**Ada Compiler Validation Summary Report**

**AdaCOMP_001, Version 1.3, VAX-11/750 host, Altera ACS 68000 14 target.**

<table>
<thead>
<tr>
<th><strong>PERFORMING ORGANIZATION NAME AND ADDRESS</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>BNI/AVF</td>
</tr>
<tr>
<td>Domaine de Voluceau - Rocquencourt</td>
</tr>
<tr>
<td>B.P. 105 - 78153 LE CHESNAY CEDEX, FRANCE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>CONTRACTING OFFICE NAME AND ADDRESS</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Ada Joint Program Office</td>
</tr>
<tr>
<td>1211 S. Fern Street, Rm. C-107</td>
</tr>
<tr>
<td>Arlington, VA 22202</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>NUMBER OF PAGES</strong></th>
<th>95</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th><strong>DISTRIBUTION STATEMENT</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Approved for public release: distribution unlimited</td>
</tr>
</tbody>
</table>

**Supplementary Notes**

**Key Words**


**Abstract**

See attached.

**DD FORM 1473 EDITION OF 1 NOV 69 IS OBSOLETE**
DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.
This Validation Summary Report presents the results and conclusions of testing performed on the AleyCOMP_005, version 1.0. Standardized tests serve as input to an Ada compiler, producing results which are evaluated by the validation team. This summary briefly states the highlights of the AleyCOMP_005, version 1.0 validation.

On-site testing was performed 31 October 1985 through 2 November 1985 at Aelys premises in La Celle Saint Cloud, France, under the auspices of the SNL (AVF), according to Ada Validation Office policies and procedures. The AleyCOMP_005, version 1.0 is hosted on SUN Workstation 2/120 and also on a SUN Workstation 2/50 operating under SUN UNIX 4.2 release 2.0. It is also hosted on SUN workstation 3/160 operating under SUN UNIX 4.2 release 3.0. The suite of tests known as the Ada Compiler Validation Capability (ACVC), Version 1.6. was used. The ACVC is used to validate conformance of a compiler to ANSI/MIL-STD-1815A Ada. The purpose of testing is to ensure that a compiler properly implements legal language constructs and that it identifies and rejects illegal language constructs. The testing also identifies behavior that is implementation dependent but permitted by the Ada Standard. Six classes of tests are used. These tests are designed to perform checks at compile time, at link time, or during execution.

The results of validation are summarized in the following table.

<table>
<thead>
<tr>
<th>RESULT</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>L</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Passed</td>
<td>60</td>
<td>777</td>
<td>961</td>
<td>16</td>
<td>0</td>
<td>8</td>
<td>1823</td>
</tr>
<tr>
<td>Failed</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Inapplicable</td>
<td>1</td>
<td>5</td>
<td>267</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>276</td>
</tr>
<tr>
<td>Anomalous</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Withdrawn</td>
<td>0</td>
<td>18</td>
<td>45</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>63</td>
</tr>
<tr>
<td>TOTAL</td>
<td>61</td>
<td>800</td>
<td>1273</td>
<td>17</td>
<td>3</td>
<td>2162</td>
<td></td>
</tr>
</tbody>
</table>

*Ada is a registered trademark of the United States Government
(Ada Joint Program Office)*
Ada COMILER VALIDATION SUMMARY REPORT:

ALSYS
AsysCOMP_001, version 1.3
VAX-11/750 host,
Altos ACS 68000 14 target

Completion of On-Site Validation:
8 November 1985

Prepared By:
BNI/AVF
Domaine de Voluceau - Rocquencourt
B.P.105 - 78153 LE CHESNAY CEDEX
FRANCE

Prepared For:
Ada Joint Program Office
United States Department of Defense
Washington, D.C.

Ada is a registered trademark of the United States Government
(Ada Joint Program Office)
Compiler Name: AlsyCOMP_W1P, version 1.3

Host Computer
VAX-11/780
under
VMS - version 4.1

Target Computer
ALTOS ACS 68000 14
under
ALTOS Operating system version 1

Testing Completed 8 November 1985 Using ACVC 1.6

This report has been reviewed and approved:

Ada Validation Facility
BNI
Nicolas Malagardis represented by Jacqueline Sidi
Domaine de Voluceau - Rocquencourt
B.P. 145 - 78153 LE CHESNAY CEDEX
FRANCE

John F. Kramer

Acting as the
Ada Validation Office (AVO)
John F. Kramer, Jr.
Institute for Defense Analyses
Alexandria, VA

Virginia L. Castor
Ada Joint Program Office (AJPO)
Virginia L. Castor
Director
Washington, D.C.

Distribution

Accession For
NTIS CRA&I
DTIC TAB
Unannounced
Justification

By
Distribution/
Availability Codes

Dist
Available and/or
Special

A-1 23

Ada is a registered trademark of the United States Government
(Ada Joint Program Office)
EXECUTIVE SUMMARY

This Validation Summary Report presents the results and conclusions of testing performed on the AlsyCOMP_001, version 1.3. Standardized tests serve as input to an Ada compiler, producing results which are evaluated by the validation team. This summary briefly states the highlights of the AlsyCOMP_001, version 1.3 validation.

On-site testing was performed 31 October 1985 through 8 November 1985 at Alsys premises in La Celle Saint Cloud - France, under the auspices of the BNH (AVF), according to Ada Validation Office policies and procedures. No precise timing information could be collected, as numerous problems arose in transferring the files from one machine to the other. This had for consequence several days delay. The AlsyCOMP_001, version 1.3 is hosted on VAX-11/750 operating under VMS version 4.1. The suite of tests known as the Ada Compiler Validation Capability (ACVC), Version 1.6, was used. The ACVC is used to validate conformance of a compiler to ANSI/MIL-STD-1815A Ada. The purpose of testing is to ensure that a compiler properly implements legal language constructs and that it identifies and rejects illegal language constructs. The testing also identifies behavior that is implementation dependent but permitted by the Ada Standard. Six classes of tests are used. These tests are designed to perform checks at compile time, at link time, or during execution.

The results of validation are summarized in the following table.

<table>
<thead>
<tr>
<th>RESULT</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>L</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Passed</td>
<td>60</td>
<td>777</td>
<td>961</td>
<td>8</td>
<td>1</td>
<td>1824</td>
<td></td>
</tr>
<tr>
<td>Failed</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Inapplicable</td>
<td>1</td>
<td>5</td>
<td>267</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>275</td>
</tr>
<tr>
<td>Anomalous</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Withdrawn</td>
<td>0</td>
<td>18</td>
<td>45</td>
<td>0</td>
<td>0</td>
<td>63</td>
<td>63</td>
</tr>
<tr>
<td>TOTAL</td>
<td>61</td>
<td>800</td>
<td>1273</td>
<td>17</td>
<td>8</td>
<td>3</td>
<td>2182</td>
</tr>
</tbody>
</table>

Ada is a registered trademark of the United States Government (Ada Joint Program Office)
Tests found to contain errors were withdrawn from Version 1.6 of the Ada Compiler Validation Capability (ACVC). When validation was completed, the following tests had been withdrawn:

```
B361050-AB  C45521A-Y-B (25 tests)  C48005C-B
C480060-B  C64103C-B
C64105C-AB  C64105F-AB
B767001A-B  B767004A-B
B774027A-B  C836005B-B
B8350070-B  BC322860-B
CA10030-AB  CA1011A-B
CA11609-B  CA1099B-B
BC1013A-B  BC3204A-D-B (4 tests)  BC3205A.D-B (4 tests)
BC34050-B  BC3503A-B
CE3403A-B  CE3604A-B
```

Some tests demonstrate that language features are not supported by an implementation. For this implementation the tests determined the following.

- SHORT_FLOAT is not supported:
  
  `BB6001C-AB.DEP  C34001F-B.DEP  C35702A-AB.DEP`

- LONG_FLOAT is not supported:
  
  `BB6001C-AB.DEP  C34001G-B.DEP  C35702B-AB.DEP`

- Representation specifications for noncontiguous enumeration representations are not allowed:
  
  `C55816A-AB.DEP`

- No other integer type other than INTEGER, SHORT_INTEGER, AND LONG_INTEGER is supported:
  
  `BB60001D-AB.DEP`

- The package SYSTEM is used by package TEXT_IO:
  
  `C56009F-B.ADA`

- The 'SIZE clause is not supported:
  
  `C87B862A-B.DEP`

- The 'STORAGE_SIZE clause is not supported:
  
  `C87B862B-B.DEP`

- The 'SMALL clause is not supported:
  
  `C87B862C-B.DEP`
Validation Summary Report  01/17/86  AlsyCOMP_001, version 1.3

Generic package bodies cannot be compiled in separate compilation files:

CA2009C--B.DEP

Pragma INLINE is not supported for procedures:

LA3044A--AB.ADA

Pragma INLINE is not supported for functions:

LA3044B--B.DEP

ACVC Version 1.6 was taken on-site via magnetic tape to Alsys premises in La Celle Saint Cloud - France. The tape was loaded, and all tests, except the withdrawn tests and any executable tests which make use of a floating point precision greater than SYSTEM.MAX_DIGITS, were compiled on VAX-11/750. Class A, C, D, and E tests were executed on the ALTOS.

On completion of testing, all results were analyzed for failed Class A, C, D, or E programs, and all Class B and L compilation results were individually analyzed.

The ACVC, Version 1.6, contains 2162 tests of which 1824 were applicable to AlsyCOMP_001, version 1.3. 21 tests were processed although inapplicable. No anomalies were found in the testing of this compiler. Testing demonstrated that all applicable tests were passed by this compiler. The AVF concluded that the results show acceptable compliance to ANSI/MIL-STD-1815A Add.
Table of contents

1- INTRODUCTION
  1.1- Purpose of this Validation Summary Report.................... 1-1
  1.2- Use of this Validation Summary Report....................... 1-2
  1.3- References............................................. 1-2
  1.4- Definition of Terms...................................... 1-3
  1.5- Configuration........................................... 1-5

2- TEST RESULTS
  2.1- ACVC Test Classes.......................................... 2-1
    2.1.1- Class A Tests........................................ 2-2
    2.1.2- Class B Tests....................................... 2-3
    2.1.3- Class C Tests....................................... 2-4
    2.1.4- Class D Tests....................................... 2-5
    2.1.5- Class E Tests....................................... 2-6
    2.1.6- Class L Tests....................................... 2-7
    2.1.7- Support Units........................................ 2-8
  2.2- Withdrawn Tests.......................................... 2-9
  2.3- Inapplicable Tests........................................ 2-12
  2.4- Implementation Characteristics................................ 2-14

3- COMPILER ANOMALIES AND NONCONFORMANCES
  3.1- Anomalies.................................................. 3-1
  3.2- Nonconformances........................................... 3-1

4- ADDITIONAL TESTING INFORMATION
  4.1- Pre-Validation............................................ 4-1
  4.2- Test Site................................................ 4-1
  4.3- Test Tape Information..................................... 4-1
  4.4- Testing Logistics........................................ 4-2
  4.5- Testing Duration......................................... 4-2

5- SUMMARY AND CONCLUSIONS

Appendix A - COMPLIANCE STATEMENT

Appendix B - TEST PARAMETERS

Appendix C - COMMAND SCRIPTS

Appendix D - COMPLETE LIST OF TESTS AND RESULTS
CHAPTER 1
INTRODUCTION

The Validation Summary Report describes how an Ada compiler conforms to the language standard. This report explains all technical terms used within and thoroughly reports the Ada Compiler Validation Capability (ACVC) test results. Ada compilers must be written according to the language specification as given in the ANSI/MIL-STD-1815A Ada. All implementation-defined features must be included for the compiler to conform to the Standard. Following the guidelines of the Standard ensures continuity between compilers. That is, the entire Standard must be implemented, and nothing can be implemented that is not in the Standard.

