ADA[1] COMPILER VALIDATION

by

Erhard Ploedereder, Ph.D.

Tartan Laboratories

477 Melwood Ave.

Pittsburgh, PA 15221

U.S.A.

Summary

This paper discusses Ada Compiler Validation from the viewpoint of both the compiler supplier and the compiler user. The objectives of the requirement for validation and the resulting benefits, as well as the limitations of validation are presented. The process of Ada compiler validation is detailed along with its various problems as well as issues relating to validation and the use of validated compilers. Solutions to these problems and issues are explained, as they are proposed in a recent AJPO draft of a revised validation policy.

Preface

The proliferation of languages and language dialects has been recognized as a major contributing factor to the high cost of software development and maintenance. Problems that arise out of such proliferation include the added expense of maintaining a software environment for each such language and dialect, the cost of educating personnel in the use of many different languages, and the increased cost of porting software to other systems.

The programming language Ada was developed by the US Department of Defense (DoD) to become the only programming language to be used in DoD mission critical computer resource (MCCR) projects. Ada has been standardized and mechanisms have been put in place to prevent the existence of language dialects. The validation of Ada compilers is an integral part of these mechanisms.

This paper discusses various aspects of the Ada compiler validation, including its objectives and limitations, the actual process of validation, the policy for the use of validated Ada compilers within U.S. DoD, and the economics of compiler validation.

1. Objectives of Validation

Past experience with standardized languages (e.g., Jovial, Pascal, Cobol), has shown that the mere existence of a standard is insufficient to prevent the emergence of language dialects, as compiler vendors tend to deviate in their compiler implementations from the respective language standard. As a consequence, software developed with one compiler often cannot be compiled with a compiler supplied by another vendor without major changes to the software. As a result, porting of software between systems employing different compilers becomes expensive, if not economically infeasible. Particularly in the military setting, large software systems have a lifespan of many years, during which enhancements and new hardware developments will make it necessary to port the software to new underlying systems.

Recognizing this problem, the U.S. DoD has taken steps to insure that the mistakes of the past are not repeated in the Ada effort. It has obtained a trademark on Ada and requires that the conformance of compilers to the language standard, ANSI/MIL-STD 1815a, be ascertained by means of the validation process, before such compilers are allowed to be called Ada compilers. The objective of this requirement is to prevent language dialects and thereby to enhance the portability of software written in Ada.

[1] Ada is a registered trademark of the United States Government, Ada Joint Program Office

2. *The Meaning of Validation*

Validation is **not** a replacement for acceptance testing by the buyer of an Ada compiler. Staying true to its objectives, the validation process tests only for conformance with the language standard. It does not address issues of usability or efficiency of the compiler and the generated code. For example, the first Ada translator ever validated was the New York University Ada interpreter, which was a remarkable achievement as an experimental system for semantic modelling and for teaching Ada, but definitely not a system to support application code written in Ada.

Moreover, validation is **not** a guarantee of compiler correctness. Because the validation process employs testing by examples as the test method, only part of the intricacies of the Ada language can be covered by the validation.

On the other hand, the test suite of the validation process is sufficiently large and well designed that broad coverage is achieved. Successful processing of the test suite is a significant and major milestone for compiler developers on the way to compiler correctness and conformance to the language standard. As such, it significantly raises the users' confidence in the correctness and completeness of Ada compilers.

Notwithstanding validation, users must evaluate the appropriateness of a validated compiler for their particular needs along all dimensions, such as code efficiency, usability and user friendliness, interaction with the environment, provision of features that are optional in the language standard, etc.

3. *The Validation Policy, Procedures, and Guidelines*

In January 1986, AJPO drafted a new validation policy, consisting of three documents:

; the general policy for compiler validation;

; the procedures for the conduct of the Ada validation process; and

; the policies and guidelines for the use of Ada compilers in DoD.

