



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



2



AD-A178 154

Ada[®] Compiler Validation Procedures and Guidelines

Version 1.1
1 January 1987

DTIC FILE COPY

Prepared by the:
Ada Joint Program Office
3D139 (1211 Fern, C107)
The Pentagon
Washington, DC 20301-3081

DTIC
SELECTED
MAR 25 1987
A

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

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETEING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Ada Validation Procedures and Guidelines Version 1.1		5. TYPE OF REPORT & PERIOD COVERED January 1987 to
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Ada Joint Program Office 3D139 1211 S.Fern, St. C-107 The Pentagon Washington, D.C. 20301-3081		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION AND ADDRESS Ada Joint Program Office 3D139 1211 S. Fern St. C-107 The Pentagon Washington, D. C.		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Ada Joint Programming Office United States Department of Defense Washington, D.C. 20301-3081		12. REPORT DATE 1 January 87
		13. NUMBER OF PAGES 25
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Ada Joint Program Office 1211 S. Fern Street C-107 The Pentagon Washington, D.C. 20301-3081		15. SECURITY CLASS (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A
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. KEYWORDS (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.		

EXECUTIVE SUMMARY

This document is intended for all Ada PROGRAMMING LANGUAGE users and compiler VENDORS who have an interest in Ada VALIDATIONS. It defines commonly used Ada VALIDATION terms; outlines the organizational structure to manage, coordinate, and direct the Ada VALIDATION process; lists steps in the process; and provides guidance to Department of Defense (DoD) program managers on the acquisition, use and maintenance of Ada COMPILERS.

Terms that are specific to this document are defined in Section 2.

The Ada CERTIFICATION SYSTEM is established by the Ada Joint Program Office (AJPO) and consists of rules of procedure and a hierarchical management structure for carrying out CONFORMITY TESTING. The Ada CERTIFICATION BODY is the governing body that directs, manages, and administers Ada VALIDATION. This body consists of the Director, AJPO, the Ada Validation Organization (AVO), and the Ada Validation Facilities (AVFs). The Director, AJPO provides overall direction and is the single point DoD authority for organizing, managing, directing and administering all matters concerning the Ada PROGRAMMING LANGUAGE and for authorizing VALIDATION CERTIFICATES. The AVO provides administrative and technical assistance to the Director and AVFs. The AVFs implement a mechanism to determine COMPLIANCE to the Ada PROGRAMMING LANGUAGE within DoD.

The rules of procedure that are outlined by the VALIDATION process are summarized here in nine steps that must be completed by a VENDOR and the Ada CERTIFICATION BODY in order to issue a VALIDATION CERTIFICATE. These steps are:

1. Obtaining the current Ada COMPILER VALIDATION CAPABILITY (ACVC) test suite;
2. Submitting a notice of intent to validate a COMPILER to an AVF;
3. Negotiating a formal agreement for VALIDATION services with an AVF;
4. Submitting a DECLARATION OF CONFORMANCE and test results to an AVF;
5. Resolving test issues with the AVF;
6. Reviewing and responding to the draft VALIDATION SUMMARY REPORT (VSR);
7. Completing on-site testing successfully;

Table of Contents

EXECUTIVE SUMMARY.....	i
TABLE OF CONTENTS.....	iii
SECTION 1: INTRODUCTION.....	1
SECTION 2: TERMS AND DEFINITIONS.....	2
SECTION 3: ORGANIZATION AND RESPONSIBILITIES.....	6
SECTION 4: BASE COMPILER VALIDATION.....	8
SECTION 5: DERIVED COMPILER VALIDATION.....	13
SECTION 6: Ada VALIDATION FACILITIES.....	17
SECTION 7. Ada COMPILER VALIDATION CAPABILITY.....	18
SECTION 8: DoD GUIDELINES FOR Ada USE.....	21
Appendix A: DECLARATION OF CONFORMANCE.....	A-1

Section 1

INTRODUCTION

The Ada PROGRAMMING LANGUAGE was initially designed to support software development for embedded computer systems. Embedded computer systems form a part of larger systems that are primarily computational, such as military weapons systems. A programming language was also needed that reduced software development and maintenance costs, provided portability of software and programmers, and encouraged sound software engineering practices for applications such as C3I, multilevel security, and nonembedded computer processing. No other programming language was designed to fully support this requirement. For this reason, the United States Department of Defense (DoD) sponsored the development of the Ada PROGRAMMING LANGUAGE, and adopted what is now the ANSI/MIL-STD-1815A Ada PROGRAMMING LANGUAGE standard.

To support implementation of the Ada PROGRAMMING LANGUAGE, DoD established the Ada Joint Program Office (AJPO). The Director, AJPO is the single point DoD authority within DoD for organizing, managing, directing and administering all matters concerning the Ada PROGRAMMING LANGUAGE. Among the responsibilities assigned to the AJPO is the function of ensuring that Ada COMPILERS faithfully implement the Ada PROGRAMMING LANGUAGE standard. To perform this function the AJPO designates the Ada Validation Organization (AVO) to provide administrative and technical support, and Ada Validation Facilities (AVFs) to perform Ada COMPILER VALIDATIONS. Collectively, these three elements constitute the Ada CERTIFICATION BODY. The Director, AJPO provides overall direction, the AVO provides administrative support to ensure consistent practices, and the AVFs conduct validation to determine COMPLIANCE to the Ada PROGRAMMING LANGUAGE.

This document outlines the process established by the AJPO to perform Ada COMPILER VALIDATIONS. VALIDATION for an Ada COMPILER is the process of CONFORMITY TESTING of a candidate compiler to the Ada PROGRAMMING LANGUAGE and, when COMPLIANCE is demonstrated through VALIDATION testing, issuing a VALIDATION CERTIFICATE. The goal of Ada VALIDATION is to ensure COMPLIANCE to the standard and prevent the proliferation of subsets, supersets, or dialects of the Ada PROGRAMMING LANGUAGE. Users are cautioned that VALIDATION does not imply or warrant the use of an Ada COMPILER for any application, nor does it provide an evaluation of compiler efficiency or performance. VALIDATION is strictly a determination of COMPLIANCE that leads to a judgment of CONFORMITY.

Section 2

TERMS AND DEFINITIONS

Terms that are specific to this document are defined in this section and are capitalized where they appear in the document. Reference to these definitions should be made whenever there is a question as to the specific meaning of a defined term.

Ada: PROGRAMMING LANGUAGE: The language defined by ANSI/MIL-STD-1815A and its successors.

Ada COMPILER: A system (in a loadable or executable code form) which translates Ada source programs into object code that, when loaded with the TARGET RUN-TIME SYSTEM, executes on a target computer in a manner that is in COMPLIANCE with the Ada PROGRAMMING LANGUAGE. The term Ada COMPILER may be used in describing language processors which are not completely conforming or are not making a claim of conformance provided that there is a precise, easily visible statement of their non-conformance at the same time and in the same context as where the term is used.

