

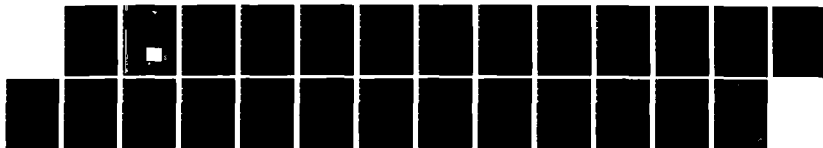
AD-A166 924

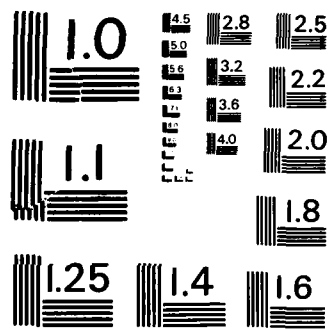
VALIDATION SUMMARY REPORT ROLM CORPORATION ROLM NSE/800 1/1
ADE ADA (TRADEMARK) (U) FEDERAL SOFTWARE TESTING CENTER
FALLS CHURCH VA 24 MAY 85 DIT/FSTC-85/505

UNCLASSIFIED

F/G 9/2

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD-A166 924

2

OFFICE OF
SOFTWARE
DEVELOPMENT
AND
INFORMATION
TECHNOLOGY

FEDERAL
SOFTWARE
TESTING
CENTER



DTIC
SELECTED
S JAN 24 1986
INT A

DTIC FILE COPY



This document has been approved
for public release and sale; its
distribution is unlimited.

86 1 24 017

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

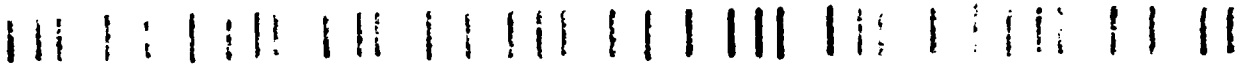
A166924

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER OIT/FSTC-85/505	12. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Validation Summary Report, Rolm Corp., Rolm MSE/800, ADE Ada Compiler Version 2.30.03.12	5. TYPE OF REPORT & PERIOD COVERED 24 May 85 ⁴ -24 May 86 ⁵	
	6. PERFORMING ORG. REPORT NUMBER OIT/FSTC-85/505	
7. AUTHOR(s)	8. CONTRACT OR GRANT NUMBER(s)	
9. PERFORMING ORGANIZATION NAME AND ADDRESS Federal Software Testing Center Office of Software Development and Info. Technology 2 Skyline Pl., Suite 1100, 5203 Leesburg Pike, Falls Church, VA		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Ada Joint Program Office 1211 S. Fern ST. Room C-107 Arlington, VA 22202		12. REPORT DATE 24 May 1985
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Federal Software Testing Center		13. NUMBER OF PAGES 17
		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) Unclassified		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Ada Programming Language, Ada Compiler Validation Summary Report, VSR, Ada Compiler Validation Cppability, ACVC, Validation Testing, Ada Validation Office, AVO, Ada Validation Facility, AVF, ANSI/MIL-STD-1815Ada, Ada Joint Program Office		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) The purpose of the report is to present the results and conclusions of performing standardized tests on the Rolm Corp. ADE Ada Compiler, which is hosted on the Rolm MSE/800, using AOS/VS Version 4.04. with version 1.5 of the ACVC Validation tests.		

DD FORM 1473
1 JAN 73EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)



Prepared for:

ROLM CORPORATION

Prepared by:

FEDERAL SOFTWARE TESTING CENTER
Office of Software Development
and Information Technology
Two Skyline Place, Suite 1100
5203 Leesburg Pike
Falls Church, VA 22041-3467

