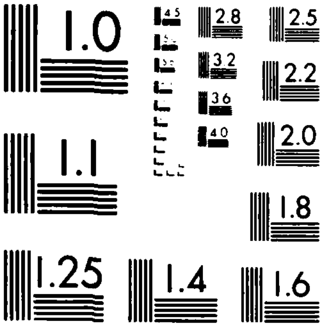
AD-A149 375 DIGITAL EQUIPMENT CORPORATION VAX ADA COMPILER VER T06-2 VAX-11/785 11/75..(U) DIGITAL EQUIPMENT CORP MAYNARD MASS 12 SEP 84 TC-84-DEC-390 1/1

UNCLASSIFIED F/G 9/2 NL

END

FILMED

DTIC


1.0
4.5
5.0
5.6
6.3
7.1
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6

Prepared for:

Digital Equipment Corporation

Ada Joint Program Office
400 Army-Navy Drive
Washington, DC 20301

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

September 12, 1984

ADA VALIDATION SUMMARY REPORT

Digital Equipment Corporation
VAX Ada Compiler Ver. T0.6-2
VAX-11/785, 11/780, 11/750,
11/730, using VMS Ver. 4.0
and MicroVAX I, using MicroVMS
Ver. 1.0

Report TC-84-DEC-390

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1 REPORT NUMBER
TC-84-Dec-390

2 GOVT ACCESSION NO.
AD-A149375

3 RECIPIENT'S CATALOG NUMBER

4 TITLE (and Subtitle)
Ada Compiler Validation Summary Report
Digital Equipment Corporation
VAX Ada Compiler Version TO.6-2

5 TYPE OF REPORT & PERIOD COVERED
~~12 September~~ 84 - 12 Sep. 85

6 PERFORMING ORG. REPORT NUMBER

7 AUTHOR(s)
Digital Equipment Corporation

8 CONTRACT OR GRANT NUMBER(s)
TC-84-Dec-390

9 PERFORMING ORGANIZATION NAME AND ADDRESS
Digital Equipment Corporation
146 Main Street ML5-5/E39
Maynard, MA 01754

10 PROGRAM ELEMENT PROJECT, TASK AREA & WORK UNIT NUMBERS

11 CONTROLLING OFFICE NAME AND ADDRESS
Federal Software Test Center
Office of Software Development and Infor. Tech.
Two Skyline Place, Suite 1100
5203 Leesburg Pike
Falls Church, VA 22041-3467

12 REPORT DATE
September 12, 1984

13 NUMBER OF PAGES

14 MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)
Ada Joint Program Office
Room 3D-139
The Pentagon
Washington, DC 20301-3081

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

Final Version

19 KEY WORDS (Continue on reverse side if necessary and identify by block number)

Ada Language, Ada Compiler Validation Capability, ACVC, Validation tests, Validation Summary Report, Digital Equipment Corporation VAX Ada Compiler Version TO.6-2, VAX 11, VMS 4.0

20 ABSTRACT (Continue on reverse side if necessary and identify by block number)

The Digital Equipment Corporation Compiler VAX Ada, version TO.6-2, for the VAX-11/785, 11/780, 11/750, 11/730 using VMS version 4.0 and MicroVAX I using MicroVMS version 1.0, was tested with version 1.4 of the ACVC validation tests. Version 1.4 of the test suite contained 2173 tests, of which 2099 were applicable to this implementation. Of the applicable tests, 11 were withdrawn due to errors in the tests. Of the remaining applicable correct tests 2099 passed, and no anomaly was discovered.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

This report has been reviewed and is approved.

John T. Stanton, for

George N. Baird, Director
Federal Software Testing Center

Thomas H. Probert

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

Robert F. Mathis

Robert F. Mathis
Director, AJPO

ABSTRACT

The Digital Equipment Corporation Compiler VAX Ada, version T0.6-2, for the VAX-11/785, 11/780, 11/750, 11/730 using VMS version 4.0 and MicroVAX I using MicroVMS version 1.0, was tested with version 1.4 of the ACVC validation tests. Version 1.4 of the test suite contained 2173 tests, of which 2099 were applicable to this implementation. Of the applicable tests, 11 were withdrawn due to errors in the tests. Of the remaining applicable correct tests 2099 passed, and no anomaly was discovered.

