

AD-A153 684

ADA VALIDATION SUMMARY REPORT ALS ADAVAX VERSION 11
VAX-11/780 USING VAX/VMS RELEASE 36(U) FEDERAL SOFTWARE
TESTING CENTER FALLS CHURCH VA 31 DEC 84
OIT/FSTC-84/581

1/1

UNCLASSIFIED

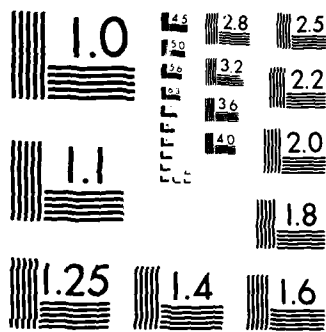
F/G 9/2

NL

END

FILED

DTN



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

2

AD-A153 684

FEDERAL BUREAU OF INVESTIGATION
 DEPARTMENT OF JUSTICE
 OFFICE OF
 SOFTWARE
 DEVELOPMENT
 AND
 INFORMATION
 TECHNOLOGY
 FEDERAL
 SOFTWARE
 TESTING
 CENTER

DTIC FILE COPY

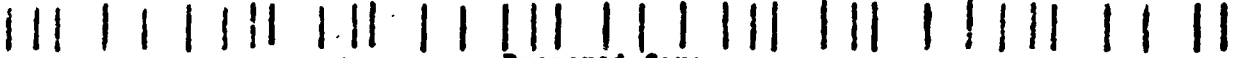


DTIC
ELECTE
 MAY 15 1985

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

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM	
1. REPORT NUMBER	17. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER	
OIT / FSTC - 84/501			
4. TITLE (and Subtitle)		5. TYPE OF REPORT & PERIOD COVERED	
Ada Validation Summary Report U.S. Army CECOM/SofTech ALS AdaVAX Version 1.1		December 31, 1984 to December 31, 1985	
		6. PERFORMING ORG. REPORT NUMBER	
7. AUTHOR(s)		8. CONTRACT OR GRANT NUMBER(s)	
Federal Software Testing Center Office of Software Development Information Technologies			
9. PERFORMING ORGANIZATION NAME AND ADDRESS		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS	
Federal Software Testing Center Office of Software Development & Information Technologies Two Skyline Place, Suite 1100 5203 Leesburg Pike Falls Church, VA 22041-3467			
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE	
Ada Validation Office Institute for Defense Analyses 1801 N. Beauregard Street Alexandria, VA 22311		December 31, 1984	
		13. NUMBER OF PAGES	
		20	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report)	
Ada Joint Program Office Room 3D-139 (1211 S. Fern, C-107) The Pentagon Washington D.C. 20301-3081		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 Compiler Validation Summary Report, VSR, Ada Language System, ALS, Validation Testing, Ada Compiler Validation Capability, ACVC, FSTC, Ada Validation Office, AVO, AVF, Ada Validation Facility, AJPO, U.S. Army CECOM, SofTech, ALS AdaVAX, VMS3.6, ANSI MIL/SDT Ada, ACVC Version 1.4			
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)-			
Under U.S. Army Contract DAAK80-C-0507, SofTech, Incorporated has developed an Ada compilation system for the Department of the Army, Headquarters USA, Communications Electronics Command. The compiler, ALS AdaVAX version 1.25, was tested against version 1.4 of the Ada Compiler Validation Capability (ACVC) test suite on a VAX-11/780 operating under VAX/VMS version 3.6. Version 1.4 of the test suite contained 2185 tests, of which 1943 were applicable to this implementation. Of the applicable tests, 74 were withdrawn due to errors in the tests. Of the remaining applicable correct tests, all (1875) passed, and no anomaly was discovered.			

