS%RSRE@UCL-CS, BRENDER@DEC-MARLBORO,
GOODENOUGH@USC-ISI, DEWAR@NYU, PLOEDEREDER@TL-20B
Subject: Re: FAST-REACTION NOTICE
In-Reply-To: ‹[USC-ECLB]22-Aug-85 15:42:27.DLEHMAN›

B74103F-B.ADA

As John has said, Paul's argument is excellent, but I have to agree with John that the wording "type or subtype declaration" in 7.4.1(4) to mean "type_declaration or subtype_declaration" is open to interpretation. The LMC should consider the issue.

CA1105B*

This is a little sticky. Sept 4 is only a week away. I am inclined to defer judgement for that short period. Technically we have been saying that a test that is referred to the LMC is withdrawn. However, since a ruling on the issue will be made within a week, I think a deferral is a safe position. It is an old test and has not been challenged through dozens of validations thus far. Taking it out and having it's intent upheld by the LMC would be diasterous for the suite's stability or the perception of the suite's stability by the implementers.

Bob K

Message 172 888 16 Sep 85
From: John B. Goodenough ‹GOODENOUGH@USC-ISI.ARPA›
To: DLEHMAN@USC-ECLB.ARPA
Cc: goodenou%wang-inst.csnet@CSNET-RELAY.ARPA, GOODENOUGH@USC-ISI.ARPA,
HILFINGER@UCB-VAX.ARPA, DEWAR@NYU.ARPA, BRENDER@DEC-MARLBORO.ARPA,
AHOOK@USC-ECLB.ARPA, KNAPPER@USC-ECLB.ARPA, CROBY@USC-ECLB.ARPA,
BWICHMANN@USC-ECLB.ARPA, PLOEDEREDER@TL-20B.ARPA,
KEVIN.PHILLIPS%RSRE@UCL-CS.ARPA
Subject: Re: LMC RESULTS re FRN of 22 AUGUST
In-Reply-To: ‹[USC-ECLB.ARPA]16-Sep-85 13:30:38.DLEHMAN›

Well, I left the AI concerned with B74103F off the agenda, so I have nothing to report on this one.

As for CA1105B, the resolution (of AI-00113) was, in essence, that the test is correct.

Sorry about B74103F; it slipped through a crack.

>p 180
Message 180 636 17 Sep 85
From: DLEHMAN@USC-ECLB.ARPA
To: JSIDI@USC-ECLB.ARPA
Cc: AHOOK@USC-ECLB.ARPA, DLEHMAN@USC-ECLB.ARPA,
    MPMyers@MIT-MULTICS.ARPA
Subject: TEST DISPUTES FOR CA1105B4 & B74103F_B

Dear Jacqueline:

    The test disputes have been resolved as follows:

CA1105B4 is judged to be correct as written: the implementer's dispute is overruled; the test must be passed.

B74103F is withdrawn from ACVC 1.6 .

---Dan

------- *

>

Message 146 1286 8 Oct 85
From: Ron Brender ‹BRENDER@MARLBORO.DEC.COM›
To: FRTMEMS: FAST-REACTION@USC-ECLB, BABCOCK@USC-ECLB, DEWAR@NYU,
GAILLY@HI-MULTICS, GOODENOUGH@ISI, goodenou%wang-inst.csnet@CSNET-RELAY,
HILFINGER@BERKELEY, PLOEDEREDER@TL-20B, PROBERT@USC-ECLB,
KRAMER@USC-ECLB, KNAPPER@USC-ECLB, AHOOK@USC-ECLB, CROBY@USC-ECLB;
Subject: BC3220B revisited

In a round about way, the implementer that challenged test BC3220B found out that I had supported the test as is, and contacted me to discuss it. While I still believe my conclusion (and the very similar supporting arguments by John Goodenough) to be the most straight forward interpretation of the LRM, I do concede that there are some subtle and fundamental points involved that would be appropriate for the LMC to review. In short, while not changing my mind per se, I would go along with a recommendation that the test be referred to the LMC for review (and be considered non-applicable to this implementer while the review is in progress).
Ron

Message 147 1176 8 Oct 85
From: KNAPPER@USC-ECLB.ARPA
To: brender@MARLBORO.DEC.COM
Cc: BABCOCK@USC-ECLB.ARPA, DEWAR@NYU.ARPA,
GAILLY@HI-MULTICS.ARPA, GOODENOUGH@USC-ISI.ARPA,
goodenou%wang-inst.csnet@CSNET-RELAY.ARPA,
HILFINGER@UCB-VAX.ARPA, PLOEDEREDER@TL-20B.ARPA,
PROBERT@USC-ECLB.ARPA, KRAMER@USC-ECLB.ARPA,
KNAPPER@USC-ECLB.ARPA, AHOOK@USC-ECLB.ARPA
Subject: Re: BC3220B revisited
In-Reply-To: ‹"MS11(2447)+GLXLIB5(0)" 12149484646.22.229.4527 at MARLBORO.DEC.COM›

Ron,

This causes a bit of a scrape. FRT decisions are binding unless the AJPO Director chooses his/her own interpretation. What really disturbs me more however is that a little persuasive "arm twisting" may be the culprit here. I'm not challenging your professional honor Ron, I may however be questioning the implementer's. I will entertain a reading from the rest of the FRT members, but I thought we were very solidly in favor of the test being correct.

Bob K
Message 149 5771 9 Oct 85
From: MPMyers@MIT-MULTICS.ARPA (PDouspis)
To: dewar@NYU.ARPA, gailly@HI-MULTICS.ARPA, goodenough@USC-ISI.ARPA,
goodenou%wang-inst.csnet@CSNET-RELAY.ARPA, hilfinger@UCB-VAX.ARPA,
ploedereder@TL-20B.ARPA, probert@USC-ECLB.ARPA,
kramer@USC-ECLB.ARPA, knapper@USC-ECLB.ARPA, ahook@USC-ECLB.ARPA,
croby@USC-ECLB.ARPA, Ichbiah@HI-MULTICS.ARPA, jsidi@USC-ECLB.ARPA,

MPMyers@MIT-MULTICS.ARPA
Subject: Disputed test BC3220B
Posted-Date: 9 Oct 85 12:00 EDT

Dispute over ACVC test BC3220B: more against this test.

We asked the Alsys group in charge of the coordination of the Ada ISO Standard to give their opinion on this test. This is their answer which seems to show that the case is not obvious from the RM.

To quote J.B. Goodenough, the test can be summarized as follows:

```
generic
  type T is (<>); package SET_OF is
  type SET is array (T) of BOOLEAN; end SET_OF;

package CHAR_SET is new SET_OF (CHARACTER);
```

Chapter 12 states the relationship between names in the generic unit and those in the instance according to the following rule:

12.3(5)
The instance is a copy of the generic unit, apart from the generic formal part; thus the instance of a generic package is a package, that of a generic procedure is a procedure, and that of a generic function is a function. For each occurrence, within the generic unit, of a name that denotes a given entity, the following list defines which entity is denoted by the corresponding occurrence within the instance.
...

12.3(13) (h)

For a name that denotes a local entity declared within the generic unit: The corresponding name denotes the entity declared by the corresponding local declaration within the instance.

This means that SET in the generic unit corresponds to SET in the instance (and reciprocally).

How, then must SET (in the instance) be understood is explained in chapter 8:

8.3(1)

The meaning of the occurrence of an identifier at a given place in the text is defined by the visibility rules and also, in the case of overloaded declarations, by the overloading rules. The identifiers considered in this chapter include any identifier other than a reserved word, an attribute designator, a pragma identifier, the identifier of a pragma argument, or an

identifier given as a pragma argument. The places considered in this chapter are those where a lexical element (such as an identifier) occurs. The overloaded declarations considered in this chapter are those for subprograms, enumeration literals, and single entries.

8.3(2)

For each identifier and at each place in the text, the visibility rules determine a set of declarations (with this identifier) that define possible meanings of an occurrence of the identifier. A declaration is said to be visible at a given place in the text when, according to the visibility rules, the declaration defines a possible meaning of this occurrence. Two cases arise.

Together with the chapter 12 excerpts this implies that the meaning of the name SET in the instance must be understood through its ties with SET in the generic unit.

Finally, chapter 4 clarifies the status of SET (in the generic unit and perforce (as upper stated) in the instance itself):

4.9(11)
A static range is a range whose bounds are static expressions. A static range constraint is a range constraint whose range is static. A static subtype is either a scalar base type, other than a generic formal type; or a scalar subtype formed by imposing on a static subtype either a static range constraint, or a floating or fixed point constraint whose range constraint, if any, is static. A static discrete range is either a static subtype or a static range. A static index constraint is an index constraint for which each index subtype of the corresponding array type is static, and in which each discrete range is static. A static discriminant constraint is a discriminant constraint for which the subtype of each discriminant is static, and in which each expression is static.

SET in the generic unit can therefore in no circumstances have a static index constraint.

Conclusion:

Since SET in the instance can be understood only through its relation with SET in the generic unit and since the latter cannot have a static index constraint, SET in the instance cannot have a static index constraint.

VA

Best regards,

Pierre Douspis
Message 154 1036 9 Oct 85

From: Ichbiah@HI-MULTICS.ARPA
To: fast-reaction@USC-ECLB.ARPA, Gailly@HI-MULTICS.ARPA,
probert@USC-ECLB.ARPA, kramer@USC-ECLB.ARPA,
knapper@USC-ECLB.ARPA, ahook@USC-ECLB.ARPA, croby@USC-ECLB.ARPA
Cc: jsidi@USC-ECLB.ARPA, Ichbiah@HI-MULTICS.ARPA
Subject: Re: BC3220B
Posted-Date: 7 Oct 85 10:05 CDT

Re: BC3220B

The rules given in 12.3(5..16) are meant to define visibility within the generic instantiation.

They should not be used for determining staticness, otherwise contradictions are bound to happen with 4.9(11): "A static subtype is either a scalar base type, other than a generic formal type ..."

If the rules of staticness were to be interpreted after a kind of "macro-expansion", the above sentence would become vacuous.

Jean Ichbiah
Message 155 1592 9 Oct 85
From: MAMyers@HI-MULTICS.ARPA (EMorel)
To: fast-reaction@USC-ECLB.ARPA, babcock@USC-ECLB.ARPA,
dewar@NYU.ARPA, gailly@HI-MULTICS.ARPA, goodenough@USC-ISI.ARPA,
goodenou%wang-inst.csnet@CSNET-RELAY.ARPA, hilfinger@UCB-VAX.ARPA,
ploedereder@TL-20B.ARPA, probert@USC-ECLB.ARPA,
kramer@USC-ECLB.ARPA, knapper@USC-ECLB.ARPA, ahook@USC-ECLB.ARPA,
croby@USC-ECLB.ARPA, Ichbiah@HI-MULTICS.ARPA
Cc: jsidi@USC-ECLB.ARPA
Subject: Re: More on BC3220B
Posted-Date: 7 Oct 85 10:08 CDT

Re: More on BC3220B

I insist on the fact that staticity of generic instantiations should be carefully studied by the LMC before imposing anything to the implementors. We already had a bad experience last year with the tests BC3205A and B which were disputed by our implementation. The FRT answer forced us to make major changes in our implementation to pass those tests, but the LMC finally decided, later on, that our arguments were correct and we were then forced to change again our implementation to come back to the previous strategy.

We would prefer not to do again the same kind of things, and we consider the case raised by BC3220B important enough to be strongly debated by the LMC before taking any decision.

In the meanwhile, we suggest that this test be withdrawn from ACVC 1.6.

Etienne Morel
Message 158 4515 9 Oct 85
From: hilfingr@ucbrenoir.Berkeley.EDU
To: MPMyers@mit-multics.arpa (PDouspis)
Cc: dewar@nyu.arpa, gailly@hi-multics.arpa, goodenough@usc-isi.arpa,
        goodenou%wang-inst.csnet@csnet-relay.arpa,
        hilfingr@ucbrenoir.Berkeley.EDU, ploedereder@tl-20b.arpa,
        probert@usc-eclb.arpa, kramer@usc-eclb.arpa, knapper@usc-eclb.arpa,
        ahook@usc-eclb.arpa, croby@usc-eclb.arpa, Ichbiah@hi-multics.arpa,
        jsidi@usc-eclb.arpa
Subject: Re: Disputed test BC3220B
In-Reply-To: Your message of Wed, 9 Oct 85 11:54 EDT.
                ‹851009155458.601827@MIT-MULTICS.ARPA›

The Alsys group's opinion is interesting, but I detect at least one
flaw in their reasoning.

[For reference, here is the disputed program fragment.

```
generic
  type T is (<>); package SET_OF is
  type SET is array (T) of BOOLEAN; end SET_OF;

package CHAR_SET is new SET_OF (CHARACTER);
```
]

Now, 12.3(9) states that CHAR_SET.T denotes CHARACTER. Thus,
the meaning established by visibility rules, etc., is clear. Turning
now to staticness per se, we have to interpret this phrase ``A static
subtype is ... a scalar base type other than a generic formal type.''
Is CHAR_SET.T a generic formal type? Notice, I ask whether CHAR_SET.T
is, not whether SET_OF.T is. If the name CHAR_SET.T denotes CHARACTER,
does it also denote a generic formal parameter?

The Alsys group quotes 12.3(13) as defining the meaning of
CHAR_SET.SET: ``For a name that denotes a local entity declared within
the generic unit: The corresponding name denotes the entity declared
by the corresponding local declaration within the instance.'' From
this, they eventually conclude that ``the meaning of the name SET in
the instance [CHAR_SET.SET] must be understood through its ties with
SET in the generic unit.'' This does not follow at all. While
12.3(13) says that CHAR_SET.SET refers to some local declaration
CORRESPONDING TO the declaration of SET_OF.SET, this says nothing
about the local declaration itself. In particular, it does not follow
that the ``meaning of SET must be understood through its ties with SET
in the generic unit,'' merely that references to SET within the
instance refer to the local declaration, just as they appear to refer
to the declaration of SET_OF.SET in the template. That is, one can
argue that SET_OF.T is a generic formal type (hence not static),
whereas CHAR_SET.T is CHARACTER (and hence static).

At first this might seem to make no sense. After all, how can we then

forbid something like this:

```
generic
    type T is range <>;
package BAD is
    type Q is range T'FIRST .. T'LAST;
    . . .
end BAD;
```

By my argument, an instance

```
    type INT is range -100 .. 100;
package NOT_SO_BAD_MAYBE is new BAD(INT);
```

is not illegal, since the bounds on Q are static. But in fact, there is no problem. While the instance is OK, the GENERIC DEFINITION has a type BAD.INT with non-static bounds, and this is wrong. This is the only interpretation of the Standard that makes sense out of what it means for something to ''be a generic formal'' whereas in the instance, it is not a generic formal. During the checking of the generic definition is for semantic consistency, we can treat identifiers as denoting generic formals.

I conclude that the Alsys analysis is faulty. On the other hand, it appears that we are again running into some problems with the definitions of generics. Notice that the Alsys analysis is consistent with a notion advanced by some of them earlier that semantic analysis of generic bodies would occur only upon instantiation. If one adopts that point of view, one can only make sense out of the Standard by ascribing generic formalness to names in the instance, since there is no other time that there really ARE names. This is another example of how the Standard is less than precise on the definition of generics and on the distinction between properties of denoted things and the names denoting them. As a mathematician, I find this disturbing, and I suspect we are going to have to agree to have this clarified by the LMC.

Message 159 7032 9 Oct 85
From: John B. Goodenough <GOODENOUGH@USC-ISI.ARPA>
To: MPMyers@MIT-MULTICS.ARPA (PDouspis)
Cc: dewar@NYU.ARPA, gailly@HI-MULTICS.ARPA, goodenough@USC-ISI.ARPA,
goodenou%wang-inst.csnet@CSNET-RELAY.ARPA, hilfinger@UCB-VAX.ARPA,
ploedereder@TL-20B.ARPA, probert@USC-ECLB.ARPA, kramer@USC-ECLB.ARPA,
knapper@USC-ECLB.ARPA, ahook@USC-ECLB.ARPA, croby@USC-ECLB.ARPA,
Ichbiah@HI-MULTICS.ARPA, jsidi@USC-ECLB.ARPA
Subject: Re: Disputed test BC3220B
In-Reply-To: <851009155458.601827@MIT-MULTICS.ARPA>

I think Pierre's argument depends on using the term "meaning" in both a technical and non-technical sense. As the citation from 8.3(2) says, the "meaning" of an identifier (in the technical sense) is the declaration it denotes. 12.3(13) as cited says that the instance CHAR_SET contains a local declaration

```
    type SET is array (T) of BOOLEAN;
```

12.3(9) says T denotes what the corresponding actual parameter denotes.