TABLE OF CONTENTS

Page

LIST OF EXHIBITS

Page

1. Introduction

1.1 Purpose of the Validation Summary Report

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

Host Machine:	VAX-11/785, 11/780, 11/750, 11/730, MicroVAX I
Operating System:	VMS Version 4.0; MicroVMS Version 1.0 for MicroVAX I
Host Disk System:	VAX-11/785 and 11/780: RA81 VAX-11/750: RK07 VAX-11/730 and MicroVAX I: RC25
Target Machine:	VAX-11/785, 11/780, 11/750, 11/730, MicroVAX I
Operating System:	VMS Version 4.0; MicroVMS Version 1.0 for MicroVAX I
Language Version:	ANSI/MIL-STD-1815A Ada
Translator Name:	VAX Ada
Translator Version:	T0.6-2
Validator Version:	1.4

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 84/08/20 through 84/08/24 at Nashua, New Hampshire 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 compiler that do not conform to the Ada standard.

- identify any unsupported language constructs required by the Ada standard.

- 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 therein.

Question 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. There were 7 class E tests in Version 1.4 of the test suite.

CUSTOMER: The agency requesting the validation (Digital Equipment Corporation).

HOST: The computer(s) on which the compiler executes are: VAX-11/785, 11/780, 11/750, 11/730, MicroVAX I.

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.

ACVC: Acronym for the Ada Compiler Validation Capability.

RM: The Ada Language Reference Manual.

IG: ACVC Implementers' Guide.

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

TARGET: The computer(s) for which the compiler generates object code are: VAX-11/785, 11/780, 11/750, 11/730, MicroVAX I.

VALIDATION: The process of validating a compiler. The term is used interchangeably with test or compiler test.

VALIDATION TESTS: The generic form used to refer to a set of test programs which evaluate how closely a compiler conforms to its language specification. In this report, the term will be used (unqualified) to mean the ACVC tests.

2. TEST ANALYSIS

A summary of tests processed, by class, is given in Exhibit 1, where:

Pr = processed.
NA = found to be inapplicable for this implementation.
Er = found to be incorrect, and withdrawn from the validation.
P = passed.
A = anomalies.
F = failed.
FE = failed to execute to completion.
FC = failed to compile successfully.
Fs = total of all failures (i.e., F+FE+FC).

Exhibit 1 shows that the Digital Equipment Corporation compiler passed all applicable correct tests.

There were 63 tests in the suite that were prc sed and found to be not applicable to the Digital Equipment Corporatior compiler (see Section 4.2.7)

In addition, 11 tests were withdrawn from the test suite because they did not conform to the ANSI/MIL-STD-1815A Standard for the Ada Language (see Section 4.2.6 for details.).

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 validaton.

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 58 tests passed.

Test Class	Pr	NA	Er	P	A	F	FE	FC	Fs
A	58	0	0	58	0	0	0	0	0
B	784	4	3	777	0	0	0	0	0
C	1262	45	8	1209	0	0	0	0	0
D	14	0	0	14	0	0	0	0	0
E	7	0	0	7	0	0	0	0	0
L	48	14	0	34	0	0	0	0	0
Total	2173	63	11	2099	0	0	0	0	0

EXHIBIT 1: Summary of Test Performance

2.2 Class B Testing

Class B tests check the ability to recognize illegal language usage. There were 784 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 whether every illegal construct in the test is detected. If all errors are not detected, a version of the test is created that contains only undetected illegal constructs. This revised version is recompiled and the results analyzed. If all errors are still not detected, the revision process is repeated until a revised test contains only a single 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 784 class B tests presented to the compiler. Of these tests 4 were found to be inapplicable to this implementation (see Section 4.2.7); 3 tests were found to be incorrect (i.e., a conforming compiler would have failed each of these tests). All 777 remaining class B tests passed.