Even though all validated Ada compilers conform to the Standard, it must be understood that some differences do exist between implementations. ANSI/MIL-STD-1815A permits some implementation dependencies, e.g., the maximum length of identifiers, the maximum values of integer types, etc. These implementation-dependent features limit the portability of programs between compilers. Other differences between compilers are due to limitations imposed on a compiler by the operating system and by the hardware. All of these dependencies are given in the report.

Validation summary reports are written according to a standardized format. Compiler users can, therefore, more easily compare the reports from several compilers when selecting a compiler for a given task. The validation report can be completed mostly from the test results produced during validation testing. Additional testing information is given at the end of the report and states problems and details which are unique for a specific compiler. The format of the validation report limits variance between reports, enhances readability of the report, and accelerates report readiness.

1.1- Purpose of this Validation Summary Report

The Validation Summary Report documents the results of the testing performed on an Ada compiler. Testing was carried out for the following purposes:

- To identify any language constructs supported by the translator that do not conform to the Ada Standard
- To identify any unsupported language constructs required by the Ada Standard
To describe the implementation-dependent behavior allowed by the Ada Standard.

Testing of this compiler was conducted by BNI according to policies and procedures established by the Ada Validation Office (AVO). Testing was conducted from 31 October 1985 through 8 November 1985 at Alsys premises in La Celle Saint Cloud - France. No precise timing information could be collected as numerous problems arose in transferring the files from one machine to the other. This had for consequence several days delay.

1.2- Use of this Validation Summary Report

Consistent with the national laws of the originating country, the Ada Validation Office may make full and free public disclosure of this report. In the United States, this is provided in accordance with the "Freedom of Information Act" (5 U.S.C. §552). The results of this validation apply only to the computers, operating systems, and compiler versions identified in this report.

The organizations represented on the signature page of this report do not represent or warrant that any statement or statements set forth in this report are accurate or complete, or that the subject compiler has no nonconformances to the Ada Standard other than those presented. This report is not intended for the purpose of publicizing the findings summarized herein.

Questions regarding this report or the validation tests should be directed to:

Ada Validation Office
Institute for Defense Analyses
1801 N. Beauregard
Alexandria VA 22311

and to:

BNI
Domaine de Voluceau - Rocquencourt
B.P. 105 - 78153 LE CHESNAY CEDEX
FRANCE

1.3- References

1.4 - Definition of Terms

Anomaly A test result that, given pre-validation analysis, is not expected during formal validation but is judged allowable under the circumstances.

ACVC The Ada Compiler Validation Capability. A set of programs that evaluates the conformance of a compiler to the Ada language specification, ANSI/MIL-STD-1815A.


Applicant The agency requesting validation.

AVF The BNI. In the context of this report, the AVF is responsible for conducting compiler validations according to established policies and procedures.

AVO The Ada Validation Office. In the context of this report, the AVO is responsible for setting policies and procedures for compiler validations.

Compiler A processor for the Ada language. In the context of this report, a compiler is any language processor, including cross-compilers, translators, and interpreters.

Failed test A test for which the compiler generates a result that demonstrates nonconformance to the Ada Standard.

Host The computer on which the compiler resides.

Inapplicable test A test that uses features of the language that a compiler is not required to support or may legitimately support in a way other than the one expected by the test.

Passed test A test for which a compiler generates the expected result.

Target The computer for which a compiler generates code.

Test A program that evaluates the conformance of a compiler to a language specification. In the context of this report, the term is used to designate a single ACVC test. The text of a program may be the text of one or more compilations.
Withdrawn test

A test that has an invalid test objective, fails to meet its test objective, or contains illegal use of the language.
1.5- Configuration

The candidate compilation system for this validation was tested under the configuration:

- **Compiler:** Alsycop_001, version 1.3
- **Test Suite:** Ada Compiler Validation Capability, Version 1.6
- **Host Computer:**
  - Machine(s): VAX-11/750
  - Operating System: VMS - version 4.1
  - Memory Size: 6 Megabytes
  - Disk System: 456 Megabytes
- **Target Computer:**
  - Machine(s): ALTOS ACS 68000 14
  - Operating System: ALTOS Operating System version 1
  - Memory Size: 1 Megabyte
  - Disk System: 40 Megabytes
CHAPTER 2

TEST RESULTS

2.1- ACVC Test Classes

Conformance to ANSI/MIL-STD-1815A is measured using the Ada Compiler Validation Capability (ACVC). The ACVC contains both legal and illegal Ada programs structured into six test classes: A, B, C, D, E, and L. Legal programs are compiled and executed while illegal programs are just compiled. Support packages are used to report the results of the legal programs. A compiler must correctly process each of the tests in the suite and demonstrate conformance to the Ada Standard by either meeting the pass criteria given for the test or by showing that the test is inapplicable to the implementation. Tests that are found to contain errors are withdrawn from the ACVC. Detailed test results are listed in the Appendix D. The results of validation testing are summarized in the following table:

<table>
<thead>
<tr>
<th>RESULT</th>
<th>TEST CLASS</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Passed</td>
<td>A  60</td>
<td>B  777</td>
</tr>
<tr>
<td>Failed</td>
<td>A  0</td>
<td>B  0</td>
</tr>
<tr>
<td>Inapplicable</td>
<td>A  1</td>
<td>B  5</td>
</tr>
<tr>
<td>Anomalous</td>
<td>A  0</td>
<td>B  0</td>
</tr>
<tr>
<td>Withdrawn</td>
<td>A  0</td>
<td>B  18</td>
</tr>
<tr>
<td>TOTAL</td>
<td>A  61</td>
<td>B  800</td>
</tr>
</tbody>
</table>

A total of 1845 tests were processed during this validation attempt. The 63 withdrawn tests in Version 1.6 were not processed, nor were 254 Class C tests that were inapplicable because they use floating point types having digits that exceed the maximum value for the implementation. All other tests were processed.

Some conventions are followed in the ACVC to ensure that the tests are reasonably portable without modification. For example, the tests make use of only the basic 55 character set, contain lines with a maximum length of 72 characters, use small numeric values, and place features that may not be supported in separate tests. However, some tests contain values that require the test to be customized according to implementation-specific values. The values used for this validation are listed in Appendix B.
2.1.1- Class A Tests

Class A tests check that legal Ada programs can be successfully compiled and executed. However, no checks are performed during execution to see if the test objective has been met. For example, a Class A test checks that reserved words of another language other than those already reserved in the Ada language) are not treated as reserved words by an Ada compiler. A Class A test is passed if no errors are detected at compile time and the program executes to produce a message indicating that it has passed. If a Class A test cannot be compiled and executed because of its size, then the test is split into a set of smaller subtests that can be processed. A split was required for 1 test:

AE2181A-B.ADA

The following table shows that all applicable Class A tests were passed:

<table>
<thead>
<tr>
<th>RESULT</th>
<th>CHAPTER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2</td>
</tr>
<tr>
<td>Passed</td>
<td>13</td>
</tr>
<tr>
<td>Failed</td>
<td>0</td>
</tr>
<tr>
<td>Inapplicable</td>
<td>0</td>
</tr>
<tr>
<td>Anomalous</td>
<td>0</td>
</tr>
<tr>
<td>Withdrawn</td>
<td>0</td>
</tr>
<tr>
<td>TOTAL</td>
<td>13</td>
</tr>
</tbody>
</table>
2.1.2- Class B Tests

Class B tests check that a compiler detects illegal language usage. Class B tests are not executable. Each test in this class is compiled and the resulting compilation listing is examined manually to verify that every syntax or semantic error in the test is detected. A Class B test is passed if every illegal construct that it contains is detected by the compiler. If one or more errors are not detected, then a version of the test is created that contains only the undetected errors. The resulting “split” is compiled and examined. The splitting process continues until all errors are detected by the compiler. Splits were required for 15 tests:

<table>
<thead>
<tr>
<th>Split</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>B32202A-B.ADA</td>
<td>B32202B-B.ADA</td>
</tr>
<tr>
<td>B33088A-B.ADA</td>
<td>B37004A-B.ADA</td>
</tr>
<tr>
<td>B43201D-B.ADA</td>
<td>B51012A-B.ADA</td>
</tr>
<tr>
<td>B63010C-B.ADA</td>
<td>B620010B-B.ADA</td>
</tr>
<tr>
<td>BA20012W-AB.ADA</td>
<td>BA20012M-AB.ADA</td>
</tr>
</tbody>
</table>

The following table shows that all applicable Class B tests were passed:

<table>
<thead>
<tr>
<th>RESULT</th>
<th>CHAPTER 2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Passed</td>
<td>35</td>
<td>72</td>
<td>83</td>
<td>113</td>
<td>70</td>
<td>55</td>
<td>49</td>
<td>91</td>
<td>56</td>
<td>88</td>
<td>147</td>
<td>18</td>
<td>777</td>
</tr>
<tr>
<td>Failed</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Inapplicable</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>Anomalous</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Withdrawn</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>18</td>
</tr>
<tr>
<td>TOTAL</td>
<td>35</td>
<td>73</td>
<td>83</td>
<td>113</td>
<td>73</td>
<td>57</td>
<td>52</td>
<td>92</td>
<td>36</td>
<td>88</td>
<td>168</td>
<td>18</td>
<td>886</td>
</tr>
</tbody>
</table>

2-3
Class C tests check that legal Ada programs can be correctly compiled and executed. Each Class C test is self-checking and produces a PASS/FAIL message indicating the result when it is executed. If a Class C test cannot be compiled because it exceeds the compiler's capacity, then the test is split into smaller subtests until all are compiled and executed. No splits were required.

The following table shows that all applicable Class C tests were passed:

<table>
<thead>
<tr>
<th>RESULT</th>
<th>CHAPTER 2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Passed</td>
<td>19 89 153</td>
<td>115</td>
<td>70</td>
<td>14</td>
<td>93 166 35</td>
<td>20</td>
<td>55 192</td>
<td>961</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Failed</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inapplicable</td>
<td>23 119 116</td>
<td>4 0 0 4 0 1 0 0 0</td>
<td>267</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Anomalous</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Withdrawn</td>
<td>0 0 27 0 4 0 0 3 7 0 0 0 4 45</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td>42 208 296 119 74 14 97 180 43 20 55 196</td>
<td>1273</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2-4
2.1.4 - Class D Tests

Class D tests check the compilation and execution capacities of a compiler. Since there are no requirements placed on a compiler by the Ada Standard for the number of identifiers permitted in a compilation, the number of units in a library, the number of nested loops in a subprogram body, and so on, a compiler may refuse to compile a Class D test. Each Class D test is self-checking and produces a PASS/FAIL message indicating the result when it is executed. If a Class D test fails to compile because the capacity of the compiler is exceeded, then the test is classified as inapplicable.

The following table shows that all applicable Class D tests were passed:

<table>
<thead>
<tr>
<th>RESULT</th>
<th>CHAPTER_1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>14</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Passed</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>9</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>17</td>
</tr>
<tr>
<td>Failed</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Inapplicable</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Anomalous</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Withdrawn</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>TOTAL</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>9</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>17</td>
</tr>
</tbody>
</table>

Capacities measured by the Class D tests are detailed in section 2.4, IMPLEMENTATION CHARACTERISTICS.
Class E tests provide information about the compiler in those areas in which the Ada Standard permits implementations to differ. Each Class E test is executable and produces messages that indicate how the Ada Standard is interpreted. However, in some cases the Ada Standard permits a compiler to detect a condition either at compile time or at execution time. And thus a Class E test may correctly fail to execute. A Class E test is passed if it fails to compile and appropriate error messages are issued, or if it executes properly and produces a message that it has passed. If a Class E test cannot be compiled and executed because of its size, then the test is split into a set of smaller subtests that can be processed. No splits were required.

The following table shows that all applicable Class E tests were passed:

<table>
<thead>
<tr>
<th>RESULT</th>
<th>CHAPTER</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Passed</td>
<td></td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Failed</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Inapplicable</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Anomalous</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Withdrawn</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>TOTAL</td>
<td></td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Information obtained from the Class E tests is detailed in section 2.4, IMPLEMENTATION CHARACTERISTICS.
Class L tests check that incomplete or illegal Ada programs involving multiple, separately compiled units are detected and not allowed to execute. Class L tests are compiled separately and execution is attempted. A Class L test passes if it is rejected at link time and the test does not execute.