Ada COMPILER VALIDATION CAPABILITY (ACVC): The set of Ada programs that tests the CONFORMITY of a candidate Ada COMPILER to the Ada PROGRAMMING LANGUAGE, as well as the documentation and tools that facilitate this CONFORMITY TESTING.

APPLICABLE ACVC TESTS: The set of tests in an ACVC version that are determined by the Ada CERTIFICATION BODY to be applicable to the VALIDATION of a specific candidate BASE COMPILER.

APPLICATION-SPECIFIC RUN-TIME LIBRARY: The subset of a Run-Time Library that is required for execution of a specific application.

BASE COMPILER: An Ada COMPILER for which a current VALIDATION CERTIFICATE exists.

BASE CONFIGURATION: The specific CONFIGURATION on which the BASE COMPILER is tested by an Ada Validation Facility (AVF) as part of the VALIDATION process.

CERTIFICATION BODY: An impartial body, governmental or non-governmental, possessing the necessary competence and reliability to operate a certification system, and in which the interests of all parties concerned with the functioning of the system are represented. [ISO GUIDE 2: 1983 and draft ISO/IEC GUIDE 2-8.23: 1986]

CERTIFICATION SYSTEM: A system having its own rules of procedure and management for carrying out conformity certification. [ISO

GUIDE 2: 1983 and draft ISO/IEC GUIDE 2-8.2.1: 1986]

COMPLIANCE: The state describing correct compilation and execution of all APPLICABLE ACVC TESTS in the applicable version of the ACVC. [Note: For purposes of this document, COMPLIANCE is a practical measure of CONFORMITY.]

COMPILER OWNER: The VENDOR who signs the DECLARATION OF CONFORMANCE and who submits a candidate Ada COMPILER for VALIDATION, or the third party which is declared by the validating VENDOR on the DECLARATION OF CONFORMANCE to be the owner of record for the candidate Ada COMPILER.

COMPUTER ARCHITECTURE: The set of physical implementations (central processing unit, memory, peripheral devices, and related hardware and software components), or logically equivalent emulator programs, which execute object code in accordance with an Instruction Set Architecture (ISA).

CONFIGURATION: A HOST ARCHITECTURE, host operating system, TARGET ARCHITECTURE, target operating system and TARGET RUN-TIME SYSTEM, (if any).

CONFORMITY: Fulfillment by a product, process or service of all requirements specified [ISO Guide 2: 1983 and draft ISO/IEC Guide 2-8.1.1: 1986]. [Note: Also see section 1.1.2 in ANSI/MIL-STD-1815A]

CONFORMITY TESTING: Testing of a candidate Ada COMPILER by the VENDOR and by an AVF to demonstrate COMPLIANCE to the Ada PROGRAMMING LANGUAGE. This testing includes checking for those characteristics that must be present in a compiler for it to be considered a compiler conforming to the Ada PROGRAMMING LANGUAGE. This also includes testing the implementation of dependent compiler characteristics that must also conform to the Ada PROGRAMMING LANGUAGE, if implemented.

DECLARATION OF CONFORMANCE: A formal statement from a VENDOR assuring that CONFORMITY is realized or attainable on the compilers, configurations and derivations for which he/she is requesting VALIDATION, REVALIDATION, or registration. See Appendix A for the format of a DECLARATION OF CONFORMANCE.

DERIVED COMPILER: One of the following:

- o A BASE COMPILER on an EQUIVALENT CONFIGURATION,
- o A MAINTAINED COMPILER on a BASE CONFIGURATION,
- o A MAINTAINED COMPILER on an EQUIVALENT CONFIGURATION, where any of these pairs originates from a BASE COMPILER and BASE CONFIGURATION pair.

EQUIVALENT CONFIGURATION: Any CONFIGURATION of the same COMPUTER ARCHITECTURE(s) and operating system for which COMPLIANCE is achievable using the same ACVC version used in the VALIDATION of the BASE COMPILER on the BASE CONFIGURATION.

GENERIC TARGET: A hardware and/or software implementation that is equivalent to, or a superset of, the REAL SYSTEM TARGET and is capable of executing all applicable ACVC tests. A GENERIC TARGET is equivalent to the REAL SYSTEM TARGET if it is an EQUIVALENT CONFIGURATION of this target. A superset of a REAL SYSTEM TARGET is one to which the REAL SYSTEM TARGET could be made equivalent by adding more memory, input-output capabilities, co-processors, and instructions.

HOST ARCHITECTURE: The COMPUTER ARCHITECTURE used for execution of an Ada COMPILER.

MAINTAINED COMPILER: A BASE COMPILER which has been changed in any way generally accepted by the software profession to constitute "maintenance" - usually meaning minor change. Complete replacement or addition of some major component of a BASE COMPILER is not considered "maintenance" in this document.

MAJOR UPGRADE: As it applies to a specific system or subsystem, is the redesign or substantial addition of hardware, the revision or addition of more than one third of the software, the redesign of the software architecture, or a substantial addition of new software functions.

PROJECT-VALIDATED COMPILER: A VALIDATED COMPILER which is baselined for a project in accordance with applicable DoD software life-cycle management policies, procedures, and practices.

REAL SYSTEM TARGET: A computer hardware component of a system that has been designed to comply with the operational form, fit, and functional specifications of the system computer resources which may execute object code generated by a VALIDATED COMPILER for an associated GENERIC TARGET. The REAL SYSTEM TARGET may possess characteristics specific to an application that have no counterpart in the GENERIC TARGET.

RESTRICTED TARGET: A REAL SYSTEM TARGET on which not all ACVC tests can be executed but which can execute object code generated by a VALIDATED COMPILER for an associated GENERIC TARGET in combination with APPLICATION-SPECIFIC RUN-TIME LIBRARIES and/or run-time support. Note: The VALIDATED COMPILER for a GENERIC TARGET cannot generate different code for the associated RESTRICTED TARGET.

REVALIDATION: The VALIDATION of a previously VALIDATED COMPILER to renew its Ada COMPILER status.

TARGET ARCHITECTURE: The COMPUTER ARCHITECTURE used for execution of object code generated by an Ada COMPILER.

TARGET RUN-TIME SYSTEM: The set of sub-programs, which may be invoked by linking, loading, and executing object code generated by an Ada COMPILER. If these sub-programs use or depend upon the services of an operating system, then the TARGET RUN-TIME SYSTEM includes those portions of that operating system.

VALIDATION: The process of checking the CONFORMITY of an Ada COMPILER to the Ada PROGRAMMING LANGUAGE and of issuing certificates indicating COMPLIANCE of those compilers that have been successfully tested.

VALIDATION CERTIFICATE: The certificate issued by authority of the Ada Joint Program Office that formally acknowledges COMPLIANCE of a BASE COMPILER.