May 24, 1985

S DTIC
ELECTE **D**
JAN 24 1986

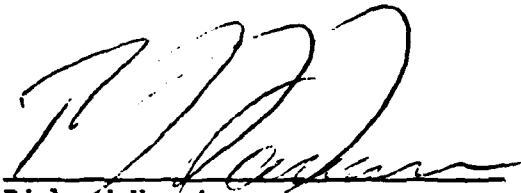
VALIDATION SUMMARY REPORT

A

ROLM CORPORATION
ROLM MSE/800
ADE Ada Compiler Version 2.30.03.12

Report OIT/FSTC-85/505

OFFICE OF SOFTWARE DEVELOPMENT
AND INFORMATION TECHNOLOGY



Richard Harrison
Acting Director
Federal Software Testing Center



Thomas H. Probert, Ph. D.
Institute for Defense Analyses



Ed Lieblein
Director, AJPO

- F -

ABSTRACT

The purpose of this Validation Summary Report is to present the results and conclusions of performing standardized tests on the Rolm Corporation ADE Ada Compiler. On-site testing was performed between 06 May 1985 and 10 May 1985 at Rolm Corporation, San Jose, California, under the auspices of an Ada Validation Facility (AVF), the Federal Software Testing Center, according to current Ada Validation Office (AVO) policies and procedures.

The Rolm Corporation ADE Ada Compiler is hosted on the Rolm MSE/800 system. The suite of tests known as the Ada Compiler Validation Capability (ACVC), Version 1.5, was used. The ACVC suite of tests is used to validate conformance of the compiler to ANSI/MIL-STD-1815A (Ada). This standard is described in the ANSI Ada Reference Manual, January 1983. Not all tests in the ACVC test suite are applicable to this specific implementation. Also, known test errors in Version 1.5 are present in some tests; these tests were withdrawn. The purpose of the testing is to ensure that the compiler properly implements legal language constructs and that it identifies, rejects from processing, and labels illegal constructs.

The Rolm Corporation Compiler ADE Ada, version 2.30.03.12, using AOS/VS Version 4.04, was tested with version 1.5 of the ACVC validation tests. Version 1.5 of the test suite contains 2050 tests, of which 1813 were applicable to this implementation. Of the applicable tests, 66 were withdrawn due to errors in the tests. Of the remaining applicable correct tests, 1813 passed.

TABLE OF CONTENTS

	Page
1. Introduction	1
1.1 Purpose of the Validation Summary Report	1
1.2 Use of the Validation Summary Report	2
1.3 References	2
1.4 Definitions of Terms	3
2. TEST ANALYSIS	5
2.1 Class A Testing	5
2.1.1 Class A Test Procedures	5
2.1.2 Class A Test Results	5
2.2 Class B Testing	5
2.2.1 Class B Test Procedures	5
2.2.2 Class B Test Results	6
2.3 Class C Testing	6
2.3.1 Class C Test Procedures	6
2.3.2 Class C Test Results	7
2.4 Class D Testing	7
2.4.1 Class D Test Procedures	7
2.4.2 Class D Test Results	7
2.5 Class E Testing	7
2.5.1 Class E Test Results	7
2.6 Class L Testing	7
2.6.1 Class L Test Procedures	7
2.6.2 Class L Test Results	7
3. COMPILER NONCONFORMANCES	8
4. ADDITIONAL INFORMATION	9
4.1 Compiler Parameters	9
4.2 Testing Information	10
4.2.1 Pre-Test Procedures	10
4.2.2 Control Files	10
4.2.3 Test Procedures	10
4.2.4 Test Analysis Procedures	10
4.2.5 Timing Information	10
4.2.6 Description of Errors in Withdrawn Tests	11
4.2.7 Description of Inapplicable Tests	12
4.2.8 Information derived from the Tests	14
5. SUMMARY AND CONCLUSIONS	17

1. Introduction

1.1 Purpose of the Validation Summary Report

This report describes the results of the validation effort for the following compiler:

Host Machine: Rolm MSE/800
Operating System: AOS/VS 4.04
Host Disk System: 6236 and 6237
Target Machines: Rolm MSE/800
Operating System: AOS/VS 4.04
Language Version: ANSI/MIL-STD-1815A Ada
Translator Name: ADE Ada
Translator Version: 2.30.03.12
Validator Version: 1.5

Testing of this compiler was conducted by the Federal Software Testing Center under the supervision of the Ada Validation Office (AVO), at the direction of the Ada Joint Program Office. Testing was conducted from May 06, 1985, through May 10, 1985, at Rolm Corporation, San Jose, CA, in accordance with AVO policies and procedures.

The purpose of this report is to document the results of the testing performed on the compiler, and in particular, to:

- . identify any language constructs supported by the compilers that do not conform to the Ada standard;
- . identify any unsupported language constructs required by the Ada standard; and
- . describe implementation-dependent behavior allowed by the standard.