The following table shows that all applicable Class L tests were passed:

<table>
<thead>
<tr>
<th>RESULT</th>
<th>CHAPTER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Passed</td>
<td>0 0 0 0 0 0 0 1 0 0 0 1</td>
</tr>
<tr>
<td>Failed</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>Inapplicable</td>
<td>0 0 0 0 0 0 0 2 0 0 0 2</td>
</tr>
<tr>
<td>Anomalous</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>Withdrawn</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>TOTAL</td>
<td>0 0 0 0 0 0 0 3 0 0 0 3</td>
</tr>
</tbody>
</table>
2.1.7- Support Units

Three packages support the self-checking features of Class C tests: REPORT, CHECK_FILE, and VAR_STRINGS. The REPORT package provides the mechanism by which executable tests report results. It also provides a set of identity functions that are used to defeat some compiler optimization strategies to cause computations to be made by the target computer instead of the compiler on the host computer. The CHECK_FILE package is used to check the contents of text files written by some of the Class C tests for Chapter 14 of the Ada Standard. The VAR_STRINGS package defines types and subprograms for manipulating varying-length character strings. The operation of these three packages is checked by a set of executable tests. These tests produce messages that are examined manually to verify that the packages are operating correctly. If these packages are not operating correctly, then validation is not attempted.

An applicant is permitted to substitute the body of package REPORT with an equivalent one if for some reason the original version provided by the ACVC cannot be executed on the target computer. Package REPORT was not modified for this validation.

All support package specifications and bodies were compiled and were demonstrated to be operating correctly.
2.2- Withdrawn Tests

Some tests are withdrawn from the ACVC because they do not conform to the Ada Standard. When testing was performed, the following 63 tests had been withdrawn for the reasons indicated:

B381059-AB:

This test requires a specific interpretation of the Ada Standard regarding whether an incomplete type can have discriminant constraints before the full type declaration; this interpretation is not fully supported by the Ada Standard or Language Maintenance Committee (LMC).

C45521A-Y-B (25 tests):

Cases C and J define the model interval for the result too narrowly.

C488055C-B:

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

C488084B-B:

This test requires a specific interpretation of the Ada Standard regarding whether an incomplete type can have discriminant constraints before the full type declaration; this interpretation is not fully supported by the Ada Standard or Language Maintenance Committee.

C64103B-B:

This test should raise CONSTRAINT_ERROR during the conversion at line 179.

C64103D-B:

This test involves a CONSTRAINT_ERROR vs. NUMERIC_ERROR issue that is to be resolved by the Language Maintenance Committee.

C64105E-AB:

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

C64105F-AB:

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

B65801A-B:

This test checks (in section C) that a function without parameters, which is equivalent to an enumeration literal in the same declarative region, is a redeclaration and as such is forbidden. According to the Ada Standard B.3(17), the explicit declaration of such a function is allowed if an enumeration literal is considered to be an implicitly declared predefined operation. The Ada Standard is not clear on this point. This issue has been referred to the Language Maintenance Committee for resolution. Since the issue cannot be resolved at this time, the test is withdrawn from Version 1.6.

B67801A-B:

Line 414 is missing the "BEGIN NULL; END:" needed to complete the block beginning at line 369 (case H).
B6788A-B:
This default name for a formal generic equality function should not be allowed to be "/=" unless an expanded name is used.

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

B74207A-B:
This test requires a specific interpretation of the Ada Standard regarding whether an incomplete type can have discriminant constraints before the full type declaration; this interpretation is not fully supported by the Ada Standard or Language Maintenance Committee.

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

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

C11811A-B:
The test objective should be reversed to be consistent with AI-00199.

C1188A-B:
A pragma ELABORATE is needed for OTHER_PKG at line 25.

C1188B-B:
A pragma ELABORATE is needed for FIRST_PKG at line 39 and for LATER_PKG at line 49.

C20809B-B:
The repetition of the main procedure after the subunit body makes the subunit body obsolete; therefore, an attempt to execute the main procedure will fail.

C20809E-B:
The repetition of the main procedure after the subunit body makes the subunit body obsolete; therefore, an attempt to execute the main procedure will fail.

C20809F-B:
The file CA20809F2-B is missing from this test suite.

B11013A-B:
The declaration of equality in lines 86-87 is illegal because the parameter type T declared in line 11 is not a limited type (Ada Standard 6.7-4).
Instantiations with types that have default discriminants are now legal (see AI-00637).

This test assumes that the staticness of instantiated generic parameters follows from the staticness of the actual parameter of the instantiation. This compiler treats all such instantiated parameters as non-static. The matter is before the LMC for resolution.

This test requires a specific interpretation of the Ada Standard regarding whether an incomplete type can have discriminant constraints before the full type declaration; this interpretation is not fully supported by the Ada Standard or Language Maintenance Committee.

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

The last case is inconsistent with AI-00050. If string argument is null, no attempt to read is made and ENDERROR is not raised.

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

A superfluous SKIP_LINE causes the input and output operations to be out of synchronization.
2.3- Inapplicable Tests

Some tests use features of the Ada language that the Ada Standard does not require a compiler to support; thus these tests may be inapplicable to a particular compiler. Others may depend on the result of another test that is either inapplicable or withdrawn. For this validation attempt, 275 tests were inapplicable for the reasons indicated:

A9102M.B.ADA:
This test is inapplicable because this implementation does not support certain pragmas such as CONTROLLED.

B86001D-AB.TST:
This test is inapplicable because this implementation has no predefined type other than INTEGER, FLOAT, SHORT_INTEGER, SHORT_FLOAT, LONG_INTEGER, LONG_FLOAT. The macro name SNAME was set to NO_SUCH_TYPE and the declaration of a procedure name NO_SUCH_TYPE is then legal.

C24113C..Y-8.DEF
C35785C..Y-8.DEF
C35806C..Y-8.DEF
C35787C..Y-8.DEF
C35788C..Y-8.DEF
C35802C..Y-8.DEF
C45241C..Y-8.DEF
C45321C..Y-8.DEF
C45421C..Y-8.DEF
C45424C..Y-8.DEF
C45621C..Y-8.DEF
C45113C..Y-12.DEF

These tests are inapplicable because this implementation limits digits to 6.

B86001CP-AB.DEF
C34001F-B.DEF
C35792A-AB.DEF:
These tests are inapplicable because this implementation does not support SHORT_FLOAT.

B86001CO-AB.DEF
C34001G-B.DEF
C35792B-AB.DEF:
These tests are inapplicable because this implementation does not support LONG_FLOAT.
These tests are inapplicable because this implementation does not support representation clauses.

This test is inapplicable because this implementation reflects the recompilation of SYSTEM at compilation-time.

This test is inapplicable because this implementation does not support instantiating missing generic bodies.

These tests are inapplicable because this implementation does not support pragma INLINE. These tests ignore the pragma and are processed correctly.

These tests are inapplicable because this implementation does not support pragma PACK. These tests ignore the pragma and are processed correctly.
2.4- Implementation Characteristics

One of the purposes of validation is to determine the behavior of a compiler in those areas of the Ada Standard that permit implementations to differ. Class D and E tests specifically check for such implementation differences. However, inapplicable tests in other classes also characterize an implementation. This compiler is characterized by the following interpretations of the Ada Standard:

- Non-graphic characters.
  Non-graphic characters are defined in the ASCII character set but are not permitted in Ada programs, even within character strings. The compiler correctly recognizes these characters as illegal in Ada compilations. The characters are not printed in the output listing.

- Capacities.
  The compiler correctly processes compilations containing loop statements nested to 65 levels, block statements nested to 65 levels, procedures nested to 10 levels, and 723 variables.

- Universal integer calculations.
  An implementation is allowed to reject universal integer calculations having values that exceed SYSTEM.MAX_INT. This implementation does not reject such calculations and processes them correctly.

- Universal real calculations.
  An implementation is allowed to reject universal real calculations having values that exceed certain precisions. This implementation does not reject such calculations and processes them correctly.

- No rounding in this compiler. The precision is arbitrarily high.

- Predefined types.
  This implementation supports the predefined types SHORT_INTEGER, LONG_INTEGER, INTEGER, FLOAT, DURATION. It does not support any other predefined numeric types.
Based literals.

An implementation is allowed to reject a based literal with value exceeding SYSTEM_MAX_INT during compilation or it may raise NUMERIC_ERROR during execution. This compiler raises NUMERIC_ERROR during execution.

Array types.

An implementation is allowed to raise NUMERIC_ERROR for an array having a 'LENGTH that exceeds STANDARD.INTEGER'LAST and/or SYSTEM_MAX_INT. When an array type is declared with an index range exceeding INTEGER values and with a component that is a null BOOLEAN array, this compiler does not raise any exception.

When an array type is declared with an index range exceeding SYSTEM_MAX_INT values and with a component that is a null BOOLEAN array, this compiler raises NUMERIC_ERROR.

A packed BOOLEAN array of length INTEGER'LAST+3 does not raise any exception. A packed two-dimensional BOOLEAN array with INTEGER'LAST+3 components does not raise any exception.

A null array with one dimension of length exceeding INTEGER'LAST does not raise any exception.

In assigning one-dimensional array types, the entire expression is evaluated before CONSTRAINT_ERROR is raised when checking whether the expression's subtype is compatible with the target's subtype. In assigning two-dimensional array types, the entire expression is not evaluated before CONSTRAINT_ERROR is raised when checking whether the expression's subtype is compatible with the target's subtype.

Discriminated types.

An incompletely declared type with discriminants may be used in an access type definition and constrained either there or in later subtype indications.

Aggregates.

When evaluating the choices of a multi-dimensional aggregate all choices are evaluated before checking against the index type.

When evaluating an aggregate containing subaggregates, all choices are not evaluated before being checked for identical bounds.
Functions.

The declaration of a parameterless function with the same profile as an enumeration literal in the same immediate scope is rejected by the implementation.

Representation clauses.

'SMALL length clauses are not supported.

Enumeration representation clauses are not supported.

Tasks.

A task object's storage size is not allowed to change after the task is activated.

Generics.

When given a separately compiled generic declaration, some illegal instantiations, and a body, the compiler rejects the body because of the instantiations.

Package CALENDAR.

TIME_OF and SPLIT are inverses when SECONDS is a non-model number.

Pragma INLINE is not supported for procedures. It is not supported for functions.

Input/output.

Package SEQUENTIAL_IO can be instantiated with unconstrained array types and record types with discriminants. Package DIRECT_IO can be instantiated with unconstrained array types and record types with discriminants without defaults.

For SEQUENTIAL_IO, DIRECT_IO and TEXT_IO more than one internal file can be associated with each external file for both reading and writing. An external file associated with more than one internal file can be deleted.

An existing text file can be opened in OUT_FILE mode, can be created in OUT_FILE mode, and can be created in IN_FILE mode.

Dynamic creation and resetting of a sequential file is allowed.

Temporary sequential files are given a name. Temporary direct files are given a name. Temporary files given names are deleted when they are closed.

2-16
CHAPTER 3

COMPILER ANOMALIES AND NONCONFORMANCES

3.1- Anomalies

An anomaly is a test result that, given the pre-validation analysis, was not expected during formal validation but which is judged allowable by the AVF and the AVO under the circumstances of the validation. No anomalies were detected in this validation attempt.

3.2- Nonconformances

Any discrepancy between expected test results and actual test results is considered to be a nonconformance. No nonconformances were detected in this validation attempt.
4.1- Pre-Validation

Prior to validation, a set of test results for ACVC 1.6 produced by Alsys COMP_001, version 1.3 was submitted to BNI by the applicant for pre-validation review. Analysis of these results demonstrated that the compiler successfully passed all applicable tests.

4.2- Test Site

Tests were compiled and executed at Alsys premises in La Coelle Saint Cloud—France.