Because all errors were not detected when compiling the original tests, the following 10 tests were modified and divided into 17 separate units by removing the detected errors; the modified tests were then resubmitted to see if the remaining errors would be detected:

B22003A.ADA B29001A-B.ADA B37301A.ADA B55A01A.ADA
B67001A-B.ADA B97101A-AB.ADA B97101E-AB.ADA B97102A-AB.ADA
B97102H-AB.ADA BC10AEA-B.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 1262 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

Of the 1262 class C tests, 1209 passed and 45 were found to be inapplicable to this implementation. Of these 8 were withdrawn because of errors in the tests. See Sections 4.2.6 and 4.2.7 for further information.

2.4 Class D Testing

Class D tests are executable tests used to check an implementation's compilation and execution capacities. There were 14 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 14 class D tests, 14 passed and 0 were found to be inapplicable to this implementation. Of these 0 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 7 class E tests passed. See Section 4.2.8 for further information.

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 48 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 48 class L tests, 14 were found to be inapplicable to this implementation (see Section 4.2.7), and 0 were withdrawn due to errors in the tests (see Section 4.2.6). The remaining 34 tests passed.

3. COMPILER NONCONFORMANCES

For this implementation there were no noncomformances or anomalies noted during the official validation. Section 4.2.9 provides a definition of an anomaly and detailed information describing an anomaly if any were noted during the official validation.

4. ADDITIONAL INFORMATION

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

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: 33
- SYSTEM.MIN_INT: -2147483648
- SYSTEM.MAX_INT: 2147483647
- predefined numeric types: INTEGER, FLOAT, SHORT_INTEGER, LONG_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: 2147483647
- TEXT_IO.FIELD'LAST: 2147483647
- illegal external file namel: "BADCHAR*"

. illegal external file name2:
"MUCHTOOLONGNAMEFORAFILEMUCHTOOLONGNAMEFORAFILE"

. SYSTEM.PRIORITY'FIRST: 0

. SYSTEM.PRIORITY'LAST: 15

4.2 Testing Information

Tests were compiled/executed at Nashua, New Hampshire.

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

Digital Equipment Corporation provided command procedures that compiled and executed tests automatically.

4.2.3 Test Procedures

All files from the version 1.4 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 all test systems. Then all executable tests were grouped into batch jobs by class and by chapter and run on the VAX-11/785, 11/780, 11/750, 11/730, MicroVAX I except for test C94004A-B.ADA which was run independently on all systems and class B tests which were run on the VAX-11/785.

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. Tests found to contain errors were withdrawn.

4.2.5 Performance Information

The real (i.e., wall clock hh:mm) time required for running was:

- VAX-11/785 16:34 (all tests time sharing with other jobs)
- VAX-11/780 23:57 (executable tests time sharing with other jobs)
- VAX-11/750 23:18 (executable tests stand-alone)
- VAX-11/730 36:53 (executable tests stand-alone)
- MicroVAX I 45:06 (executable tests. Total time for two stand-alone systems)

4.2.6 Description of Errors in Withdrawn Tests

The following tests in version 1.4 of the ACVC did not conform to the ANSI Ada standard and were withdrawn for the reasons given below:

In line 66 of B43201B-B, the 'others' choice in the aggregate which begins on the previous line is marked as being okay. However, VAX Ada correctly reports that the applicable index constraint at line 47 is not static. (The first discrete range is static, but the second is not; this does not satisfy the requirements for a static index constraint in the Ada standard, 4.9(11)).

B43203B-B checks the legal use of an 'others' choice in array aggregates. In line 80, the test incorrectly indicates that the use of 'others' should be diagnosed in error. VAX Ada correctly does not find an error in this line.

The accept statement in B950BAA-B at line 38 is marked as begin correct. However, VAX Ada correctly gives an error indicating that mode 'in' is explicitly given for parameter P2 in this line but the mode is implicit in the corresponding entry formal parameter at line 19.