Pierre says "The meaning of the name SET in the instance must be understood through its ties with SET in the generic unit." But here "meaning" is not being used in the sense of what declaration is denoted by T or SET, but rather in the informal sense. The tie to the template is stated by 12.3(13), and that's the only tie.

It is probably relevant, however, to raise broader issues here of how the interpretation of declarations in a template can change from interpretations of the "corresponding" declarations in an instance. For example, consider:

```
generic
     type T is private;
     CONS : T;
package P is
     subtype ST is T;
     X : ST := 5;    -- illegal
     C : ST := CONS;      -- C is non-static
     type DER_ST is new ST;    -- = and /= are declared
end P;

package NP is new P (INTEGER, 3);

X1 : NP.ST := 5;     -- legal
C1 : NP.ST := NP.C; -- C1 is non-static
type DER1_ST is new NP.ST;     -- + etc. declared implicitly
```

The declaration of P.X is illegal because inside the template, P.ST is a private type and no implicit conversions from universal_integer to ST are visible. P.C is non-static since P.ST is a generic formal type and since P.CONS is not declared by an object declraation. P.DER_ST only gets the equality and inequality operators because the class of parent type P.ST is private.

The effect of instantiating NP is to create and elaborate an instance equivalent to the following:

```
-- a constant %CONS% is created that has the value 3
package NP is
     subtype ST is INTEGER;
     C : constant ST := %CONS%;
     type DER_ST is new ST;
end NP;
```

Here I have replaced the formal parameter names of the template with the bindings created for the instance, in accordance with 12.3(7 9).

Since NP.ST denotes the type INTEGER, the declaration of X1 is legal, and in particular, the use of the numeric literal is legal.

NP.C is nonstatic (more precisely, cannot be used in a static expression) because its initialization expression is not static; %CONS% is not a constant

explicitly declared by an object declaration, so C1 is also nonstatic.

The derivation from NP.ST includes an implicit declaration of the arithmetic and relational operators because NP.ST denotes an integer type. Surely we must say that NP.ST denotes an integer type if we are going to allow the use of literals in the initialization of X1.

Now what about the declaration:

```
Y : NP.DER_ST := 5;
```

Are there any conversion operations declared for NP.DER_ST? No; the rules in 12.3 do not allow additional declarations to be created within the template. Thus the declaration of Y is illegal.

DER1_ST declares an integer type, since NP.ST is an integer type, and so the integer arithmetic operators (and others) are declared implicitly for DER1_ST, even though no such operators were declared for NP.DER_ST, which has exactly the same form of parent type. Within the template, P.ST is a private type; within the instance, NP.ST is an integer type.

Now consider the declaration:

```
type DER2 is new NP.DER_ST;
```

What operations are declared for DER2, i.e., to what class of types does NP.DER_ST belong? The parent type of NP.DER_ST is NP.ST, which is an integer type, so DER2 is also an integer type and has integer operations declared for it. Thus

```
X2 : DER2 := 5;
```

is legal even though

```
X1 : NP.DER_ST := 5;
```

is not (no implicit conversions are declared for NP.DER_ST).

This effect may be surprising, but it is just an interesting "theorem" derived from the rules.

So much for legalisms.

Etiene Morel brings up the case history of BC3205A and 5B, which involved instantiations with types whose discriminants have default values. As I recall, the arguments that eventually resulted in the LMC interpretation were methodological, and were presented by Bell Tech Ops, not by Alsys. With respect to BC3220B, So far, we haven't seen any methodological arguments saying that even if the RM is clear, it is wrong because it forbids critically useful capability to a programmer, nor have we seen arguments saying that the RM is wrong because it is unimplementable. These are the kinds of arguments that convince the LMC to make interpretations that run counter to what the RM appears to say.

In fact, from a methodological point of view, I would say that the test case shows a very USEFUL capability that is supported by generic units. Why should this ability not be supported?

The examples with the NP package show that the current rules are clear enough to provide interpretations of unusual cases. I certainly wouldn't object to having the LMC affirm these interpretations. The NP cases show, however, that the "meaning" (in the non-technical sense) of declarations in the instance is often different from the "meaning" in the template, and that we should not be surprised that certain usages that would be illegal inside a template are legal with respect to an instantiation. In particular, I would find it very hard to justify why X1 : NP.ST := 5 should be declared to be illegal on the basis that NP.ST is in some sense a "private" type for which no implicit conversions are defined. And if the declaration of X1 is legal, it seems to me that the other cases I have considered must also have the interpretations I have given.

From the AVO viewpoint, it is probably worth considering how many implementations have validated under version 1.6 without challenging this test. Given that the RM seems to support the test, and the absence of other challenges, I don't see why the AVO should withdraw the test.

Of course, the LMC can also discuss these issues, since they do involve some careful reasoning based on the rules of the RM. I'll be happy to put this topic (in its expanded form) on the LMC agenda, since, as Ron Brender notes, it does involve some careful reasoning.

Message 166 2845 10 Oct 85
From: Dancy@HI-MULTICS.ARPA
To: goodenou%wang-inst.csnet@CSNET-RELAY.ARPA, hilfinger@UCB-VAX.ARPA, ploedereder@TL-20B.ARPA, probert@USC-ECLB.ARPA, kramer@USC-ECLB.ARPA, knapper@USC-ECLB.ARPA, ahook@USC-ECLB.ARPA, croby@USC-ECLB.ARPA, dewar@NYU.ARPA, gailly@HI-MULTICS.ARPA, goodenough@USC-ISI.ARPA
Cc: MPMyers@MIT-MULTICS.ARPA
Subject: Re: Disputed test BC3220B
Posted-Date: 10 Oct 85 10:19 CDT

Re: BC3220B

As far as I am concerned, the matter is not that clear. For instance, is there anything wrong with the following interpretation?

12.3(17) says:

"For the elaboration of a generic instantiation, each expression supplied as an explicit generic actual parameter is first evaluated, as well as each expression that appears as a constituent of a variable name or entry name supplied as an explicit generic actual parameter; these evaluations proceed in some order that is not defined by the language. Then, for each omitted generic association (if any), the corresponding default

expression or default name is evaluated; such evaluations are performed in the order of the generic parameter declarations. Finally, the implicitly generated instance is elaborated. The elaboration of a generic instantiation may also involve certain constraint checks as described in later subsections."

This indicates that an instance of a generic unit only exists after the elaboration of a generic instantiation: this instance is implicitly generated at elaboration time after the evaluation of any expression, default name, and so on needed for this generation. Thus there is no instance at compile time, and speaking of such things as "static in the instance" does not make sense.

In fact, all the compile time requirements for a generic instantiation are in terms of matches, and not in terms of exact

identification of parameters, since some of them are typically nonstatic (indexed components, entries of a family...).

Note:
This rule applies in all cases, and it can only be considered an optimization to generate the instance earlier in simple cases. Since an optimization can not have such effects as rendering a subtype static

```
subtype NS is INTEGER range 1 .. IDENT_INT(1)-IDENT_INT(1);
--NS is not static, even if the compiler can determine that its
--upper bound is 0
```

one cannot say that in BC3220B's simple case, the compiler must be able to make such an illegal optimization.

Best regards
H. Dancy

Message 51 2360 12 Sep 85
From: MAMyers@HI-MULTICS.ARPA (Douspis)
To: AHOOK@USC-ECLB.ARPA, JSIDI@USC-ECLB.ARPA
Cc: MAMyers@HI-MULTICS.ARPA
Subject: Validation Tests (1.6) BC3220B_B.ADA

Test Reference: BC3220B_B.Ada

Test Intent:

This test intends to show that when a formal type is used to declare an array index, then after an instantiation where the formal type is associated with a static discrete type the index subtype of the instantiated array is considered static (outside the instantiated unit).

Our Rational:

We disagree with the purpose of the test. Our compiler considers that the instantiation of SET at line 21 creates a type ENUM.SET which is NOT static since the corresponding type in the template is not static.

According to us, a type can be static only if it is an explicitly declared type, as opposed to an instantiated type.

Before making any decision, we suggest that the LMC carefully examine this point since it could be a case where you can't see the wood for the trees. Take the following example:

```
(1)  generic
        type T is (<>);
     package SET_OF is
        type SET is array (T) of BOOLEAN;
     end SET_OF;
     subtype INT is INTEGER range 1..10;
     package INT_SET is new SET_OF(INT);

(2)  generic
        LOW  : INTEGER;
        HIGH : INTEGER;
     package SET_OF is
        type SET is array(LOW..HIGH) of BOOLEAN;
     end SET_OF;
     package INT_SET is new SET_OF(1, 10);
```

If the instantiation of (1) creates a static SET, I assume that the one of (2) does also. But the later case clearly violates 4.9(6) since the matching of LOW and HIGH with respectively 1 and 10 does not generate explicitly declared constant.

Thinking more on that point could probably lead to a lot of other surprises. We believe that our implementation which assumes that staticness is not changed by an instantiation is more regular and more safe for the user.

Conclusion:

We suggest that this test be withdrawn, pending a LMC decision.

Best regards,

Pierre Douspis (Alsys)

Message 79 2206 17 Sep 85
From: DLEHMAN@USC-ECLB.ARPA
To: FAST-REACTION@USC-ECLB.ARPA, KNAPPER@USC-ECLB.ARPA, AHOOK@USC-ECLB.ARPA, DEWAR@NYU.ARPA, PLOEDEREDER@TL-20B.ARPA, BWICHMANN@USC-ECLB.ARPA, goodenou%wang-inst.csnet@CSNET-RELAY.ARPA, GOODENOUGH@USC-ISI.ARPA, BRENDER@DEC-MARLBORO.ARPA, KEVIN.PHILLIPS%RSRE@UCL-CS.ARPA, HILFINGER@UCB-VAX.ARPA, CROBY@USC-ECLB.ARPA
Cc: DLEHMAN@USC-ECLB.ARPA
Subject: FRN 85-09-17

FRT Members:

Test BC3220B_B is disputed; the implementer's argument follows.

This test checks that when a formal type is used to declare an array index, then after an instantiation where the formal type is associated with a static discrete type the index subtype of the instantiated array is considered static (outside of the instantiated unit).

But we consider the instantiation of SET at line 21 to create a type ENUM.SET that is NOT static, since the corresponding type in the template is not static. A type can be static only if it is an explicitly declared type--not an instantiated type.

Lest you not see the wood for the trees [I may wait 'til Autumn to return THIS resolution!], consider the following example.

```
        subtype INT is INTEGER range 1..10;
        package INT_SET is new SET_OF(INT);

(2)     generic
        LOW  : INTEGER;
        HIGH : INTEGER;
        package SET_OF is
        type SET is array(LOW..HIGH) of BOOLEAN;
        end SET_OF;
        package INT_SET is new SET_OF(1, 10);
```

if the instantiation of (1) creates a static SET, presumably so does the one of (2). But the latter case clearly violates LRM 4.9/6, since the matching of LOW and HIGH with 1 and 10, resp., does not generate explicitly declared constants.

Thinking more on that point could probably lead to a lot of other surprises. Our implementation assumes that staticness is not changed by an instantiation; this is more regular and safe for a user.

This matter should be referred to the LMC.

---Dan
------- *

Message 80 1143 17 Sep 85
From: DLEHMAN@USC-ECLB.ARPA
To: FAST-REACTION@USC-ECLB.ARPA, KNAPPER@USC-ECLB.ARPA,
AHOOK@USC-ECLB.ARPA, DEWAR@NYU.ARPA,
PLOEDEREDER@TL-20B.ARPA, BWICHMANN@USC-ECLB.ARPA,
goodenou%wang-inst.csnet@CSNET-RELAY.ARPA,
GOODENOUGH@USC-ISI.ARPA, HILFINGER@UCB-VAX.ARPA,
CROBY@USC-ECLB.ARPA, KEVIN.PHILLIPS%RSRE@UCL-CS.ARPA,
BRENDER@DEC-MARLBORO.ARPA
Cc: DLEHMAN@USC-ECLB.ARPA
Subject: FRN 85-09-17 CORRECTION

Folks.

the FRN 85-09-17 contains two examples, the first of which was not fully given in my previous message due my confusion re exactly where I'd gotten in my $TEXT-INPUT-etc. file(dead host). That example should read:

```
(1)  generic
     type T is (<>);
     package SET_OF is
     type SET is array (T) of BOOLEAN;
     end SET_OF;

     subtype INT is INTEGER range 1..10;
     package INT_SET is new SET_OF(INT);
```

---Dan
------- *

Message 86 1285 18 Sep 85
From: KNAPPER@USC-ECLB.ARPA
To: DLEHMAN@USC-ECLB.ARPA
Cc: FAST-REACTION@USC-ECLB.ARPA, KNAPPER@USC-ECLB.ARPA,

AHOOK@USC-ECLB.ARPA, DEWAR@NYU.ARPA,
PLOEDEREDER@TL-20B.ARPA, BWICHMANN@USC-ECLB.ARPA,
goodenou%wang-inst.csnet@CSNET-RELAY.ARPA,
GOODENOUGH@USC-ISI.ARPA, BRENDER@DEC-MARLBORO.ARPA,
KEVIN.PHILLIPS%RSRE@UCL-CS.ARPA,
HILFINGER@UCB-VAX.ARPA, CROBY@USC-ECLB.ARPA
Subject: Re: FRN 85-09-17
In-Reply-To: ‹[USC-ECLB.ARPA]17-Sep-85 12:12:46.DLEHMAN›

I really can't find anything in the RM to support the implementer's claim that "A type can be static only if it is an explicitly declared type--not an instantiated type." RM 12.1.1(3) states that a generic formal object of mode "in" is a constant and RM 12.3.1(1) states "If a generic unit has a generic formal object of mode 'in', a check is made that the value of the expression belongs to the subtype denoted by the type mark, as for an explicit constant declaration." Based on this, I do not agree with the implementer that RM 4.9(6) is violated.

Bob K
Message 90 1912 19 Sep 85
From: Ron Brender ‹BRENDER at DEC-MARLBORO.ARPA›
To: FRTMEMS: FAST-REACTION at USC-ECLB, BABCOCK at USC-ECLB,
DEWAR at NYU, GAILLY at HI-MULTICS, GOODENOUGH at ISI,
goodenou%wang-inst.csnet at CSNET-RELAY, HILFINGER at BERKELEY,
PLOEDEREDER at TL-20B, PROBERT at USC-ECLB, KRAMER at USC-ECLB,
KNAPPER at USC-ECLB, AHOOK at USC-ECLB, CROBY at USC-ECLB;
Subject: Two FRT items

FRT members:

1) Re the 1750a with the simulated disk in memory...

Grumble... I guess I would have to concede that it is an acceptable implementation restriction for the external files to disappear when the main program completes, although I'm not completely happy with the idea.

I sure hope the implementer went to all of the trouble to simulate files in memory because it was perceived as important to the application domain, rather than in order to pass validation. The latter means that the AVO is not getting the right message out and ought to be a matter of concern.

2) Re BC3220B

I think the implementer is wrong. While type SET is declared within the generic in terms of a generic formal type, in the instantiation the semantics of that type is explained in terms of a "copy" in which the occurence of the formal type name is understood to DENOTE the actual subtype ENUM, which is definitely static. Moreover, the rule regarding the use of others is interpreted in AI-00310 in terms of the staticness of the corresponding index constraint (that is,

on a constraint by constraint basis rather than on the staticness of the array subtype as a whole).