1.2 Use of the Validation Summary Report

The Ada Validation Office may make full and free public disclosure of this report in accordance with the "Freedom of Information Act" (5 U.S.C. #552). The results of the validation are only for the purpose of satisfying United States Government requirements, and apply only to the computers, operating systems, and compiler version identified in this report.

The Ada Compiler Validation Capability is used to determine insofar as is practical, the degree to which the subject compiler conforms to the Ada standard. Thus, this report is necessarily discretionary and judgemental. The United States Government does not represent or warrant that the statements, or any one of them, set forth in this report are accurate or complete, nor that the subject compiler has no other nonconformances to the Ada standard. This report is not meant to be used for the purpose of publicizing the findings summarized herein.

Any questions regarding this report or the validation tests should be sent to the Ada Validation Office at:

Ada Joint Program Office
Room 3D 139 (400 Army Navy Drive)
Pentagon
Washington, D.C. 20301

1.3 References

Reference Manual for the Ada Programming Language, ANSI/MIL-STD-1815A, February 1983.

Ada Validation Organization: Policies and Procedures, Mitre Corporation, June 1982, PB 83-110601.

Ada Compiler Validation Implementers' Guide, SofTech, Inc., October 1980.

The Ada Compiler Validation Capability, Computer, Vol. 14, No. 6, June 1981.

Using the ACVC Tests, SofTech, Inc. November 1981.

Ada Compiler Validation Plans and Procedures, SofTech, Inc. November 1981.

1.4 Definitions of Terms

Class A tests are passed if no errors are detected at compile time. Although these tests are constructed to be executable, no checks can be performed at run-time to see if the test objective has been met; this distinguished Class A from Class C tests. For example, a Class A test might check that keywords of other languages (other than those already reserved in Ada) are not treated as reserved words by an Ada implementation.

Class B tests are illegal programs. They are passed if all the errors they contain are detected at compile-time (or link-time) and no legal statements are considered illegal by the compiler.

Class L tests consist of illegal programs whose errors cannot be detected until link time. They are passed if errors are detected prior to beginning execution of the main program.

Class C tests consist of executable self-checking programs. They are passed if they complete execution and do not report failure.

Class D tests are capacity tests. Since there are no firm criteria for the number of identifiers permitted in a compilation, number of units in a library, etc., a compiler may refuse to compile a class D test. However, if such a test is successfully compiled, it should execute without reporting a failure.

Class E tests provide information about an implementation's interpretation of the Standard. Each test has its own pass/fail criterion.

ACVC: Acronym for the Ada Compiler Validation Capability.

AVO: The Ada Validation Office. In the context of this report the AVO is responsible for directing compiler validation.

CHECK or
CHECKTEST: An automated tool defined by the FSTC and developed by the AVF that produces summary test results by reading compiler output in a spool file. This tool is available on the ACVC distribution tapes from the AVF.

CUSTOMER: The agency requesting the validation (Rolm Corporation).

FSTC: Federal Software Testing Center. In the context of this report the FSTC conducts Ada validations under contract to the AVO as a satellite facility.

HOST: The computer on which the compiler executes (Rolm MSE/800).

IG: ACVC Implementers' Guide.

RM: The Ada Language Reference Manual.

STANDARD: The standard for the Ada language, ANSI/MIL-STD-1815A.

TARGET: The computer for which the compiler generates object code (Rolm MSE/800).

VALIDATION: The process of testing a compilation system to certify that it conforms to the standard.

VALIDATION TESTS: The set of test programs used to detect non-conformances in compilation systems. In this report, the term will be used (unqualified) to mean the ACVC tests.

2. TEST ANALYSIS

The following table shows that the Rolm Corporation ADE Ada compiler passed all applicable tests.

	A	B	C	D	E	L	Support	Total
Processed	58	753	1206	14	7	9	3	2050
Inapplicable	1	9	158	3	0	0	0	171
Withdrawn	0	1	65	0	0	0	0	66
Passed	57	743	983	11	7	9	3	1813
Failed	0	0	0	0	0	0	0	0

171 tests in the suite were found to be not applicable to the Rolm Corporation implementation.

In addition, 66 tests were withdrawn from the test suite because they did not conform to ANSI/MIL-STD-1815A, the Ada Language Standard.