VALIDATION SUMMARY REPORT (VSR): A report produced by an Ada VALIDATION FACILITY (AVF) containing results that are observed from testing a specific Ada COMPILER or grouping of Ada COMPILERS.

VALIDATED COMPILER: A BASE COMPILER for which a VALIDATION CERTIFICATE is in effect or a DERIVED COMPILER that is currently registered with the Ada Joint Program Office (AJPO). A BASE COMPILER and any derivation of that compiler will be considered a VALIDATED COMPILER only as long as the VALIDATION CERTIFICATE for the BASE COMPILER is in effect. Any compiler that does not have a current VALIDATION CERTIFICATE and qualifies as a DERIVED COMPILER must be registered with the AJPO to be a VALIDATED COMPILER.

VENDOR: The supplier of an Ada COMPILER. [Note: A VENDOR becomes an AVF client when the VENDOR enters into a contract for Ada VALIDATION services.]

Section 3

ORGANIZATION AND RESPONSIBILITIES

3.0 ORGANIZATION. The organizations that govern, direct, manage, and administer Ada VALIDATIONS on behalf of the AJPO constitute the Ada CERTIFICATION BODY. This section specifies the responsibilities of each organization or party that is involved in the Ada VALIDATION process.

3.1 RESPONSIBILITIES. The following are the specific responsibilities assigned to each party or organization in the Ada CERTIFICATION BODY:

3.1.1 The Director, AJPO:

- a. Is responsible for establishing and maintaining the Ada VALIDATION process, and is the governing official for this document.
- b. Establishes the period of time after which a VALIDATION CERTIFICATE expires.
- c. Designates an AVO and delegates to it the authority to implement the Ada VALIDATION process.
- d. Approves the establishment of AVFs to perform the actual VALIDATIONS in accordance with AJPO procedures.
- e. Has final decision authority over disputes raised by VENDORS concerning VALIDATION issues.

3.1.2 The AVO:

- a. Recommends for AJPO approval, changes to the VALIDATION procedures established in this document.
- b. Ensures that each AVF performs VALIDATIONS consistent with the VALIDATION process established by the AJPO.
- c. Maintains accurate records pertaining to each VALIDATION and to the VALIDATION process.
- d. Maintains the list of VALIDATED COMPILERS.
- e. Advises the Director, AJPO on all VALIDATION issues.
- f. Resolves, where appropriate, disputes and issues between an AVF and a VENDOR.

- g. Reviews all VALIDATION SUMMARY REPORTS (VSRs) prepared by AVFs, and coordinates with AVFs to ensure that an accurate and comprehensive VSR is produced for each BASE COMPILER.
- h. Prepares VALIDATION CERTIFICATES for signature by the Director, AJPO.

3.1.3 A chartered AVF:

- a. Follows the VALIDATION process established by the AJPO.
- b. Is responsible for conducting VALIDATION in a timely and impartial manner.
- c. Produces the VSR.
- d. Forwards disputes raised by a VENDOR to the AVO for review and analysis, with final resolution to be provided by the AJPO, if necessary.

3.2 VENDOR RESPONSIBILITIES. VENDORS are clients of the AVF and are serviced by the Ada CERTIFICATION BODY in matters concerning Ada VALIDATION. In requesting services of the Ada CERTIFICATION BODY, VENDORS are to:

- a. Provide accurate and sufficient information to perform a VALIDATION.
- b. Nominate DERIVED COMPILERS for registration by the AJPO, and provide sufficient information and rationale to support registration of these compilers at the time of their nomination.

3.3 RELATIONSHIP BETWEEN THE AVF AND THE CLIENT. VENDORS are to notify an AVF, in writing, of their desire to have a VALIDATION conducted. VENDORS will negotiate and enter into a contract with an AVF for services to be performed during a projected period of time. When a VENDOR and an AVF have entered into a contract, the VENDOR becomes a client of the AVF; and, both parties assume responsibilities for conducting the VALIDATION. All members of the CERTIFICATION BODY cooperate to ensure that the service to each VENDOR is effective and efficient. Technical or procedural issues arising from an AVF and VENDOR relationship are referred to a higher level of the CERTIFICATION BODY only when those issues cannot be resolved at a lower level.

Section 4

BASE COMPILER VALIDATION

4.0 VALIDATION OF BASE COMPILERS. Ada COMPILERS are validated and periodically revalidated using the VALIDATION process defined in this document. VALIDATION is successfully completed when the APPLICABLE ACVC TESTS are run against a candidate Ada COMPILER and an AVF reviews the test results and judges them to demonstrate compiler CONFORMITY to the Ada PROGRAMMING LANGUAGE. The issuance of a VALIDATION CERTIFICATE is the official confirmation and evidence of successful completion of the VALIDATION process. VALIDATION TESTING of an Ada COMPILER is performed by an AVF as authorized by the Director, AJPO. The results of VALIDATION testing are documented in a VSR. This section details the steps and procedures established by the AJPO for performing a VALIDATION.

4.1 STEPS AND PROCEDURES. There are nine steps in Ada VALIDATION that a VENDOR and the CERTIFICATION BODY must successfully complete in order to qualify a compiler as an Ada BASE COMPILER for which a VALIDATION CERTIFICATE is issued. These steps are:

1. Obtain the current ACVC test suite, (VENDOR). The ACVC is the only official test suite that can be used by VENDORS and AVFs to conduct CONFORMITY TESTING. The ACVC is distributed exclusively by AVFs and is used by VENDORS to prepare their DECLARATION OF CONFORMANCE.

2. Submit a request for VALIDATION testing, (VENDOR). Informal communication between a VENDOR and an AVF is encouraged. However, VENDORS must notify the AVF, in writing, that they intend to become a client and desire to be scheduled for VALIDATION TESTING. An AVF schedules and commits resources on a first-come-first-served basis; therefore, VENDORS should ask for AVF services as soon as they can accurately project the need. This written notification should include the following information:

- a. Target dates for submission of a DECLARATION OF CONFORMANCE and test results, and the commencement of on-site testing.
- b. The ACVC version that is being used by the VENDOR to prepare a declaration of CONFORMANCE.
- c. The CONFIGURATION(s) and candidate Ada COMPILER(s) to be tested.
- d. The location of on-site testing.

- e. The length of time (estimate) required for on-site testing.
- f. The point of contact for further information.

When VENDORS project target dates for submission of a DECLARATION OF CONFORMANCE and for on-site testing, they must consider that it takes an AVF generally three months (90 days) from receipt of the DECLARATION OF CONFORMANCE and testing materials to send a final VSR to the AVO.

In acknowledging receipt of the VENDOR's notification, the AVF advises the VENDOR of known resource constraints that affect the desired schedule. When an AVF expects to have scheduling problems, the AVO should be consulted concerning possible redistribution of workload among other AVFs.

