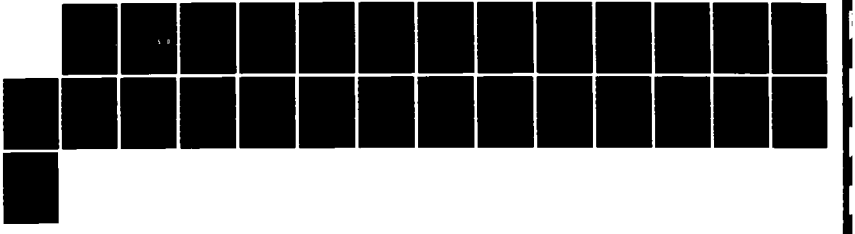
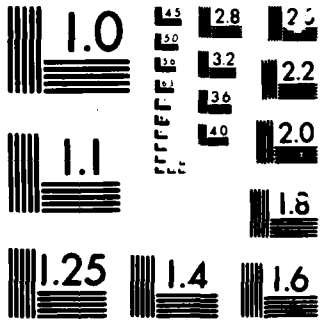
AD-A167 214 ADA (TRADE NAME) VALIDATION SUMMARY REPORT: HONEYWELL LARGE SYSTEMS GCOS-.. (U) FEDERAL SOFTWARE MANAGEMENT SUPPORT CENTER FALLS CHURCH VA 20 DEC 85 1/1

UNCLASSIFIED F/G 9/2 NL





MICROCOPY CHART

AD-A167 214

(2)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1 REPORT NUMBER / 2 GOVT ACCESSION NO. / 3 RECIPIENT'S CATALOG NUMBER	
4 TITLE (and Subtitle) Ada Validation Summary Report: Honeywell Large Systems, GCOS-8 Compiler, Version 3.1, DPS-88 Using SR2300 July 1985 Release IFAD B.4 SMAS B0 Version and the DPS-88.	5 TYPE OF REPORT & PERIOD COVERED 20 Dec. '85 to 20 Dec. '86
	6 PERFORMING ORG. REPORT NUMBER
7 AUTHOR(s) Federal Software Management Support Center	8 CONTRACT OR GRANT NUMBER(s)
9 PERFORMING ORGANIZATION NAME AND ADDRESS Federal Software Management Support Center Office of Software Development & Information Technology Two Skyline Place, Suite 1100-5203 Leesburg Pike Falls Church, VA 22041-3467	10 PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11 CONTROLLING OFFICE NAME AND ADDRESS Ada Joint Program Office 1211 S. Fern Street, Rm. C-107 Arlington, VA 22202	12 REPORT DATE 20 December 1985
	13 NUMBER OF PAGES 24
14 MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Federal Software Management Support Center	15 SECURITY CLASS (of this report) UNCLASSIFIED
	15a DECLASSIFICATION DOWNGRADING SCHEDULE

16 DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

DTIC ELECTE APR 3 0 1986 S D

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, Ada Compiler Validation Capability, ACVC, Validation Testing, Ada Validation Office, AVO, Ada Validation Facility, AVF, ANSI/MIL-STD-1815A, Ada Joint Program Office, AJPO.

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

See attached.

DTIC FILE COPY

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

86 4 29 134

ABSTRACT

The purpose of this Validation Summary Report is to present the results and conclusions of performing standardized tests on the Honeywell GCOS-8 compiler version 3.1. On-site testing was performed by the Federal Software Management Support Center (FSMC)--an Ada Validation Facility--in accordance with current Ada Validation Office policies and procedures, on 2 - 4 December, 1985 at Honeywell Large Systems in Phoenix, Az.

The suite of tests known as the Ada Compiler Validation Capability (ACVC), Version 1.6, 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 1.6 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 Honeywell compiler GCOS-8 version 3.1 using SR2300 was tested with version 1.6 of the ACVC validation tests. Version 1.6 of the test suite contains 2162 tests of which 66 were withdrawn and 175 were inapplicable to this implementation. All of the 1921 remaining tests were passed.

The Honeywell Compiler was tested on the system listed in the table below. The following table represents the tested target/host relationships:

Host	Target	Series of Testing	Site of Execution
DPS-88	DPS-8	Full ACVC	Phoenix,Az.