The first document establishes the policies for the validation process applicable to Ada compilers both for general trade and for U.S. DoD applications. The second document details the procedures necessary to establish a compiler as a validated compiler. The third document integrates the requirement for validation of compilers with the constraints of life-cycle management procedures in U.S. DoD projects. While this third document relates specifically to the use of Ada in U.S. DoD, similar issues will arise in general trade usage of Ada compilers. It therefore can act as a model for general trade companies in adopting similar policies and guidelines.

In the subsequent sections, the provisions of the draft policy are described in more detail. Since, at this time, the draft is still under discussion and the policies and procedures not officially issued, changes may occur that are not reflected in this paper and may contradict its contents.

4. *The Components of the Validation Process*

The three organizational components of the validation process are the Ada Joint Program Office (AJPO), the Ada Validation Organization (AVO), and the Ada Validation Facilities (AVF). The technical component is the Ada Compiler Validation Capability (ACVC). These components and their role are briefly described in the following sections.

4.1. The Ada Joint Program Office (AJPO)

The Director of the AJPO formulates the validation policy, issues charters to AVF, and is ultimately responsible for all components of the validation process. Validation certificates are issued or authorized by AJPO after successful completion of the validation process.

4.2. The Ada Validation Organization (AVO)

AJPO has chartered an Ada Validation Organization (AVO) with the task to establish detailed guidelines and procedures for the process of Ada compiler validations, and to supervise Ada Validation Facilities (AVF) in their conduct of the validations. The AVO consults to AJPO on validation issues that arise from compiler validations. Furthermore, the AVO maintains the lists of validated compilers.

4.3. The Ada Validation Facilities (AVF)

The AVF are impartial organizations that are chartered by AJPO to conduct validations. Such charters are in effect for three years and are renewable. AVF are bound to adhere to the validation policies issued by AJPO and to the procedural guidelines established by the AVO. The AVF act as the interface between suppliers of compilers that desire validation of their compilers and the certification body comprising AJPO, AVO, and AVF.

Currently, five AVF are in existence: two in the U.S., and one each in Germany, Britain, and France. Their addresses are listed in Appendix B.

4.4. Ada Compiler Validation Capability (ACVC)

The AJPO has procured the Ada Compiler Validation Capability (ACVC) as the technical means to ascertain conformance of Ada compilers to the language standard. The core element of the ACVC is a large set of test programs to be submitted to the compiler to be validated. Currently, this test suite consists of about 2400 programs, each of which checks particular aspects of the language. About half of the tests are programs that contain errors that a compiler must detect. The remainder of the tests are correct Ada and must be executable after compilation. Some of the tests examine the correct implementation of optional features of the language, which a compiler may or may not support; the applicability of these tests depends on the individual compiler. The tests containing errors usually contain a large number of variations of the same class of error. Each of these errors must be recognized by the compiler. Since error recovery occasionally will mask subsequent errors, such tests may have to be split into multiple tests to isolate individual error situations, thus further increasing the number of test examples used in the validation process. The result of the tests containing errors is obtained by examining the listings produced by the compiler. Executable tests, on the other hand, are carefully designed to condense an indication of success or failure into a small print-out produced by the execution of the test.

The ACVC is augmented by software tools that facilitate the evaluation of test results, and by the Implementers' Guide which is a detailed commentary on ramifications of the Ada language semantics and on the corresponding ACVC tests.

At any given time, three versions of the ACVC test suite are in existence:

1. the development version: this version is not released to the public. New tests and corrected tests are added to this version.

2. the field-test version: this version is available to the public. During the first three months of its release, individual tests may be corrected. After three months, this version is frozen and can be optionally selected by a compiler implementor to be used in the formal validation process.

3. the released version: this frozen version of the test suite is generally used during formal validation. Tests in the frozen versions that are found to be incorrect are withdrawn and, after being corrected, are added to the development version.