4.3- Test Tape Information

A test tape containing ACVC Version 1.6 was taken on-site by the validation team. This tape contained all tests applicable to this validation as well as all tests inapplicable to this validation except for any Class C tests that require floating-point precision exceeding the maximum value supported by the implementation. Tests that were withdrawn from ACVC 1.6 were not written to the tape. Tests that make use of values that are specific to an implementation were customized before being written to the tape. Any split tests were also included on the test tape so that no editing of the test files was necessary when the validation team arrived on-site.

The test files were mounted on the VAX. Only one directory was used. The format of these test tape was the same as the ACVC distribution tapes.
4.4- Testing Logistics

Once all tests had been loaded to disk, processing was begun using command scripts provided by ALSYS. The text of these scripts are given in Appendix C.

The output of the host machine was on tape. It was then transferred to the target disk using a standard communication line. The operation of loading the target and executing the tests did not depend on the host. This is due to the fact that each machine has its own operating system. The results of execution were transferred back to the VAX to be forwarded to the BNI on tape for analysis.

The compiler supports various options that control its operation. The compiler was tested with the following option settings.

The following options were used:

- error_limit=999: extension of the implicit number of errors before abortion
- line=120: line length
- short: no compilation listing
- long: compilation listing
- banner: banner for each test
- nosummary: no recapitulation of errors

The B tests were compiled with the options: error_limit=999, line=120, long, banner, nosummary.

The other tests that do not execute were compiled with the options: error_limit=999, line=120, long, banner, nosummary.

The tests that do execute were compiled with the options: error_limit=999, line=120, short, banner, nosummary.

The tests were run in the following order: A, B, C, D, E and L.

One Ada library per ACVC chapter was used.

4.5- Testing Duration

The ACVC has not been designed for use in measuring compiler performance. The information reported here thus merely describes the duration of the on-site testing for conformity, and is not necessarily an indication of the subject system's performance.
The validation started on the 31 October 1985. It finished on the 8 November 1985. No precise timing information could be collected as numerous problems arose in transferring the files from one machine to the other. This had for consequence several days delay.
The BNI identified 1845 of the 2162 tests in ACVC version 1.6 to be processed during the validation of AlsyCOMP_001, version 1.0. Excluded were 254 tests requiring too great a floating-point precision, and the 63 withdrawn tests. 21 tests were determined to be inapplicable after they were processed. The remaining 1824 tests were passed by the compiler.

The BNI concludes that these results demonstrate acceptable conformance to the Ada Standard.
The only allowed implementation dependencies correspond to implementation-dependent progtms and attributes, to certain machine-dependent conventions as mentioned in Chapter 13 of MIL-STD-1815A, and to certain allowed restrictions on representation classes. The implementation-dependent characteristics of the AisyCOMP_001, version 1.3 are described in the following sections which discuss topics one through eight as stated in Appendix F of the Ada Standard.

(1) Implementation-Dependent Progtms

None.

(2) Implementation-Dependent Attributes

None.
(3) Package SYSTEM

The specification for package SYSTEM is

```pascal
package SYSTEM is
    type ADDRESS is private;
    type NAME is ('UNIX');

    SYSTEM_NAME : constant NAME := UNIX;
    STORAGE_UNIT : constant := 8;
    MEMORY_SIZE : constant := 2^24 - 1;

    -- System-Dependent Named Numbers:
    MIN_INT      : constant := -(2^31);
    MAX_INT      : constant := 2^31-1;
    MAX_DIGITS   : constant := 6;
    MAX_MANTISSA : constant := 31;
    FINE_DELTA   : constant := 2**1.0**-31;
    TICK         : constant := 1.0;

    -- Other System-Dependent Declarations
    subtype PRIORITY is INTEGER range 1..127;
end SYSTEM;
```

(4) Representation Clause Restrictions

Representation clauses specify how the types of the language are to be mapped onto the underlying machine. The following are restrictions on representation clauses.

Address Clause
Not accepted

Length Clause
Not accepted

Enumeration Representation Clause
Not accepted

Record Representation Clause
Not accepted

A-2
(5) Conventions

No implementation-generated names.

(6) Address Clauses

Not accepted.

(7) Unchecked Conversions

The following are restrictions on unchecked conversions, including those depending on the respective sizes of objects of the source and target.

They should have the same size.

(8) Input-Output Packages

The following are implementation-dependent characteristics of the input-output packages.

SEQUENTIAL_IO Package

Declare file type and applicable operations for files of this type.

There is no restriction in the use of sequential Input/Output.

DIRECT_IO Package

type COUNT is range 0..2_147_483_647;

TEXT_IO Package

type COUNT is range 0..2_147_483_647;

subtype FIELD is INTEGER range 0..255;
(9) Package STANDARD

    type INTEGER is range -32768 .. 32767;
    type SHORT_INTEGER is range -128 .. 127;
    type LONG_INTEGER is range -2_147_483_648 .. 2_147_483_647;

    type FLOAT is digits 6 range

    No other additional predefined floating point types

    type DURATION is delta 0.002 range -86,400.0 .. 86,400.0;

    No other predefined types

(10) File Names

    File names make no use of conventions except those of the operating system.
APPENDIX B

TEST PARAMETERS

Certain tests in the ACVC make use of implementation-dependent values, such as the maximum length of an input line and invalid file names. A test that makes use of such values is identified by the extension .TST in its file name. Actual values to be substituted are identified by names that begin with a dollar sign. A value is substituted for each of these names before the test is run. The values used for this validation are given below.

<table>
<thead>
<tr>
<th>Name and Meaning</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$MAX_IN_LEN</td>
<td>255</td>
</tr>
</tbody>
</table>

Maximum input line length permitted by the implementation.

$BIG_ID1
Identifier of size MAX_IN_LEN with varying last character.

$BIG_ID2
Identifier of size MAX_IN_LEN with varying last character.
<table>
<thead>
<tr>
<th>Name and Meaning</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$BIG_ID3</td>
<td>Identifier of size \texttt{MAX_LEN} with varying middle character.</td>
</tr>
<tr>
<td></td>
<td>$X_{2345678901234567890123456789012345} \ldots$</td>
</tr>
<tr>
<td></td>
<td>\texttt{AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA}</td>
</tr>
<tr>
<td></td>
<td>\texttt{AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA}</td>
</tr>
<tr>
<td></td>
<td>\texttt{AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA}</td>
</tr>
<tr>
<td></td>
<td>\texttt{AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA}</td>
</tr>
<tr>
<td>$BIG_ID4</td>
<td>Identifier of size \texttt{MAX_LEN} with varying middle character.</td>
</tr>
<tr>
<td></td>
<td>$X_{2345678901234567890123456789012345} \ldots$</td>
</tr>
<tr>
<td></td>
<td>\texttt{AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA}</td>
</tr>
<tr>
<td></td>
<td>\texttt{AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA}</td>
</tr>
<tr>
<td></td>
<td>\texttt{AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA}</td>
</tr>
<tr>
<td></td>
<td>\texttt{AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA}</td>
</tr>
<tr>
<td>$NEG_BASED_INT$</td>
<td>A based integer literal whose highest order non-zero bit falls in the sign bit position of the representation for \texttt{SYSTEM_MAX_INT}.</td>
</tr>
<tr>
<td></td>
<td>\texttt{16#FF_FF_FD#}</td>
</tr>
<tr>
<td>$BIG_INT_LIT$</td>
<td>An integer literal of value 298 with enough leading zeroes so that it is \texttt{MAX_LEN} characters long.</td>
</tr>
<tr>
<td></td>
<td>\texttt{0} \ldots \texttt{298}</td>
</tr>
<tr>
<td>$BIG_REAL_LIT$</td>
<td>A real literal that can be either of floating or fixed point type, has value 690.0, and has enough leading zeroes to be \texttt{MAX_LEN} characters long.</td>
</tr>
<tr>
<td></td>
<td>\texttt{0} \ldots \texttt{690.0} \texttt{E1}</td>
</tr>
<tr>
<td>$EXTENDED_ASCII_CHARS$</td>
<td>A string literal containing all the ASCII characters with printable graphics that are not in the basic 55 Ada character set.</td>
</tr>
<tr>
<td></td>
<td>\texttt{&quot;abcdefghijklmnopqrstuvwxyz&quot;}$\texttt{$#[]}$ \texttt{&amp; {} \texttt{'} \texttt{}}$</td>
</tr>
</tbody>
</table>
$NON_ASCII_CHAR_TYPE (NON_NULL)
An enumerated type definition for a character type whose literals are the identifier NON_NULL and all non-ASCII characters with printable graphics.

$BLANKS
Blanks of length MAX_IN_LEN - 20

$MAX_DIGITS 6
Maximum digits supported for floating point types.

$NAME NO_SUCH_TYPE
A name of a predefined numeric type other than FLOAT, INTEGER, SHORT_FLOAT, SHORT_INTEGER, LONG_FLOAT, LONG_INTEGER, or DURATION.

$INTEGER_FIRST -32768
The universal integer literal expression whose value is INTEGER'FIRST.

$INTEGER_LAST 32767
The universal integer literal expression whose value is INTEGER'LAST.

$LESS_THAN_DURATION -100.000.0
A universal real value that lies between DURATION'BASE'FIRST and DURATION'FIRST or any value in the range of DURATION.

$GREATER_THAN_DURATION 100.000.0
A universal real value that lies between DURATION'BASE'LAST and DURATION'LAST or any value in the range of DURATION.
## Validation Summary Report

**01/17/86 AlsyCOMP_001, version 1.3**

<table>
<thead>
<tr>
<th>Name and Meaning</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$LESS_THAN_DURATION_BASE_FIRST</td>
<td>$-100_000_000.0$</td>
</tr>
<tr>
<td>The universal real value that is less than DURATION'BASE'FIRST.</td>
<td></td>
</tr>
<tr>
<td>$GREATER_THAN_DURATION_BASE_LAST</td>
<td>$100_000_000.0$</td>
</tr>
<tr>
<td>The universal real value that is greater than DURATION'BASE'LAST.</td>
<td></td>
</tr>
<tr>
<td>$COUNT_LAST</td>
<td>2,147,483,647</td>
</tr>
<tr>
<td>Value of COUNT'LAST in TEXT_IO package.</td>
<td></td>
</tr>
<tr>
<td>$FIELD_LAST</td>
<td>255</td>
</tr>
<tr>
<td>Value of FIELD'LAST in TEXT_IO package.</td>
<td></td>
</tr>
<tr>
<td>$FILE_NAME_WITH_BAD_CHARS</td>
<td>ABCDEFGHIJKLMNOPQRSTUVWXYZ</td>
</tr>
<tr>
<td>An illegal external file name that either contains invalid characters or is too long.</td>
<td></td>
</tr>
<tr>
<td>$FILE_NAME_WITH_WILD_CARD_CHAR</td>
<td>123456789012345</td>
</tr>
<tr>
<td>An external file name that either contains a wild card character or is too long.</td>
<td></td>
</tr>
<tr>
<td>$ILLEGAL_EXTERNAL_FILE_NAME1</td>
<td>BAD_CHARACTER++</td>
</tr>
<tr>
<td>Illegal external file name.</td>
<td></td>
</tr>
<tr>
<td>$ILLEGAL_EXTERNAL_FILE_NAME2</td>
<td>MUCH_TOO_LONG_NAME_FOR_A_FILE</td>
</tr>
<tr>
<td>Illegal external file name.</td>
<td></td>
</tr>
</tbody>
</table>
APPENDIX C

COMMAND SCRIPTS
C-2
FILE = fsearch("temp.ini")
if FILE == "" then goto wait
retry:
  on transfer=locked obj_file 'file'
close obj_file
idle = 0
set nonon
kernel send 'file'
del 'file'
sut on
auto rnon
!
wait:
if idle < 4 then goto failed ! don't wait more than 4 hours
idle = idle + 1
wait 10!1 ! wait one minute.
auto rnon
!
slocked:
  wait 10!
  auto retry
  goto
!
sfailed:
  write sysout "'0 object file found after 2 hours"
! cleanup:
  on error then exit
  kernel logout
  dealloc line_to_target
!--------------------------- That's all ---------------------------
Start a compile qualification train

Syntax: start_qualif [list [product] [traces [options]]]

$ submit -c

* The qualification job = qualif_train.bat
* The translation jobs = to send the objects to the target
* - to receive the results from the target

Six batch queues are necessary:

- qualif*batch 1
- qualif*cebu 1
- qualif*batch 1
- qualif*batch 1
- qualif*batch 1
- qualif*batch 1

These batch queues must only have one entry each.