C37011A-B checks that sliding occurs for initialization of record components that are of an array type. However, 3.2.1(16) of the Ada standard is interpreted as restricting sliding (more precisely, an implicit array subtype conversion) to the initialization of a complete array object (that is, one declared by an "object declaration"; a record component is declared by a "component declaration" according to 3.7(2)). VAX Ada correctly raises CONSTRAINT_ERROR for the elaboration of the object declaration at lines 27 and 42.

A test assumed that when a constrained array type was declared using a range, e.g., 1..10, that the index subtype was INTEGER instead of INTEGER range 1..10: C52102A-AB and C52102B-AB.

In C87B26B-B at line 119, the attribute STORAGE_SIZE is applied to a prefix which is an object of an access type. This is illegal; STORAGE_SIZE is allowed only with a prefix that is an access type/subtype ("is", not "is appropriate for") or for a task type/subtype or object of a task type.

These tests attempt a comparison of a null string literal with a variable. Evaluation of the string literal correctly raises CONSTRAINT_ERROR because the lower bound of the literal is INTEGER'FIRST. The test does not include an exception handler for this execution, and execution terminates abnormally: C52104G-AB.ADA and C52104Q-AB.ADA.

A test assumed that the size of the attribute of a type would always be equal to the size attribute of an object of that type: C74206A-B.

An array aggregate with named associations and an (others) choice must be qualified in an assignment context: C87B31A-B.

4.2.7 Description of Inapplicable Tests

There were 74 tests that were not processed. These tests were:

B43201B-B.ADA B43203B-B.ADA B52004D-AB.DEP B55B09C-AB.DEP
B86001CP-AB.DEP B86001CS-AB.DEP B950BAA-B.ADA C24113H-B.DEP
C24113I-B.DEP C24113J-B.DEP C24113K-B.DEP C24113L-B.DEP
C24113M-B.DEP C24113N-B.DEP C24113O-B.DEP C24113P-B.DEP
C24113Q-B.DEP C24113R-B.DEP C24113S-B.DEP C24113T-B.DEP
C24113U-B.DEP C24113V-B.DEP C24113W-B.DEP C24113X-B.DEP
C24113Y-B.DEP C34001E-B.DEP C34001F-B.DEP C35702A-AB.DEP
C37011A-B.ADA C52102A-AB.ADA C52102B-AB.ADA C52104G-AB.ADA
C52104Q-AB.ADA C55B07A-AB.DEP C74206A-B.ADA C87B26B-B.ADA
C87B31A-B.ADA C87B62B-B.DEP CE2102D-B.DEP CE2102E-B.DEP
CE2102F-B.DEP CE2102G-B.DEP CE2107A-B.DEP CE2107B-B.DEP
CE2107C-B.DEP CE2107D-B.DEP CE2107E-B.DEP CE2110B-B.DEP
CE2111D-B.DEP CE2401D-B.DEP CE2407A-B.ADA CE3104A-B.ADA
CE3109A-B.ADA CE3111B-B.DEP CE3111C-B.DEP CE3114B-B.DEP

CE3115A-B.DEP CE3202A-B.ADA LA3004A0-AB.DEP LA3004A1-AB.DEP
LA3004A2-AB.DEP LA3004A3-AB.DEP LA3004A4-AB.DEP
LA3004A5-AB.DEP LA3004A6M-AB.DEP LA3004B0-B.DEP
LA3004B1-B.DEP LA3004B2-B.DEP LA3004B3-B.DEP LA3004B4-B.DEP
LA3004B5-B.DEP LA3004B6M-B.DEP

Because the implementation did not support LONG_INTEGER, SHORT_FLOAT 8 tests were inapplicable:

LONG_INTEGER	C34001E,F, B52004D, C55B07A-AB, B55B09C-AB, B86001CS-AB
SHORT_FLOAT	C34001F-B, C35702A-AB, B86001CP-AB

LA3004A*-AB and LA3004B*-B were inapplicable because they required support for the INLINE pragma.