The field-test and released versions are available to the public. Periodically the released version is retired and replaced by the field-test version; the development version becomes the field-test version and a new development version is created. Currently, this change of versions occurs in six-month intervals on June 10th and December 10th of each year. Consequently, for formal validation, implementors have a choice between the field-test version and the released version from September 10th to December 10th, and from March 10th to June 10th. During the remainder of the year only the current released version of the ACVC test suite can be used in formal validation. The overlap of ACVC versions admissible in validations has been introduced to permit compiler suppliers to reduce their risk of failing to complete the validation process before the released version expires; by choosing the frozen field-test version for validation, potential delays

can be absorbed more easily.

As the ACVC is rapidly maturing, it is expected that, in the near future, the release cycle will be lengthened so that the released version of the ACVC test suite has a lifetime of nine or twelve months.

5. The Validation Process

Ada compiler validation requires that all applicable tests of the ACVC test suite must be passed successfully by the compiler to be validated. Thus, validation testing is a pass/fail decision. The evaluation of the test results is done by an impartial body, i.e., an AVF, in two major steps: first, the AVF examines the test results as they were submitted to the AVF by the compiler supplier; at this stage, testing issues that the supplier or the AVF might raise, such as the applicability of individual tests, are resolved. Second, the AVF witnesses the on-site testing of the compiler, the results of which formally determine the success of the validation. Normally, any causes for potential failure of the validation testing are discovered and corrected during the first step so that on-site testing becomes a mere formality.

The applicability and correctness of individual ACVC tests may be challenged by the compiler supplier. Applicability issues arise for tests that examine optional features of the language not supported by the compiler. Correctness issues relate in most cases to tests for which the test designers interpreted the language standard differently than the compiler writers. These challenges are forwarded by the AVF to the AVO which, with the help of a group of language experts, determines whether the compiler writers' interpretation is permissible by the language standard. If so, the test is withdrawn from the test suite. The compiler supplier is informed about the disposition on the challenge typically within two weeks.

After successful validation, a Validation Certificate is issued by AJPO to the supplier of the compiler. This certificate is valid for one year and entitles the supplier of the compiler to market the compiler as a validated Ada compiler. The Validation Certificate attests to the successful completion of validation testing for a specific compiler and specified host and target systems, for which the full ACVC test suite was run as part of the on-site testing. The validated compiler for the specified host and target system is referred to as a "Base Compiler".

In detail, the validation process consists of ten steps that must be successfully completed in order to obtain a Validation Certificate for a Base Compiler. These ten steps are explained in Appendix A.

6. Revalidation

In order to preserve the status of a Base Compiler as a validated Ada compiler after expiration of its Validation Certificate, the supplier must re-submit the compiler for validation prior to the expiration of the Validation Certificate.

Such revalidations have a dual purpose: first, they guarantee that changes to compilers do not cause a gradual deviation from the language standard. Second, the ACVC test suite is continuously enhanced to provide more comprehensive coverage of the language standard. The requirement of revalidation insures that compilers conform to the standard even in those areas that were not covered by the ACVC version that was in effect at the time the compiler was initially validated.

It is expected that, as compilers and the ACVC gradually mature, the validity of the Validation Certificate will be extended to two years, thus reducing the effort and cost of revalidations.

7. Status of Maintained Compilers

Ada compilers will undergo maintenance changes and enhancements, in particular during the initial period after their first release. As it is economically and administratively infeasible to require formal revalidation after each change to a compiler, as well as undesirable to discourage maintenance upgrades by such a requirement, AJPO has adopted the policy that the validation status of a compiler automatically extends to its revisions. This policy relies on the mandate for annual revalidation to detect deviations from the standard and on the fact that it is in the best interest of the compiler suppliers to remain conforming to the language standard.

8. *Compilers for Multiple Configurations*

A particular problem with which compiler validation is confronted is the issue of compilers for multiple host and target system configurations. Typical examples are compilers hosted on and targeted for system architectures that are compatible in their instruction set; for such compilers, it can be reasonably expected that a compiler tested on one such configuration will generally execute correctly for any other configuration within the family of these architectures. This expectation is more difficult to justify if there are minor differences in the instruction set architecture or the operating system. A similar complication arises from compiler switches that influence execution characteristics of the compiler or the code generated by the compiler.