$ default_product = "001"
$ default_traces = "6x0"
$ default_options = "/ILOC/WU/ASM/XC/LIN=12/XK/UP=099"
$ default_cpu = "60"
$ default_debug = "n"
$ default_bind_traces = "n"
$ default_qualification = "n"

$ if not eq $1 then inq p1 "List file"
$ n1 = */p1[1,]"list"
$ if "search(\""p1\")" then goto check_p2
$ write systoutput "File \"p1\" not found" $ exit

$ check_p2:
$ if not eq $1 then inq p2 "Product name (\"default_product\")"
$ if not eq $2 then p2 = default_product
$ altissetproduct 'n2'
$ qualification = p3
$ if not eq $3 then inquire - qualification "qualifization (\"default_qualification\")"
$ if not eq $4 then qualification = default_qualification
$ if not eq $5 then inquire debug "Debug option (\"default_debug\")"
$ if not eq $6 then debug = default_debug
$ traces = p5
$ if not eq $7 then inquire traces "Compile traces (\"default_traces\")"
$ if not eq $8 then traces = default_traces
$ if not eq $9 then goto qualif
$ validation =
$ options = default_options
$ cpu_line = default_cpu
$ default_bind_traces = 445
$ inquire bind_traces "Bind traces (\"default_bind_traces\")"
$ if bind_traces = 445 then bind_traces = default_bind_traces

C-4
`%0123456789101112131415161718192021222324252627282930313233343536373839404142434445464748495051525354555657585960616263646566676869707172737475767778798081828384858687888990`
```c

/ln="temp"_listname_long=
/ln="temp"("""n"""n"""n"""n"""n"""n"""n"""n"""n"""n"""n"""n"""n"""n"""n"""n"""n"""n"""n"""n"""n"""n"""n"""n"""n"
"""n"""n"""n"""n"""n"""n"""n"""n"""n"""n"""n"""n"""n"""n"
"""n"""n"""n"""n"""n"
 QUEUE="""n"""n"""n"""n"
 QUEUE="""n"""n"""n"
 QUEUE="""n"""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
 QUEUE="""n"
Start a complete qualification train

Syntax: start-qualification [list [product [traces [options]]]]

SIU: the qualification job:
- qualifier_train.bat

SIU: the transmission job:
- to send the objects to the target
- to receive the results from the target

Six batch queues are necessary:
- for the qualification batches
- for the send batches
- for the receive batches
- for the receive results batches

These batch queues must only have one entry each.

default_product = "null2"
default_traces = "traces"
default_options = "/integr_test/ps/lin/pslin/pslin=120/pslin=99"
default_cou defaultrun = "n"
default_edi n = "n"
default_out_traces = "n"
default_qualification = "n"

if p1.len == 0 then inquire p1 "List file"
pl = fspatp(pl, "List file")

if fsearch("".","", "" then goto check_p2

write sys$output "File "pl not found"

exit

check_p2:
if p2.len == 0 then inquire p2 "Product_Name ("default_product")"
if p2.len == 0 then p2 = default_product
set product 'n2'

qualification = p2
if n3.len == 0 then inquire qualification "Qualification ("default_qualification")"
if qualification.len == 0 then qualification = default_qualification

if n4.len == 0 then inquire debug_option "Debug Option ("default_debug")"
if n5.debug.len == 0 then debug = default_debug

traces = p6
if traces.len == 0 then inquire traces "Compile Traces ("default_traces")"
if p6.len == 0 then traces = default_traces

if p3 = 0 then goto qualify

validation:

options = default_options
cou_time = default_cou

default_bind_traces = 445
inquire out_traces "Bind Traces ("default_bind_traces")"
if p7.len == 0 then bind_traces = default_bind_traces
```plaintext
$!
$ narr "om"
$!
$ qualif:
$!
$ | options = ""
$ | then inquire options "options ("default_options")"
$ | options = "default_options"
$!
$ | cpu_time = ""
$ | then inquire cpu_time "max cpu_time ("default_cpu")"
$ | cpu_time = "default_cpu"
$!
$ | inquire bind_traces "bind traces ("default_bind_traces")"
$ | bind_traces = "default_bind_traces"
$!
$ | both:
$!
$ | set mess /notac/noacv/notacv/notacv
$!
$ | if debus then goto nc
$!
$ | not debug
$!
$   send_queue = "send\$batch"
$   receive_queue = "receive\$batch"
$   qualif_queue = "qualif\$batch"
$!
$   define temp ""\$trnlim(\$temp)"
$!
$   goto begin
$!
$ | opt:
$!
$   send_queue = "send\$debug"
$   receive_queue = "receive\$debug"
$   qualif_queue = "qualif\$debug"
$!
$   define temp ""\$trnlim(\$debug)"
$!
$ | begin:
$!
$   listname = \$parse (""\$1", ",\$4,\$5")
$!
$ | set mess /mac/ev/text/i
$!
$ | submit/knee/nonprint
$   /log\$temp\$1\$listname\$log =
$ | /param("\$p2",\$3\$debug")
$ | /name="\$listname"
$ | /queue="\$send_queue"
$ | prom\$send\$debug
$!
$ !
$ | submit/knee/nonprint/notify
$ | /log\$temp\$1\$listname\$log =
$ | /param("\$p1",\$p2",\$traces",\$options", =
$ | "cpu_time",\$debug",\$p3",\$bind_traces")  =
$ | /name="\$listname"
$ | /queue="\$qualif_queue"
$ | prom\$qualif\$train\$notacv
```

C-8
Cette commande traite les tests de classe A+B+C+D+E+F+G+H+I+J+K et L.

A : Tests corrects doivent seulement être compilés, pas bindés.
B : Tests échoués, ils contiennent des erreurs, pas de bind.
C : Tests executables, doivent être compilés et bindés avec succès.
D : Tests de performance
E : comme C mais spécifique d'une implémentation
I : Tests Interactifs
K : Tests d'information
L : Tests contenant une erreur ou étant à détecter au bind.

Pour être validé le compilateur doit fournir les résultats suivants:
les tests qui lui sont soumis, de plus les erreurs détectées
devraient être collées attenues.

<table>
<thead>
<tr>
<th>Classe Compilateur</th>
<th>B</th>
<th>C</th>
<th>A</th>
<th>L</th>
<th>E</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>fail</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>1</td>
<td>fail</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>5</td>
<td>fail</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>2</td>
<td>fail</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>1</td>
<td>fail</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>1</td>
<td>fail</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>

Les tests de classes J et K n'interviennent pas directement dans le
processus de validation.

---

Class: Compilateur

Syntax: susrot qualif_train,mat -

/var=list, product_name, traces, options, max_cpu, debug or not,
qualification or validation)

Define the product and the logical names defined for the qualification:

$ used [/productization]login.com-

user1:setproduct 'p2'

$ kermit := "user1[/productization,command]$kermit.exe"

$ product_version_name = alsys_product
$ product_name = $parse(alsys_product,"name")

$ version = $parse(alsys_product,"version")

$ define product_name 'product_name'
$ define tests user1[/productization,tests]
$ edibase := susrot[/productization,commands$edit_base,exe]
$ create_base := susrot[/productization,commands$create_base,exe]

C-10
```c
no_check = 0
err_level = 1 // success value
valid_options = "/EXEC/TRACE/UTIL/10/IN/SIN"/CPU/PRINT"
valid_class = "A/CD/DE/IK/L"
set a CPU limit, traces and options.
max_cpu = "1/00000" // maximum CPU time in minutes
traces = "1/00000"
options = "1/00000"
debug_option = "1/00000"
qualification = "1/00000"
bind_traces = "1/00000"
separators
short_separ = "="
short_separator = "="
killed = "1/00000"

Open all the files
list_name = "product_domain(result)" + fspath(pl, + "NAME")

1. the list file with the complete name of each test
open/read list_file 'p1'

2. the report file that contains the result of each compilation
rpt_name = list_name + ".rpt"
open/write report file 'rpt_name'
rpt_name = fspath(rpt_name) // get full name, including version number

3. the check file that contains the names of the programs for which the
! compilation failed and a manual check is required.
check_name = list_name + ".chk"
open/write check file 'check_name'

4. the error file which contains all the compilation errors
error_name = list_name + ".err"
create 'error_name'
error_name = fspath(error_name) // get full name
open/append error file 'error_name'

Write a header for the report file and the error file
line = fspath("150< 10n + system_product, fspath())
report line
write error_file line
report ""
write error_file ""
report "Test report for", pl
write error_file "Error file for", pl
report ""
write error_file ""
report separator
write error_file separator
report ""
write error_file ""
line = fspath(
  "1332", 10'S + IAS + IAS + IAS", -
```
**FILE NAME**, "CLASS", "Compilation", "bind", "Validator"