Ron

Message 91 3388 19 Sep 85
From: John B. Goodenough ‹GOODENOUGH@USC-ISI.ARPA›
To: DLEHMAN@USC-ECLB.ARPA
Cc: FAST-REACTION@USC-ECLB.ARPA, KNAPPER@USC-ECLB.ARPA, AHOOK@USC-ECLB.ARPA, DEWAR@NYU.ARPA, PLOEDEREDER@TL-20B.ARPA, BWICHMANN@USC-ECLB.ARPA, goodenou%wang-inst.csnet@CSNET-RELAY.ARPA, GOODENOUGH@USC-ISI.ARPA, BRENDER@DEC-MARLBORO.ARPA, KEVIN.PHILLIPS%RSRE@UCL-CS.ARPA, HILFINGER@UCB-VAX.ARPA, CROBY@USC-ECLB.ARPA
Subject: Re: FRN 85-09-17
In-Reply-To: ‹[USC-ECLB.ARPA]17-Sep-85 12:12:46.DLEHMAN›

BC3220B

Let me repeat a portion of the test here:

```
generic
    type T is (<>);
package Set_Of is
    type Set is array (T) of Boolean;
end Set_Of;

package Char_Set is new Set_Of (Character);
```

The question here is whether the index constraint of Char_Set.Set is static because the actual generic parameter is static.

4.9(11) says "a static index constraint is an index constraint for which each index subtype of the corresponding array type is static, and in which each discrete range is static."

In the test, both the index subtype and the discrete range are defined by the formal generic parameter T. Since T is a generic formal type, it is clear that the array has a non-static subtype at least within the generic unit, and this is checked by the test.

On the other hand, after the instantiation, we must ask "what is the index subtype of Char_Set.Set?" Certainly it is not a generic formal type. The index subtype is Character, which is certainly a static subtype, and so is the discrete range, since the bounds are given by the static expressions, CHARACTER'FIRST and CHARACTER'LAST.

This is the basis for the test, and the implementer's arguments don't invalidate this basis. In fact both of the implementer's arguments are wrong.

The implementer's first argument concerns whether the type Char_Set.Set is static, but an array type is never static. The real issue is whether the index constraint is static or not.

The implementer's second argument revolves around this example:

```
(2)  generic
     LOW  : INTEGER;
     HIGH : INTEGER;
     package SET_OF is
     type SET is array(LOW..HIGH) of BOOLEAN;
     end SET_OF;
     package INT_SET is new SET_OF(1, 10);
```

LOW and HIGH can't be considered static bounds because they are not entities declared by a constant_declaration, either before or after the instantiation. INT_SET.SET is not a type that has a static index constraint.

The implementer seems to be concerned that the properties of an entity declared within a template might differ within the template and outside it. But this is already possible:

```
generic
    type U is (<>);
package GP is
    subtype SU is U;
    ...
end GP;
```

Within GP, SU is a non-static type because U is a formal generic type. After instantiation, SU denotes the type denoted by U (3.3.2/6 "... the [declared] subtype is the same as that denoted by the type mark [given in the subtype indication]"). So, let's look at an instantiation:

```
package ST is new GP (INTEGER);
```

Since INTEGER is static, and since ST.SU denotes INTEGER, ST.SU also is a static subtype. So the "staticness" of SU changes after the instantiation.

In short, the test is okay.

Message 101 895 24 Sep 85
From: hilfingr%ucbrenoir@Berkeley.EDU (Paul Hilfinger)
To: GOODENOUGH@usc-isi.arpa, FAST-REACTION@usc-eclb.arpa,
KNAPPER@usc-eclb.arpa, AHOOK@usc-eclb.arpa, DEWAR@nyu-acf2.arpa,
PLOEDEREDER@tl-20b.arpa, BWICHMANN@usc-eclb.arpa,
goodenou%wang-inst.csnet@csnet-relay.arpa,
BRENDER@dec-marlboro.arpa,
KEVIN.PHILLIPS%RSRE@ucl-cs.arpa, CROBY@usc-eclb.arpa,
DLEHMAN@usc-eclb.arpa
Subject: Re: FRN 85-09-17

I agree with John. Test BC3220B appears to be correct.

Paul

Message 262 4083 1 Oct 85
From: KNAPPER@USC-ECLB.ARPA
To: dlehman@USC-ECLB.ARPA
Subject: n
[BABCOCK@USC-ECLB.ARPA: Invalid/not applicable ACVC tests]

Dan,

These came in from Dave Babcock at Rolm. You may wish to check whether Rolm/DG is in the midst of any validation attempts before you send them out, but since the recent activity has been light I see no problem with sending it out right away.

Bob K

Begin forwarded message
Received: By USC-ECLB.ARPA via direct-append with Hermes; 30 Sep 85 11:05:28-PDT
Date: 30 Sep 1985 11:05-PDT
From: BABCOCK@USC-ECLB.ARPA
To: Knapper@USC-ECLB.ARPA
Cc: Babcock@USC-ECLB.ARPA
Subject: Invalid/not applicable ACVC tests
Message-ID: ‹[USC-ECLB.ARPA]30-Sep-85 11:05:27.BABCOCK›
Sender: BABCOCK@USC-ECLB.ARPA

We believe the following ACVC tests are invalid and/or inapplicable to our implementation for the reasons indicated. I'd like you to "run them up the flag" and see what happens. Thanks, Dave Babcock.

C48008A

At line 255, the statement "VCB := NEW TB(4)" is executed, where TB is a record type with a single (integer subtype) discriminant, and VCB has type access TB(3). The assignment therefore raises CONSTRAINT_ERROR due to a mismatch of discriminants. However, line 259 checks that the default initial value for TB's single record field was not evaluated before raising the exception, declaring failure if it was evaluated.

This test does not agree with the LRM. LRM 4.8, paragraph 6 states that evaluation of an allocator includes default initializations of the object created before returning the access value designating it. LRM 5.2, paragraph 3 states that evaluation of the LHS and RHS of an assignment statement (the target variable and the source expression) takes place before that constraint check. Therefore, the default initialization of the object designated by the new access value must occur before the CONSTRAINT_ERROR is raised, and the test is the opposite of that required by the language definition.

CA1105B4

Lines 7-9 are "with CA1105B3M; separate (CA1105B3M); package body CA1105B4 is ...". The with clause naming the subunit's parent unit is clearly unnecessary; none of the LRM rules (or its examples) require the with clause. Furthermore LRM 10.2, paragraph 6, says that visibility within the subunit is that which "would be obtained at the place of the corresponding body stub... if the with clauses ... of the subunit were appended to the context clause of the parent unit." Therefore, the subunit has visibility into both its parent unit and the units named by its with clauses. This creates a homographic definition in the test case at hand, since both the with'ed unit and the parent unit have the same name.

CE2110B

This test opens a single external file twice using multiple internal files FILE1 and FILE2, for both SEQUENTIAL_IO and DIRECT_IO. The DIRECT_IO test, if the opens succeed, attempts (at line 87) to delete FILE1. On a system which does not allow deletion of a file which is still open on another channel, this raises an exception. The test responds to this by attempting to close FILE2 (at line 94) and then delete FILE1 again (at line 95).

The initial attempt to delete FILE1 first closes FILE1 and then attempts to delete the external file (per LRM 14.2.1 paragraph 12). By the time the external file delete fails, the close operation has severed the connection between the internal file handle and the external file closed (per LRM 14.2.1 paragraph 9). Therefore, the second delete of FILE1 raises an unhandled exception because FILE1 is no longer associated with an external file. (Note that the SEQUENTIAL_IO test avoids this problem by avoiding the second delete of FILE1.)

End forwarded message

Message 315 3764 16 Oct 85
From: DLEHMAN@USC-ECLB.ARPA
To: FAST-REACTION@USC-ECLB.ARPA, KNAPPER@USC-ECLB.ARPA,
BWICHMANN@USC-ECLB.ARPA,
HILFINGER@UCB-VAX.ARPA, BRENDER@MARLBORO.DEC.COM,
GOODENOUGH@USC-ISI.ARPA, goodenou%wang-inst.csnet@CSNET-RELAY.ARPA,
DEWAR@NYU.ARPA, PLOEDEREDER@TL-20B.ARPA,
KEVIN.PHILLIPS%RSRE@UCL-CS.ARPA
Cc: DLEHMAN@USC-ECLB.ARPA, AHOOK@USC-ECLB.ARPA, CROBY@USC-ECLB.ARPA,
SYKESD@WPAFB-JALCF.ARPA

Subject: FRN 85-10-16

FRT Members:

Below are implementer disputes for ACVC 1.6.

---Dan
------- *

We believe the following ACVC tests are invalid and/or inapplicable to our implementation for the reasons indicated.

C48008A

At line 255, the statement "VCB := NEW TB(4)" is executed, where TB is a record type with a single (integer subtype) discriminant, and VCB has type access TB(3). The assignment therefore raises CONSTRAINT_ERROR due to a mismatch of discriminants. However, line 259 checks that the default initial value for TB's single record field was not evaluated before raising the exception, declaring failure if it was evaluated.

This test does not agree with the LRM. LRM 4.8, paragraph 6 states that evaluation of an allocator includes default initializations of the object created before returning the access value designating it. LRM 5.2, paragraph 3 states that evaluation of the LHS and RHS of an assignment statement (the target variable and the source expression) takes place before that constraint check. Therefore, the default initialization of the object designated by the new access value must occur before the CONSTRAINT_ERROR is raised, and the test is the opposite of that required by the language definition.

CA1105B4

Lines 7-9 are "with CA1105B3M; separate (CA1105B3M); package body CA1105B4 is ...". The with clause naming the subunit's parent unit is clearly unnecessary; none of the LRM rules (or its examples) require the with clause. Furthermore LRM 10.2, paragraph 6, says that visibility within the subunit is that which "would be obtained at the place of the corresponding body stub... if the with clauses ... of the subunit were appended to the context clause of the parent unit." Therefore, the subunit has visibility into both its parent unit and the units named by its with clauses. This creates a homographic definition in the test case at hand, since both the with'ed unit and the parent unit have the same name.

CE2110B

This test opens a single external file twice using multiple internal files FILE1 and FILE2, for both SEQUENTIAL_IO and DIRECT_IO. The DIRECT_IO test, if the opens succeed, attempts (at line 87) to delete FILE1. On a system which does not allow deletion of a file which is still open on another channel, this raises an exception. The test responds to this by attempting to close FILE2 (at line 94) and then delete FILE1 again (at line 95).

The initial attempt to delete FILE1 first closes FILE1 and then attempts to delete the external file (per LRM 14.2.1 paragraph 12). By the time the external file delete fails, the close operation has severed the connection between the internal file handle and the external file closed (per LRM 14.2.1 paragraph 9). Therefore, the second delete of FILE1 raises an unhandled exception because FILE1 is no longer associated with an external file. (Note that the SEQUENTIAL_IO test avoids this problem by avoiding the second delete of FILE1.)

End forwarded message

Message 319 1610 16 Oct 85
From: hilfingr@ucbrenoir.Berkeley.EDU
To: DLEHMAN@usc-eclb.arpa
Cc: FAST-REACTION@usc-eclb.arpa, KNAPPER@usc-eclb.arpa,
BWICHMANN@usc-eclb.arpa, Hilfinger@UCB-VAX.Berkeley.EDU,
BRENDER%marlboro.DEC@decwrl.ARPA, GOODENOUGH@usc-isi.arpa,
goodenou%wang-inst.csnet@csnet-relay.arpa, DEWAR@nyu.arpa,
PLOEDEREDER@tl-20b.arpa, KEVIN.PHILLIPS%RSRE@ucl-cs.arpa,
AHOOK@usc-eclb.arpa, CROBY@usc-eclb.arpa, SYKESD@wpafb-jalcf.arpa
Subject: Re: FRN 85-10-16
In-Reply-To: Your message of 16 Oct 1985 09:05-PDT.
‹[USC-ECLB.ARPA]16-Oct-85 09:05:09.DLEHMAN›

C48008A:

The implementor seems to be correct.

CA1105B4:

I thought we already ruled on this one (is there any filtering of these disputes going on at the IDA end to catch things we've discussed previously?) I recall arguing that the implementor had a good point. We discussed this at the LMC, I think, and I'll leave it to John to remember with what result.

CE2110B:
I am inclined to agree with the implementor on this point. We can argue about what the Standard says (it's vague on this point), but the behavior specified by the implementor is not entirely unreasonable.

Message 334 1503 21 Oct 85
From: KNAPPER@USC-ECLB.ARPA
To: DLEHMAN@USC-ECLB.ARPA
Cc: FAST-REACTION@USC-ECLB.ARPA, KNAPPER@USC-ECLB.ARPA,
BWICHMANN@USC-ECLB.ARPA,
HILFINGER@UCBVAX.BERKELEY.EDU,
GOODENOUGH@USC-ISI.ARPA,
goodenou%wang-inst.csnet@CSNET-RELAY.ARPA,
DEWAR@NYU.ARPA, PLOEDEREDER@TL-20B.ARPA,
KEVIN.PHILLIPS%RSRE@UCL-CS.ARPA, AHOOK@USC-ECLB.ARPA,
CROBY@USC-ECLB.ARPA, SYKESD@WPAFB-JALCF.ARPA,
brender@MARLBORO.DEC.COM
Subject: Re: FRN 85-10-16
In-Reply-To: ‹[USC-ECLB.ARPA]16-Oct-85 09:05:09.DLEHMAN›

C48008A

I agree with the implementer.

CA1105B4

I remember this as a previous dispute also. I do not however remember the LMC resolution. The implementer has a good arguement and I cannot find anything to counter it. (Paul, we have been sort of given a "no editing" directive WRT the disputes. If a test is on the withdrawn list already, I believe it isn't sent out, but as long as an implementer is challenging a test that has not yet been withdrawn we'll send it out again.)

CE2110B

My reading of the test sides me with the implementer. I believe that the second DELETE is extraneous and should be removed.The intent of the test is not changed if the DELETE is taken out. The test should be revised.

Bob Knapper
Message 338 2418 21 Oct 85
From: John B. Goodenough ‹GOODENOUGH@USC-ISI.ARPA›
To: DLEHMAN@USC-ECLB.ARPA
Cc: FAST-REACTION@USC-ECLB.ARPA, KNAPPER@USC-ECLB.ARPA,
BWICHMANN@USC-ECLB.ARPA,
HILFINGER@UCB-VAX.ARPA, BRENDER@MARLBORO.DEC.COM,
GOODENOUGH@USC-ISI.ARPA.

goodenou%wang-inst.csnet@CSNET-RELAY.ARPA, DEWAR@NYU.ARPA,
PLOEDEREDER@TL-20B.ARPA, KEVIN.PHILLIPS%RSRE@UCL-CS.ARPA,
AHOOK@USC-ECLB.ARPA, CROBY@USC-ECLB.ARPA, SYKESD@WPAFB-JALCF.ARPA
Subject: Re: FRN 85-10-16
In-Reply-To: ‹[USC-ECLB.ARPA]16-Oct-85 09:05:09.DLEHMAN›

C48008A

The implementer's argument is wrong, but his conclusion is probably acceptable. His argument is wrong because the issue of constraint checks for the assignment operation does not arise. The question is whether, when the allocator is evaluated, the ALLOCATOR raises CONSTRAINT_ERROR (because the designated object has the wrong discriminant value) before or after an object has been created and whether, if an object is created, it can be initialized. AI-00150 (which has been approved by everyone except the AJPO) says it is undefined whether this check is performed before or after creation of an object. Presumably, if an object is created, default initializations may be performed as well, although this is really a subject for interpretation. Given that a ruling has been issued saying that the time of object creation is somewhat vague, I would go along with an interpretation that the default initializations can be performed before the check is made (i.e., I would not object too strongly if the test were considered incorrect). But an LMC interpretation is needed here to make invalidating this test really kosher.

CA1105B4

AI-00113 says this test is correct. This AI has been approved by the LMC, subject to letter ballot, but I haven't heard any objection so far to the conclusions of the Commentary, so I think the AVO has a reasonable basis for approving this test. (Also this is one of the oldest tests in the suite; all validated compilers up to now pass it, and users have been observed to actually write such context clauses.)