In addition, the vendor certifies that the GCOS-8 Ada compiler using the SR2300 operating system is compatible with the following Honeywell equipment:

DPS-8/47 DPS-8/49

DPS-8/52 DPS-8/62

DPS-8/70 DPS-8/90

Ada* COMPILER VALIDATION SUMMARY REPORT:

Honeywell LARGE SYSTEMS
GCOS-8 Compiler
Version 3.1
DPS-88
Using SR2300 July 1985 Release IFAD B.4 SMAS BO Version
and the DPS-88

December 20, 1985

Prepared by:

Federal Software Management Support Center
Office of Software Development
and Information Technology
Two Skyline Place, Suite 1100
5203 Leesburg Pike
Falls Church, VA 22041-3467

Prepared for:

Honeywell Large Systems
13430 N. Black Canyon Highway
Phoenix, Arizona 85029

Ada Joint Program Office
1211 South Fern St.
Arlington, Va 22202

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

1366

for L Arnold Johnson
Richard C. Harrison
Director
Federal Software Management Support Center

John F Kramer
Dr. John F. Kramer, Jr.
Institute for Defense Analyses

Virginia L. Castor
Virginia L. Castor
Director
Ada Joint Program Office

TABLE OF CONTENTS

		Page
1.	Introduction	1
	1.1 Purpose of the Validation Summary Report	1
	1.2 Host to Target Relationship Table	2
	1.3 Use of the Validation Summary Report	3
	1.4 References	3
	1.5 Definitions of Terms	4
2.	TEST ANALYSIS	6
	2.1 Class A Testing	6
	2.1.1 Class A Test Procedures	6
	2.1.2 Class A Test Results	6
	2.2 Class B Testing	6
	2.2.1 Class B Test Procedures	6
	2.2.2 Class B Test Results	7
	2.3 Class C Testing	7
	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 Testing Procedures	8
	2.5.2 Class E Test Results	8
	2.6 Class L Testing	8
	2.6.1 Class L Test Procedures	9
	2.6.2 Class L Test Results	9
	2.7 Subset Testing	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 On-Site Data Collection	12
	4.2.4 Test Analysis Procedures	13
	4.2.5 Timing Information	13
	4.2.6 Description of Errors in Withdrawn Tests	13
	4.2.7 Description of Inapplicable Tests	17
	4.2.8 Information derived from the Tests	19
5.	SUMMARY AND CONCLUSIONS	21

1. Introduction

1.1 Purpose of the Validation Summary Report

This report describes the results of the validation testing for the compiler designated as GCOS-8 Version 3.1 for the following configurations:

Host Machines: DPS-88

Operating System: SR2300 July Release 1985 IFAD B.4 SMAS BO

Host Disk Systems: MSU 0451

Target Machines: DPS-88

Language Version: ANSI/MIL-STD-1815A Ada

Translator Name: GCOS-8, 3.1

Validation Test Version: 1.6

The Ada compiler and the Ada linker currently produce assembly source code (GMAP) for the DPS-88. This GMAP source code for each Ada domain and link is assembled via the GMAP assembler before each compilation unit is linked or executed.

The DPS-8 Ada compiler is based on an Ada compiler system supplied to Honeywell Information Systems by Dansk Datamatik Center of Lyngby, Denmark.

Testing of this compiler was conducted by the Federal Software Management Support Center under the supervision of the Ada Validation Office (AVO), at the direction of the Ada Joint Program Office. Testing was conducted from 2 - 4 December, 1985 at Honeywell in Phoenix, Az. All testing was performed 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 Host To Target Relationship Table

The Honeywell GCOS-8 compiler was tested on those systems listed in the table below. The following table represents the tested target/host relationships:

Host	Target	Series of Testing	Site of Execution
DPS-88	DPS-88	Full ACVC	Phoenix, Az

In addition, the vendor certifies that the GCOS-8 Ada compiler using the SR2300 operating system is compatible with the following Honeywell equipment:

DPS-8/47	DPS-8/49
DPS-8/52	DPS-8/62
DPS-8/70	DPS-8/90

1.3 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 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
1211 South Fern Street
Arlington, VA 22202

1.4 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.

1.5 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 Federal Software Management Support Center (FSMC) and developed by the AVF that produces summary test results by reading compiler output in a spool file.

CUSTOMER: The agency requesting the validation Honeywell Large Systems in Phoenix, Az.