```
report line
report ""
!
report "-- options" = "options"
report "-- class" = "class"
report "-- traces" = "traces"
report "-- debug" = "debug_option"
report "-- qualification" = "qualification"
report "-- bind_traces" = "bind_traces"
report"
```
the "urdyn" is the default one. given in the audek set command

if err_test = 1 compiler error handling
err_level = strtrim("err_level","LW"))
killed = strtrim("killed","LW")
if killed eqs. "YES" then err_level = 6
err_level = ord level: success; even number: failure.
ok = failing, -3 warning, -4 crashed, -5 immediate abort.
on error then note unexpected ! avoid loop on error goto err_test
!
compiled:
open/aperror/error=retry_for_rpt report_file 'rpt_name'
binding_result = " ---"
open error:
if not append/error=retry_for error_file 'error_name'
!
if not qualification then goto compile_result ! validation => don't
!
delete listings and don't write in error_file
!
listing = fsparse ("product_domain\results\lits",filename)
set messege /notext/noservice/nosave/nofac
define listing 'listing'
set messege /text/yes/ic/fac
!
if err_level eq 1 then goto del_out_text
!
other cases:
!
if compile errors or crashed => append the listing to the error file
!
close error_file
set noon
append 'listing' 'error_name'
set on
write error_file ""
write error_file "The above error was for ", dir_name, filename
write error_file "error_file"!
!
de1_out_text:
delete 'listing'
!
compile_result:
!
switch on the compilation result
!
result 6:
comp_result = " ABORTED"
if killed eqs. "YES" then comp_result = " ABORTED"
goto no_good!
!
result 2:
comp_result = " FAILED"

check_deviant:
  if class ~eq. "a" then goto no_good
  !
  ! class ~eq. deviant --> listing must be checked manually
  ! actually, for the validation it is possible to compare
  ! automatically with a previous set of results
  !
  ! validated = "?"
  *
  * write check_file dir_name, filename
  *
  * no_check = nb_check + 1
  goto write_result
  !

result_4:
  comp_result = "# C/SHLP"
  !
  no_good:
    validated = "NU"
    nb_invalid = nb_invalid + 1
    goto write_result
    !

  ! no error in compilation
  !
result_3:
  comp_result = " warning"
  goto check_if_deviant
  !

result_2:
  comp_result = " success"
  !
  check_if_deviant:
    if class ~eq. "A" then goto no_good
    ! compare the *lis files with HP reference
    validated := "--"
    !
    if class ~eq. "A" then goto ls_validated
    if class ~eq. "I" then goto interactif
    goto do_bin
    !

ls_validated:
  validated = "YES"
  goto write_result
  !

interactif:
  validated = "--"
  goto write_result
  !

  !
  do_bin:
    object_name = main_program + ".FN"
  goto binds_it
  !

unknown_result:
  report "-- Unknown error level : ", err_level
  comp_result = " ??????"

C-15
valiacted = """"
write_result:
line = "#fa(extract("", input_line) result, validate)"
report_line:
if (err_level = eq. 4) or (err_level = eq. 6) then goto case_re
goto loop

$class = """
$coff_result = """
$"----------------------------------
$set noon
$set mess /notext/nosv/notac/noid
$proc_name = "b" + (fex + 1, main_program) ! B for Bind
$set process/name="""" + proc_name"
$set on
$set mess /text/nosv/notac/noid
on error then goto bind_error
if bind_traces eqs. "" then -
a a b b k b i n ' m a i n _ p r o g r a m ' -
"informational/warning -
"list=result"main_program".bnd -
"out=temp"main_program".o
$!" -----------------------------------
$! bind is successful
case_attr: ! "RAC" causes new traces to be considered
if not debug_option then goto trace_end
if traces = after equal
   report "-- Traces = ", traces
   goto loop
!
trace_end:
write sysoutput "#### option trace not allowed ! not debug compiler."
goto loop
!
case_cnv:
max_cpu = after equal
report "-- max_cpu = ", max_cpu
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
If nb_total > 0 then goto close_files
per_fail = (nb_invalid + 1C) / nb_total
per_check = (nb_check + 1C) / nb_total
nb_success = nb_total - nb_invalid - nb_check
per_success = 100 - per_fail - per_check

line = fffino("!12c !4c !5m !_(!12m)"); "success":nb_success,per_success)
report line
line = fffino("!13c !4c !5m !/_(!12m)"); "failed":nb_invalid,per_fail)
report line
line = fffino("!12c !4c !5m !_(!12m)"); "to be checked":nb_check,per_check)
report line
!
close_file:
close report_file
close list_file
close error_file
close check_file
!
If nb_invalid 0 then delete 'error_name'
If nb_check 0 then delete 'check_name';0
!
the_end:
set now
!
If debug_option then exit 1 ! debug_option = y
update base 'rpt_name' 'product_name'
exit 1
!
unexpected:
report "Unexpected error"
goto loop
!
open error handler
!
open_error_handler:
!
open error 1:
write syslog "Unable to open ", pi
exit
!
open error 2:
close list_file
write syslog "Unable to open ", rpt_name
exit
!
open error 3:
close list_file
close report_file
write syslog "Unable to open ", check_name
exit
!
open error 4:
close list_file
close report_file
close check_file
write syslog "Unable to open ", error_name
exit
!

C-19
$retry_for_ro:
$ wait 0:0:30 ! wait 30 seconus
$ goto compiled
$
$retry_for_err:
$ wait 0:0:30 ! wait 30 seconus
$ goto open_err
$
$retry_again:
$ wait 0:0:30
$ goto reopen
$
$!===============================================
$! That's all folks
$!===============================================

6-Nov-198
$2$ Create a subprocess to run the compiler, limits the cpu time and the size of the log file crash.

$2$ ! $2$ syntax : compile_i source options [max_cpu]

$2$ ! The maximum cpu time must be in minutes. The default is 16 minutes.

$2$ ! If the subprocess is killed then the symbol "killed" is set to "YES", else it is set to "NO". The subprocess communicates with the parent process with the logical name "err_level" in the group table.

$2$ !

$2$ ! set nonzero

$2$ !

$2$ ! source = "" $2$

$2$ ! options = "" $2$

$2$ ! traces = "" $2$

$2$ ! debug_option = "" $2$

$2$ ! qualification = "" $2$

$2$ ! max_cpu = 60 * f#integer(pk) $2$

$2$ ! if max_cpu .eq. 0 then max_cpu = 60

$2$ !

$2$ ! priy = f#getpri( "CPU1" )

$2$ ! set mess/noid/nostr/fac/notxt $2$

$2$ ! define/jac killed "YES" $2$

$2$ ! proc_name = "" + f#extract(u13, f#sparsel(source,""""MA""E") ) ! A for Ada

$2$ ! set mess/i/sev/fac/text $2$

$2$ ! spawn/nosec/nolog/process=""proc_name"" =

$2$ ! procrun_compiler.bat =

$2$ ! source = "" $2$

$2$ ! options = "" $2$

$2$ ! traces = "" $2$

$2$ ! debug_option = "" $2$

$2$ ! qualification = "" $2$

$2$ !

$2$ ! Search the BIN of the subprocess

$2$ !

$2$ ! set mess/notext/nolog/nosec/noloc $2$

$2$ ! context = 0 $2$

$2$ !

$2$ ! loop1:

$2$ ! pid = f#pid(context)

$2$ ! name = facetjni(nid, "PPCHAW")

$2$ ! if name .eqs proc_name then goto loop2

$2$ ! if context .ne. 0 then goto loop1

$2$ !

$2$ ! write sys$comput "Sub-process not found"

$2$ ! goto crash_ok

$2$ ! Watch at the log file and at the cpu time

$2$ ! loop2:

$2$ ! cpu = f#getjni(nid, "CPU1")

$2$ ! if severity .eq. 0 then goto terminated

$2$ ! cpu = cpu / 100

$2$ ! if not. max_cpu then goto kill_it

$2$ ! does not exist any more

$2$ ! put cpu time in second

C-21
S "wait 0:0:10"
S "go to loop"
S !
S "kill_it:
S "stop/id=pid"
S "go to crash_ok"
S !
S "terminated:
S "define/job killed "no"
S !
S "crash_ok"
S "set on"
S "set mess/text/10/fac/serv"
S "exit"
$ set roam
$!$ define sys*output temporary
$!
$ source = """
$ options = """
$ debug_option = """
$ trace = """
$ qualification = """
$!
$ if debug_option then goto debug
$!
$ not debug
$! define setopt
$! set debug
$! goto compile
$!
$ debug:
$! define setopt
$! set debug
$!
$ compile:
$!
$ nrv = fiset_rv("","");
$ set mess/notext/novii/note
$ set working_set /quot=":C\:/xten=20";
$ define/job err_level """
$ set mess/text/ld/seq/fac
$!
$ define/user systere error systput
$!
$ source_name = fenames(source,"name");
$ nom_listing = "resui" + source_name + ",lis"
$ option_lis = /lis=now_listing"
$ options = p2 + option_lis
$!
$ if not qualification then goto after
$! nom_nlag = "resui" + source_name + "aia"
$ option_dag = /dag=now_nlag"
$ options = option + option_dag
$!
$ after:
$!
$ if debug_option .anu. traces .nes. "" then options = options + "/trace=(" + traces + ")"
$!
$! define compile 'source' 'options'
$!
$ err = &severity
$ set mess/notext/novii/note
$ define/job err_level ""err"
$ set mess/text/ld/seq/fac
$!
$! reassign sys*output
$! delete temporary_file.log;
shell -- not C shell -- script to link, execute an object file and to create report file

1 must be a file name without suffix.

_dir=/usr/voosalif/debug

test_dir

the object file is linked.

```bash
ld -o exe/$1.o obj/$1.o rts/libada.d.a -lc 2>ldtraces/$1.l
```

rm obj/$1.o

the exe file is executed, the results of its execution is in the same named file under the directoru

test_dir/results.
1 stands for results, .t stands for traces
the execution is performed in the directory tmp if temp files are created.

test_dir/tmp

```bash
../exe/$1.e >../results/$1.l 2>../traces/$1.t
```

is determined whether the test passed or failed by searching the string FAILED or PASSED in the result file.

test_dir

```bash
琛 -c PASSED results/$1.l >/dev/null
```

cho '$1 passed'
The execution listing is no more necessary; remove it.
NO! now keep it for validation.

rm results/$1.l

```bash
琛 -c FAILED results/$1.l >/dev/null
```

cho '$1 failed'

cho '$1 strange'

The initial object file and the exe file are removed.

```bash
exe/$1.e
```
all script to run tests as soon as they are transmitted by kermit.

e argument $1 is the name of the train; usually a date such as ser_27.
e files $1.kr and $1 ks contain the outputs of kermit (kert for safety);
e file $1 res contains one status line per test executed.

more handup
** 1

1? 'Give an identifier as parameter (train name)'

_dir=/usr/qualif/debug

test_dir/stats

d $test_dir/cbu ; time /usr/tools/kermit ruilb /dev/tty11 $600 ) \
tee $1.kr \ $test_dir/com/runthem 2>$1.err \ tee $1.res \ $test_dir/com/send_it >$1 ks ) 1

'Transfer begins. cat $1.res and $1.err to see how it goes..."
Reads messages emitted by Kermit on standard input (through the pipeline) and executes the corresponding files transferred from the VAX. The results are written by adarun on standard output.

```bash
_dirm=/usr/local/debu*

test_dir=/om

azd the Kermit version:
`/dev/null

azd the first file name:
kermit msg vax:name as name

loop, while receiving a file name, execute the previous file, already transmitted from the host machine:
```

```bash
e [ $msg = Receiving ]

exit if $name is null:
if [ ! $name ]
then echo 'Unexpected kermit message'
exit 1

wait for another file name (or for the message 'done' ?):
read kermit msg vax:name as newname
then 
# execute the previous file:
name="expr $name : '/([0-9-:]*)'" # remove the suffix:
adarun $name
name=$newname

Treat the last transmitted one:
exit if $name is null:
if [ ! $name ]
then echo 'Unexpected kermit message'
exit 1

adarun $name

$msd != 'done.' ]