CE2110B

I haven't had a chance yet to look over the test in detail. I'll send my response tomorrow. (The implementer's argument sounds plausible.)

Message 339 2156 22 Oct 85
From: Ron Brender ‹BRENDER@MARLBORO.DEC.COM›
To: FRTMEMS: FAST-REACTION@USC-ECLB, BABCOCK@USC-ECLB, DEWAR@NYU,
KEVIN.PHILLIPS%RSRE@UCL-CS, GAILLY@HI-MULTICS, GOODENOUGH@ISI,
goodenou%wang-inst.csnet@CSNET-RELAY, HILFINGER@BERKELEY,
AHOOK@USC-ECLB, KNAPPER@USC-ECLB, KRAMER@USC-ECLB, DLEHMAN@USC-ECLB,
PLOEDEREDER@TL-20B, PROBERT@USC-ECLB, CROBY@USC-ECLB,
SYKESD@WPAFB-JALCF, BWICHMANN@USC-ECLB;
Subject: FRNs 85-10-15 and 85-10-16

Re FRN 85-10-15

C64103A I think Goodenough is right on here. The test is okay.

A62006D, B62006C, B85007C
The tests are okay as Goodenough argues.

C94004B I agree with Goodenough. The use of 'TERMINATED is legal.

C94004A, -B, -C
I am a little surprized that John would so readily take these to the LMC as I thought this question had been resolved in an already approved AI (in favor of the test, against the implementer). But I can't find such, so will certainly go along with LMC review.

LA3004A*, -B*
The tests are okay and the implementer has acceptably passed them.

It surprizes me that this one even need go to the FRT. The behavior cited by the implementor has certainly been accepted before by the AVO in earlier validations.

CA3006D The language IS a little strange in this regard, but I remember the discussions that lead up to the wording now in the LRM. The test is certainly in line with the intent of the LRM.

B48003B, B64003A
The tests and the implementation are both okay, though I too have complained that the comments in B64003A in particular need improvement.

Re FRN 85-10-16

C48008A The implementer has a point. LMC review is in order.

CA1105B This has been discussed by the LMC, with a result that supports the test as is.

CE2110B The implementer is correct. The test needs revision to more delicately clean up after the possible exceptions.

Message 350 959 23 Oct 85
From: dewar@NYU-ACF2.ARPA
To: AHOOK@USC-ECLB, BABCOCK@USC-ECLB, BRENDER@MARLBORO.DEC.COM,
BWICHMANN@USC-ECLB;, CROBY@USC-ECLB, DEWAR@NYU, DLEHMAN@USC-ECLB,
FRTMEMS:FAST-REACTION@USC-ECLB, GAILLY@HI-MULTICS, GOODENOUGH@ISI,
KEVIN.PHILLIPS%RSRE@UCL-CS, KNAPPER@USC-ECLB, KRAMER@USC-ECLB,
PLOEDEREDER@TL-20B, PROBERT@USC-ECLB, SYKESD@WPAFB-JALCF,
HILFINGER@berkeley, goodenou%wang-inst.csnet@CSNET-RELAY
Subject: Re: FRNs 85-10-15 and 85-10-16

I support Ron Brender's comments on these tests (which if I remember all my mail correctly, means that I also support John Goodenough's comments!)

Message 366 4506 28 Oct 85
From: DLEHMAN@USC-ECLB.ARPA
To: BABCOCK@USC-ECLB.ARPA
Cc: KNAPPER@USC-ECLB.ARPA, AHOOK@USC-ECLB.ARPA, DLEHMAN@USC-ECLB.ARPA
Subject: THREE DISPUTES--C48008A, CA1105B, & CE2110B

Dave:

I am sorry for the delay in submitting the disputes of yours to the FRT --Bob forwarded them to me promptly, but since they were not disputes from a validatee, they were overlooked (too long) before getting looked over. Anyway, forwarded are the substantive responses received re your disputes.

---Dan LEHMAN, IDA
------ *

Begin forwarded messages
Subject: Re: FRN 85-10-16
In-Reply-To: Your message of 16 Oct 1985 09:05-PDT.
‹[USC-ECLB.ARPA]16-Oct-85 09:05:09.DLEHMAN›

C48008A:

The implementor seems to be correct.

CA1105B4:

I thought we already ruled on this one (is there any filtering of these disputes going on at the IDA end to catch things we've discussed previously?) I recall arguing that the implementor had a good point. We discussed this at the LMC, I think, and I'll leave it to John to remember with what result.

CE2110B:
I am inclined to agree with the implementor on this point. We can argue about what the Standard says (it's vague on this point), but the behavior specified by the implementor is not entirely unreasonable.

Subject: Re: FRN 85-10-16
In-Reply-To: ‹[USC-ECLB.ARPA]16-Oct-85 09:05:09.DLEHMAN›

C48008A

I agree with the implementer.

CA1105B4

I remember this as a previous dispute also. I do not however remember the LMC resolution. The implementer has a good arguement and I cannot find anything to counter it.

CE2110B

My reading of the test sides me with the implementer. I believe

that the second DELETE is extraneous and should be removed.The intent of the test is not changed if the DELETE is taken out. The test should be revised.

Subject: Re: FRN 85-10-16
In-Reply-To: ‹[USC-ECLB.ARPA]16-Oct-85 09:05:09.DLEHMAN›

C48008A

The implementer's argument is wrong, but his conclusion is probably acceptable. His argument is wrong because the issue of constraint checks for the assignment operation does not arise. The question is whether, when the allocator is evaluated, the ALLOCATOR raises CONSTRAINT_ERROR (because the designated object has the wrong discriminant value) before or after an object has been created and whether, if an object is created, it can be initialized. AI-00150 (which has been approved by everyone except the AJPO) says it is undefined whether this check is performed before or after creation of an object. Presumably, if an object is created, default initializations may be performed as well, although this is really a subject for interpretation. Given that a ruling has been issued saying that the time of object creation is somewhat vague, I would go along with an interpretation that the default initializations can be performed before the check is made (i.e., I would not object too strongly if the test were considered incorrect). But an LMC interpretation is needed here to make invalidating this test really kosher.

CA1105B4

AI-00113 says this test is correct. This AI has been approved by the LMC, subject to letter ballot, but I haven't heard any objection so far to the conclusions of the Commentary, so I think the AVO has a reasonable basis for approving this test. (Also this is one of the oldest tests in the suite; all validated compilers up to now pass it, and users have been observed to actually write such context clauses.)

CE2110B

I haven't had a chance yet to look over the test in detail. I'll send my response tomorrow. (The implementer's argument sounds plausible.)

[No response was received; concurrence was presumed. --DL]

Re FRN 85-10-16

C48008A **The implementer has a point. LMC review is in order.**

CA1105B **This has been discussed by the LMC, with a result that supports the test as is.**

C32110B **The implementer is correct. The test needs revision to more delicately clean up after the possible exceptions.**

End forwarded messages

Message 368 434 29 Oct 85
From: BABCOCK@USC-ECLB.ARPA
To: DLEHMAN@USC-ECLB.ARPA
Subject: Re: THREE DISPUTES--C48008A, CA1105B, & CE2110B
In-Reply-To: ‹[USC-ECLB.ARPA]28-Oct-85 08:01:54.DLEHMAN›

Thanks for the info.
Dave Babcock
ROLM Mil-Spec Computers
›

Message 336 774 27 Nov 85
From: DLEHMAN@USC-ISIF.ARPA
To: BABCOCK@USC-ISIF.ARPA
Cc: DLEHMAN@USC-ISIF.ARPA
Subject: C48008A

Dave:

You raised the dispute re this test, so I thought that you'd like to know that the November LMC meeting concluded that this test should be withdrawn (from 1.7, at this time). Referenced is AI-00397/01.

Tests C94004A, --B, --C WERE ALSO RULED INCORRECT (subject to a letter ballot (which could reverse this decision)), leaving the behavior of an Ada program undefined after termination of the main program (AI-00399/01).

---Dan
------- *

Message 246 4271 1 Nov 85
From: DLEHMAN@USC-ECLB.ARPA
To: FAST-REACTION@USC-ECLB.ARPA, KNAPPER@USC-ECLB.ARPA, AHOOK@USC-ECLB.ARPA,
    DEWAR@NYU.ARPA, HILFINGER@UCBVAX.BERKELEY.EDU, BWICHMANN@USC-ECLB.ARPA,
    goodeonu%wang-inst.csnet@CSNET-RELAY.ARPA,
    PLOEDEREDER@TL-20B.ARPA, GOODENOUGH@USC-ISI.ARPA, CROBY@USC-ECLB.ARPA
Cc: DLEHMAN@USC-ECLB.ARPA,
    SYKESD@WPAFB-JALCF.ARPA
Subject: FRN 85-11-01

FRT Members:

Two disputes have been received; they are given below. Re C64103A, this test has been considered earlier for a similar dispute concerning line 66; Paul's response to that (cryptic) dispute found a possible error (i.e., for some implementations), but the previous disputer dropped the dispute and passed the test. (Ref. FRN of 9 Aug, response of 16 Aug). This current dispute seems rather cryptic, too.

---Dan
------- *

CA5004B-B

The REPORT package must contain a PRAGMA ELABORATE for TEXT_IO before any of the TEXT_IO facilities are invoked in this test's execution.

C64103A-B

In lines 65-67 and 155-157 the NUMERIC_ERROR execption need not be raised, sinc the values LG_FLOAT'LARGE and 0.1 are within the range of the selected predefined base type LONG_FLOAT for LARGE and SMALL, resp..

[pertinent sections of the test are included below--DL]

-- C64103A-B.ADA

-- CHECK THAT THE APPROPRIATE EXCEPTION IS RAISED AS REQUIRED FOR TYPE
-- CONVERSIONS ON IN OUT SCALAR VARIABLES. IN PARTICULAR:
--   (A) NUMERIC_ERROR IS RAISED FOR NUMERIC TYPES BEFORE THE CALL WHEN
--   THE ACTUAL VALUE IS OUTSIDE THE RANGE OF THE FORMAL PARAMETER'S
--   BASE TYPE.
--   (B) NUMERIC_ERROR IS RAISED FOR NUMERIC TYPES AFTER THE CALL WHEN
--   THE FORMAL PARAMETER'S VALUE IS OUTSIDE THE RANGE OF THE ACTUAL
--   VARIABLE'S BASE TYPE.

-- CPP 7/2/84

```
---
. . .
---
          DECLARE
               TYPE SM_FLOAT IS DIGITS 1;
               TYPE LG_FLOAT IS DIGITS SYSTEM.MAX_DIGITS;
               LARGE : LG_FLOAT := LG_FLOAT'LARGE;

               PROCEDURE P2 (X : IN OUT SM_FLOAT) IS
               BEGIN
                    FAILED ("EXCEPTION NOT RAISED BEFORE CALL -P2 (A)");
               END P2;
          BEGIN
               COMMENT ("CHECK INPUT OF FLOATING POINT TYPES (A)");
               IF LG_FLOAT (SM_FLOAT'BASE'LARGE) < LG_FLOAT'BASE'LARGE
                    THEN
                         P2 (SM_FLOAT (LARGE));
               ELSE
                    COMMENT ("NOT APPLICABLE -P2 (A)");
               END IF;
          EXCEPTION
               WHEN NUMERIC_ERROR =>
                    NULL;
               WHEN CONSTRAINT_ERROR =>
                    FAILED ("CONSTRAINT_ERROR RAISED INSTEAD OF " &
                            "NUMERIC_ERROR -P2 (A)");
               WHEN OTHERS =>
                    FAILED ("WRONG EXCEPTION RAISED -P2 (A)");
          END;
---
. . .
---
          DECLARE
               TYPE SM_FLOAT IS DIGITS 1;
               TYPE LG_FLOAT IS DIGITS SYSTEM.MAX_DIGITS;
               SMALL : SM_FLOAT := 0.1;
               LARGE : LG_FLOAT := LG_FLOAT'LARGE;

               PROCEDURE P2 (X : IN OUT LG_FLOAT) IS
               BEGIN
                    X := LARGE;
               END P2;
          BEGIN
               COMMENT ("CHECK OUTPUT OF FLOATING POINT TYPES (B)");
               IF LG_FLOAT (SM_FLOAT'BASE'LARGE) <
                    LG_FLOAT'BASE'LARGE THEN
                         P2 (LG_FLOAT (SMALL));
                         FAILED ("EXCEPTION NOT RAISED AFTER CALL " &
                                 "-P2 (B)");
               ELSE
                    COMMENT ("NOT APPLICABLE -P2 (B)");
               END IF;
          EXCEPTION
               WHEN NUMERIC_ERROR =>
```

```
                    NULL;
               WHEN CONSTRAINT_ERROR =>
                    FAILED ("CONSTRAINT_ERROR RAISED INSTEAD OF " &
                            "NUMERIC_ERROR -P2 (B)");
               WHEN OTHERS =>
                    FAILED ("WRONG EXCEPTION RAISED -P2 (B)");
          END;
```

Message 247 1544 2 Nov 85
From: hilfingr%renoir@BERKELEY.EDU
To: DLEHMAN@usc-eclb.arpa
Cc: FAST-REACTION@usc-eclb.arpa, KNAPPER@usc-eclb.arpa, AHOOK@usc-eclb.arpa,
DEWAR@nyu.arpa, HILFINGER@ucb-vax.berkeley.edu,
BWICHMANN@usc-eclb.arpa, goodeonu%wang-inst.csnet@csnet-relay.arpa,
PLOEDEREDER@tl-20b.arpa, GOODENOUGH@usc-isi.arpa,
CROBY@usc-eclb.arpa,
SYKESD@wpafb-jalcf.arpa, hilfingr@renoir.berkeley.edu
Subject: Re: FRN 85-11-01
In-Reply-To: Your message of 1 Nov 1985 17:06-PST.
‹[USC-ECLB.ARPA] 1-Nov-85 17:06:10.DLEHMAN›

CA5004B-B

This is not, strictly speaking, a dispute, since the REPORT package is not precisely a part of the validation suite. Modifications of REPORT such as requested are perfectly in order.

C64103A-B

This test seems cuckoo to me. Since when does

    LG_FLOAT (SM_FLOAT'BASE'LARGE) ‹ LG_FLOAT'BASE'LARGE

imply (according to the Standard) that SM_FLOAT(LG_FLOAT'LARGE) is not in the range of SM_FLOAT?

I don't understand the implementor's comment either. He seems to think that it matters that LARGE and SMALL are within the base type for LG_FLOAT. That is irrelevant, of course.

Paul

Message 249 844 3 Nov 85
From: John B. Goodenough ‹GOODENOUGH@USC-ISI.ARPA›
To: DLEHMAN@USC-ECLB.ARPA
Cc: FAST-REACTION@USC-ECLB.ARPA, KNAPPER@USC-ECLB.ARPA, AHOOK@USC-ECLB.ARPA,
DEWAR@NYU.ARPA, HILFINGER@UCBVAX.BERKELEY.EDU, BWICHMANN@USC-ECLB.ARPA,
goodeonu%wang-inst.csnet@CSNET-RELAY.ARPA, PLOEDEREDER@TL-20B.ARPA,
GOODENOUGH@USC-ISI.ARPA, CROBY@USC-ECLB.ARPA, SYKESD@WPAFB-JALCF.ARPA
Subject: Re: FRN 85-11-01
In-Reply-To: ‹[USC-ECLB.ARPA] 1-Nov-85 17:06:10.DLEHMAN›

C64103A

The implementer is correct. The test should be referencing 'LAST rather than 'LARGE.

CA5004B-B

I agree with Paul. The requested modification of the REPORT package is allowed under test procedures.

JBG

Message 259 859 6 Nov 85
From: BWICHMANN@USC-ECLB.ARPA
To: DLEHMAN@USC-ECLB.ARPA
Subject: C64103A-B

Please forward as necessary. (You could perhaps tell me how to forward directly. Is sending a message to FAST-REACTION@USC-ECLB.ARPA sufficient?)