Clearly, it is quite impossible to perform compiler validations for all host and target configurations for which a compiler is designed, and for all switch settings of the compiler, due to the combinatorial explosion of all permutations of these influencing parameters. On the other hand, it is equally impossible to decide with certainty whether these parameters will alter the outcome of validation testing without actually performing the test. In a strict sense, it is therefore impossible for the AVF to attest to the successful completion of the validation testing, unless the tests are executed on the specific configuration.

The draft validation policy addresses this issue by introducing the concept of a "Registered Derived Compiler": A compiler supplier who has successfully validated a Base Compiler can register with AJPO additional compilers derived from this Base Compiler. Thus registered compilers are considered as equally validated compilers. This status is tied to the validation status of the Base Compiler and expires with the expiration date of the Validation Certificate for the Base Compiler. A registered derived compiler can be marketed as a validated Ada compiler.

As part of the registration process, the supplier has to assert that the compiler is a - possibly modified - version of the Base Compiler and that the compiler conforms to the Ada language standard. AJPO or AVO may require substantiating information for this assertion, such as a hardware vendor's statement guaranteeing the equivalence of the instruction set architecture or the results of running ACVC tests. However, final judgement on trusting a Registered Compiler to pass all applicable ACVC tests lies with the prospective buyer of the compiler.

The AVO maintains the list of Registered Compilers. If, at some later time, it is established that a Registered Compiler does not pass an ACVC test that the Base Compiler passed successfully, the compiler will loose its status as a Registered Compiler, unless the supplier corrects the problem. Any such challenges, which may be brought to the attention of the AJPO by any current or prospective user of the compiler, will be noted in the list of Registered Compilers after ascertaining the validity of those challenges.

The intention of this policy is to avoid the combinatorial explosion of formal validations, without requiring AVF or AVO to render judgement on whether a compiler is likely to pass validation testing on some host and target configuration without running the test suite on this configuration. It relies on the market place to force Registered Compilers to adhere to the language standard as well. Given that registration of compilers is dependent on the existence of an already validated Base Compiler by the same supplier, it is reasonable to assume that any potential deviations of Registered Compilers from the language standard are minor, unintentional, and easy to correct.

9. *Life-Cycle Management of Validated Compilers*

The mandate for the use of validated compilers in projects is generally in conflict with the practice of baselining compilers for extended periods of time. Since the validation status of a compiler expires after one year, annual revalidation is necessary. Due to changes in the ACVC test suite, it may be the case that the baselined compiler fails revalidation; an upgrade of the baseline would be required. For project stability or contractual reasons, such upgrades may be highly undesirable.

A revised policy for the use of Ada compilers in DoD projects has therefore been drafted that reconciles the requirement for compiler validation with the need for baselining compilers. This policy introduces the concept of a "Project-Validated Compiler".

A Project-Validated Compiler is a validated compiler that has been baselined in accordance with DoD life-cycle management policies. Such a compiler maintains its status as a Project-Validated Compiler throughout

the duration of the project. Only major system upgrades require that the baselined compilers also be upgraded to Validated Compilers. The rationale of this project-specific extension of the validation status is that, as the ACVC is maturing rapidly, it is very unlikely that a validated compiler would exhibit grave deviations from the language standard. The benefit of preventing minor accidental deviations by enhancements to the ACVC is low compared to the complications that arise from the risk of forcing an upgrade of the baselined compiler at inopportune stages of a project. Naturally, program managers are encouraged to plan for periodic upgrades of the baseline to a compiler with a Validation Certificate in effect.

While the Ada compiler used during development of MCCR software is not required to be a Project-Validated compiler, the compiler used for the software delivered for operational testing must be a Project-Validated Compiler. The acceptance testing upon delivery of such software will ascertain that the compiler passes all applicable tests of an ACVC version equal to or, optionally, more recent than the version that was in effect at the time of baselining the compiler. Depending on the software volume, such testing may range from internal testing to formal validation. The requirement for such testing is automatically satisfied if the compiler is a validated Base Compiler.