At the request of the VENDOR, the AVF makes every effort to keep confidential both the intent to become a client and the schedule for VALIDATION. However, this confidentiality cannot be guaranteed nor is it allowed to interfere with normal review/approval procedures described in this document for draft VSRs.

3. Negotiate a formal agreement, (AVF/Client). An AVF requires a formal agreement with a VENDOR, and may require payment in advance for the analytical and testing services that will be performed. Preparation of the test report is included in this fee. Once a formal agreement is negotiated between a VENDOR and an AVF, the VENDOR becomes a client of the AVF and may be serviced by the Ada CERTIFICATION BODY. A VENDOR who is in the process of self-testing may discover ACVC tests that could be incorrectly written or which are inapplicable for the compiler and CONFIGURATION under test. As soon as the VENDOR becomes an AVF client, a list of all ACVC tests that are doubtful from the client's perspective are provided by the client to the AVF, along with the client's rationale for including each test on this list.

4. Submit a DECLARATION OF CONFORMANCE and pre-VALIDATION test results, (Client). The AVF will provide the client with the format for a DECLARATION OF CONFORMANCE, which is defined in Appendix A of this document, and a list of required pre-VALIDATION information. This list includes the information required by Appendix F of the Ada PROGRAMMING LANGUAGE standard and other implementation specific information determined by the AVF. The AVF prescribes the format in which this information is to be delivered.

After the client runs the ACVC on the candidate Ada COMPILER, the client provides the AVF with a DECLARATION OF CONFORMANCE and ACVC test results. The ACVC must be run in a manner agreed upon with the AVF, and test results must be delivered to the AVF no later than two months (60 days) prior to the scheduled on-site test date. The actual date for delivery of ACVC test results by the client is specified in the contract between the client and the AVF. During this two-month period there is sufficient time for a thorough analysis of pre-validation test results, resolution of all testing issues, and preparation of a draft VSR. It should be noted that, if a two-month period is not provided for analysis, any problems that arise may cause the AVF to reschedule VALIDATION. Such a change in schedule may even necessitate the use of a later ACVC version and thus the resubmission of test results.

When received, the AVF acknowledges receipt of the client's DECLARATION OF CONFORMANCE and test results. In the event a VENDOR successfully completes step 4 in this VALIDATION procedure prior to the expiration date of the current ACVC, and that the expiration date is subsequently reached before the VALIDATION process is complete, then the VENDOR is allowed to proceed with the original ACVC provided that:

- a. the AVF has had sufficient time to determine that there are no failures of APPLICABLE ACVC TESTS, and
- b. the VENDOR has no test disputes pending at the time the ACVC expired that are subsequently decided against the VENDOR's position on the test.

5. Resolve test issues, (AVF/CLIENT/AVO/AJPO). The AVF thoroughly analyzes all client supplied test results and materials that support the client's DECLARATION OF CONFORMANCE. Included in this analysis is a review of the list of tests considered by the client to be inapplicable to the CONFIGURATION being tested, and the specific BASE CONFIGURATION(S) used by the client to test the candidate Ada COMPILER. If the AVF and the client cannot agree on the applicability of specific tests, or the interpretation of test results, or cannot mutually agree to definitions, such as BASE CONFIGURATION, then the issue(s) is referred to the AVO for review. The AVO then analyzes all issues with the help of the AVF, the client, and any other technical body the AVO considers advisable. Information resulting from this review is then presented to the Director, AJPO for consideration and final resolution. The Director, AJPO decision on disputed tests or test methods is final. On-site testing will not take place until all test issues have been resolved. When the resolution goes against the client's position, the client may renegotiate with the AVF to reschedule the VALIDATION or to amend or terminate the contract.

6. Preparation and review of the draft VSR, (AVF/AVO/AJPO). The AVF prepares a draft VSR based on information supplied by the client. This includes a DECLARATION OF CONFORMANCE, a description of all ACVC tests that were processed on the BASE COMPILER, a table showing the class and category of all ACVC tests (e.g., total number of class C tests passed, failed, withdrawn, or inapplicable, etc.), and a description of the testing environment (e.g., designation of CONFIGURATIONs tested, testing completion date). The draft VSR also reflects any decisions made concerning disputed test issues. Implementation dependent options are also detailed in the VSR.

The client, the AVO, and AJPO will receive copies of the draft VSR for concurrent review and comment. This review will be completed as soon as possible with comments provided directly to the AVF. The AVF coordinates with the AVO resolution of remaining issues prior to on-site testing that arise as a result of this review. Once all issues are resolved and documented in the draft VSR, the draft VSR is referred to as the annotated VSR and the annotated VSR becomes the basis on which on-site testing is conducted and test results analyzed. When necessary, on-site testing may begin prior to completion of AVO and AJPO review of the draft VSR.

7. On-site testing, (AVF). The AVF conducts the CONFORMITY TESTING of the client's compiler at the location designated by the client in accordance with the annotated VSR. The client submits to the AVF a refined estimate of time required for testing two weeks (14 days) before AVF personnel are scheduled to arrive at the client's site. AVF test personnel prepare the ACVC for the client's specific implementation. This customization of the ACVC for a client includes: (a) parameters that are implementation dependent; and (b) insertion of special fields/records used by the AVF to ensure the integrity of the test suite. The AVF, during on-site testing, executes the identical APPLICABLE ACVC TESTS that the client executed during pre-VALIDATION testing. It is the AVF that determines those inapplicable tests to execute during on-site testing in order to confirm their inapplicable status. If the AVF's test results are challenged by the client, the matter is to be referred to the AVO for resolution, or if resolution cannot be obtained, to the AJPO.

8. Issuing a VALIDATION CERTIFICATE, (AJPO). The Director, AJPO authorizes the issuance of a VALIDATION CERTIFICATE after the results of AVF testing demonstrate that a VENDOR's candidate Ada COMPILER is in COMPLIANCE with the Ada PROGRAMMING LANGUAGE. When an AVF completes on-site VALIDATION testing and test results confirm CONFORMITY for a candidate Ada COMPILER, it notifies the AVO that on-site test results were identical to those agreed to in the annotated VSR.

If on-site testing was based on a draft VSR, then the VSR review process described above must first be completed before a certificate is issued. Test results must be analyzed based on an annotated VSR. Once all issues concerning the VSR have been resolved to the AVO's and AJPO's satisfaction, a VALIDATION CERTIFICATE is prepared by the AVO for an Ada COMPILER, and issued by the authority of the Director, AJPO.

The VALIDATION CERTIFICATE uniquely identifies a version of the BASE COMPILER and the nomenclature of the BASE CONFIGURATION, as well as the version of the ACVC under which the testing was performed. It is issued to the VENDOR who is the COMPILER OWNER of the Ada COMPILER. In most cases, the VENDOR and the client for the compiler are the same. However, the client must ensure that the AVF is provided accurate information for inclusion on the VALIDATION CERTIFICATE, including the COMPILER OWNER. VALIDATION CERTIFICATES are delivered to the client by the AVF.