C64103A-B.ADA

The test is indeed incorrect. Since NUMERIC_ERROR need never be raised, a major change is needed to execute the tests only if MACHINE_OVERFLOWS is true. If one then replaces LARGE by SAFE_LARGE the test may be correct! Note that the use of LAST is not satisfactory since this could be an 'infinite' value as with some modes in the IEEE standard. I would be happy to review any revision of this test (not all 500 that might be revised!).

Brian Wichmann.

Message 280 750 12 Nov 85
From: DLEHMAN@USC-ECLB.ARPA
To: FSTC-AVF@USC-ECLB.ARPA
Cc: AHOOK@USC-ECLB.ARPA, DLEHMAN@USC-ECLB.ARPA
Subject: HONEYWELL GCOS6 DISPUTES

John:

Test C64103A-B is ruled inapplicable for the reasons given in the previous Honeywell dispute of this test(the Big Systems people gave an intelligible reason; the GCOS6 argument was not to the point, presuming that they have the same trouble as their colleagues).

The REPORT pacakage(-package) may be modified with pragma ELABORATE as necessary for CA5004B-B.

---DAN

Message 250 2402 4 Nov 85
From: ima!inmet!ada-uts!wkb%cca-unix.arpa@cca-unix.arpa
To: hillm@WPAFB-JALCF.ARPA
Cc: dlehman@USC-ECLB.ARPA
Subject: ACVC CA5004B dispute

Received: by inmet.uucp (4.12/inmet) id AA21986; Mon, 4 Nov 85 11:06:08 est
Date: Mon, 4 Nov 85 11:06:08 est
Message-Id: ‹8511041606.AA21986@inmet.uucp›

Nov 4, 1985

Warren Berger
Intermetrics, Inc.

Mr. Michael Hill
HILLM@WPAFB-JALCF

Dear Mr. Hill,

As we discussed, Intermetrics anticipates the validation of our AIE compiler by December 1985. We are currently using Version 1.6 of the ACVC tests to debug the compiler. To avoid time consuming resolution of disputes at pre-validation time, we would like to begin to resolve some of our exceptions with certain ACVC tests as soon as possible.

We believe that the ACVC test CA5004B-B.ADA from Test Suite Version 1.6 is incorrect because there exists a legal elaboration order that causes PROGRAM_ERROR to be raised.

The objective of the test is:

```
-- CA5004B-B.ADA
-- CHECK THAT PRAGMA ELABORATE IS ACCEPTED AND OBEYED EVEN IF THE UNIT
--     NAMED IN THE PRAGMA DOES NOT YET HAVE A BODY IN THE LIBRARY OR IF
--     ITS BODY IS OBSOLETE.
```

Our compiler raises PROGRAM_ERROR when REPORT.TEST is called from the HEADER package body. The package body has a pragma ELABORATE (REPORT), but REPORT does not pragma ELABORATE TEXT_IO. The PROGRAM_ERROR exception is raised because the body of TEXT_IO had not been elaborated when the REPORT package body tried to use one of its routines.

We believe that there is a legal sequence of elaborations that raises PROGRAM_ERROR and the test doesn't allow for this.
Please let us know of your decision. Thank you.

Sincerely,

Warren Berger
Intermetrics, Inc.
Ada Systems Division
733 Concord Ave
Cambridge, MA 02146
(617) 661-1840

Arpanet: ima!inmet!ada-uts!wkb@CCA-UNIX.ARPA
Message 251 1641 4 Nov 85
From: hillm@wpafb-jalcf
To: DLEHMAN@USC-ECLB
Cc: SYKESD@WPAFB-JALCF
Subject: Dispute of CA5004B-B.ADA

Dan,

You should have received a copy of this dispute but I am forwarding anyway.

Please forward to the FRT.

Thanks,

Mike Hill
AVF-WPAFB/SofTech Inc.
^L
We believe that the ACVC test CA5004B-B.ADA from Test Suite Version 1.6 is incorrect because there exists a legal elaboration order that causes PROGRAM_ERROR to be raised.

The objective of the test is:
-- CA5004B-B.ADA
-- CHECK THAT PRAGMA ELABORATE IS ACCEPTED AND OBEYED EVEN IF THE UNIT
-- NAMED IN THE PRAGMA DOES NOT YET HAVE A BODY IN THE LIBRARY OR IF
-- ITS BODY IS OBSOLETE.

Our compiler raises PROGRAM_ERROR when REPORT.TEST is called from the HEADER package body. The package body has a pragma ELABORATE (REPORT), but REPORT does not pragma ELABORATE TEXT_IO. The PROGRAM_ERROR exception is raised because the body of TEXT_IO had not been elaborated when the REPORT package body tried to use one of its routines.

We believe that there is a legal sequence of elaborations that raises PROGRAM_ERROR and the test doesn't allow for this.
Please let us know of your decision. Thank you.

Sincerely,

Warren Berger
Intermetrics, Inc.
Ada Systems Division
733 Concord Ave
Cambridge, MA 02146
(617) 661-1840
Arpanet: ima!inmet!ada-uts!wkb@CCA-UNIX.ARPA
Message 253 4556 4 Nov 85
From: ima!inmet!ada-uts!wkb%cca-unix.arpa@cca-unix.arpa
To: hillm@WPAFB-JALCF.ARPA
Cc: dlehman@USC-ECLB.ARPA
Subject: ACVC CA5004B dispute

Received: by inmet.uucp (4.12/inmet) id AA04535; Mon, 4 Nov 85 18:26:00 est
Date: Mon, 4 Nov 85 18:26:00 est
Message-Id: ‹8511042326.AA04535@inmet.uucp›

Nov 4, 1985

Warren Berger
Intermetrics, Inc.

Mr. Michael Hill
HILLM@WPAFB-JALCF

Dear Mr. Hill,

A previous letter from me earlier today described the problem that we have with ACVC test CA5004B. Basically, our compiler raises PROGRAM_ERROR because the test does not force TEXT_IO to be elaborated before the package body HEADER in the test. We have modified the test to force the elaboration of all necessary I/O packages and the modified version now passes. Is the modification acceptable? We definitely prefer a solution to the problem which withdraws the test or which fixes it, to the addition of pragmas to REPORT and/or to our Run-Time system.

Please get back to me as soon as possible. Thank you.
The modified test follows.

Sincerely,

Warren Berger
Intermetrics, Inc.
Ada Systems Division
733 Concord Ave
Cambridge, MA 02146
(617) 661-1840
Arpanet: ima!inmet!ada-uts!wkb@CCA-UNIX.ARPA

```
-- CA5004B-B.ADA

-- CHECK THAT PRAGMA ELABORATE IS ACCEPTED AND OBEYED EVEN IF THE UNIT
--    NAMED IN THE PRAGMA DOES NOT YET HAVE A BODY IN THE LIBRARY OR IF
--    ITS BODY IS OBSOLETE.
-- CHECK THAT MORE THAN ONE NAME IS ALLOWED IN A PRAGMA ELABORATE.

-- BHS 8/03/84
-- JRK 9/20/84

-------------------------------------------------------------------

PACKAGE HEADER IS

     PROCEDURE WRONG (WHY : STRING);

END HEADER;
```

```
with text_io, unix_io, char_io;
WITH REPORT; USE REPORT;
PRAGMA ELABORATE (REPORT);
pragma elaborate (text_io, unix_io, char_io);
PACKAGE BODY HEADER IS

     PROCEDURE WRONG (WHY : STRING) IS
     BEGIN
          FAILED ("PACKAGE WITH " & WHY & " NOT ELABORATED " &
                  "CORRECTLY");
     END WRONG;

BEGIN

     TEST ("CA5004B", "PRAGMA ELABORATE IS ACCEPTED AND OBEYED " &
                      "EVEN WHEN THE BODY OF THE UNIT NAMED IS " &
                      "MISSING OR OBSOLETE");

END HEADER;

--------------------------------------------------------------------

PACKAGE CA5004B0 IS

     I : INTEGER := 1;

     FUNCTION F RETURN BOOLEAN;

END CA5004B0;


PACKAGE BODY CA5004B0 IS

     FUNCTION F RETURN BOOLEAN IS
     BEGIN
          RETURN TRUE;
     END F;

END CA5004B0;

--------------------------------------------------------

PACKAGE CA5004B0 IS          -- OLD BODY NOW OBSOLETE.

     I : INTEGER := 2;
     B : BOOLEAN := TRUE;

     FUNCTION F RETURN BOOLEAN;

END CA5004B0;

--------------------------------------------------------
```

```
PACKAGE CA5004B1 IS

     J : INTEGER := 3;

     PROCEDURE P (X : INTEGER);

END CA5004B1;                    -- NO BODY GIVEN YET.

-----------------------------------------------------------

WITH HEADER; USE HEADER;
WITH CA5004B0, CA5004B1;
USE CA5004B0, CA5004B1;
PRAGMA ELABORATE (HEADER, CA5004B0, CA5004B1);
PACKAGE CA5004B2 IS

        WRONG ("NO BODY");
     END IF;

END CA5004B2;

-----------------------------------------------------

WITH REPORT, CA5004B2;
USE REPORT, CA5004B2;
PROCEDURE CA5004B IS
BEGIN

     RESULT;

END CA5004B;

-----------------------------------------------------

PACKAGE BODY CA5004B0 IS

     FUNCTION F RETURN BOOLEAN IS
     BEGIN
          RETURN FALSE;
     END F;

BEGIN

     I := 4;

END CA5004B0;

-----------------------------------------------------

PACKAGE BODY CA5004B1 IS

     PROCEDURE P (X : INTEGER) IS
     BEGIN
          NULL;
```

```
     END P;

BEGIN

     J := 5;

END CA5004B1;
```

Message 256 2724 5 Nov 85
From: DLEHMAN
To: FAST-REACTION@USC-ECLB.ARPA, KNAPPER@USC-ECLB.ARPA,
    HILFINGER@UCBVAX.BERKELEY.EDU,
    BRENDER@MARLBORO.DEC.COM, GOODENOUGH@USC-ISI.ARPA,
    goodenou%wang-inst.csnet@CSNET-RELAY.ARPA,
    DEWAR@NYU.ARPA, BWICHMANN@USC-ECLB.ARPA,
    PLOEDEREDER@TL-20B.ARPA
Cc: dlehman@USC-ECLB.ARPA,
    AHOOK@USC-ECLB.ARPA, CROBY@USC-ECLB.ARPA,
    SYKESD@WPAFB-JALCF.ARPA
Subject: ACVC CA5004B dispute

FRT Members:

A second implementer has had problems with the order of elaboration with CA5004B-B. The implementer requests that a modification TO THE TEST be allowed, instead of to pkg. REPORT. Is the implementer's request permissable?

I have included part of the implementer's message; I highlighted the implementer's modifications with asterisks.

---Dan
------- *

A previous letter from me earlier today described the problem that we have with ACVC test CA5004B. Basically, our compiler raises PROGRAM_ERROR because the test does not force TEXT_IO to be elaborated before the package body HEADER in the test. We have modified the test to force the elaboration of all necessary I/O packages and the modified version now passes. Is the modification acceptable? We definitely prefer a solution to the problem which withdraws the test or which fixes it, to the addition of pragmas to REPORT and/or to our Run-Time system.

```
-- CA5004B-B.ADA

-- CHECK THAT PRAGMA ELABORATE IS ACCEPTED AND OBEYED EVEN IF THE UNIT
--    NAMED IN THE PRAGMA DOES NOT YET HAVE A BODY IN THE LIBRARY OR IF
--    ITS BODY IS OBSOLETE.
-- CHECK THAT MORE THAN ONE NAME IS ALLOWED IN A PRAGMA ELABORATE.

-- BHS 8/03/84
```

```
-- JRK 9/20/84

-----------------------------------------------------------

PACKAGE HEADER IS

     PROCEDURE WRONG (WHY : STRING);

END HEADER;


with text_io, unix_io, char_io; * * * * * * * *  * * * * * * * * * * * * * *
WITH REPORT; USE REPORT;
PRAGMA ELABORATE (REPORT);
pragma elaborate (text_io, unix_io, char_io); * * * * * * * * * * ** * * *
PACKAGE BODY HEADER IS

     PROCEDURE WRONG (WHY : STRING) IS
     BEGIN
          FAILED ("PACKAGE WITH " & WHY & " NOT ELABORATED " &
                  "CORRECTLY");
     END WRONG;

BEGIN

     TEST ("CA5004B", "PRAGMA ELABORATE IS ACCEPTED AND OBEYED " &
                      "EVEN WHEN THE BODY OF THE UNIT NAMED IS " &
                      "MISSING OR OBSOLETE");

END HEADER;

-----------------------------------------------------------------------

PACKAGE CA5004B0 IS
Message 257  1384  5 Nov 85
From: John B. Goodenough ‹GOODENOUGH@USC-ISI.ARPA›
To: DLEHMAN@USC-ECLB.ARPA
Cc: GOODENOUGH@USC-ISI.ARPA
Subject: Re: ACVC CA5004B dispute
In-Reply-To: ‹[USC-ECLB.ARPA] 5-Nov-85 16:19:17.DLEHMAN›

I don't see why the test modification should be considered acceptable.
The body of HEADER does not use any text_io functions, so it should not
be necessary to say WITH TEXT_IO (or anything else).  Moreover, if the
pragma ELABORATE is being obeyed correctly, and if the body of the
REPORT package contains PRAGMA ELABORATE(TEXT_IO), then the body of TEXT_IO
should be elaborated before the body of the report package, which should be
elaborated before the body of HEADER, so the addition of PRAGMA
ELABORATE (TEXT_IO) in the HEADER body should have no effect at all.
There is, of course, even less reason for requiring non-standard packages
like unix_IO to be named in a context clause or an elaborate pragma.

In short, given the modification of the report package body, I see no
reason to accede to the implementer's request, or to withdraw the test.
```

If in fact, the TEXT_IO body uses UNIX_IO and CHAR_IO, then the TEXT_IO body had better have a pragma ELABORATE for these units to ensure against program_error.

Message 258 1362 6 Nov 85
From: hilfingr%renoir@BERKELEY.EDU (Paul Hilfinger)
To: DLEHMAN@usc-eclb.arpa
Cc: FAST-REACTION@usc-eclb.arpa, KNAPPER@usc-eclb.arpa,
HILFINGER@ucb-vax.berkeley.edu, BRENDER@marlboro.dec.com,
GOODENOUGH@usc-isi.arpa, goodenou%wang-inst.csnet@csnet-relay.arpa,
DEWAR@nyu.arpa, BWICHMANN@usc-eclb.arpa, PLOEDEREDER@tl-20b.arpa,
AHOOK@usc-eclb.arpa, CROBY@usc-eclb.arpa, SYKESD@wpafb-jalcf.arpa
Subject: Re: ACVC CA5004B dispute
In-Reply-To: Your message of 5 Nov 1985 16:19-PST.
‹[USC-ECLB.ARPA] 5-Nov-85 16:19:17.DLEHMAN›

Re: CA5004B-B:

Why does the implementor prefer not to put extra pragmas on REPORT? Why does he need the lower case lines? Why isn't the pragma ELABORATE(REPORT) sufficient, in combination with pragmas on REPORT? Barring reasonable explanations of these, I am not in favor of allowing the addition of the lower case lines the implementor has specified; their potential semantic change is too large.

Paul
Message 261 981 6 Nov 85
From: KNAPPER@USC-ECLB.ARPA
To: DLEHMAN@USC-ECLB.ARPA
Cc: FAST-REACTION@USC-ECLB.ARPA, KNAPPER@USC-ECLB.ARPA,
HILFINGER@UCBVAX.BERKELEY.EDU,
GOODENOUGH@USC-ISI.ARPA,
goodenou%wang-inst.csnet@CSNET-RELAY.ARPA,
DEWAR@NYU.ARPA, BWICHMANN@USC-ECLB.ARPA,
PLOEDEREDER@TL-20B.ARPA, AHOOK@USC-ECLB.ARPA,
CROBY@USC-ECLB.ARPA, SYKESD@WPAFB-JALCF.ARPA
Subject: Re: ACVC CA5004B dispute
In-Reply-To: ‹[USC-ECLB.ARPA] 5-Nov-85 16:19:17.DLEHMAN›

I find it unreasonable to modify an executable test in any fashion other than to perhaps split it because of a capacity limitation or some for some other truely implementation dependent reason. The modification of REPORT is much prefered and I believe should be the course taken here.