As a risk reduction strategy, project managers should consider using validated compilers as early as possible in a project. Maintenance and enhancement revisions of these compilers should be periodically checked for conformance with the language standard by using the ACVC as the test mechanism, so that potential acceptance problems at the time of delivery of software for operational testing are prevented.

10. Compilers for Embedded Targets

Embedded targets pose another challenge to the validation process: such targets may not be able to execute the ACVC tests, due to lack of hardware, in particular of appropriate output devices; the hardware for such targets may not exist yet; the hardware may be so restricted that it cannot support the mandatory features of the Ada language standard. In all these cases, the validation process cannot be applied to the compiler targeted at the embedded system and, hence, no Validation Certificate can be issued for the compiler.

Since it is nevertheless desirable to use compilers conforming to the Ada language standard in developing software for such embedded targets, the proposed DoD policy addresses this issue and arrives at a compromise that alleviates the problem and achieves the objective of conformance to the language standard:

For many embedded systems, the application code is first developed using a "simulated target", i.e., a simulator or an extended hardware configuration that provides more capabilities than the real embedded target system. If the compiler for the simulated target is a (project-)validated compiler, then the compiler for the real embedded target is also regarded as a project-validated compiler, provided that

1. the compiler for the real embedded target is derived from the project-validated compiler for the simulated target, and

2. all mandatory features of the Ada language standard that can be supported by the real target

4. any additions or changes to the run-time support maintain conformance of the execution of application code to the semantics of the Ada language standard.

From the viewpoint of practicality, the compiler for the real embedded target and any run-time modifications are unlikely to deviate from the language standard in an incompatible fashion (as opposed to a subsetting of the run-time support), since the application code translated by the validated compiler for the simulated target, on which the application code is developed and tested, cannot differ significantly from the code produced by the compiler for the real embedded target. Otherwise such earlier testing would be rendered quite useless. Consequently, the conditions enumerated above are sufficient to ensure a maximum of portability and reusability achievable for the application software.

11. The Economics of Compiler Validation

Without doubt, the validation process is expensive for the supplier of an Ada compiler and ultimately for the user of such compilers, since the cost of validation will be reflected in the cost of the compiler to the user. Typical AVF fees for the validation process are $12,000 to $15,000. Suppliers of validated Ada compilers have, however, estimated the total cost of a formal validation in the range from $50,000 to $100,000, due to the internal cost of preparing for the validation. While this estimate seems high, given that the monitoring of results of running the ACVC tests should be accounted for as part of the internal quality assurance for the compiler rather than of validation, a non-trivial effort is spent on administrating the validation process, on conducting the on-site testing, on making sure that the supplier's interpretation of the applicability of individual ACVC tests conforms to the AVO's official position, and on adjusting the compiler in areas where the AVO does not share the supplier's opinion. Extrapolating from these high estimates, the cost of revalidation of a compiler is likely to range from $25,000 to $50,000, provided that neither the compiler nor the ACVC test suite has changed significantly since the previous validation. Even for Registered Derived Compilers, the cost of ascertaining that the compilers indeed are capable of passing all applicable ACVC tests can be substantial to the compiler supplier.

For the user of Ada compilers, the economic benefits of using validated compilers can be considerable. First, the DoD mandate for using validated Ada compilers in MCCR projects makes it impossible to successfully compete for DoD contracts without access to a validated Ada compiler. Second, for the user concerned with porting Ada software between systems, the use of validated compilers is a first measure to reduce the cost of such porting by eliminating the problem with subset or superset compiler implementations. While compliance of compilers to the language standard is not a panacea for solving the porting problem, it eliminates one of the major hurdles encountered in the past with other languages.

12. The Impact of Validation on the Ada Effort

The requirement for validation has been blamed for the long delays in getting Ada compilers to the market place. Historically, the first compilers marketed for comprehensive languages such as COBOL or PL/1 processed only subsets of the respective language, thereby enabling first experiences to be gained with these new languages before compilers for the full languages appeared on the market. This introduction strategy is a two-edged sword, however, since the subset compilers typically remain in use, thus creating significant porting problems. Furthermore, by the time compilers for the full language emerge, programming styles have been established that perpetuate a limited utilization of the full capabilities of the language.