FSMC:	Federal Software Management Support Center. In the context of this report the FSMC conducts Ada validations under contract to the AVO as a satellite facility.
HOST:	The computer on which the compiler executes.
IG:	ACVC Implementors' Guide.
RM:	The Ada Language Reference Manual.
STANDARD:	The standard for the Ada language, ANSI/MIL-STD-1815A (1983).
SUBSET TESTS:	A grouping of ACVC tests selected by the FSMC. Each chapter in the ACVC is represented in the subset by between 4 to 7 tests. The subset is used for statistical sampling of the various host and target hardware configurations.
TARGET:	The computers for which the compiler generates object code.
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 Honeywell GCOS-8 compiler passed all applicable tests.

	A	B	C	D	E	L	Total
In suite	61	800	1273	17	8	3	2162
Inapplicable	0	4	169	2	0	0	175
Withdrawn	0	19	47	0	0	0	66
Passed	61	777	1057	15	8	3	1921
Failed	0	0	0	0	0	0	0

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.

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 applicable tests passed.

2.2 Class B Testing

Class B tests check the ability to recognize illegal language usage. All applicable class B tests passes.

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 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 are still not detected, the revision process is repeated until a revised test contains only a single previously undetected illegal construct.

A class 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

Of all the B tests presented to compiler, 1 was found to be inapplicable to this implementation (see Section 4.2.7); 19 tests were found to be incorrect (i.e., a conforming compiler would have failed each of these tests). All remaining class B tests passed.

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

B22003A.ADA	B29001A.ADA	B33001A.ADA	B33004A.ADA
B35101A.ADA	B37301A.ADA	B37302A.ADA	B51001A.ADA
B53009A.ADA	B54A01C.ADA	B55A01A.ADA	B61001R.ADA
B61001W.ADA	B66001C.ADA	B67001C.ADA	B67001D.ADA
B91001A.ADA	B91002A.ADA	B91002B.ADA	B91002C.ADA
B91002D.ADA	B91002E.ADA	B91002F.ADA	B91002G.ADA
B91002H.ADA	B91002I.ADA	B91002J.ADA	B91002K.ADA
B91002L.ADA	B950ABA.ADA	B950ABB.ADA	B97101A.ADA
B97101E.ADA	B97102A.ADA	B97103E.ADA	B97104G.ADA
BC10AEA.ADA	BC10AEB.ADA	BC10AED.ADA	BC10AFA.ADA
BC1202A.ADA	BC1202B.ADA	BC12ACA.ADA	BC20ABA.ADA

All illegal constructs were detected.

2.3 Class C Testing

Class C tests check that legal Ada programs are correctly compiled and executed by an implementation.

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 which were applicable passed.

2.4 Class D Testing

Class D tests are executable tests used to check an implementation's compilation and execution capacities.

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

All applicable class D tests passed.

2.5 Class E Testing

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

2.5.1 Class E Testing Procedures

Each class E test is separately compiled and executed. The tests are self-checking and produce pass/fail messages.

2.5.2 Class E Test Results

All of the applicable tests 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.

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 checked to see if they are correct and applicable to the implementation. Tests that are inapplicable or that do not conform to the Ada standard are withdrawn.

2.6.2 Class L Test Results

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

2.7 Subset Testing

There were no subset tests run for this validation.

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: 126 characters.
- maximum digits value for floating point types: 17
- SYSTEM.MIN_INT: -2_361_183_241_434_822_606_848
- SYSTEM.MAX_INT: 2_361_183_241_434_822_606_847
- predefined numeric types: INTEGER, FLOAT, LONG_INTEGER and LONG_FLOAT.

  INTEGER_FIRST: -34_359_738_368

  INTEGER_LAST: 34_359_738_367
- LONG_INTEGER'FIRST: -2_361_183_241_434_822_606_848
- LONG_INTEGER'LAST: 2_361_183_241_434_822_606_847
- Source character set: ASCII
- Extended ASCII chars: abcdefghijklmnopqrstuvwxyz !$%?@[\]^'{}~"
- non-ascii char type: (NON_NULL)
- TEXT_IO.COUNT'LAST: 34_359_738_367
- FIELD'LAST: 75
- illegal external file name1: f@lename
- illegal external file name2: nametoolongforafilename

. SYSTEM.PRIORITY'FIRST: 7

. SYSTEM.PRIORITY'LAST: 7

4.2 Testing Information

Complete ACVC test runs were compiled/executed at Honeywell Large Systems in Phoenix, Az.

4.2.1 Pre-Test Procedures

Prior to on-site 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 was prepared and brought to the testing site.

For pre-validation, Honeywell Large Systems compiled the ACVC on the DPS-88 in Phoenix, Az. and executed them on the DPS-88.