2.1 Class A Testing

Class A tests check that legal Ada programs can be successfully compiled. These tests are executed but contain no executable self-checking capabilities. There were 58 class A test programs processed in this validation.

2.1.1 Class A Test Procedures

Each class A test was separately compiled and executed. However, the only purpose of execution is to produce a message indicating that the test passed.

2.1.2 Class A Test Results

Successful compilation and execution without any error messages indicates the tests passed. All 57 applicable tests passed.

2.2 Class B Testing

Class B tests check the ability to recognize illegal language usage. There were 743 applicable class B tests processed.

2.2.1 Class B Test Procedures

Each Class B test was separately compiled. The resulting test compilation listings are manually examined to see if every illegal construct in the test is detected. If some errors are not detected, a version of the program test is created that contains only undetected illegal constructs. This revised version is recompiled and the results analyzed. If some errors still are not detected, the revision process is repeated until a revised test contains only a single previously undetected illegal construct.

A B test is considered to fail only if a version of the test containing a single illegal construct is accepted by the compiler (i.e., an illegal construct is not detected) or a version containing no errors is rejected (i.e., a legal construct is rejected).

2.2.2 Class B Test Results

There were 753 class B tests presented to the compiler. Of these tests 9 were found to be inapplicable to this implementation (see Section 4.2.7); 1 test was found to be incorrect (i.e., a conforming compiler would have failed each of these tests). All 743 remaining class B tests passed.

Because all errors were not detected when compiling the original tests, the following 19 tests were modified by removing the detected errors; the modified tests were then resubmitted to see if the remaining errors would be detected.

B26005A.ADA	B26005AA0.ADA	B26005AA1.ADA	B26005AA2.ADA
B26005AA3.ADA	B44001AA0_B.ADA	B44001A_B.ADA	B64004A.ADA
B64004AA0.ADA	B67001AA0_B.ADA	B67001AA1_B.ADA	B67001A_B.ADA
B97101AA0_AB.ADA	B97101AA1_AB.ADA	B97101AA2_AB.ADA	B97101A_AB.ADA
B97101EA0_AB.ADA	B97101EA1_AB.ADA	B97101E_AB.ADA	

All illegal constructs were eventually detected except in some tests that were withdrawn because of errors in the tests (see Section 4.2.8).

2.3 Class C Testing

Class C tests check that legal Ada programs are correctly compiled and executed by an implementation. There were 983 class C tests processed in this validation attempt.

2.3.1 Class C Test Procedures

Each Class C test is separately compiled and executed. The tests are self-checking and produce PASS/FAIL messages. Any 'failed' tests are individually checked to see if they are correct and if they are applicable to the implementation. Any tests that are inapplicable or that do not conform to the Ada Standard are withdrawn.

2.3.2 Class C Test Results

All class C tests were processed except those tests requiring a floating point precision exceeding SYSTEM.MAX_DIGITS.

2.4 Class D Testing

Class D tests are executable tests used to check an implementation's compilation and execution capacities. There were 11 class D tests used in this validation.

2.4.1 Class D Test Procedures

Each class D test is separately compiled and executed. The tests are self-checking and produce PASS/FAIL messages.

2.4.2 Class D Test Results

Of the 11 applicable class D tests, 11 passed and 3 were found to be inapplicable to this implementation. Of these 3 were withdrawn because of errors in the tests. See section 4.2.7 for further information.

2.5 Class E Testing

Class E tests are executable tests that provide information about an implementation's interpretation of the Standard in areas where the Standard permits implementations to differ. Each test has its own pass/fail criterion. There were 7 class E tests used in this validation.

2.5.1 Class E Test Results

All class E test results passed.

2.6 Class L Testing

Class L tests check that incomplete or illegal Ada programs involving multiple separately compiled source files are detected at link time and are not allowed to execute. There were 9 Class L test programs processed in this validation attempt.

2.6.1 Class L Test Procedures

Each Class L test is separately compiled and execution is attempted. The tests produce FAIL messages if executed. Any "failed" tests are individually checked to see if they are correct and if they are applicable to the implementation. Any tests that are inapplicable or that do not conform to the Ada standard are withdrawn.

2.6.2 Class L Test Results

Of the 9 class L tests, none were found to be inapplicable to this implementation, and none were withdrawn due to errors in the tests. All nine L tests passed.