9. Issuing the final VSR, (AVF/AVO/AJPO). The fully agreed upon annotated VSR is used by the AVF to prepare the final VSR for signature. The AVF manager signs the final VSR and forwards it to the AVO for signature. The AVO, in turn, signs the VSR and forwards it to the Director, AJPO for final signature. A fully signed copy of the final VSR is then returned to the AVF.

4.2 CERTIFICATE EXPIRATION. VALIDATION CERTIFICATES expire in accordance with the time period established by the Director, AJPO. This time period is currently one year after the date of issue. An automatic extension, valid until adjudication of a pending REVALIDATION, shall occur whenever a VENDOR has submitted his COMPILER for REVALIDATION in a timely manner as defined in step two of these procedures.

4.3 VALIDATION INFORMATION AVAILABILITY: The AJPO ensures that a publicly available list of Ada COMPILERS is maintained. This list identifies the Ada COMPILERS for which VALIDATION CERTIFICATES have been issued. The AJPO also ensures that the final VSR is made publicly available for compilers on this list through an appropriate government information distribution facility. A separate list of registered Ada DERIVED COMPILERS is also maintained by the AJPO and is publicly available. The Ada Information Clearinghouse can provide assistance in obtaining or accessing any of the above information.

4.4 VALIDATION LIMITATIONS. A compiler is judged as having either passed or failed in VALIDATION. If a compiler passes VALIDATION, then it demonstrates that the product, as tested by the ACVC, conforms to the Ada PROGRAMMING LANGUAGE. Testing for characteristics other than those specified by the Ada PROGRAMMING LANGUAGE, such as performance or suitability for particular applications, is outside the scope of VALIDATION testing.

Section 5

DERIVED COMPILER VALIDATION

5.0 DERIVED COMPILERS. Once an Ada COMPILER has a current VALIDATION CERTIFICATE, then variations or modifications to that Ada COMPILER and/or its configuration result in a DERIVED COMPILER. A DERIVED COMPILER must conform to the Ada PROGRAMMING LANGUAGE in every respect and by the same measure as does the BASE COMPILER from which it was derived. In order to be considered a DERIVED COMPILER, and therefore acquire status as an Ada VALIDATED COMPILER, a VENDOR must register DERIVED COMPILERS with the AJPO. This section details the process and procedures governing the registration of DERIVED COMPILERS.

5.1 REGISTRATION OF A DERIVED COMPILER. VENDORS may request to register a DERIVED COMPILER while they are clients of an AVF and are engaged in the VALIDATION of the BASE COMPILER or, subsequently, after the AVF-client relationship has been terminated by successful completion of VALIDATION of the BASE COMPILER, but prior to expiration of the VALIDATION certificate for the BASE COMPILER from which the subject compiler was derived. As part of this registration, the VENDOR:

- a. identifies the BASE COMPILER and BASE CONFIGURATION from which the DERIVED COMPILER originates; and,
- b. affirms that the DERIVED COMPILER is in COMPLIANCE with the same version of the ACVC used in the VALIDATION of the BASE COMPILER.

5.2 REGISTRATION DURING VALIDATION. When VENDORS are clients of an AVF and desire to register a DERIVED COMPILER, the following steps must be completed:

1. Submit a DECLARATION OF CONFORMANCE (VENDOR). The VENDOR submits to the AVF a statement signed by an authorized VENDOR official that lists the CONFIGURATION that is declared to be an EQUIVALENT CONFIGURATION. This statement also includes the nomenclature identifying the candidate compiler. If the VENDOR supplies only the compiler, then the VENDOR must obtain either a statement from a responsible official of the hardware manufacturer or technical documentation which affirms that the COMPUTER ARCHITECTURE is equivalent to that of the BASE CONFIGURATION originally used to demonstrate COMPLIANCE.

2. Review the registration request (AVF). The AVF reviews the information supplied by clients and provides technical advice to them, as required, to ensure that the requested registration is consistent with the definition of a DERIVED COMPILER. The client's request for registration of a DERIVED COMPILER is then submitted to the AJPO, via the AVO, concurrent with submission of the final VSR for the BASE COMPILER.

5.3 POST VALIDATION REGISTRATION. VENDORS may request registration of a DERIVED COMPILER during the period a VALIDATION CERTIFICATE for the BASE COMPILER is in force. The AJPO accepts requests for registration after a VALIDATION CERTIFICATE has been issued and the final VSR has been approved. The steps listed for registration during VALIDATION must be completed as well for post VALIDATION registration. A second final VSR for the BASE COMPILER need not be submitted. VENDORS are to forward these requests to the AVF which conducted the VALIDATION of the subject BASE COMPILER. The AJPO may refuse a registration request if the cycle for updating the list coincides with expiration of the VENDOR's VALIDATION CERTIFICATE for the BASE COMPILER. Requests are to include the following information:

- a. Expiration date and serial number (where applicable) of the VALIDATION CERTIFICATE for the BASE COMPILER.
- b. Identification of the compiler version to be registered.
- c. Identification of the authorized VENDOR representative responsible for COMPLIANCE of the DERIVED COMPILER.
- d. Rationale to substantiate the validity of the registration.

5.4 MAINTENANCE OF REGISTRATION. The registration of a DERIVED COMPILER expires when the related VALIDATION CERTIFICATE expires. The AJPO maintains and makes publicly available a list of all registered DERIVED COMPILERS; a description of their configurations; their relation to a BASE COMPILER; and of any information supplied by the VENDOR in substantiation of compliance to the standard. When an AVF believes that COMPLIANCE is questionable, it may require the VENDOR to provide specific evidence of testing against the ACVC version used in VALIDATION of the BASE COMPILER.

5.5 DERIVED COMPILER STATUS. DERIVED COMPILERS that are registered are considered VALIDATED COMPILERS. This status expires no later than the VALIDATION CERTIFICATE of the associated BASE COMPILER. No VALIDATION CERTIFICATE will be issued for a DERIVED COMPILER.

5.6 CHALLENGES. The status of a DERIVED COMPILER can be challenged if it is determined to fail an applicable ACVC test of the ACVC version used to validate the BASE COMPILER. Applicability of such tests is documented in the VSR for the BASE COMPILER. When a user or potential buyer desires to test whether or not a registered compiler is in COMPLIANCE with the Ada PROGRAMMING LANGUAGE he/she must do the following:

1. obtain at challenger's own expense a copy of the applicable VSR and a copy of the ACVC test suite used during VALIDATION of the BASE COMPILER;

2. conduct testing at challenger's own expense under the conditions specified in the VSR, using the appropriate ACVC, and gather, in the same format as the VSR, data to substantiate a challenge;

3. submit to the AVF which conducted the original VALIDATION, with a copy to the AJPO, a formal challenge to the validity of the DERIVED COMPILER. Documentation for this challenge shall include test results that substantiate the challenge. In addition, identify to both the AJPO and the AVF the VENDOR who offered the DERIVED COMPILER (for sale or use).