DTIC
ELECTED
1985



Prepared for:

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



Report OIT/PSTC-84/501

Approved For	
Prepared By	X
Checked By	
Reviewed By	
Classified	
Controlled	
Excluded	
Other	
Notes	
and/or	
Initial	

A-1

Ada® COMPILER VALIDATION
SUMMARY REPORT

DEPARTMENT OF THE ARMY,
COMMUNICATIONS ELECTRONICS
COMMAND
ALS AdaVAX Version 1.1
VAX-11/780
using VAX/VMS Release 3.6

December 31, 1984

Ada is a registered trademark of the U.S.
Government (Ada Joint Program Office)

OFFICE OF SOFTWARE DEVELOPMENT
AND INFORMATION TECHNOLOGY

ABSTRACT

Under US Army Contract DAAK80-C-0507, SofTech, Incorporated has developed an Ada compilation system for the Department of the Army, Headquarters USA, Communications Electronics Command. The compiler, ALS AdaVAX version 1.25, was tested against version 1.4 of the Ada Compiler Validation Capability (ACVC) test suite on a VAX-11/780 operating under VAX/VMS version 3.6. Version 1.4 of the test suite contained 2185 tests, of which 1943 were applicable to this implementation. Of the applicable tests, 74 were withdrawn due to errors in the tests. Of the remaining applicable correct tests all (1875) passed, and no anomaly was discovered.

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	7
2.2.1 Class B Test Procedures	7
2.2.2 Class B Test Results	7
2.3 Class C Testing	8
2.3.1 Class C Test Procedures	8
2.3.2 Class C Test Results	8
2.4 Class D Testing	8
2.4.1 Class D Test Procedures	8
2.4.2 Class D Test Results	8
2.5 Class E Testing	8
2.5.1 Class E Test Results	9
2.6 Class L Testing	9
2.6.1 Class L Test Procedures	9
2.6.2 Class L Test Results	9
3. COMPILER NONCONFORMANCES	10
4. ADDITIONAL INFORMATION	11
4.1 Compiler Parameters	11
4.2 Testing Information	12
4.2.1 Pre-Test Procedures	12
4.2.2 Control Files	12
4.2.3 Test Procedures	12
4.2.4 Test Analysis Procedures	12
4.2.5 Performance Information	13
4.2.6 Description of Errors in Withdrawn Tests	13
4.2.7 Description of Inapplicable Tests	16
4.2.8 Information Derived from the Tests	18
4.2.9 Anomalous Test Results	19
5. SUMMARY AND CONCLUSIONS	20

LIST OF EXHIBITS

	Page
1. Summary of Test Performance	6

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-11/780
Operating System: VAX/VMS Version 3.6
Host Disk System: RP07
Target Machine: VAX-11/780
Operating System: VAX/VMS Version 3.6
Language Version: ANSI/MIL-STD-1815A Ada
Translator Name: ALS AdaVAX
Translator Version: 1.25
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 December 3, 1984 through December 7, 1984 at Waltham, Massachusetts 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 used for the validation should be sent to:

Validation Program Manager
Federal Software Testing Center
5203 Leesburg Pike, Suite 1100
Falls Church, VA 22041-3467

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.

Class B tests are illegal programs. These tests pass 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. These tests pass if errors are detected prior to beginning execution of the main program.

Class C tests consist of executable self-checking programs. These tests pass 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.

CUSTOMER: The agency requesting the validation (Department of the Army).

HOST: The computer(s) on which the compiler executes is/are VAX-11/780.

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
(ANSI/MIL-STD-1815A).

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 is/are
VAX-11/780.

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 specifi-
cation. 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 ALS AdaVAX compiler passed all applicable correct tests.

There were 236 tests in the suite that were processed and found to be not applicable to the ALS AdaVAX compiler (see Section 4.2.7).

In addition, 74 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 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. Of the 58 class A tests, 56 passed and 1 was found to be inapplicable. Of these 1 was withdrawn due to errors in the test.

- . 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. ALS AdaVAX 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. ALS AdaVAX rejected the body because of the instantiations.
- . 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.

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 ALS AdaVAX 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. ALS AdaVAX rejected all of these strings.
- . D29002K-B.ADA: This test declares 1000 identifiers and was passed by ALS AdaVAX.
- . 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. ALS AdaVAX passed all these tests.
- . 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. ALS AdaVAX 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.