3. COMPILER ANOMALIES AND NONCONFORMANCE

There were no nonconformances to the Ada standard detected in this validation. The compiler passed all applicable correct tests.

4. ADDITIONAL INFORMATION

This section describes in more detail how the validation was concluded.

4.1 Compiler Parameters

Certain tests do not apply to all Ada compilers, e.g., compilers are not required to support several predefined floating point types, and so tests must be selected based on the predefined types an implementation actually supports. In addition, some tests are parameterized according to the maximum length allowed by an implementation for an identifier (or other lexical element; this is also the maximum line length), the maximum floating point precision supported, etc. The implementation dependent parameters used in performing this validation were:

- . maximum lexical element length: 120 characters.
- . maximum digits value for floating point types: 15
- . SYSTEM.MIN_INT: -2147483648
- . SYSTEM.MAX_INT: 2147483647
- . predefined numeric types: INTEGER, FLOAT.
- . INTEGER'FIRST: -2147483648
- . INTEGER'LAST: 2147483647
- . source character set: ASCII
- . extended ascii chars: abcdefghijklmnopqrstuvwxyz
!\$%?[\]^'() ";
- . non-ascii char type: (NON_NULL)
- . TEXT_IO.COUNT'LAST: 2147483646
- . TEXT_IO.FIELD'LAST: 2147483646
- . illegal external file name1: "bad-character*^"
- . illegal external file name2: "much_much_much_too_long_
name_for_a_file"
- . SYSTEM.PRIORITY'FIRST: 1
- . SYSTEM.PRIORITY'LAST: 10

4.2 Testing Information

Tests were compiled/executed at Rolm Corporation, San Jose, CA.

4.2.1 Pre-Test Procedures

Prior to testing, appropriate values for the compiler-dependent parameters were determined. These values were used to adapt tests that depend on the values. A magnetic tape containing the adapted tests [and split versions of some class B tests (see section 2.2.2)] was prepared and brought to the testing site.

4.2.2 Control Files

Rolm Corporation provided command procedures that compiled and executed tests automatically.

4.2.3 Test Procedures

All files from the version 1.5 tape were read onto disk. The package REPORT and the procedure CHECK-FILE were first compiled and the corresponding library file saved. The tests checking the REPORT package and CHECK-FILE procedure were executed on the Rolm MSE/800. Then all tests were grouped into batch jobs by class and by chapter and run on the Rolm MSE/800.

All applicable results were correct from the Rolm MSE/800 under AOS/VS.

All applicable executable modules were processed on the Rolm MSE/800 AOS/VS successfully.

4.2.4 Test Analysis Procedures

On completion of testing, all results were analyzed for failed Class A, C, D, E, or L programs, and all class B compilation results were individually analyzed. Analysis procedures are described for each test class in chapter 2.

4.2.5 Timing Information

The real (i.e., wall clock) time required for running all the tests on the Rolm MSE/800 was approximately 32 hours and 51 minutes.

4.2.6 Description of Errors in Withdrawn Tests

The following tests in version 1.5 of the ACVC did not conform to the ANSI Ada standard and were withdrawn for the reasons given below:

- . B66001A-B: Test checks (in section G) that a parameterless function that is equivalent to an enumeration literal in the same declarative region is a redeclaration and, as such, is forbidden. According to RM 8.3(17), the explicit declaration of such a function is allowed if an enumeration literal is considered to be an implicitly declared predefined operation. The RM is not clear on this point. This issue has been referred to the Language Maintenance Committee for resolution. Since the issue cannot be resolved at this time, the test is withdrawn from Version 1.5. (Please note that this test may be considered correct and may appear in the future Versions of the ACVC, including Version 1.6.)
- . C38104A-B: An incomplete type with discriminants was constrained before its full declaration occurred. An implementation is allowed to reject such subtype indications because of an ambiguity in the language.
- . C43103B-B: A non-null range had a bound that was outside the index subtype.
- . C43206A-B, C43207A-B, C43207B-B, C43214A-B: CONSTRAINT_ERROR is raised if one dimension of a multidimensional aggregate has non-null bounds that do not belong to the index subtype, even if the aggregate specifies a null array.
- . C45321*-B, C45521*-B: Incorrect values were used for values assigned to variables having a floating point subtype.
- . C52001B-AB: An equality comparison for nonmodel numbers (e.g., 23.4 = 23.4) has an implementation defined value.
- . C52007A-B: A comparison of INTEGER'LAST with SYSTEM.MAX_INT will raise NUMERIC_ERROR if SYSTEM.MAX_INT exceeds INTEGER'LAST, since the implicit conversion of SYSTEM.MAX_INT to INTEGER will raise NUMERIC_ERROR.
- . C52102A-AB, C52102B-AB: The result of concatenating slices of an array of characters had an upper bound that did not belong to the array's index subtype because the array was declared to have an index subtype 1..10 (or 1..9) instead of subtype INTEGER.