5.6.1 RESOLUTION OF CHALLENGES. When a challenge is made, the AJPO annotates the compiler on the registration list indicating: (a) there is a challenge to the compiler's COMPLIANCE status; (b) the date of the challenge; and (c) the target date for resolution of the challenge; and further, stipulates that a challenge in and of itself does not constitute non-COMPLIANCE of the annotated compiler. All information gathered or submitted to the AVF and AJPO concerning a challenge is made immediately available for review by all parties having an interest in the compiler's status.

In making the determination of COMPLIANCE vs. non-COMPLIANCE, the AVF may request that the challenged VENDOR submit the results of his/her in-house VALIDATION testing to demonstrate full COMPLIANCE. Further, the AVF may request from the challenged VENDOR special testing, which may consist of a subset of the ACVC. The complete and timely cooperation from the owner of the challenged compiler is required in investigating and resolving disputed issues. When a challenge is fully substantiated to the satisfaction of the AVF, AVO, and AJPO, the VENDOR who owns the compiler is requested to correct the cited compiler deficiencies and to demonstrate to the AVF that such deficiencies have been corrected, or cease and desist from offering the compiler for sale or use as an Ada VALIDATED COMPILER. If deficiencies can not be corrected or are not corrected within a 90 day time period, then the AJPO annotates the DERIVED COMPILER list to reflect that the subject compiler is no longer considered as having a VALIDATED COMPILER status. Any substantiated challenge or resolution to a challenge will be announced through appropriate public documents and reflected by annotating the DERIVED COMPILER list.

5.7 VALIDATION OF MULTIPLE COMPILERS. If the client desires a VALIDATION CERTIFICATE for a BASE COMPILER on several BASE CONFIGURATIONS, rather than registering each modified compiler or change in configuration as a DERIVED COMPILER, then at least one member of each configuration must be provided for AVF testing with the full ACVC. These BASE CONFIGURATIONS are placed on a certificate, while derived configurations may be registered separately at the client's request. When different operating systems, major changes to an operating system, or modified compilers are presented to the AVF, the compiler is tested under each operating system, even if the COMPUTER ARCHITECTURE is identical. In such cases, detailed test methods may vary on a case by case basis as negotiated by the AVF and AVO to ensure rigid testing while minimizing unnecessary effort.

5.8 MAINTAINED COMPILERS AS DERIVED COMPILERS. Maintenance changes to a VALIDATED COMPILER do not affect its status as a VALIDATED COMPILER, provided that the compiler retains CONFORMITY to the Ada PROGRAMMING LANGUAGE. Individual versions of a MAINTAINED COMPILER must be distinguished through the use of incremental version/release control numbers assigned by the developer. Such a version is considered a DERIVED COMPILER, and is subject to the procedures pertaining to DERIVED COMPILERS when it is first registered or subsequently registered with the AJPO. A MAINTAINED COMPILER is not a validated Ada COMPILER when it fails an APPLICABLE ACVC TEST that was passed by the associated BASE COMPILER during VALIDATION testing.

5.9 REGISTRATION LIMITATIONS. The registration of a DERIVED COMPILER by the AJPO does not imply or support VENDOR claims that the compiler conforms to the Ada PROGRAMMING LANGUAGE. Registration of DERIVED COMPILERS provides a public record of those compilers that VENDORS claim are modifications to Ada COMPILERS, which in spite of the modifications, remain compliant to the Ada PROGRAMMING LANGUAGE in every manner as the BASE COMPILER.

5.10 EQUIVALENT CONFIGURATION REGISTRATION. It is not the intention of the AJPO to focus on each specific CONFIGURATION within a family of EQUIVALENT CONFIGURATIONS. These EQUIVALENT CONFIGURATIONS do not have to be individually registered as DERIVED COMPILERS to be considered VALIDATED provided a BASE COMPILER on one of the configurations exist. Registration of a series of EQUIVALENT CONFIGURATIONS, produced by the same manufacturer, will be accepted.

Section 6

Ada VALIDATION FACILITIES ESTABLISHMENT AND OPERATION

6.0 AVF PURPOSE. Ada Validation Facilities (AVFs) are established by the Director, AJPO to conduct Ada VALIDATIONS. These facilities execute the VALIDATION procedures contained in this document for the purpose of ensuring that the Ada PROGRAMMING LANGUAGE standard is properly and faithfully implemented in Ada COMPILERS. This section discusses the establishment and operation of AVFs.

6.1 ESTABLISHMENT OF AVFs. The Director, AJPO may issue an AVF charter to a requesting organization that represents the government of the United States of America (USA) and has a Memorandum of Understanding (MOU) with the AJPO concerning Ada VALIDATION or represents a foreign government that has an MOU with the USA concerning the establishment of AVFs. AVFs thus chartered operate under the terms of the governing MOU and the procedures established in this document, and are chartered to conduct VALIDATIONS on behalf of the AJPO.

6.2 AVF CHARTER. An AVF charter carries both the authority to conduct VALIDATION testing as well as the obligation to fully support the CERTIFICATION SYSTEM established by the AJPO.

6.3 CHARTER RENEWAL. An AVF charter remains in effect for two years from the date of issue. However, a charter can be removed by the Director, AJPO in less than two years for due cause. Renewal requests are submitted by AVFs to the Director, AJPO for re-evaluation and approval at least six months prior to the end of the two-year period.

6.4 AUDIT OF OPERATIONS. The AVO has oversight authority over all AVF practices for the purpose of maintaining a uniform process for VALIDATION of Ada COMPILERS. The Director, AJPO may direct an impartial body to conduct an audit of AVF operations at any time and may require an audit prior to issuing or renewing an AVF charter. Audits are conducted in accordance with procedures established by the Director, AJPO at the time of the audit, and are tailored to reflect the purpose of the audit. AVF managers meet periodically with the Director, AJPO and/or with the AVO to discuss the VALIDATION process and audit procedures, to share ideas, and to resolve problems encountered in conducting the VALIDATION of Ada COMPILERS.

Section 7

Ada COMPILER VALIDATION CAPABILITY DEVELOPMENT, MANAGEMENT, AND LIFE CYCLE CONTROL

7.0 ACVC PURPOSE. The Ada COMPILER VALIDATION CAPABILITY (ACVC) is the test suite used to test COMPLIANCE and determine CONFORMITY of an Ada COMPILER. This section outlines the development approach to changing tests, managing the test suite configuration, and controlling the life cycle maintenance of the ACVC.