cho 'Unexpected end of kermit transmission'
exi 1
```
APPENDIX D

COMPLETE LIST OF TESTS AND RESULTS

This Appendix presents a complete list of the ACVC test files used in the validation attempt, presented in order by ACVC Implementers' Guide section and objective. Each test name indicates the class of the test and which test objective in the ACVC Implementers' Guide applies to the test.

Each test has a name that identifies the section of the Ada Standard addressed by the test objective. The name of a test is interpreted according to the table below, where the first column indicates the character position in the name and the second column, the meaning of that position:

<table>
<thead>
<tr>
<th>POS</th>
<th>MEANING</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Test class: A, B, C, D, E, L.</td>
</tr>
<tr>
<td>2</td>
<td>Implementers' Guide chapter number (in hexadecimal)</td>
</tr>
<tr>
<td>3</td>
<td>Implementers' Guide section number within a chapter (in hexadecimal)</td>
</tr>
<tr>
<td>4</td>
<td>Implementers' Guide subsection number (in hexadecimal)</td>
</tr>
<tr>
<td>5-6</td>
<td>Implementers' Guide Test Objective number (in decimal)</td>
</tr>
<tr>
<td>7</td>
<td>Test sequence letter</td>
</tr>
<tr>
<td>8</td>
<td>[Optional] Compilation sequence digit or letter</td>
</tr>
<tr>
<td>9</td>
<td>[Optional] Main program designator in the case of a test having multiple compilation units.</td>
</tr>
</tbody>
</table>

Characters 8 and 9 are only present for tests that consist of several separately compiled units. A series of separately compiled units is counted as one test for reporting purposes. The eighth character indicates
the order in which the units are to be compiled, with unit 0 being compiled first. The ninth character is only present for a file containing a main program for a test comprising multiple files and is always M.

The suffix -AB means the test was written prior to release of the ANSI Standard and is also valid for the version of Ada published in July 1980.

The suffix -B means the test was written specifically for the ANSI Standard. Tests without a suffix have not yet had their names revised to -AB.

A file name ending with the extension .TST indicates that the test depends on one or more of the implementation-dependent parameters listed in Appendix B. A file name ending with .DEP indicates that the test is not necessarily applicable to all implementations because it depends upon the support of language features that a compiler may legally not implement.

The result for each file in ACVC Version 1.6 is given in the following pages, where:

- P indicates Passed.
- F indicates Failed.
- N/A indicates Not Applicable to this implementation.
- W indicates Withdrawn due to test errors.
- C indicates Compiled without error.
- A indicates Anomalous.

A test may comprise several separate compilation units contained in two or more files; the names of such files are indented under the name of the test. The letter 'M' indicates which of these files contains the main procedure.
Support Units

- CHECK_FILE-B.ADA
- REPORT_SPEC-AB.ADA
- REPORT_BODY-B.ADA
- VAR_STRINGS_SPEC.ADA
- VAR_STRINGS_BODY.ADA
- CZ1101A-AB.ADA
- CZ1102A-AB.ADA
- CZ1103A-B.ADA
- CZ1201A-AB.ADA
- CZ1201B-AB.ADA
- CZ1201C-AB.ADA
- CZ1201D-AB.ADA
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>File</th>
<th>Dep File</th>
</tr>
</thead>
<tbody>
<tr>
<td>A21001A</td>
<td>ADA P</td>
<td>B23002A</td>
<td>C24113C-B.DEP</td>
</tr>
<tr>
<td>A22002A</td>
<td>ADA P</td>
<td>B23003D-AB.TST</td>
<td>C24113D-B.DEP</td>
</tr>
<tr>
<td>A26004A</td>
<td>TST P</td>
<td>B23003E-AB.TST</td>
<td>C24113E-B.DEP</td>
</tr>
<tr>
<td>A29002A-B</td>
<td>ADA P</td>
<td>B23003F-AB.TST</td>
<td>C24113F-B.DEP</td>
</tr>
<tr>
<td>A29002B-B</td>
<td>ADA P</td>
<td>B23004A.ADA</td>
<td>C24113G-B.DEP</td>
</tr>
<tr>
<td>A29002C-B</td>
<td>ADA P</td>
<td>B23004B.ADA</td>
<td>N/A</td>
</tr>
<tr>
<td>A29002D-B</td>
<td>ADA P</td>
<td>B24001A.ADA</td>
<td>C24113H-B.DEP</td>
</tr>
<tr>
<td>A29002E-B</td>
<td>ADA P</td>
<td>B24001B.ADA</td>
<td>C24113I-B.DEP</td>
</tr>
<tr>
<td>A29002F-B</td>
<td>ADA P</td>
<td>B24001C.ADA</td>
<td>C24113J-B.DEP</td>
</tr>
<tr>
<td>A29002G-B</td>
<td>ADA P</td>
<td>B24005A.ADA</td>
<td>C24113K-B.DEP</td>
</tr>
<tr>
<td>A29002H-B</td>
<td>ADA P</td>
<td>B24005B.ADA</td>
<td>C24113L-B.DEP</td>
</tr>
<tr>
<td>A29002I-B</td>
<td>ADA P</td>
<td>B24104A.ADA</td>
<td>C24113M-B.DEP</td>
</tr>
<tr>
<td>A29002J-B</td>
<td>ADA P</td>
<td>B24104B.ADA</td>
<td>C24113N-B.DEP</td>
</tr>
<tr>
<td>B22001A-AB.TST</td>
<td></td>
<td>B24104C.ADA</td>
<td>C24113O-B.DEP</td>
</tr>
<tr>
<td>B22001B-AB.TST</td>
<td></td>
<td>B26002A.ADA</td>
<td>C24113P-B.DEP</td>
</tr>
<tr>
<td>B22001C-AB.TST</td>
<td></td>
<td>B26005A.ADA</td>
<td>C24113Q-B.DEP</td>
</tr>
<tr>
<td>B22001D-AB.TST</td>
<td></td>
<td>B29001A-AB.TST</td>
<td>C24113R-B.DEP</td>
</tr>
<tr>
<td>B22001E-AB.TST</td>
<td></td>
<td>C23001A.ADA</td>
<td>C24113S-B.DEP</td>
</tr>
<tr>
<td>B22001F-AB.TST</td>
<td></td>
<td>C23003A.TST</td>
<td>C24113T-B.DEP</td>
</tr>
<tr>
<td>B22001G-AB.TST</td>
<td></td>
<td>C24002A.ADA</td>
<td>C24113U-B.DEP</td>
</tr>
<tr>
<td>B22001H-AB.ADA</td>
<td></td>
<td>C24002B.ADA</td>
<td>C24113V-B.DEP</td>
</tr>
<tr>
<td>B22001I-AB.TST</td>
<td></td>
<td>C24002C.ADA</td>
<td>C24113W-B.DEP</td>
</tr>
<tr>
<td>B22001J-AB.TST</td>
<td></td>
<td>C24003A.TST</td>
<td>C24113X-B.DEP</td>
</tr>
<tr>
<td>B22001K-AB.TST</td>
<td></td>
<td>C24003B.TST</td>
<td>C24113Y-B.DEP</td>
</tr>
<tr>
<td>B22001L-AB.TST</td>
<td></td>
<td>C24003C.TST</td>
<td>C24113Z-B.DEP</td>
</tr>
<tr>
<td>B22001M-AB.TST</td>
<td></td>
<td>C24102A.ADA</td>
<td>C24600A-AB.ADA</td>
</tr>
<tr>
<td>B22001N-AB.TST</td>
<td></td>
<td>C24102B.ADA</td>
<td>C24600B-AB.ADA</td>
</tr>
<tr>
<td>B22003A-ADA</td>
<td></td>
<td>C24102C.ADA</td>
<td>C24700A-AB.ADA</td>
</tr>
<tr>
<td>B22004A-ADA</td>
<td></td>
<td>C24103A.ADA</td>
<td>C24700B-AB.ADA</td>
</tr>
<tr>
<td>B22004B-ADA</td>
<td></td>
<td>C24113A-B.DEP</td>
<td>E24101A-B.TST</td>
</tr>
<tr>
<td>B22004C-ADA</td>
<td></td>
<td>C24113B-B.DEP</td>
<td>N/A</td>
</tr>
<tr>
<td>File Name</td>
<td>Dep</td>
<td>File Name</td>
<td>Dep</td>
</tr>
<tr>
<td>-----------</td>
<td>-----</td>
<td>-----------</td>
<td>-----</td>
</tr>
<tr>
<td>C35706P-B</td>
<td>N/A</td>
<td>C35706K-B</td>
<td>N/A</td>
</tr>
<tr>
<td>C35706Q-B</td>
<td>N/A</td>
<td>C35706L-B</td>
<td>N/A</td>
</tr>
<tr>
<td>C35706R-B</td>
<td>N/A</td>
<td>C35706M-B</td>
<td>N/A</td>
</tr>
<tr>
<td>C35706T-B</td>
<td>N/A</td>
<td>C35706O-B</td>
<td>N/A</td>
</tr>
<tr>
<td>C35706U-B</td>
<td>N/A</td>
<td>C35706P-B</td>
<td>N/A</td>
</tr>
<tr>
<td>C35706V-B</td>
<td>N/A</td>
<td>C35706W-B</td>
<td>N/A</td>
</tr>
<tr>
<td>C35706X-B</td>
<td>N/A</td>
<td>C35706Q-B</td>
<td>N/A</td>
</tr>
<tr>
<td>C35707A-B</td>
<td>N/A</td>
<td>C35707B-B</td>
<td>N/A</td>
</tr>
<tr>
<td>C35707C-B</td>
<td>N/A</td>
<td>C35707D-B</td>
<td>N/A</td>
</tr>
<tr>
<td>C35707E-B</td>
<td>N/A</td>
<td>C35707F-B</td>
<td>N/A</td>
</tr>
<tr>
<td>C35707G-B</td>
<td>N/A</td>
<td>C35707H-B</td>
<td>N/A</td>
</tr>
<tr>
<td>C35707I-B</td>
<td>N/A</td>
<td>C35707J-B</td>
<td>N/A</td>
</tr>
<tr>
<td>C35707K-B</td>
<td>N/A</td>
<td>C35707L-B</td>
<td>N/A</td>
</tr>
<tr>
<td>C35707M-B</td>
<td>N/A</td>
<td>C35707N-B</td>
<td>N/A</td>
</tr>
<tr>
<td>C35707O-B</td>
<td>N/A</td>
<td>C35707P-B</td>
<td>N/A</td>
</tr>
<tr>
<td>C35707Q-B</td>
<td>N/A</td>
<td>C35707R-B</td>
<td>N/A</td>
</tr>
<tr>
<td>C35707S-B</td>
<td>N/A</td>
<td>C35707T-B</td>
<td>N/A</td>
</tr>
<tr>
<td>C35707U-B</td>
<td>N/A</td>
<td>C35707V-B</td>
<td>N/A</td>
</tr>
<tr>
<td>C35707W-B</td>
<td>N/A</td>
<td>C35707X-B</td>
<td>N/A</td>
</tr>
<tr>
<td>C35707Y-B</td>
<td>N/A</td>
<td>C35707Z-B</td>
<td>N/A</td>
</tr>
<tr>
<td>C35708A-B</td>
<td>N/A</td>
<td>C35708B-B</td>
<td>N/A</td>
</tr>
<tr>
<td>C35708C-B</td>
<td>N/A</td>
<td>C35708D-B</td>
<td>N/A</td>
</tr>
<tr>
<td>C35708E-B</td>
<td>N/A</td>
<td>C35708F-B</td>
<td>N/A</td>
</tr>
<tr>
<td>C35708G-B</td>
<td>N/A</td>
<td>C35708H-B</td>
<td>N/A</td>
</tr>
<tr>
<td>C35708I-B</td>
<td>N/A</td>
<td>C35708J-B</td>
<td>N/A</td>
</tr>
</tbody>
</table>
Chapter 4

AlayCOMP_001, version 1.3  6/17/86  Validation Summary Report

D-7
<p>| C45101H-AB.ADA | P | C45321H-B.DEP | N/A | C45424H-B.DEP | N/A |
| C45101I.ADA | P | C45321L-B.DEP | N/A | C45424K-B.DEP | N/A |
| C45103A-AB.ADA | P | C45321M-B.DEP | N/A | C45424L-B.DEP | N/A |
| C45103B-AB.ADA | P | C45321N-B.DEP | N/A | C45424M-B.DEP | N/A |
| C45103C-AB.ADA | P | C45321O-B.DEP | N/A | C45424N-B.DEP | N/A |