For Ada, DoD consciously decided on a strategy in which only compilers for the full language are acceptable for software development. Within this constraint, the availability of the ACVC has helped the implementors, by providing an extensive and well designed test set, as much as hindering them by forcing them into adherence to minute and sometimes pathological details of the Ada language standard and into the time-consuming process of formal validation.

By the end of 1985, more than 15 compilers were listed by AJPO as formally validated compilers, many of which are hosted on and targeted to a variety of systems. A significant number of additional compilers are expected to be validated in 1986. It is safe to say that the Ada effort is now past any initial delays that may have been caused by the mandate to validate Ada compilers.

The requirement of validation has also led to a situation in which compiler suppliers concentrate first and foremost on the task of obtaining validation status for their compilers, before concentrating on making their products sufficiently efficient and user-friendly to be viable tools in the development of Ada software. This situation must be considered a real danger to the casual public perception of the value of Ada compiler validation, as a validated compiler is by no means guaranteed to be a compiler usable in real application programming. Therefore it cannot be emphasized enough that evaluation of Ada compilers for their respective application domain remains a prime responsibility of the user. Validation can be one decision criterion in this evaluation, but surely not the only one.

Initial concerns about the combinatorial explosion of compiler validations with its cost and probable scarceness of AVF availability should be resolved by the proposed revised validation policies.

On the positive side, the requirement for validation has provided the user with an initial screening of the market place for Ada compilers: since the investment for getting an Ada compiler into a validatable state is

very high, the market place has focused on the viable compiler suppliers likely to provide good products and continued product support. Moreover, the publicized list of validated compilers, and the Validation Summary Reports (see Appendix A), which are publicly available, provide the user with valuable information about the products offered in the market place.

The validation process for Ada has raised the awareness of the suppliers as well as the users for the importance of compliance of a compiler with the programming language standards.

In the long run, DoD's persistence in its "no dialects"-policy for Ada and its validation requirement will have made a significant contribution to the portability of Ada programs and to the potential emergence of an Ada software component market which would be considerably less viable without such a standard. Compared to the cost reductions that can result from these facts, the cost of validation, especially when distributed over a large user base, is a minor expense that can be easily justified by the derived benefits.

APPENDIX A
THE TEN STEPS TO VALIDATION

A.1. Step 1: Self-Testing

The supplier of an Ada compiler must obtain a copy of the ACVC test suite for in-house testing of the compiler. The test suite is distributed by the AVF for a nominal charge covering the cost of distribution. The suppliers compile and execute the test suite using their compilers and, typically, will proceed to step 2 only when compliance of the compilers with the test suite has essentially been established.

A.2. Step 2: Notice of Intent to Validate

The suppliers must notify an AVF that they wish to be scheduled for formal validation, and establish a contract with the AVF for its services. The AVF will schedule validations on a first-come-first-served basis. It is therefore advisable for suppliers to contact an AVF as early as possible, once a target date for full compliance with the test suite can be internally established.

In planning for the validation, there are two dates that are of crucial importance to the suppliers:

- the expiration date of the test suite version that the supplier wishes to use for formal validation; and
- in the case of a revalidation, the expiration date of the existing validation certificate.

As there is generally a 90-day lead-time for an AVF in preparing the formal validation, as well as potential resource constraints of the AVF due to other scheduled validations, suppliers must plan well in advance to complete the validation process before these two crucial dates.

Formal validation testing with an already expired ACVC version is permissible only if a declaration of conformity (see step 4) has been made by the supplier prior to expiration of the ACVC version and no test disputes that are pending are decided against the supplier's position.

A.3. Step 3: Contract Negotiations with the AVF

A formal contract needs to be negotiated between the supplier and the AVF, in which schedules and payments to the AVF are agreed upon. As a major part of the work will be performed by the AVF prior to the on-site testing, advance payments will generally be required.