C87B62B-B was inapplicable because the test presumes that the attribute STORAGE_SIZE cannot legally yield zero or negative value. In VAX Ada, the value zero is returned when no representation clause has been specified for an access type or task type. In the case of an access type, NO storage is "reserved" for a collection that lacks a representation clause.

CE2401D-B was inapplicable because the implementation does not support operations on direct files of unconstrained array and record types unless the maximun record length for elements of the file is specified by an implementation dependent FORM option.

C24113H-B through C24113Y-B were inapplicable because these tests contain source lines that exceed the VAX Ada implementation limit of 120 characters.

CE2102D-B, E, CE2407A-B and CE3109A-B were inapplicable because the implementation does not allow creation of a file with mode IN_FILE.

CE2107A-B, B, C, D, E, CE2111D-B, CE3111B-B, C, CE3114B-B and CE3115A-B.DEP were inapplicable because this implementation does not support file sharing when more than one internal file is associated with the same external file and the internal files are opened with mode OUT_FILE.

CE2101G-B and CE3104A-B were inapplicable because the implementation does not support resetting a file from mode IN_FILE to either OUT_FILE or INOUT_FILE.

CE2102F-B was inapplicable because the implementation DOES support direct files with mode INOUT_FILE.

CE3202A-B was inapplicable since the FORM string returned by VAX Ada for the I/O funciton FORM includes a "time stamp" identifying when the information about the file was obtained; this time stamp has a granularity of 10 milliseconds. Thus this test either passes or fails depending on whether or not the two calls of FORM at line 32 occur within the same 10-millisecond interval; similar results occur for the two calls at line 36.

Results for inapplicable class D and E tests are given in Section 4.2.8.

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. Raising NUMERIC_ERROR at run-time is preferred, since it makes programs compilable for a wider variety of implementations (and the numeric literal might occur in an unexecutable portion of code). This test showed that VAX Ada raised a NUMERIC_ERROR at run-time for a compilation unit containing an integer literal exceeding SYSTEM.MAX_INT.

- B26005A.ADA: This test contains all the ASCII control characters in string literals. VAX Ada rejected all of these strings.

- D29002K-B.ADA: This test declares 1000 identifiers and was passed by VAX Ada.

- D4A002A-AB.ADA, D4A002B.ADA, D4A004A-AB.ADA, and D4A004B.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. VAX Ada passed all these tests.

- C52103X-B.ADA, 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 thest types are declared, or it may accept the type and object declarations. VAX Ada raised NUMERIC_ERROR.

- A series of tests (D55A03*-AB.ADA) check to see what level of loop nesting is allowed by an implementation. Tests containing 65 or fewer nested loops passed.

- 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 insuring 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 remain in execution 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. VAX Ada allows 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. VAX Ada rejected the body because of the instantiatons.

- CE2106A-B.DEP and CE3110A-B.DEP: These tests confirmed that dynamic creation and deletion of files is supported.

- 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: This test confirmed that more than one internal file could be associated with the same external file provided that both internal files were opened for reading.

4.2.9 Anomalous Test Results

An anomaly is test behavior that suggests the implementation does not conform to the Standard. The test behavior is not, however, considered to demonstrate nonconformance to the Standard for purposes of this validation attempt. The reasons for discounting an anomalous test result vary, depending on the anomaly. The existence of an anomaly sometimes means that new tests should be added to the suite to demonstrate the suspected nonconformance in a more convincing manner. Sometimes, when the anomaly is the result of procedural errors during the validation attempt, the existence of an anomaly requires revisions in validation procedures.

No anomalies were noted for this validation.

5. SUMMARY AND CONCLUSIONS

The Ada Validation Office identified 2173 of the ACVC version 1.4 tests as being being potentially applicable to the validation of the Digital Equipment Corporation compiler hosted on the VAX-11/785, 11/780, 11/750, 11/73C MicroVAX I. Of these, 11 were withdrawn due to test errors, and 63 were determined to be inapplicable after they were processed. The compiler passed the remaining 2099 tests.

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