- . C52103X-B: A test assumed that a slice would be performed even if it raised NUMERIC_ERROR.
- . C55B15A-B: If SYSTEM.MAX_INT is greater than INTEGER'LAST, the discrete range INTEGER range -SYSTEM.MAX_INT + 10 .. -SYSTEM.MAX_INT will raise NUMERIC_ERROR.
- . C87B10A-B: Literal values were used that were outside an integer based type for some implementations.
- . B87B23B-B: A tricky case of overload resolution marked OK was actually ambiguous.
- . C930BDA-B: An attempt to activate a task before its body is elaborated should raise TASKING_ERROR, not PROGRAM ERROR.
- . C95008A: It was possible for an entry call to call a terminated task, depending on the implementation.
- . C95009A: An unintended race condition in a tasking test allowed a null access value to be dereferenced before the access variable was assigned the access value of an allocated task.
- . CE3103A-B: A test would print a failed message if RESET raised USE_ERROR.
- . CE3804E-B: A test contained a nonmodel number for which an equality comparison was expected to always yield true.

4.2.7 Description of Inapplicable Tests

There were 3 tests that were not processed because SYSTEM.MAX_DIGITS was 15. These tests were:

D4A002B D4A004A D4A004B.

Because the implementation did not support LONG_INTEGER, SHORT_FLOAT, LONG_FLOAT, LONG_LONG_INTEGER, the following tests were inapplicable:

C24113L thru C24113Y inclusive

C34001E no LONG_INTEGER type
 C34001F no SHORT_FLOAT type
 C35702A no SHORT_FLOAT type
 C35705L thru C35705Y inclusive

C35706L thru C35706Y inclusive

C35707L thru C35707Y inclusive

C35708L thru C35708Y inclusive

C35802L thru C35802Y inclusive

C45241L thru C45241Y inclusive

C45321L thru C45321Y inclusive

C45421L thru C45421Y inclusive

C45424L thru C45424Y inclusive

C45521L thru C45521Y inclusive

C45621L thru C45621Z inclusive

B52004D - no LONG_INTEGER type

B55B09C - no LONG_INTEGER type

C55B07A - no LONG_INTEGER type

C55B16A - Enumeration Representation Specification not supported

C55B16A-AB was inapplicable because it required support for explicitly specifying the representation of an enumeration type.

LA2004A*-AB and LA3004B*-B were inapplicable because they required support for the INLINE pragma.

AE2101C-B, CE2201D-B, CE2201E-B, and CE2401D-B were inapplicable because the implementation's version of SEQUENTIAL_IO, DIRECT_IO and TEXT_IO did not allow for instantiation with unconstrained array and record types.

CE2102D-B, CE2102E-B, CE2102F-B, and CE2102G-B were inapplicable because the implementation does not support modes IN_FILE, OUT_FILE, and INOUT_FILE, and also the procedures RESET and DELETE.

B86001DOM and B86001DT were inapplicable because the implementation does not support additional predefined data types.

CE3111B-B__DEP through CE3111E, CE3114B and CE3115A were inapplicable because the implementation's I/O is buffered by the operating system which does not permit opening the same physical file as two logical files.

AE2201D, BC3205F, CE22010, CE2201E, CE2202A and CE2401D were inapplicable because unconstrained array I/O is not supported in the implementation.

4.2.8 Information derived from the Tests

Processing of the following tests indicated support as described below for a variety of implementation options examined by the tests.