7.1 ACVC TEST SUITE DEVELOPMENT. Each version of the ACVC has a development cycle (currently twelve months) divided into two periods: laboratory development and field testing. During the laboratory development period, tests are added, removed, corrected, and improved. During the field-test period, the test suite is available for public inspection. An ACVC version becomes the official ACVC test suite version to be used for all VALIDATIONS when the AJPO releases it for use. The official period of use is normally twelve months.

7.2 VERSION RELEASE. After the development cycle is complete, the new version of the ACVC test suite is issued for use by all AVFs and their clients in all VALIDATIONS. Each version expires on a publicly announced schedule. The current release schedule is 1 December for the field test version and 1 June for the official version, with the ACVC test suite expiring twelve months after release. VENDORS are required to use the same single version of the test suite for pre-VALIDATION and VALIDATION. Table 7.1 provides a representative schedule of an ACVC release.

7.3 MANAGEMENT OF THE ACVC TEST SUITE. The AJPO approves each version of the ACVC prior to its release for use in VALIDATIONS, and ensures that the test suite and procedures for VALIDATION testing are publicly available. The AVF at ASD/SIOL, Wright-Patterson Air Force Base, Ohio, 45433-6503 is designated the ACVC Maintenance Organization (AMO). The AMO maintains a subscription service for issuing the ACVC and provides to other AVFs a list of tests withdrawn from the test suite. Each AVF maintains a subscription service for its own clients of record.

7.4 QUALITY ASSURANCE OF THE ACVC. During the field-test period, all reported test errors and disputed tests are reported to and researched by the AMO; tests found to contain errors are either corrected or removed from the test suite. During the twelve month period when a test suite version is used for CONFORMITY testing, the only change permitted to the suite is the removal of tests whose interpretation of the Ada PROGRAMMING LANGUAGE is judged by the AJPO to be incorrect or dubious. The AVO collects reports from the Ada user and compiler developer communities concerning implementation anomalies.

When a question of CONFORMITY can be detected and resolved by adding a suggested test to the next ACVC version, strong consideration will be given to implementing it. Contributions to the ACVC are solicited on a no-cost basis by each AVF. Proposed modifications to the ACVC generated in this way are entered into the same review cycle as are modifications proposed through the regular AMO maintenance process, and are included into the next ACVC test suite after approval by the AJPO.

Table 7.1

Table Showing ACVC Life-cycle

June 85	Dec 85	June 86	Dec 86	June 87	Dec 87	June 88
------------	-----------	------------	-----------	------------	-----------	------------

ACVC Revision 1.7

```

-----
|                                     |
|      OPERATIONAL                    |
|                                     |
-----

```

ACVC Revision 1.8

```

-----
|           | FIELD |           |
| DEVELOP  | TEST  | OPERATIONAL |
|           |           |           |
-----

```

ACVC Revision 1.9

```

-----
|           | FIELD |           |
| DEVELOP  | TEST  | OPERATIONAL |
|           |           |           |
-----

```

7.5 ACVC LIFE-CYCLE CONTROL. When a particular version of an ACVC is released for a field-test period, the tests contained in that version become the field-test baseline. At the end of the field-test period, an official baseline list of tests included in the next ACVC version to be released is published by the AMO. The official baseline can be changed only by actions approved by the AJPO. When tests in the official baseline are found to contain errors, they are removed from the ACVC by the AMO. The AJPO, AVO, and other AVFs are advised of this action by the AMO. The AVO ensures that appropriate action is taken to address tests removed from an ACVC version. When tests are corrected or replaced, they are incorporated into a subsequent ACVC version on a schedule published by the AMO.

The following are the procedures to report and resolve a suspect ACVC test:

1. Disputed ACVC tests. If a client, VENDOR, or user believes that a test is in error, he/she should send a written statement giving the test name and a description of the problem to the validating AVF or AMO, as appropriate. Supplementary materials, such as compilation listings or execution results, that illustrate the test error should be included when possible.

2. Corrective action. When tests in a released ACVC test suite version are challenged as being in error, the AVF or AMO receiving the challenge refers it to the AVO. The AVO analyzes the information surrounding the challenged test with the advice of a group of Ada PROGRAMMING LANGUAGE experts. A decision on these disputes is reached by the AJPO as soon as possible, and the AVF referring the challenge is notified of the decision. If the decision is that a test is to be withdrawn from that ACVC version, the AMO generates and distributes to all AVFs a new list of withdrawn tests. Tests that have been withdrawn from the ACVC by the AJPO are not to be executed during on-site testing.

Section 8

DoD GUIDELINES FOR Ada COMPILER USE

8.0 PURPOSE. The purpose of this section is to establish guidelines for ensuring CONFORMITY to the Ada PROGRAMMING LANGUAGE on DoD projects that develop and maintain Ada software.

8.1 SCOPE. This document integrates guidelines for the VALIDATION and control of Ada COMPILERS with DoD life-cycle management practices. These guidelines are intended to aid program managers in the initial acquisition of an Ada COMPILER, control of that compiler through the software development phase, the transition of an Ada COMPILER from development to maintenance activities, and control of an Ada COMPILER during maintenance.

8.2 OVERALL CONCEPT. A program manager identifies the requirement for a VALIDATED Ada COMPILER as an action within the context of project milestones. After identification, a VALIDATED COMPILER is baselined in accordance with applicable DoD software life-cycle management policies, procedures, and practices. This baselined VALIDATED COMPILER then becomes a PROJECT-VALIDATED COMPILER for the lifetime of that project.

8.3 GENERAL GUIDANCE.

8.3.1 USE OF PROJECT-VALIDATED Ada COMPILERS. Ada software used in DoD systems that is delivered for operational testing, deployment and maintenance should be compiled with a PROJECT-VALIDATED COMPILER. A project may have more than one PROJECT-VALIDATED COMPILER.

a. A PROJECT-VALIDATED COMPILER retains its status as a PROJECT-VALIDATED COMPILER for the duration of a project and under any contractual arrangement that requires use of this PROJECT-VALIDATED COMPILER. This status is retained regardless of the VALIDATION status of the compiler under the general VALIDATION policies and procedures.

b. Maintenance of a PROJECT-VALIDATED COMPILER does not affect its status as a PROJECT-VALIDATED COMPILER if the modified compiler is baselined and passes all applicable tests of the ACVC in a version equal to or more recent than the ACVC version that was in effect at the time of the baselining of the PROJECT-VALIDATED COMPILER. When a system software release has been compiled on several PROJECT-VALIDATED COMPILERS, ACVC testing requirements apply to each of these compilers.