Bob K
Message 264 991 7 Nov 85
From: DLEHMAN@USC-ECLB.ARPA
To: HILLM@WPAFB-JALCF.ARPA, SYKESD@WPAFB-JALCF.ARPA,
CHITWOODG@WPAFB-JALCF.ARPA

Cc: AHOOK@USC-ECLB.ARPA, DLEHMAN@USC-ECLB.ARPA
Subject: CA5004B & VAR_STRINGS

Dear Mike, Dave, & Georgeanne:

Re the recent questions raised by Intermetrics, we have decided that --without further justification to do otherwise--CA5004B should NOT be altered, that instead package REPORT should contain the necessary PRAGMA to elaborate the needed IO packages. I know that some of the FRT comments have been forwarded to Intermetrics, and they likely anticipate our decision; perhaps they have further arguments.

The VAR_STRINGS package may be considered inapplicable.

---Dan
------- *

p.s.: --received the FSU/AFATL VSRs this a.m., thanks, DL **

Message 266 845 8 Nov 85
From: sykesd@wpafb-jalcf
To: dlehman@USC-ECLB
Cc: SYKESD@WPAFB-JALCF
Subject: CA5004B

Dan,

Intermetrics feels that test CA5004B is not correct and is not sure this was considered in the last FRT review. Please submit the following arguments.

Intermetrics chooses an elaboration order that elaborates TEXT_IO body after the package body of PACKAGE HEADER. Intermetrics states that no elaboration order is required by the LRM and that the elaboration order that they chose does not force the elaboration of the TEXT_IO body at any time.

Therefore Intermetrics claims that their results of a PROGRAM_ERROR is legal and the test CA5004B is illegal.

Thanks,
Mike Hill

Message 288 3151 14 Nov 85 ANSWERED
From: hillm@wpafb-jalcf
To: DLEHMAN@USC-ECLB
Cc: SYKESD@WPAFB-JACLF
Subject: CA5004B dispute

Dan,

A message from Intermetrics follows which shows how they handled CA5004B. Is this acceptable?

Thanks.
Mike Hill
^L
From: ima!inmet!ada-u 13-NOV-1985 17:16:41
To: HILLM
Subj: (Network Mail)

Return-Path: ‹ima!inmet!ada-uts!wkb%cca-unix.arpa@cca-unix.arpa›
Received: from CCA-UNIX.ARPA by wpafb-jalcf ; 13 Nov 85 17:16:31 EST
Received: by CCA-UNIX.ARPA (4.12/4.7)
id AA26500; Wed, 13 Nov 85 17:04:10 est
Date: Wed, 13 Nov 85 17:04:10 est
From: ima!inmet!ada-uts!wkb%cca-unix.arpa@cca-unix.arpa
Message-Id: ‹8511132204.AA26500@CCA-UNIX.ARPA›
Sender: ada-uts!wkb%cca-unix.arpa@cca-unix.arpa
Subject: CA5004B dispute
To: hillm@WPAFB-JALCF.ARPA

Received: by inmet.uucp (4.12/inmet) id AA25668; Wed, 13 Nov 85 00:19:50 est
Date: Wed, 13 Nov 85 00:19:50 est
Message-Id: ‹8511130519.AA25668@inmet.uucp›

Nov 12, 1985

Warren Berger
Intermetrics, Inc.

Mr. Michael Hill
HILLM@WPAFB-JALCF

Dear Mr. Hill,

We still believe that CA5004B is incorrect and that it should be withdrawn, however, we are able to demonstrate that our compiler passes the objective of the test. We do this by adding a pragma elaborate statement to the Report Package body (and a related WITH statement.) In this demonstration, we have added the two lines to the Report Package body, and for convenience, have changed "REPORT" to "REPORT2" in the report package specification and body, and in the source of the test. When these three units are compiled, and then CA5004B is linked and executed, the test passes.

Here is the modified part of the Report Package, note that the lower case is just to highlight the changes:

```
-- REPORT2_BODY-B.ADA
WITH TEXT_IO;
USE TEXT_IO;
with unix_io, char_io;
pragma elaborate (text_io, unix_io, char_io);
PACKAGE BODY REPORT2 IS
```

The additional "with"ed units are required since we must force the elaboration of all necessary I/O packages, not just TEXT_IO. We could avoid these "with"s by adding pragmas to the Run-time System, however, their addition would require a recompilation of our entire I/O Run-time System which we understand would require us to rerun all the

ACVC's. Since both the recompilation and the ACVC rerun would require substantial resources and since we believe the test is incorrect, we currently do not intend to take any action that would force this recompilation and rerun to occur.

Will the above modification acceptably demonstrate that we pass the objective of the test?

Please get back to me as soon as possible. Thank you.

Sincerely,

Warren Berger
Intermetrics, Inc.
Ada Systems Division
733 Concord Ave
Cambridge, MA 02146
(617) 661-1840
Arpanet: ima!inmet!ada-uts!wkb@CCA-UNIX.ARPA

55

Message 301 1419 19 Nov 85
From: sykesd@wpafb-jalcf
To: DLEHMAN@USC-ECLB
Cc: SYKESD@WPAFB-JALCF, CHITWOODG@WPAFB-JALCF
Subject: Intermetrics pre-val problems

Dan,

Here is the wording of the info we got from Intermetrics:

With the exception of test BA1101C2m, all the tests were modified and rerun to demonstrate that additional errors from the compiler were results of cascading errors. Test BA1101C2M had separate piece ba1101c4 split into two pieces, BA1101C4-X.ADA and BA1101C4-Y.ADA, because of the way we process compilation units in the same file that have the same name. In this case, the second unit in file BA1101C4.ADA is named exactly the same as the first (error containing) unit, so the second unit is inserted into the library with that name. The fact that the first unit with the same name contained an error is no longer available when the status of the compilation is reported. When the file is split into two units compiled separately, the test passes.

There is also the issue of truncating file names that are too long. As I told you on the phone, they volunteered this piece of info. It was not evident that they were doing the truncation from the results that we received.

Thanks for your help. If you need me, just call.

Dave Sykes.
Message 307 1647 20 Nov 85
From: DLEHMAN@USC-ECLB.ARPA
To: FAST-REACTION@USC-ECLB.ARPA, KNAPPER@USC-ECLB.ARPA, HILFINGER@UCBVAX.BERKELEY.EDU, BRENDER@MARLBORO.DEC.COM, GOODENOUGH@USC-ISI.ARPA, goodenou%wang-inst.csnet@CSNET-RELAY.ARPA, DEWAR@NYU.ARPA, PLOEDEREDER@TL-20B.ARPA, BWICHMANN@USC-ECLB.ARPA, DANCY@HI-MULTICS.ARPA
Cc: AHOOK@USC-ECLB.ARPA, DLEHMAN@USC-ECLB.ARPA, CROBY@USC-ECLB.ARPA, SYKESD@WPAFB-JALCF.ARPA

FRT Members:

Please welcome Henry (Henri?) DANCY, of Alsys, to the FRT. He will be assisted by Mike WOODGER (at the same NET address).

Below is a dispute concerning the "split" processing of a "B" test. (--of ACVC 1.6)

---Dan

------- *

With the exception of test BA1101C2m, all the tests were modified and rerun to demonstrate that additional errors from the compiler were results of cascading errors. Test BA1101C2M had separate piece ba1101c4 split into two pieces, BA1101C4-X.ADA and BA1101C4-Y.ADA, because of the way we process compilation units in the same file that have the same name. In this case, the second unit in file BA1101C4.ADA is named exactly the same as the first (error containing) unit, so the second unit is inserted into the library with that name. The fact that the first unit with the same name contained an error is no longer available when the status of the compilation is reported. When the file is split into two units compiled separately, the test passes.

Message 309 1800 20 Nov 85
From: DLEHMAN@USC-ECLB.ARPA
To: FAST-REACTION@USC-ECLB.ARPA, KNAPPER@USC-ECLB.ARPA,
HILFINGER@UCBVAX.BERKELEY.EDU,
GOODENOUGH@USC-ISI.ARPA, goodenou%wang-inst.csnet@CSNET-RELAY.ARPA,
DEWAR@NYU.ARPA, PLOEDEREDER@TL-20B.ARPA, BWICHMANN@USC-ECLB.ARPA,
DANCY@HI-MULTICS.ARPA
Cc: AHOOK@USC-ECLB.ARPA, DLEHMAN@USC-ECLB.ARPA, CROBY@USC-ECLB.ARPA,
SYKESD@WPAFB-JALCF.ARPA,
BRENDER@MARLBORO.DEC.COM
Subject: FRN 85-11-20

[This is a duplicate fothe previous FRN--but I've supplied a SUBJECT here (and misplaced Ron in the CC--sorry!)]

FRT Members:

Please welcome Henry (Henri?) DANCY, of Alsys, to the FRT. He will be assisted by Mike WOODGER (at the same NET address).

Below is a dispute concerning the "split" processing of a "B" test. (--of ACVC 1.6)

---Dan
------- *

With the exception of test BA1101C2m, all the tests were modified and rerun to demonstrate that additional errors from the compiler were results of cascading errors. Test BA1101C2M had separate piece ba1101c4 split into two pieces, BA1101C4-X.ADA and BA1101C4-Y.ADA, because of the way we process compilation units in the same file that have the same name. In this case, the second unit in file BA1101C4.ADA is named exactly the same as the first (error containing) unit, so the second unit is inserted into the library with that name.

The fact that the first unit with the same name contained an error is no longer available when the status of the compilation is reported. When the file is split into two units compiled separately, the test passes.

Message 318 1826 25 Nov 85
From: hilfingr@renoir.berkeley.edu (Paul Hilfinger)
To: DLEHMAN@usc-eclb.arpa
Cc: FAST-REACTION@usc-eclb.arpa, KNAPPER@usc-eclb.arpa,
HILFINGER@ucbvax.berkeley.edu, GOODENOUGH@usc-isi.arpa,
goodenou%wang-inst.csnet@csnet-relay.arpa, DEWAR@nyu.arpa,
PLOEDEREDER@tl-20b.arpa, BWICHMANN@usc-eclb.arpa,
DANCY@hi-multics.arpa, AHOOK@usc-eclb.arpa, CROBY@usc-eclb.arpa,
SYKESD@wpafb-jalcf.arpa, BRENDER%marlboro.DEC@decwrl.dec.com
Subject: Re: FRN 85-11-20
In-Reply-To: Your message of 20 Nov 1985 16:41-PST.
‹[USC-ECLB.ARPA]20-Nov-85 16:41:57.DLEHMAN›

I'm sorry; your message is too cryptic for me to make any sort of decision. It sounds as if these guys are saying that when two units in the same file are compiled, the first of which is illegal, the "status of the compilation" (whatever that is) only reflects the second, legal, compilation unit. Why on earth is this an FRT matter? Does the compiler produce an error message for the illegal unit or not? Are you asking whether, in addition to producing an error message, the compiler must also return an appropriate error code to the operating stem? The answer, of course, is that the Standard says absolutely nothing about such details.

Paul
Message 319 1680 26 Nov 85 ANSWERED
From: sykesd@wpafb-jalcf
To: DLEHMAN@USC-ISIF
Cc: SYKESD@WPAFB-JALCF
Subject: That #@%&! split

Dan.

The text of JBG's message is below. I talked to John Kelly and he thinks the split is OK since there's no way that the programmer can get into trouble by this compiler's approach.

I didn't get a hold of JBG, but his message below seems to indicate that he understood what happened in the split. It all makes sense to me now.

Dave Sykes

From: GOODENOUGH 25-NOV-1985 23:08:44
To: SYKESD

Subj: (Network Mail)

Return-Path: ‹GOODENOUGH@USC-ISI.ARPA›
Received: from USC-ISI.ARPA by wpafb-jalcf ; 25 Nov 85 23:08:30 EST
Date: 25 Nov 1985 23:07:34 EST
Subject: Re: FRN 85-11-20
From: John B. Goodenough ‹GOODENOUGH@USC-ISI.ARPA›
To: DLEHMAN@USC-ECLB.ARPA
cc: FAST-REACTION@USC-ECLB.ARPA, KNAPPER@USC-ECLB.ARPA,
HILFINGER@UCBVAX.BERKELEY.EDU, GOODENOUGH@USC-ISI.ARPA,
goodenou%wang-inst.csnet@CSNET-RELAY.ARPA, DEWAR@NYU.ARPA,
PLOEDEREDER@TL-20B.ARPA, BWICHMANN@USC-ECLB.ARPA, DANCY@HI-MULTICS.ARPA,
AHOOK@USC-ECLB.ARPA, CROBY@USC-ECLB.ARPA, SYKESD@WPAFB-JALCF.ARPA,
BRENDER@MARLBORO.DEC.COM
In-Reply-To: ‹[USC-ECLB.ARPA]20-Nov-85 16:41:57.DLEHMAN›

It appears to me that the implementation probably doesn't diagnose the first error when the two package bodies are compiled together in the same compilation. Since the repetition of the body is just to make some later units legal, there certainly seems to be no problem in splitting this test.

Message 304 5729 15 Oct 85
From: sykesd@wpafb-jalcf
To: DLEHMAN@USC-ECLB
Cc: SYKESD@WPAFB-JALCF, AHOOK@USC-ECLB
Subject: Test disputes from VERDIX

Dan or Audrey,%

The following is a list of test disputes from VERDIX. Please/ forward to the FRT and provide a response as quickly as possible./

Thanks,#
Mike Hill$
SofTech/AVF-WPAFB%

TEST: C64103A#
COMMENT: This test contains the following declarations:+

```
TYPE FINEFIXED IS DELTA SYSTEM.FINEDELTA+
     RANGE -1.0 .. 1.0;(
LGC : CONSTANT := 2 * FINEFIXED'BASE'LARGE;,
TYPE COARSEFIXED IS DELTA 2 * SYSTEM.FINEDELTA,
     RANGE -LGC .. LGC;(
```

Since we only support one fixed point type, FINEFIXED and/ COARSEFIXED will be derived from the same predefined fixed point/ type. The value for FINEFIXED'BASE'LARGE is the upper bound of/ the predefined fixed point type. Twice the upper bound of the/ predefined fixed point type is greater than the upper bound of/ the predefined fixed point type, so the predefined fixed point/ type cannot be selected. Therefore, no type can be selected..

TEST: A62006D, B62006C, B85007C'
COMMENT: These tests appear to conflict. They all contain/ expressions of the form REC.COMP'POSITION (among others) where/ REC is an OUT record formal parameter. In A62006D, lines 52-57/ no error is expected, whereas in B62006C, B85007C lines 116, 120,/ 124 and error is expected. This should be allowed according to/ 4.1(9), since the evaluation of the prefix only determines the/ entity and requires no reading of the out parameter.,

TEST: C94004B#
COMMENT: At line 63 of the test there is a misuse of the terminated/ attribute (T'TERMINATED) where T is an access to a task. RM/ 9.9(1) only allows the prefix of the TERMINATED attribute to be a/ task object of value. The confusion is caused by a conflicting/ definition in Appendix A(47). Our understanding is that RM 9.9/ takes precedence.%

TEST: C94004A, C94004B, C94004C'
COMMENT: In both of these tests there is a library package which/

activates a task. The main program correctly terminates without/ waiting for completion of the task. Since our tasking system is/ implemeted as a single UNIX process, when the main program/ terminates, all other tasks are terminated (as allowed by RM/ 9.4(13)). Consequently, the task body never calls REPORT.RESULT/ procedure. What then is the criteria for passing these tests?.

TEST: LA3004A6M$
COMMENT: The subprograms LA3004A2 and LA3004A3 are compiled as subprogram/ bodies which introduce implicit subprogram declarations with a/ dependency on LA3004A0. The effect of recompiling the package/ body LA3004A0, makes the implicit subprogram declarations and/

subprogram bodies for LA3004A2 and LA3004A3 obsolete, due to the/ inline pragma dependence on LA3004A0. Therefore, a with on/ either LA3004A2 or LA3004A3 is no longer possible and results in/ error diagnostics during the compilation of LA3004A6M. SOLUTION:/ Introduce explicit subprogram declarations for LA3004A2 and/ LA3004A3.$

TEST: LA3004B6M$
COMMENT: See comment pertaining to LA3004A6M.)