4.2.2 Control Files

Honeywell provided command procedures that compiled and executed tests automatically.

4.2.3 On-site Data Collection

All test hardware was located in Phoenix, Az

All applicable tests in the ACVC were compiled on the DPS-88 and executed on a DPS-88.

4.2.4 Test Analysis Procedures

On completion of testing the base system, all results were analyzed for failed Class A, C, D, E, and L tests, 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 elapsed time required for compiling the non-executable tests and compiling, linking, and running the executable tests.

DPS-88	36:47	Full suite - on the DPS-88 targeted to the DPS-88.

4.2.6 Description of Errors in Withdrawn Tests

The following tests in version 1.6 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 1.6. (Please note that this test may be considered correct and may appear in the future Versions of the ACVC, including Version 1.6.)

- BC1013A-B: The declaration of equality in lines 86-87 is illegal because the parameter type T declared in line 11 is not a limited type (LRM 6.7-4).

- C45521A, C45521B, ... C45521Y (25 tests) : Cases C and I define the model interval for the result too narrowly.

- C48005C-B: Lines 38 and 63 of this test should check that the value of the designated object is null.

- C64103C-B: This test should raise CONSTRAINT_ERROR during the conversion at line 179.

- C64103D-B: This test involves a CONSTRAINT_ERROR vs. NUMERIC__ERROR issue that is to be resolved by the Language Maintenance Committee.

- C64105E-AB: For case E, ensure that non-null dimensions of formal and actual parameters belong to both index subtypes (see AI-00313).

- C64105F-AB: For case E, ensure that non-null dimensions of formal and actual parameters belong to both index subtypes (See AI-00313).

- B67001A-B: Line 414 is missing the "BEGIN NULL; END;" needed to complete the block beginning at line 389 (case H).

- B67004A-B: The default name for a formal generic equality function should not be allowed to be "/=" unless an expanded name is used.

- C93005A, C93005B, and C93005C: These tests contain a declaration of an interger variable whose initialization is solely for the purpose of raising an exception. Some compilers will not raise this exception due to their optimization.

- C93007B-B: This test should check for PROGRAM_ERROR rather than TASKING_ERROR (See AI-000149).

- CA1011A*-B: The test objective should be reversed to be consistent with AI-00199.

- CA1108A-B: A pragma ELABORATE is needed for OTHER_PKG at line 25.

- CA1108B-B: A pragma ELABORATE is needed for FIRST_PKG at line 39 and for LATER_PKG at line 49.

- CA2009B-B, CA2009E-B: The repetition of the main procedure after the subunit body makes the subunit body obsolete; therefore, an attempt to execute the main procedure will fail.

- CA2009F*-B: The file CA2009F2-B is missing from the test suite.

- BC3204A-B, BC3204B-B, BC3204C*-B, BC3204D-B, BC3205A-B, BC3205B-B, BC3205C-B, BC3205D*-B, BC3405B-B: Instantiations with types that have default discriminants are now legal (AI-00037).

- CE3603A-B: The last case is inconsistent with AI-00050. If the string argument is null, no attempt to read is made and END_ERROR is not raised.

- CE3604A-B: Cases 5, 8, 9, and 11 are inconsistent with AI-00050. SKIP__LINE is called only if the end of the output string has not been met.

- CE3704M-B: A superfluous SKIP_LINE causes the input and output operations to be out of synchronization.

- B38105B-AB: This test requires a specific interpretation of the Language Reference Manual (LRM) regarding whether an incomplete type can have discriminate constraints before the full type declaration; this interpretation is not fully supported by the LRM or Language Maintenance Committee.

- C48006B-B: This test requires a specific interpretation of the Language Reference Manual (LRM) regarding whether an incomplete type can have discriminate constraints before the full type declaration; this interpretation is not full fully supported by the LRM or Language Maintenance Committee.

- B74103F-B: This test hinges on whether or not a generic formal type declaration declares a type. This matter will be debated by the Language Maintenance Committee in November.

- B74207A-B: This test requires a specific interpretation of the Language Reference Manual (LRM) regarding whether an incomplete type can have discriminate constraints before the full type declaration; this interpretation is not fully supported by the LRM or Language Maintenance Committee.

- CA1003B-AB: A compilation that contains an illegal compilation unit may now be rejected as a whole (see AI-00255/05).

- BC3503A-B: This test requires a specific interpretation of the Language Reference Manual (LRM) regarding whether an incomplete type can have discriminate constraints before the full type declaration; this interpretation is not fully supported be the LRM or Language Maintenance Committee.