- . E24101A-B.TST: If a based integer literal has a value exceeding SYSTEM.MAX_INT, an implementation may either reject the compilation unit at compile time or raise NUMERIC_ERROR at run time. This test showed that the Rolm Corporation compiler raises NUMERIC_ERROR at run time.
- . B26005A.ADA: This test contains all the ASCII control characters in string literals. The system replaced the control characters corresponding to format effectors with a space in the listing file. All occurrences were identified with a diagnostic message by the Rolm Corporation compiler.
- . D29002K-B.ADA: This test declares 713 identifiers and was passed by the Rolm Corporation compiler.
- . E36202A-B.ADA and E36202B-B.ADA: These tests declare multidimensional null BOOLEAN arrays in which LENGTH of one dimension exceeds INTEGER'LAST and SYSTEM.MAX_INT, respectively. An implementation can accept this, or it can raise NUMERIC_ERROR or STORAGE_ERROR at run time. The Rolm Corporation compiler did accept the declarations and raised NUMERIC_ERROR during execution.
- . D4A002A-AB.ADA and D4A002B.ADA: These tests contain universal integer calculations requiring 32 and 64 bits of accuracy, i.e., values that exceed SYSTEM.MAX_INT are used. An implementation is allowed to reject programs requiring such calculations. The Rolm Corporation compiler passed all four tests.
- . E43211B-B.ADA: If a bound in a non-null range of a non-null aggregate does not belong to an index subtype, then all choices may or may not be evaluated before CONSTRAINT_ERROR is raised. The Rolm Corporation compiler evaluates all choices before CONSTRAINT_ERROR is raised.
- . E43212B-B.ADA: This test examines whether or not all choices are evaluated before subaggregates are checked for identical bounds. The Rolm Corporation compiler evaluates all subaggregates first.

- . E52103Y-B.ADA, C52104X-B.ADA, C52104y-B.ADA: These tests declare BOOLEAN arrays with INTEGER'LAST+3 components. An implementation may raise NUMERIC_ERROR at the type declaration or STORAGE_ERROR when array objects of these types are declared, or it may accept the type and object declarations. The Rolm Corporation compiler raised NUMERIC_ERROR for null array with one dimension of length greater than INTEGER'LAST in E52103Y-B.
- . A series of tests (D55A03*-AB.ADA) checks to see what level of loop nesting is allowed by an implementation. Tests containing up to 65 nested loops passed without exceeding the implementation's capacity.
- . D56001B-AB.ADA contains blocks nested 65 levels deep. This test was passed.
- . C94004A-B.ADA: This test checks to see what happens when a library unit initiates a task and a main program terminates without ensuring that the library unit's task is terminated. An implementation is allowed to terminate the library unit task or it is allowed to leave the task in execution. This test showed that such library tasks do terminate when the main program terminates.
- . CA1012A4M-B.DEP: This test checks whether an implementation requires generic library unit bodies to be compiled in the same compilation as the generic declaration. The Rolm Corporation compiler does allow generic declarations and bodies to be compiled in completely separate compilations.
- . BC3204C*-B.ADA and BC3205D*-B.ADA: These tests contain a separately compiled generic declaration, some instantiations and a body. An implementation must reject either the instantiations or the body. The Rolm Corporation compiler generated errors when compiling the generic package body.
- . CE2106A-B.DEP and CE3110A-B.DEP: These tests confirm that dynamic creation and deletion of files is supported.
- . CE2107*.DEP: These tests showed that more than one internal file may be associated with the same external file.
- . CE2110B-B.DEP: This test confirmed that an external file associated with more than one internal file can be deleted.

- . EE3102C-B.ADA: This test confirmed that an Ada program can open an existing file in OUT_FILE mode, and can create an existing file in either OUT_FILE or IN_FILE mode.
- . CE3111A-B.DEP showed that two internal files may read the same external file.
- . CE3111B-B.DEP and CE3111C-B.DEP showed that the Rolm Corporation compiler does allow two internal TEXT_IO files to be associated with the same external file when one or both internal files are opened for writing.

5. SUMMARY AND CONCLUSIONS

The Ada Validation Facility (AVF) identified 2050 of the ACVC version 1.5 tests as being potentially applicable to the validation of the Rolm Corporation compiler hosted on the Rolm MSE/800. Of these, 66 were withdrawn due to test errors, and 171 were determined to be inapplicable before they were processed. The compiler passed the remaining 1813 tests.

The AVF considers these results to show acceptable compliance to the February 1983 ANSI Ada Reference Manual.

END

OTIC

6-86