TEST: CA3006D*$
COMMENT: These tests appear to expect a compiler/linker to discard a user/ package body just because that body WITHs an obsolete unit. We/ believe that linkers should give an error in this situation,/ forcing users to either remove the body or recompile it. We/ believe that in general it is highly unlikely that the user would/ want to discard the code. After all, if a package that does not/ require a body is GIVEN a body, then it's a fair bet that the/ body is doing something useful and that its absence will cause/ some error. For example:'

```
package will_be_recompiled is(
    crucial : integer := 1;(
end;#
---------------------------------)
package needs_no_body is'
    v : integer;%
end;#
---------------------------------)
with will_be_recompiled;'
package body needs_no_body is(
begin#
    v := will_be_recompiled.crucial;)
end;#
```

Suppose willberecompiled is now altered and recompiled:-

```
package will_be_recompiled is(
    crucial : integer := 10000;(
```

end;*

Surely the user will be dismayed if his variable v is now/ unintialized, or incorrectly initialized. These are the very/ kind of errors that Ada is designed to prevent.+

TEST: B48003B*
COMMENT: At line 55, the initial value should be an allocator.-

TEST: B64003A*
COMMENT: Line 67 calls function F as if it were a procedure. Based on the/ comments, this appears to be unintentional.+
^L
Message 308 6908 15 Oct 85
From: DLEHMAN@USC-ECLB.ARPA
To: FAST-REACTION@USC-ECLB.ARPA, KNAPPER@USC-ECLB.ARPA,
BWICHMANN@USC-ECLB.ARPA,
goodenou%wang-inst.csnet@CSNET-RELAY.ARPA, GOODENOUGH@USC-ISI.ARPA,
HILFINGER@UCB-VAX.ARPA, BRENDER@MARLBORO.DEC.COM,
DEWAR@NYU.ARPA, KEVIN.PHILLIPS%RSRE@UCL-CS.ARPA,
PLOEDEREDER@TL-20B.ARPA
Cc: DLEHMAN@USC-ECLB.ARPA, AHOOK@USC-ECLB.ARPA, CROBY@USC-ECLB.ARPA,
SYKESD@WPAFB-JALCF.ARPA
Subject: FRN 85-10-15

FRT Members:

Below are a vendor's arguments against tests from ACVC 1.6. Note that C64103A is here disputed for reasons not previously put forth (the previous dispute was dropped by a vendor). Also, tests C94004A, --B, & --C are again submitted. Although John supported these tests (and others concurred) previously, the previous disputer intends to raise the issue for the LMC and I thought that the FRT ought to reconsider the matter.

---Dan
------- *
postscript

The addressees listed in this message are the REAL Fast-Reaction Team. The Alsys-produced list of recent messages is NOT correct (odd how Alsys omitted Ron from their discussions!?). Dave SYKES of the ACVC Maintenance Organization has been included as an observer so that that effort is kept abreast of needed corrections or modifications to ACVC tests in a more timely manner.

(The tests C94004* are disputed again by this implementer, not merely issued for reconsideration.)

--DL **

%
TEST: C64103A
*COMMENT: This test contains the following declarations:

```
TYPE FINEFIXED IS DELTA SYSTEM.FINEDELTA
     RANGE -1.0 .. 1.0;
LGC : CONSTANT := 2 * FINEFIXED'BASE'LARGE;
TYPE COARSEFIXED IS DELTA 2 * SYSTEM.FINEDELTA
     RANGE -LGC .. LGC;
```

Since we only support one fixed point type, FINEFIXED and COARSEFIXED will be derived from the same predefined fixed point type. The value for FINEFIXED'BASE'LARGE is the upper bound of the predefined fixed point type. Twice the upper bound of the predefined fixed point type is greater than the upper bound of the predefined fixed point type, so the predefined fixed point type cannot be selected. Therefore, no type can be selected.

TEST: A62006D, B62006C, B85007C
'COMMENT: These tests appear to conflict. They all contain expressions of the form REC.COMP'POSITION (among others) where REC is an OUT record formal parameter. In A62006D, lines 52-57 no error is expected, whereas in B62006C, B85007C lines 116, 120, 124 and error is expected. This should be allowed according to 4.1(9), since the evaluation of the prefix only determines the entity and requires no reading of the out parameter.

TEST: C94004B
*COMMENT: At line 63 of the test there is a misuse of the terminated attribute (T'TERMINATED) where T is an access to a task. RM 9.9(1) only allows the prefix of the TERMINATED attribute to be a task object of value. The confusion is caused by a conflicting definition in Appendix A(47). Our understanding is that RM 9.9 takes precedence.

%
TEST: C94004A, C94004B, C94004C
'COMMENT: In both of these tests there is a library package which activates a task. The main program correctly terminates without waiting for completion of the task. Since our tasking system is implemeted as a single UNIX process, when the main program terminates, all other tasks are terminated (as allowed by RM 9.4(13)). Consequently, the task body never calls REPORT.RESULT procedure. What then is the criteria for passing these tests?

TEST: LA3004A6M
$COMMENT: The subprograms LA3004A2 and LA3004A3 are compiled as subprogram bodies which introduce implicit subprogram declarations with a dependency on LA3004A0. The effect of recompiling the package body LA3004A0, makes the implicit subprogram declarations and

subprogram bodies for LA3004A2 and LA3004A3 obsolete, due to the inline pragma dependence on LA3004A0. Therefore, a with on either LA3004A2 or LA3004A3 is no longer possible and results in

error diagnostics during the compilation of LA3004A6M. SOLUTION: Introduce explicit subprogram declarations for LA3004A2 and LA3004A3.

$
 TEST: LA3004B6M
$COMMENT: See comment pertaining to LA3004A6M.

 TEST: CA3006D*
$COMMENT: These tests appear to expect a compiler/linker to discard a user package body just because that body WITHs an obsolete unit. We believe that linkers should give an error in this situation, forcing users to either remove the body or recompile it. We believe that in general it is highly unlikely that the user would want to discard the code. After all, if a package that does not require a body is GIVEN a body, then it's a fair bet that the body is doing something useful and that its absence will cause some error. For example:

```
package will_be_recompiled is
    crucial : integer := 1;
end;
--------------------------------
package needs_no_body is
    v : integer;
end;
--------------------------------
with will_be_recompiled;
package body needs_no_body is
begin
    v := will_be_recompiled.crucial;
end;
```

Suppose willberecompiled is now altered and recompiled:

```
package will_be_recompiled is
    crucial : integer := 10000;
end;
```

Surely the user will be dismayed if his variable v is now unintialized, or incorrectly initialized. These are the very kind of errors that Ada is designed to prevent.

 TEST: B48003B
*COMMENT: At line 55, the initial value should be an allocator.

 TEST: B64003A
*COMMENT: Line 67 calls function F as if it were a procedure. Based on the comments, this appears to be unintentional.
^L

---

nd forwarded message

essage 320 2850 17 Oct 85

From: hilfingr@ucbrenoir.Berkeley.EDU
To: DLEHMAN@usc-eclb.arpa
Cc: FAST-REACTION@usc-eclb.arpa, KNAPPER@usc-eclb.arpa,
BWICHMANN@usc-eclb.arpa, goodenou%wang-inst.csnet@csnet-relay.arpa,
GOODENOUGH@usc-isi.arpa, Hilfinger@UCB-VAX.Berkeley.EDU,
BRENDER%marlboro.DEC@decwrl.ARPA, DEWAR@nyu.arpa,
KEVIN.PHILLIPS%RSRE@ucl-cs.arpa, PLOEDEREDER@tl-20b.arpa,
AHOOK@usc-eclb.arpa, CROBY@usc-eclb.arpa, SYKESD@wpafb-jalcf.arpa
Subject: Re: FRN 85-10-15
In-Reply-To: Your message of 15 Oct 1985 12:30-PDT.
‹[USC-ECLB.ARPA]15-Oct-85 12:30:55.DLEHMAN›

C64103A:

The implementor is correct by 3.5.9(7), which explicitly requires ''at least one'' anonymous fixed-point type.

A62006D, B62006C, B85007C:

The implementor has neglected the fact that in, e.g., ACC_ARR1'FIRST in B62006C, ''the entity'' is ACC_ARR1.all, whose determination requires THE VALUE (a pointer) of ACC_ARR1. Hence, he is incorrect.

C94004B:

This isn't so clear. Notice that the last sentence of 9.5(4) uses the phrase ''[selected component] whose prefix DENOTES the task object'' for a place where we intend that an access value would be acceptable. Similar phrasing is used in 9.9(3), but with 'designates' instead of 'denotes.' If the implementor is really pressed for time, I suppose you can rule this test inapplicable for him, and it should be submitted to the LMC for clarification for the reasons he gives. It is, however, my opinion that the intention of the Standard is to make the test legal as it stands.

C94004A-C:

I see no reason to change my previous response.

LA3004A6M, LA3004B6M:

The implementor appears to be correct here.

CA3006D*:

As you know, I have a semi-political position on this one. I believe that the Standard never did have any business specifying how the library behaves and how recompilation behaves--just that the final

program that is executed must have appropriate versions of everything. A sufficiently tortuous reading of the Standard can be used to back up this contention, and recently the LMC has shown an inclination to go along. Therefore, I side with the implementor on this one.

B48003B:
I suppose that would be better, although it does not invalidate the test.

B64003A:
I agree. Again, the test is not invalidated.

Message 336 2133 21 Oct 85
From: KNAPPER@USC-ECLB.ARPA
To: DLEHMAN@USC-ECLB.ARPA
Cc: FAST-REACTION@USC-ECLB.ARPA, KNAPPER@USC-ECLB.ARPA,
BWICHMANN@USC-ECLB.ARPA,
goodenou%wang-inst.csnet@CSNET-RELAY.ARPA,
GOODENOUGH@USC-ISI.ARPA,
HILFINGER@UCBVAX.BERKELEY.EDU, DEWAR@NYU.ARPA,
KEVIN.PHILLIPS%RSRE@UCL-CS.ARPA,
PLOEDEREDER@TL-20B.ARPA, AHOOK@USC-ECLB.ARPA,
CROBY@USC-ECLB.ARPA, SYKESD@WPAFB-JALCF.ARPA,
brender@MARLBORO.DEC.COM
Subject: Re: FRN 85-10-15...
In-Reply-To: ‹[USC-ECLB.ARPA]15-Oct-85 12:30:55.DLEHMAN›

C64103A

The implementer is correct.

A62006D, B62006C, B85007C

I agree with Paul.

C94004B

My reading of "designates" and "denotes" is as if they are synonomous. I believe the test is therefore correct. It can be argued however that those two words are not synonomous. We have a language issue here, but it's English not Ada. If the LMC wants to discuss the English I'll vote to withdraw the test until then.

C94004A, B, C

RM 9.4(13) says that "the language does not define whether such

tasks are required to terminate." My reading of this is that the language does not say that those tasks MUST terminate. The implementer is saying that those tasks MUST terminate. I therefore disagree with the implementer.

LA3004A6M. B6M

I agree with the implementer.

CA3006D*

The implementer and Paul have very valid points. I also believe the intent of the Standard for the library and on recompilation is that the final executable object be constructed from the appropriate versions of it's component modules. The Standard shouldn't be concerned with trying to specify how thatfinal object is to be built. I side with the implementer.

B48003B, B64003A

As long as the new errors do not mask the original error to be detected then thses B tests are not invalidated. They should however be revised.
Message 337 6417 21 Oct 85
From: John B. Goodenough ‹GOODENOUGH@USC-ISI.ARPA›
To: DLEHMAN@USC-ECLB.ARPA
Cc: FAST-REACTION@USC-ECLB.ARPA, KNAPPER@USC-ECLB.ARPA,
BWICHMANN@USC-ECLB.ARPA,
goodenou%wang-inst.csnet@CSNET-RELAY.ARPA, GOODENOUGH@USC-ISI.ARPA,
HILFINGER@UCB-VAX.ARPA, BRENDER@MARLBORO.DEC.COM, DEWAR@NYU.ARPA,
KEVIN.PHILLIPS%RSRE@UCL-CS.ARPA, PLOEDEREDER@TL-20B.ARPA,
AHOOK@USC-ECLB.ARPA, CROBY@USC-ECLB.ARPA, SYKESD@WPAFB-JALCF.ARPA
Subject: Re: FRN 85-10-15
In-Reply-To: ‹[USC-ECLB.ARPA]15-Oct-85 12:30:55.DLEHMAN›

C64103A

I think the implementer is incorrect here, under the rules of AI-00325. The Standard does not say that an implementation is allowed to support exactly one predefined fixed point type besides DURATION (and in fact, DURATION could be derived from this type). It seems to me that the implementer is here attempting to not support fixed point at all, and it is only the current weakness of the tests that have allowed him to get away with it so far. There should be no implementation difficulty in supporting the COARSE_FIXED type. Let me repeat the declarations here:

```
TYPE FINE_FIXED IS DELTA SYSTEM.FINE_DELTA
     RANGE -1.0 .. 1.0;
LGC : CONSTANT := 2 * FINE_FIXED'BASE'LARGE;
TYPE COARSE_FIXED IS DELTA 2 * SYSTEM.FINE_DELTA
     RANGE -LGC .. LGC;
```

Suppose that FINE_DELTA is 2**(-31). Then LGC will be 2*(1.0 - 2**(-31)), but since COARSE_FIXED has a delta that is twice as big, the model number mantissa

for LGC is the same as the mantissa for FINE_FIXED'BASE'LARGE, i.e., only the scale factor has changed. If this implementation supports only one fixed point type, then it presumably must reject

```
type F is delta 0.5 range -2.0 .. 2.0;
```

since the anonymous fixed point type from which F is derived must have a SMALL of 0.5, so this anonymous type cannot possibly be the same as the one used for FINE_FIXED.

In short, I am not convinced that the implementer has a sound argument here. Approving his argument as presented is equivalent to approving a compiler that does not support useful fixed point capability.

A62006D, B62006C, B85007C

These tests are all correct and do not conflict. The relevant RM paragraph is 4.1(4), which simply forbids any attribute prefixes from having an access type if the prefix is "a name that denotes a formal parameter of mode out or a subcomponent thereof." The rule has nothing to do with whether the prefix is evaluated or not. These tests check for this subtle distinction.

C94004B

There is no problem with this test. Section 3.8.2(3) allows prefixes of the attribute TERMINATED to be a value of an access type. It is this paragraph that makes the summary in Appendix A(47) correct.

C94004A, 4B, 4C

I only point out that 9.4(13) is a note and not part of the standard, so the implementer's reliance on this statement cannot be considered in interpreting the Standard (except, perhaps, as an indication of intent). Although I think the implementer is incorrect in his interpretation of the standard, this issue will be put on the LMC agenda for its November meeting.

LA3004A6M, B6M

The class L tests exist for the convenience of the tester, in that link time and compile time are usually distinguished by implementations. An L test means no one should be concerned if no error message is produced at "compile" time, because implementations will probably produce one at "link" time. In fact, in this case, an implementation may produce an error message at compile time, for the reasons indicated. But in any case, it is never an error to produce a "compile time" error messge for an L test. (These tests have been modified in version 1.7, and will hopefully cause less confusion for that release.)

CA3006D*

This test is in conformance with the requirements of the Standard. 10.3(5) says "If a compilation unit is successfully recompiled, the compilation units potentially affected by this change are obsolete and must be recompiled UNLESS THEY ARE NO LONGER NEEDED." It is clear that CA3006D1's body is made obsolete

when CA3006D0 is recompiled, and since the body is optional, it cannot be included in the set of secondary units associatied with the main program.

The implementer argues that users should be forced to "either remove the body or recompile it." But how is the body to be removed? Presumably by recompiling the package specification. The body then disappears because of the sentence just quoted from 10.3(5). The RM does not distinguish obsolesence of a body due to recompiling its specification and obsolescence due to recompiling a library unit with'd by the body.