- CE2107E-B: This test has a variable, TEMP_HAS_TRUE, that needs to be given an initial value of TRUE.

- C35904A-B: The elaborations of the subtype declarations for SFX3 and SFX4 in this test raise NUMERIC_ERROR in some implementations. The exception is raised on the conversion of the real literals 2.0 and 5.0 to the base type of FIX.

- BA2001E*-AB: LRM 10.2(5) states that "simple names of all subunits that have the same ancestor library unit must be distinct identifiers." This test requires that the above conditions be checked when the stub is declared; but since the LRM uses the term "subunit", it is not clear that the check must be made then, as opposed to when the subunit is compiled. (There may be an LMC ruling regarding this issue.)

- BC3220B-B: This test assumes that instantiated types may be static. This assumption has been questioned, and the matter will be considered by the LMC.

4.2.7 Description of Inapplicable Tests

B52004E, B55B09D, B86001CR, C34001D, and C55B07B were inapplicable because the implementation does not support SHORT_INTEGER.

B86001CP, C34001F, and C35702A were inapplicable because the implementation does not support SHORT_FLOAT.

The following tests were inapplicable because they exceed the accuracy of the max-digits definition of the target implementation:

C35705N through C35705Y (12)
C35706N through C35706Y (12)
C35707N through C35707Y (12)
C35708N through C35708Y (12)
C35802N through C35802Y (12)
C45241N through C45241Y (12)
C45321N through C45321Y (12)
C45421N through C45421Y (12)
C45424N through C45424Y (12)
C45621N through C45621Z (13). (=121 tests, total)

C24113I through C24113M were inapplicable because the implementation does not support the line length specified in the tests.

C24113N through C24113Y were inapplicable due to the definition of max-digits for this implementation.

C96005B - This test checks to find a difference between DURATION'BASE'FIRST and DURATION'FIRST. If no difference exists (as is the case in the implementation) the test is inapplicable.

CE2401D - This test fails because the maximum element size allowed for external sequential files is 2**18-1.

D29002K, D64005G - Tests exceeds compiler storage capacity for this implementation.

C64103A - This test is invalid in that LG_FLOAT'LARGE may be in the range of SM_FLOAT safe numbers and therefore NUMERIC_ERROR should not be raised. See LRM 4.5.7 para 7 and 8.

CA2009C0 - Compiled but not executed because the compiler uses the restriction allowed by ADA LRM P10.3(9)..."An implementation may also require that subunits of a generic unit be part of the same compilation".

CE2107A,CE2107B,CE2107C,CE2107D,CE3114B,CE3115A, CE2110B,CE2111D, CE3111A,CE3111B,CE3111C,CE3111D,CE3111E - Test execution states "NOT APPLICABLE" because multiple internal files may not access the same external or temporay file.

CE2108A,CE2108C,CE3112A - Test execution states "NOT APPLICABLE" because temporary files have no name.

C87B62A - This test is inapplicable because 'SIZE clause is not supported.

C87B62C - This test is inapplicable because 'SMALL clause is not supported.

CE2102D, CE2102E, CE2102F - These tests check that the appropriate exceptions are raised when attempting I/O operation modes IN__FILE, OUT__FILE, and INOUT_FILE, respectfully, when not supported; however, Honeywell GCOS-8 supports these modes.

CE2102G - This test checks that the appropriate exceptions are raised for procedures RESET and DELETE when not supported; however, Honeywell GCOS-8 supports these procedures.

CE2102C - This test is inapplicable because illegal file names may not exist.

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 compiler did not reject the compilation unit at compile 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 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 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 compiler passed these 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 compiler did not evaluate 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 compiler evaluates all subaggregates for identical bounds.

- 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 compiler did not raise 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. This test showed that such library tasks continued to execute even after the main program terminates and then terminated appropriately by themselves.

- 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 compiler does allow generic declarations and bodies to be compiled in completely separate compilations.

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

5. SUMMARY AND CONCLUSIONS

ACVC version 1.6 comprises 2162 tests, of which 66 were withdrawn due to errors. The Federal Software Management Support Center (FSMC), an AVF, identified 1921 of the remaining tests to be applicable to Honeywell Large Systems GCOS-8 AdA compiler. The compiler passed all of these tests, compiled on the DPS-88 targeted to the DPS-88.

The FSMC considers these results to show acceptable conformity to the Ada Language Standard.

END

FILMED

6-86

DTIC