- . CE2102D-B, E, F and G were inapplicable because the implementation does support modes IN_FILE, OUT_FILE, and INOUT_FILE, and also the procedures RESET and DELETE.
- . CE2107A-B, B, C, D, E, CE2110B-B, D, CE3111B-B, C, CE3114B-B and CE3115A-B 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.
- . CE2410A-B was inapplicable because of the file specification in the call to TEXT_IO.CREATE.
- . CE3605A-B was inapplicable because this test requires the implementation to support text lines greater than 255 characters.
- . CE3708A-B was inapplicable because this test at line 24 requires the implicit conversion of the literal 36382 into INTEGER (due to ident_int). This conversion causes NUMERIC_ERROR to be raised because the implementation cannot represent the value as an INTEGER.

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

4.2.7 Description of Inapplicable Tests

The following tests in version 1.4 of the ACVC were considered inapplicable for this implementation for the reasons given below:

- . C24113F-B, C24113G-B, C24113L through C24113Y-B, C35705F through C35705Y-B, C35706F through C35706Y-B, C35707F through C35707Y-B, C35708F through C35708Y-B, C35802F through C35802Y-B, C45241F through C45241Y-B, C45421F through C45421Y-B, C45424F through C45424Y-B, and C45621F through C45621Z-B were inapplicable because they contain tests that exceed the machine capacity using DIGITS precision greater than MAX_DIGITS = 9.
- . C24113H through C24113K-B were inapplicable because these tests contain literals that exceed the maximum length of 120 characters supported by the ALS AdaVAX compiler.
- . B52004E-AB, B55B09D-AB, B86001CR-AB, C34001D-B and C55B07B-AB were inapplicable because the compiler does not support SHORT_INTEGER.
- . C34001F-B, C35702A-AB and B86001CP-AB were inapplicable because the compiler does not support SHORT_FLOAT.
- . B86001DT-AB was inapplicable because the compiler does not support LONG_LONG_INTEGER.
- . C35508B-B was inapplicable because of optimization performed by the ALS AdaVAX compiler.
- . C86001E-B was inapplicable because package SYSTEM is used by package TEXT_IO.
- . C910AHA-B was inapplicable because it attempts to assign SPYNUMB outside range of NATURAL.
- . AE2101C-B, CE2201D-B, CE2201E-B, CE2202A-B and CE2401D-B were inapplicable because this implementation does not allow instantiation of DIRECT_IO or SEQUENTIAL_IO with unconstrained array types and record types with discriminants.

- . In C55B15A-B the CONSTRAINT_ERROR in line 89 should be changed to allow NUMERIC_ERROR to be raised if LOWERBOUND is less than INTEGER'LAST.
- . In C87B10A-B there are literal values that are outside an integer base type for some implementations.
- . 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.
- . An array aggregate with named associations and an 'others' choice must be qualified in an assignment context: C87B31A-B.
- . In C930BDA-B TASKING_ERROR should be raised; not PROGRAM_ERROR.
- . C94004A-B assumes report elaborated before package body.
- . In C95008A it was possible for an entry call to call a terminated task, depending on the implementation.
- . In 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.
- . The exception handler in lines 87 to 89 does not reflect that exception INCOMPLETE is raised by inner exception handlers for USE_ERROR. These exceptions will be handled by the OTHERS choice (incorrectly) resulting in "failed". An additional exception handler "WHEN INCOMPLETE => RAISE;" should be added before line 88: CE3103A-B.
- . In CE3804E-B the value 1.35 is incorrectly used for a model number.