A.4. Step 4: Declaration of Conformity

The compiler supplier is required to provide a Declaration of Conformity to the AVF. In this declaration, the supplier asserts that the compiler to be validated conforms to the Ada language standard. Implementation-specific information, such as required by Appendix F of ANSI/MIL-STD-1815A, must be supplied to the AVF.

Furthermore, the results of running the ACVC tests on the compiler must be submitted to the AVF for evaluation at least 60-days prior to on-site testing, as well as a list of tests that the supplier believes to be incorrect or inapplicable to the compiler to be validated.

The AVF will carefully evaluate the submitted material, resolve any test disputes and issues in coordination with the supplier, and prepare a draft test report.

A.5. *Step 5: Resolution of Test Issues*

If the supplier and the AVF disagree on test methods or the applicability of tests, the AVF will refer these issues to the AVO for a binding decision.

The AVO will arrive at a decision within two workweeks and convey this decision to the AVF. The AVO is assisted in this process by the "Fast Reaction Team", a group of Ada language experts.

A.6. *Step 6: On-site Testing*

The AVF will conduct the conformity testing of the compiler at the location designated by the supplier. This test consists of running all applicable tests of the ACVC test suite. The compiler passes the conformance testing, if all tests are processed according to the language standard. A disposition on the outcome of the on-site testing will, however, not be made until step 8 is completed.

A.7. *Step 7: Report Preparation*

The AVF will collect and evaluate the results of on-site testing and then issue a draft Validation Summary Report (VSR) within 30 days of the on-site testing.

A.8. *Step 8: Review of Draft Report*

The supplier, AVO, and AJPO receive the draft VSR for concurrent review and comment. Any comments must be submitted to the AVF within two workweeks of receipt of the draft VSR. The AVF will then forward a final VSR to the AVO within two workweeks. If disagreements on the contents of the VSR arise, the Director of AJPO will decide after hearing arguments.

A.9. *Step 9: Approval Review*

The AVO will review the VSR for proper reflection of all comments made in Step 8, and then forward the VSR to the Director of AJPO for signature.

A.10. *Step 10: Issuing the Validation Certificate*

A certificate of conformity, called the "Validation Certificate", will be issued to the compiler supplier by the AJPO when the Director of AJPO has signed the VSR. This certificate lists the compiler and describes the configuration on which the compiler was tested on-site by the AVF. The Validation Certificate and the VSR are then made publicly available. The compiler is added to the list of validated compilers.

APPENDIX B
THE ADA VALIDATION FACILITIES

The following AVF are chartered by AJPO to perform Ada compiler validations. The ACVC test suite can be obtained from any AVF or directly from the Wright-Patterson Ada Validation Facility, which acts as the central maintainer of the test suite.

Wright-Patterson Ada Validation Facility
ASD/SIOL
Wright-Patterson AFB, OH 45433-6503
Tel: (513) 255-4472
Contact: Georgeane Chitwood

Federal Software Testing Center (FSTC)
Office of Software Development & Information Technology
Two Skyline Place, Suite 1100
5203 Leesburg Pike
Falls Church, VA 22041-3467
Tel: (703) 756-6153
Contact: Arnold Johnson

Bureau d'Orientation de la Normalisation en Informatique
Domain de Voluceau-Rocquencourt
B.P. 105
F-78153 Le Chesney, FRANCE
Tel: 33-3-955-2535
Contact: Jacqueline Sidi

IABG-AVF
Industrieanlagen-Betriebsgesellschaft (IABG)
Dept. SZT
Einsteinstrasse
D-8012 Ottobrunn, Fed. Rep. of Germany
Tel: 49-89-6008-3090
Contact: Helmut Hummel

Ministry of Defense (PE)
EQD "Aquila"
Golf Road
Bromley, Kent BR 1 2JB, United Kingdom
Tel: 01-467 2600 (Ext. 6056)
Contact: Kevin E. Phillips