I think the correctness of this test is independent of any discussions about the role of the library and library consistency. There is a semantic effect here, i.e., in the absence of any action by the programmer, is the optional body to be included in the set of elaborated units or not? 10.3(5) says the unit is not supposed to be included, and this is what the test checks.

The sentence in 10.3(5) was included to allow the deletion of useless package bodies. It is a side-effect of the sentence that bodies might unintentionally be deleted as well. It is perfectly okay for an implementation to announce that such a body has been made obsolete and will not be included when an attempt is made to execute the main program, but it is not consistent with the Standard to REFUSE to execute such a main program, and that is what the test checks for. The implementation must allow execution of the main program and must not include the obsolete, optional body.

I'll put this one on the LMC agenda too, since I think this effect is worth discussing. But I think it is sufficient if an implementation gives a warning message. (I have yet to be convinced that these optional bodies are really useful in practice.)

B48003B, B64003A

The tests are, of course, not invalidated even though they contain unintended errors. The implementer's comments are gratefully received by the ACVC team, who will revise the test to ensure that the intended errors are actually checked for.

Message 339 2156 22 Oct 85
From: Ron Brender ‹BRENDER@MARLBORO.DEC.COM›
To: FRTMEMS: FAST-REACTION@USC-ECLB, BABCOCK@USC-ECLB, DEWAR@NYU,
KEVIN.PHILLIPS%RSRE@UCL-CS, GAILLY@HI-MULTICS, GOODENOUGH@ISI,
goodenou%wang-inst.csnet@CSNET-RELAY, HILFINGER@BERKELEY,
AHOOK@USC-ECLB, KNAPPER@USC-ECLB, KRAMER@USC-ECLB, DLEHMAN@USC-ECLB,
PLOEDEREDER@TL-20B, PROBERT@USC-ECLB, CROBY@USC-ECLB,
SYKESD@WPAFB-JALCF, BWICHMANN@USC-ECLB;
Subject: FRNs 85-10-15 and 85-10-16

Re FRN 85-10-15

C64103A I think Goodenough is right on here. The test is okay.

A62006D, B62006C, B85007C
The tests are okay as Goodenough argues.

C94004B I agree with Goodenough. The use of 'TERMINATED is legal.

C94004A, -B, -C
I am a little surprized that John would so readily take these to the LMC as I thought this question had been resolved in an already approved AI (in favor of the test, against the implementer). But I can't find such, so will certainly go along with LMC review.

LA3004A*, -B*
The tests are okay and the implementer has acceptably passed them.

It surprizes me that this one even need go to the FRT. The behavior cited by the implementor has certainly been accepted before by the AVO in earlier validations.

CA3006D The language IS a little strange in this regard, but I remember the discussions that lead up to the wording now in the LRM. The test is certainly in line with the intent of the LRM.

B48003B, B64003A
The tests and the implementation are both okay, though I too have complained that the comments in B64003A in particular need improvement.

Re FRN 85-10-16

C48008A The implementer has a point. LMC review is in order.

CA1105B This has been discussed by the LMC, with a result that supports the test as is.

CE2110B The implementer is correct. The test needs revision to more delicately clean up after the possible exceptions.

Message 350 959 23 Oct 85 UNSEEN
From: dewar@NYU-ACF2.ARPA
To: AHOOK@USC-ECLB, BABCOCK@USC-ECLB, BRENDER@MARLBORO.DEC.COM, BWICHMANN@USC-ECLB;, CROBY@USC-ECLB, DEWAR@NYU, DLEHMAN@USC-ECLB, FRTMEMS:FAST-REACTION@USC-ECLB, GAILLY@HI-MULTICS, GOODENOUGH@ISI, KEVIN.PHILLIPS%RSRE@UCL-CS, KNAPPER@USC-ECLB, KRAMER@USC-ECLB, PLOEDEREDER@TL-20B, PROBERT@USC-ECLB, SYKESD@WPAFB-JALCF, HILFINGER@berkeley, goodenou%wang-inst.csnet@CSNET-RELAY
Subject: Re: FRNs 85-10-15 and 85-10-16

I support Ron Brender's comments on these tests (which if I remember all my mail correctly, means that I also support John Goodenough's comments!)
Message 352 433 24 Oct 85
From: BWICHMANN@USC-ECLB.ARPA
To: DLEHMAN@USC-ECLB.ARPA
Subject: C64103A

Dear Dan, For you to forward again!

C64103A

I agree with the comments of John Goodenough. If implementors need not past this test, then fixed point barely exists, certainly not in a useful form.

Message 357 1361 25 Oct 85
From: DLEHMAN@USC-ECLB.ARPA
To: FAST-REACTION@USC-ECLB.ARPA, KNAPPER@USC-ECLB.ARPA, HILFINGER@UCBVAX.BERKELEY.EDU, BRENDER@MARLBORO.DEC.COM, GOODENOUGH@USC-ISI.ARPA, goodenou%wang-inst.csnet@CSNET-RELAY.ARPA, DEWAR@NYU.ARPA, PLOEDEREDER@TL-20B.ARPA, BWICHMANN@USC-ECLB.ARPA
Cc: DLEHMAN@USC-ECLB.ARPA, AHOOK@USC-ECLB.ARPA, CROBY@USC-ECLB.ARPA, SYKESD@WPAFB-JALCF.ARPA
Subject: RE Test C64103A's Dispute

FRT Members:

This test may indeed be correct, as John argues, but I fail to see the pertinence of his arguments to the vendor's dispute.

The vendor argues that this test's declaration "LG_C: constant:= 2 * FINE_FIXED'BASE'LARGE;" exceeds the BOUNDS of the PREDEFINED fixed-point type (anonymous).

John's argument pertains to matters of precision.

The implementation, presumably, gives very different results for "F_F'BASE'LARGE" and "F_F'LARGE"--the former being a number in the xxx-illions, the latter being near 1.0 (what John calculates for the former).

I think that the issue here concerns LRM 3.5.9(8-10), 3.3.1(4), and the operation of the attribute BASE.

---Dan

Message 359 2800 25 Oct 85
From: John B. Goodenough ‹GOODENOUGH@USC-ISI.ARPA›
To: DLEHMAN@USC-ECLB.ARPA
Cc: FAST-REACTION@USC-ECLB.ARPA, KNAPPER@USC-ECLB.ARPA, HILFINGER@UCBVAX.BERKELEY.EDU, BRENDER@MARLBORO.DEC.COM, GOODENOUGH@USC-ISI.ARPA, goodenou%wang-inst.csnet@CSNET-RELAY.ARPA, DEWAR@NYU.ARPA, PLOEDEREDER@TL-20B.ARPA, BWICHMANN@USC-ECLB.ARPA, AHOOK@USC-ECLB.ARPA, CROBY@USC-ECLB.ARPA, SYKESD@WPAFB-JALCF.ARPA
Subject: Re: RE Test C64103A's Dispute
In-Reply-To: ‹[USC-ECLB.ARPA]25-Oct-85 11:08:43.DLEHMAN›

There undoubtedly is some set of bounds values that an implementation is allowed to reject for a fixed point type definition. The question is whether the implementation is being "reasonable" in its rejection. In a much earlier validation dispute, an implementer failed one of the case statement tests

because he ran out of memory space because he was using a jump table implementation for case statements. It was decided that this was not a reasonable implementation for certain useful kinds of case statement, and the implementer had to do a little (or maybe a lot) more work.

Now I suspect that the implementation in question is prepared to support any fixed point type declarations whose delta is greater than or equal to DURATION'SMALL, and whose upper and lower bound values fit within 31 bits when this level of precision is used. Such an anonymous fixed point type can be used for lots of fixed point type declarations. I was incorrect in my previous statement that the implementation would probably reject:

```
type FF is delta 0.5 range -2.0 .. 2.0;
```

because this could be derived from such an anonymous fixed point type. (In this case, the base type has a lot more precision and range than is actually required.)

Since DURATION'SMALL could be as large as 0.020, this would imply, however, that the declaration

```
type F2 is delta 0.001 range -1.0 .. 1.0;
```

would be rejected, although the model numbers can be easily represented in 31 bits or less.

So the real question is whether the implementer has an acceptable implementation of fixed point in terms of AI-00325. I was probably too harsh before in saying that having only one anonymous fixed point type was virtually useless, but I think it is safe to say that if I have hypothesized correctly, the implementer's support for fixed point is not very useful. The limitation to one predefined type doesn't seem to be fully justified by 1.1.2 anyway.

I would like to hear more implementer response to these arguments, and particularly, any arguments concerning the difficulty of providing better support for fixed point types.

Message 254 1683 5 Nov 85
From: sykesd@wpafb-jalcf
To: DLEHMAN@USC-ECLB
Cc: SYKESD@WPAFB-JALCF
Subject: ACVC disputes/questions

Dan,

The following text contains some comments/disputes with the ACVC test suite forwarded to me by Ron Tischler of Tandem Computers. Please forward to the FRT. To my knowledge, these disputes are NOT with respect to an imminent validation.

Thanks,
Mike Hill
^L

C43215A: This test expects a constraint error to be raised within a function that returns an unconstrained array type, but the constraint error should only be raised when the return value is assigned to an object whose constraint doesn't match, which happens after the function returns its value. In other words, the handler for the constraint error is in the wrong place.

CA2009A and CA2009D: These tests have comments saying that you could refuse to compile these tests if your compiler instead passes CA2009B and CA2009E, respectively. However, CA2009B and CA2009E have been withdrawn because they were illegal. How does this affect CA2009A and CA2009D?

D4A002A, D4A002B, D4A002C, D4A002D: These tests are called "capacity tests", which a compiler may refuse to compile, but I think it at least goes against the spirit of universal types for a compiler to refuse to compile them. Maybe a capacity test for "universal integer" might use expressions as big as 2 ** 255, but I think implementations are expected to do more with universal integers than simply treating them as ordinary integers.

--------end of list
Message 281 1985 12 Nov 85
From: DLEHMAN@USC-ECLB.ARPA
To: FAST-REACTION@USC-ECLB.ARPA, KNAPPER@USC-ECLB.ARPA,
    DEWAR@NYU.ARPA, PLOEDEREDER@TL-20B.ARPA,
    HILFINGER@UCBVAX.BERKELEY.EDU, BRENDER@MARLBORO.DEC.COM,
    GOODENOUGH@USC-ISI.ARPA, goodenou%wang-inst.csnet@CSNET-RELAY.ARPA,
    BWICHMANN@USC-ECLB.ARPA
Cc: AHOOK@USC-ECLB.ARPA, DLEHMAN@USC-ECLB.ARPA, CROBY@USC-ECLB.ARPA,
    SYKESD@WPAFB-JALCF.ARPA
Subject: FRN 85-11-12

Dear FRT:

I am forwarding some disputes and questions raised by a NON-validating implementer (at least "non-" at this time).

---Dan
------- *

C43215A: This test expects a constraint error to be raised within a function that returns an unconstrained array type, but the constraint error should only be raised when the return value is assigned to an object whose constraint doesn't match, which happens after the function returns its value. In other words, the handler for the constraint error is in the wrong place.

CA2009A and CA2009D: These tests have comments saying that you could refuse to compile these tests if your compiler instead passes CA2009B and CA2009E, respectively. However, CA2009B and CA2009E have been withdrawn because they were illegal. How does this affect CA2009A and CA2009D?

D4A002A, D4A002B, D4A002C, D4A002D: These tests are called "capacity tests", which a compiler may refuse to compile, but I think it at least goes against the spirit of universal types for a compiler to refuse to compile them. Maybe a capacity test for "universal integer" might use expressions as big as 2 ** 255, but I think implementations are expected to do more with universal integers than simply treating them as ordinary integers.

--------end of list

End forwarded message

Message 290 1862 14 Nov 85
From: John B. Goodenough ‹GOODENOUGH@USC-ISI.ARPA›
To: DLEHMAN@USC-ECLB.ARPA
Cc: FAST-REACTION@USC-ECLB.ARPA, KNAPPER@USC-ECLB.ARPA, DEWAR@NYU.ARPA, PLOEDEREDER@TL-20B.ARPA, HILFINGER@UCBVAX.BERKELEY.EDU, BRENDER@MARLBORO.DEC.COM, GOODENOUGH@USC-ISI.ARPA, goodenou%wang-inst.csnet@CSNET-RELAY.ARPA, BWICHMANN@USC-ECLB.ARPA, AHOOK@USC-ECLB.ARPA, CROBY@USC-ECLB.ARPA, SYKESD@WPAFB-JALCF.ARPA
Subject: Re: FRN 85-11-12
In-Reply-To: ‹[USC-ECLB.ARPA]12-Nov-85 16:44:00.DLEHMAN›

C43215A

The situation covered by the test is addressed in Commentary AI-00019, which says, "For positional aggregates, a check is made that the index bounds belong to the corresponding index subtype; CONSTRAINT_ERROR is raised if this check fails." Since the positional aggregate is written in the return statement, the check must be made in the context of the function, and the CONSTRAINT_ERROR handlers are in the right place, i.e., the test is correct.

The commentary has been approved by WG9 and the ADA Board.

Distribution List for IDA Memorandum Report M-157

Ms. Virginia Castor (5 copies)
Director, Ada Joint Program Office
1211 Fern St., Room C-107
Arlington, VA 22305

Defense Technical Information Center (2 copies)
Cameron Station
Alexandria, VA 22314

DoD-IDA Management Office
1801 N. Beauregard St.
Alexandria, VA 22311

CSED Review Panel

Dr. Dan Alpert, Director
Center for Advanced Study
University of Illinois
912 W. Illinois Street
Urbana, Illinois 61801

Dr. Barry W. Boehm
TRW Defense Systems Group
MS 2-2304
One Space Park
Redondo Beach, CA 90278

Dr. Ruth Davis
The Pymatuning Group, Inc.
2000 N. 15th Street, Suite 707
Arlington, VA 22201

Dr. Larry E. Druffel
Rational Machines
1501 Salado Drive
Mountain View, CA 94043

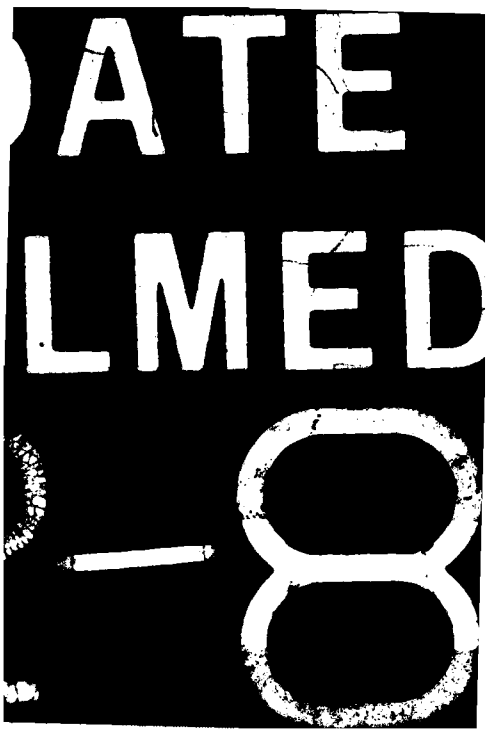
Mr. Neil S. Eastman, Manager
Software Engineering & Technology
IBM Federal Systems Division
6600 Rockledge Drive
Bethesda, MA 20817

Dr. C.E. Hutchinson, Dean
Thayer School of Engineering
Dartmouth College
Hanover, NH 03755

Mr. Oliver Selfridge
45 Percy Road
Lexington, MA 02173

IDA

Mr. Seymour Deitchman, HQ
Mr. Robin Pirie, HQ
Dr. Jack Kramer, CSED
Dr. John Salasin, CSED
Dr. Robert Winner, CSED
Ms. Audrey A. Hook, CSED (2 copies)
Mr. R. Danford Lehman, CSED (2 copies)
Ms. Katydean Price, CSED (2 copies)
IDA Control & Distribution Vault (2 copies)


ATE
LMED
-8