- . 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: C38104A-B.
- . The (model) interval used in the test of C (lines 151-152) is too narrow: C45321A,B,..Y-B.
- . The (model) interval used in the test of C (lines 181-182) is too narrow: C45521A,B,..Z-B.
- . The number 23.4 used in lines 28 and 33 is neither a model number of the float subtype FLT nor the anonymous type derived in line 15 (LRM 3.5.7(11)). A model number should have been used instead of 23.4 (e.g. 23.5): C52001B-AB.
- . In line 76, INTEGER'LAST is compared with SYSTEM.MAX_INT without allowing (by a special exception handler) the implicit conversion of SYSTEM.MAX_INT to INTEGER (before comparison) to raise NUMERIC_ERROR. This is an unintended omission in the test program. Line 136 may also (correctly) raise NUMERIC_ERROR when trying to implicitly convert W_LIT to INTEGER: C52007A-B.
- . 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.
- . 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 exception, and execution terminates abnormally: C52104G-AB.ADA and C52104Q-AB.ADA.
- . In C52103X-B the slice assignment in lines 125 to 127 may raise NUMERIC_ERROR in the evaluation of the slices or the length test, prior to assignment. Hence, the check performed in lines 147 to 173 may fail because no values have been assigned to the four elements of ARR42 that are tested. The check in lines 147 to 173 should be performed only if no exceptions are raised during the slice assignment of the lines 125 to 127.

. illegal external file name2:
MUCH-TOO-LONG-NAME-FOR-A-FILE

. SYSTEM.PRIORITY'FIRST: 1

. SYSTEM.PRIORITY'LAST: 15

4.2 Testing Information

Tests were compiled/executed at Waltham, Massachusetts.

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

SofTech, Inc., under contract to the Department of the Army, Communications Electronics Command, 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 four (4) VAX-11/780 used for testing the ALS AdaVAX compiler. Then all the remaining tests were grouped into batch jobs by class and by chapter and processed by four (4) VAX-11/780s.

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. 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: 9
- . SYSTEM.MIN_INT: -2147483648
- . SYSTEM.MAX_INT: 2147483647
- . predefined numeric types: INTEGER, LONG_INTEGER, FLOAT, LONG_FLOAT
- . INTEGER'FIRST: -32768
- . INTEGER'LAST: 32767
- . source character set: ASCII
- . extended ascii chars: "abcdefghijklmnopqrstuvwxyzi!\$?@[\]^'{}~"
- . non-ascii char type: (NON_NULL)
- . TEXT_IO.COUNT'LAST: 32767
- . TEXT_IO.FIELD'LAST: 32767
- . illegal external file name: BAD-CHARACTER"



3. COMPILER NONCONFORMANCES

For this implementation there were no nonconformances 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.

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 60 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

All 60 class L tests passed.

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 5 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 776 remaining class B tests passed.

Because all errors were not detected when compiling the original tests, the following 30 tests were modified and divided into 131 separate units by removing the detected errors; the modified tests were then resubmitted to see if the remaining errors would be detected:

B22003A.ADA	B23002A.ADA	B29001A-B.ADA	B33004A.ADA
B36101A-AB.ADA	B36102A.ADA	B36201A-B.ADA	B37004C-B.ADA
B37004H-B.ADA	B37301A.ADA	B37301B.ADA	B37303A.ADA
B44001A-B.ADA	B45205A-AB.ADA	B54A25A-AB.ADA	B55A01A-AB.ADA
B61005A-B.ADA	B64101A-B.ADA	B65001A.ADA	B67001A-B.ADA
B74207A-B.ADA	B97101A-AB.ADA	B97101E-AB.ADA	B97102A-AB.ADA
BA2001E.ADA	BC10AEB-B.ADA	BC1202B-AB.ADA	BC1202C-AB.ADA
BC1202D-AB.ADA	BC1207A-B.ADA		

Test Class	Pr	NA	Er	P	A	F	FE	FC	Fs
A	58	1	1	56	0	0	0	0	0
B	784	5	3	776	0	0	0	0	0
C	1262	230	70	962	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	60	0	0	60	0	0	0	0	0
Total	2185	236	74	1875	0	0	0	0	0

EXHIBIT 1: Summary of Test Performance

5. SUMMARY AND CONCLUSIONS

The Ada Validation Office identified 2185 of the ACVC version 1.4 tests as being being potentially applicable to the validation of the Department of the Army, Communications Electronics Command, ALS AdaVAX compiler hosted on the VAX-11/780. Of these, 74 were withdrawn due to test errors, and 236 were determined to be inapplicable after they were processed. The compiler passed the remaining 1875 tests.

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

END

FILMED

6-85

DTIC