8.3.2 Ada COMPILERS FOR RESTRICTED TARGETS. An Ada COMPILER used to generate object code for a RESTRICTED TARGET is considered to be a PROJECT-VALIDATED COMPILER if and only if all of the following conditions are satisfied:

- a. The PROJECT-VALIDATED COMPILER for the GENERIC TARGET is a fully conforming implementation of the Ada PROGRAMMING LANGUAGE (CONFORMITY), even though its use may be solely for the development of applications software for RESTRICTED TARGETS.
- b. All mandatory features of the Ada PROGRAMMING LANGUAGE that can be supported by hardware, or are emulated on the RESTRICTED TARGET, are supported by the compiler for the RESTRICTED TARGET. (i.e., Compilers for the RESTRICTED TARGET should not be arbitrarily constrained to subset implementations of the Ada PROGRAMMING LANGUAGE.)
- c. All APPLICATION-SPECIFIC RUN-TIME LIBRARIES for RESTRICTED TARGETS are contained within that application and do not affect the Ada COMPILER for the GENERIC TARGET or the RESTRICTED TARGET when used to generate code for other applications.

8.3.3 Ada COMPILERS USED DURING SOFTWARE DEVELOPMENT. Compilers used to develop Ada software do not have to be PROJECT-VALIDATED COMPILERS throughout the entire development phase. The Program Manager determines when in the development phase a compiler will be placed under baseline control, at which time the compiler must satisfy the definition of a VALIDATED COMPILER in order to become a PROJECT-VALIDATED COMPILER. A review of the CONFORMITY status of a PROJECT-VALIDATED COMPILER should be conducted prior to any Ada application software release and reported to the Program Manager. When because of modifications to the compiler, the CONFORMITY status of a PROJECT-VALIDATED COMPILER becomes uncertain to the Program Manager, it should be retested. Considerations for retesting include the scope of change to the PROJECT-VALIDATED COMPILER, a change in the BASE CONFIGURATION, and the size of the Ada application software release.

8.4 Ada COMPILER USE DURING DEPLOYMENT AND MAINTENANCE. At each baseline milestone in the maintenance cycle, the testing of a PROJECT-VALIDATED COMPILER to ensure CONFORMITY should be accomplished as it was done during software development. This testing can be waived if the PROJECT-VALIDATED COMPILER is identical to one of the previous baseline milestones, or if it has been replaced by an unmodified BASE COMPILER for which a VALIDATION CERTIFICATE is in effect. At a minimum, PROJECT-VALIDATED COMPILERS should be upgraded or replaced in conjunction with MAJOR UPGRADES.

8.5 DETAILED GUIDANCE.

8.5.1 SOFTWARE DEVELOPMENT WITH A VALIDATED ADA COMPILER.

8.5.1.1 In the initial acquisition of an Ada Compiler, if the compiler selected for use on a project is not a VALIDATED COMPILER, the program manager is responsible for requiring that a VALIDATION CERTIFICATE for this compiler be obtained in accordance with the formal VALIDATION process.

8.5.1.2 Ada software may be developed prior to obtaining a VALIDATED COMPILER and baselining this compiler as a PROJECT-VALIDATED COMPILER. However, use of a VALIDATED or PROJECT-VALIDATED COMPILER at the earliest practical time is encouraged to reduce risk and potential problems during the acceptance of the software for operational testing.

8.5.1.3 When a VALIDATED COMPILER has been baselined for a project, configuration control procedures must be established to ensure complete documentation of changes made to the PROJECT-VALIDATED COMPILER and for derivations from it. Program managers are encouraged to ascertain at periodic intervals that maintenance changes and derivations have not affected the capability of the PROJECT-VALIDATED COMPILER to pass all applicable tests of the ACVC in a version equal to or more recent than used in its initial VALIDATION.

8.5.1.4 After expiration of a VALIDATION CERTIFICATE for a PROJECT-VALIDATED COMPILER, a program manager should ensure that the compiler stays in CONFORMITY throughout the remaining life of the project by taking the following actions:

- a. Re-test the PROJECT-VALIDATED COMPILER and DERIVED COMPILERS using the ACVC version used to originally establish the CONFORMITY of the BASE COMPILER whenever the CONFORMANCE of the compiler is suspect. This periodic retesting should be scheduled as part of project baseline milestones and the results incorporated into project documentation. A program manager determines whether this testing is done by project personnel or by an AVF. Cost, schedules, extent of maintenance on the compiler, contractual obligations, as well as the expected size of the software release, are considerations in determining periodic retesting.
- b. Ensure that all APPLICATION-SPECIFIC RUN-TIME LIBRARIES and run-time support for RESTRICTED TARGETS are developed and documented separately from that of the compiler, and that the support does not affect the CONFORMITY of the Ada COMPILER used to generate object code.

- c. Ensure that planned program product improvement (P3I) actions are incorporated into project baseline milestones and contracts well in advance of the projected action, as these (P3I) actions may result in the acquisition of a replacement compiler.

8.5.2 TRANSITION TO MAINTENANCE. Program managers should perform ACVC testing on their PROJECT-VALIDATED COMPILERS to establish the CONFORMITY status of PROJECT-VALIDATED COMPILERS prior to transition of an Ada application program to a maintenance activity.

Appendix A

DECLARATION OF CONFORMANCE (SAMPLE)

Compiler Implementer: _____
Ada Validation Facility: _____
Ada Compiler Validation Capability (ACVC) Version: _____

Base Configuration

Base Compiler Name: _____ Version: _____
Host Architecture - ISA: _____ OS&VER #: _____
Target Architecture - ISA: _____ OS&VER #: _____

Derived Compiler Registration

Derived Compiler Name: _____ Version: _____
Host Architecture - ISA: _____ OS&VER #: _____
Target Architecture - ISA: _____ OS&VER #: _____

Implementer's Declaration

I, the undersigned, representing (Corporation) have implemented no deliberate extensions to the Ada Language Standard ANSI/MIL-STD-1815A in the compiler(s) listed in this declaration. I declare that (Owner's Name) is the owner of record of the Ada language compiler(s) listed above and, as such, is responsible for maintaining said compiler(s) in conformance to ANSI/MIL-STD-1815A. All certificates and registrations for Ada language compiler(s) listed in this declaration shall be made only in the owner's corporate name.

(Implementer's Signature and Title) (Dated)

Owner's Declaration

I, the undersigned, representing (Corporation) take full responsibility for implementation and maintenance of the Ada compiler(s) listed above, and agree to the public disclosure of the final Validation Summary Report. I further agree to continue to comply with the Ada trademark policy, as defined by the Ada Joint Program Office. I declare that all of the Ada language compilers listed, and their host/target performance are in compliance with the Ada Language Standard ANSI/MIL-STD-1815A. I have reviewed the Validation Summary Report for the compiler(s) and concur with the contents.

(Owner's Signature and Title) (Dated)

This document is part of the Validation Summary Report (VSR), Appendix A, for initial validations and must be submitted for each derived compiler registration during or subsequent to initial validation.

END

4-87

DTIC