| C45104A.ADA | P | C45321P-B.DEP | N/A | C45424O-B.DEP | N/A |
| C45105A-AB.ADA | P | C45321Q-B.DEP | N/A | C45424P-B.DEP | N/A |
| C45105B-B.ADA | P | C45321R-B.DEP | N/A | C45424Q-B.DEP | N/A |
| C45106A.ADA | P | C45321S-B.DEP | N/A | C45424R-B.DEP | N/A |
| C45201A.ADA | P | C45321T-B.DEP | N/A | C45424S-B.DEP | N/A |
| C45201B.ADA | P | C45321U-B.DEP | N/A | C45424T-B.DEP | N/A |
| C45202A-AB.ADA | P | C45321V-B.DEP | N/A | C45424U-B.DEP | N/A |
| C45208A.ADA | P | C45321W-B.DEP | N/A | C45424V-B.DEP | N/A |
| C45226A.ADA | P | C45321X-B.DEP | N/A | C45424W-B.DEP | N/A |
| C453228A.ADA | P | C45321Y-B.DEP | N/A | C45424X-B.DEP | N/A |
| C453228B.ADA | P | C45321Z-B.DEP | N/A | C45424Y-B.DEP | N/A |
| C453228C.ADA | P | C45342A-AB.ADA | P | C45424Z-B.DEP | N/A |
| C453228D.ADA | P | C45343A-AB.ADA | P | C45525A-B.D.AA | P |
| C453228E-AB.ADA | P | C45343A-AB.ADA | P | C45525B-B.DEP | W |
| C45421A-AB.ADA | P | C45345A-AB.ADA | P | C45525C-B.DEP | W |
| C45421B-AB.ADA | P | C45345B-AB.ADA | P | C45525D-B.DEP | W |
| C45421C-AB.ADA | P | C45345C-AB.ADA | P | C45525E-B.DEP | W |
| C45421D-AB.ADA | P | C45345D-AB.ADA | P | C45525F-B.DEP | W |
| C45421E-AB.ADA | P | C45461A.ADA | P | C45525G-B.DEP | W |
| C45421F-AB.ADA | P | C45461B-AB.ADA | P | C45525H-B.DEP | W |
| C45421G-AB.ADA | P | C45461C-AB.ADA | P | C45525I-B.DEP | W |
| C45421H-AB.ADA | P | C45461D-AB.ADA | P | C45525J-B.DEP | W |
| C45421I-AB.ADA | P | C45461E-AB.ADA | P | C45525K-B.DEP | W |
| C45421J-AB.ADA | P | C45461F-AB.ADA | P | C45525L-B.DEP | W |
| C45421K-AB.ADA | P | C45461G-AB.ADA | P | C45525M-B.DEP | W |
| C45421L-AB.ADA | P | C45461H-AB.ADA | P | C45525N-B.DEP | W |
| C45421O-AB.ADA | P | C45461J-AB.ADA | P | C45525P-B.DEP | W |
| C45421Q-AB.ADA | P | C45462A-AB.ADA | P | C45525Q-B.DEP | W |
| C45421R-AB.ADA | P | C45462B-AB.ADA | P | C45525R-B.DEP | W |
| C45421S-AB.ADA | P | C45462C-AB.ADA | P | C45525S-B.DEP | W |
| C45421T-AB.ADA | P | C45462D-AB.ADA | P | C45525T-B.DEP | W |
| C45421U-AB.ADA | P | C45462E-AB.ADA | P | C45525U-B.DEP | W |
| C45421V-AB.ADA | P | C45462F-AB.ADA | P | C45525V-B.DEP | W |
| C45421W-AB.ADA | P | C45462G-AB.ADA | P | C45525W-B.DEP | W |
| C45421X-AB.ADA | P | C45462H-AB.ADA | P | C45525X-B.DEP | W |
| C45421Y-AB.ADA | P | C45462I-AB.ADA | P | C45525Y-B.DEP | W |
| C45421Z-AB.ADA | P | C45462J-AB.ADA | P | C45525Z-B.DEP | W |
| C45424A-AB.ADA | P | C45462K-AB.ADA | P | C45525A-DEP | N/A |
| C45424B-AB.ADA | P | C45462L-AB.ADA | P | C45525B-DEP | N/A |
| C45424C-AB.ADA | P | C45462M-AB.ADA | P | C45525C-DEP | N/A |
| C45424D-AB.ADA | P | C45462N-AB.ADA | P | C45525D-DEP | N/A |
| C45424E-AB.ADA | P | C45462P-AB.ADA | P | C45525E-DEP | N/A |
| C45424F-AB.ADA | P | C45462Q-AB.ADA | P | C45525F-DEP | N/A |
| C45424G-AB.ADA | P | C45462R-AB.ADA | P | C45525G-DEP | N/A |
| C45424H-AB.ADA | P | C45462S-AB.ADA | P | C45525H-DEP | N/A |
| C45424I-AB.ADA | P | C45462T-AB.ADA | P | C45525I-DEP | N/A |
| C45424J-AB.ADA | P | C45462U-AB.ADA | P | C45525J-DEP | N/A |
| C45424K-AB.ADA | P | C45462V-AB.ADA | P | C45525K-DEP | N/A |
| C45424L-AB.ADA | P | C45462W-AB.ADA | P | C45525L-DEP | N/A |
| C45424M-AB.ADA | P | C45462X-AB.ADA | P | C45525M-DEP | N/A |
| C45424N-AB.ADA | P | C45462Y-AB.ADA | P | C45525N-DEP | N/A |
| C45424O-AB.ADA | P | C45462Z-AB.ADA | P | C45525O-DEP | N/A |</p>
<table>
<thead>
<tr>
<th>File</th>
<th>Status</th>
<th>Checksum</th>
<th>File</th>
<th>Status</th>
<th>Checksum</th>
</tr>
</thead>
<tbody>
<tr>
<td>C45621P.DEP</td>
<td>N/A</td>
<td>C480058-B.ADA</td>
<td>P</td>
<td>C48009H-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>C45621Q.DEP</td>
<td>N/A</td>
<td>C48005C-B.ADA</td>
<td>W</td>
<td>C480091-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>C45621R.DEP</td>
<td>N/A</td>
<td>C48006A-B.ADA</td>
<td>P</td>
<td>C48009J-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>C45621S.DEP</td>
<td>N/A</td>
<td>C48006F-B.ADA</td>
<td>W</td>
<td>C48010A-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>C45621T.DEP</td>
<td>N/A</td>
<td>C48007A-B.ADA</td>
<td>P</td>
<td>C48012A-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>C45621U.DEP</td>
<td>N/A</td>
<td>C48007B-B.ADA</td>
<td>P</td>
<td>C4A001A.ADA</td>
<td>P</td>
</tr>
<tr>
<td>C45621V.DEP</td>
<td>N/A</td>
<td>C48007C-B.ADA</td>
<td>P</td>
<td>C4A003A.ADA</td>
<td>P</td>
</tr>
<tr>
<td>C45621W.DEP</td>
<td>N/A</td>
<td>C48008A-B.ADA</td>
<td>P</td>
<td>C4A011A.ADA</td>
<td>P</td>
</tr>
<tr>
<td>C45621X.DEP</td>
<td>N/A</td>
<td>C48008B-B.ADA</td>
<td>P</td>
<td>C4A018A-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>C45621Y.DEP</td>
<td>N/A</td>
<td>C48008C-B.ADA</td>
<td>P</td>
<td>C4A013A.ADA</td>
<td>P</td>
</tr>
<tr>
<td>C45621Z.DEP</td>
<td>N/A</td>
<td>C48008D-B.ADA</td>
<td>P</td>
<td>D4A002A-AB.ADA</td>
<td>P</td>
</tr>
<tr>
<td>C48004A-B.ADA</td>
<td>P</td>
<td>C48009A-B.ADA</td>
<td>P</td>
<td>D4A002B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>C48004B-B.ADA</td>
<td>P</td>
<td>C48009B-B.ADA</td>
<td>P</td>
<td>D4A004A-AB.ADA</td>
<td>P</td>
</tr>
<tr>
<td>C48004C-B.ADA</td>
<td>P</td>
<td>C48009C-B.ADA</td>
<td>P</td>
<td>D4A004B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>C48004D-B.ADA</td>
<td>P</td>
<td>C48009D-B.ADA</td>
<td>P</td>
<td>E43211B-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>C48004E-B.ADA</td>
<td>P</td>
<td>C48009E-B.ADA</td>
<td>P</td>
<td>E43212B-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>C48004F-B.ADA</td>
<td>P</td>
<td>C48009F-B.ADA</td>
<td>P</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C48005A-B.ADA</td>
<td>P</td>
<td>C48009G-B.ADA</td>
<td>P</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Chapter 5

| A54001A-AB.ADA | B54A27D-AB.ADA | B50002B-AB.ADA | P |
| A54002A-AB.ADA | B54A01B-AB.TST | B50002C-AB.ADA | P |
| A55b12A-AB.ADA | B54A01C-AB.ADA | B50003A-AB.ADA | P |
| A56b13A-AB.ADA | B54A02B-AB.ADA | B50003B-AB.ADA | P |
| A58b14A-AB.ADA | B54A02C-AB.ADA | B50001A-AB.ADA | P |
| B51001A-AB.ADA | B54A02D-AB.ADA | B50001C-AB.ADA | P |
| B51903A-AB.ADA | B54A04A-AB.ADA | B50001D-AB.ADA | P |
| B51904A-AB.ADA | B54A04B-AB.ADA | B50001E-AB.ADA | P |
| B51904B-AB.ADA | B54A04C-AB.ADA | B50001F-AB.ADA | P |
| B52902A-AB.ADA | B55A01A-AB.ADA | B50001G-AB.ADA | P |
| B52902B-AB.ADA | B55A01B-AB.ADA | B50001H-AB.ADA | P |
| B52902C-AB.ADA | B55A01C-AB.ADA | B50001I-AB.ADA | P |
| B52902D-AB.ADA | B55A01D-AB.ADA | C51002A-AB.ADA | P |
| B52902E-AB.ADA | B55A01E-AB.ADA | C51004A-AB.ADA | P |
| B52902F-AB.ADA | B55A01F-AB.ADA | C52001A-AB.ADA | P |
| B52902G-AB.ADA | B55A01G-AB.ADA | C52001B-AB.ADA | P |
| B52902H-AB.ADA | B55A01H-AB.ADA | C52001C-AB.ADA | P |
| B52902I-AB.ADA | B55A01I-AB.ADA | C52005A-AB.ADA | P |
| B52903C-AB.ADA | B55A01J-AB.ADA | C52005B-AB.ADA | P |
| B52904A-AB.ADA | B55A01K-AB.ADA | C52005C-AB.ADA | P |
| B52904B-AB.ADA | B55A01L-AB.ADA | C52005D-AB.ADA | P |
| B52904C-AB.ADA | B55A01M-AB.ADA | C52005E-AB.ADA | P |
| B52904D-AB.DEP | B55A01N-AB.ADA | C52006F-AB.ADA | P |
| B52904E-AB.DEP | B55A01O-AB.ADA | C52007A-AB.ADA | P |
| B52906A-AB.ADA | B55A01P-AB.ADA | C52008A-AB.ADA | P |
| B53001A-AB.ADA | B55A01Q-AB.ADA | C52008B-AB.ADA | P |
| B53001B-AB.ADA | B55A01R-AB.ADA | C52009A-AB.ADA | P |
| B53002A-AB.ADA | B55A01S-AB.ADA | C52009B-AB.ADA | P |
| B53002B-AB.ADA | B55A01T-AB.ADA | C52018A-AB.ADA | P |
| B53002C-AB.ADA | B55A01U-AB.ADA | C52011A-AB.ADA | P |
| B53004A-AB.ADA | B55A01V-AB.ADA | C52011B-AB.ADA | P |
| B53009A-AB.ADA | B55A01W-AB.ADA | C52012A-AB.ADA | P |
| B53009B-AB.ADA | B55A01X-AB.ADA | C52012B-AB.ADA | P |
| B53009C-AB.ADA | B55A01Y-AB.ADA | C52013A-AB.ADA | P |
| B54A01A-AB.ADA | B55A09C-AB.DEP | C52181A-AB.ADA | P |
| B54A01B-AB.ADA | B55A09D-AB.DEP | C52182A-AB.ADA | P |
| B54A01C-AB.ADA | B55B12B-AB.ADA | C52182B-AB.ADA | P |
| B54A01D-AB.ADA | B55B12C-AB.ADA | C52182C-AB.ADA | P |
| B54A01E-AB.ADA | B55B14B-AB.ADA | C52182D-AB.ADA | P |
| B54A01F-AB.ADA | B55B18A-AB.ADA | C52183A-AB.ADA | P |
| B54A01G-AB.ADA | B55E01A-AB.ADA | C52183B-AB.ADA | P |
| B54A01H-AB.ADA | B55E01C-AB.ADA | C52183C-AB.ADA | P |
| B54A01I-AB.ADA | B55E01D-AB.ADA | C52183F-AB.ADA | P |
| B54A01J-AB.ADA | B55E01E-AB.ADA | C52183G-AB.ADA | P |
| B54A01K-AB.ADA | B55E01F-AB.ADA | C52183H-AB.ADA | P |
| B54A01L-AB.ADA | B55E01G-AB.ADA | C5183K-AB.ADA | P |
| B54A05A-AB.ADA | B55E01H-AB.ADA | C52183L-AB.ADA | P |
| B54A05B-AB.ADA | B57001A-AB.ADA | C52183M-AB.ADA | P |
| B54A06A-AB.ADA | B57001B-AB.ADA | C52183P-AB.ADA | P |
| B54A06B-AB.ADA | B57001C-AB.ADA | C52183Q-AB.ADA | P |
| B54A02A-AB.ADA | B57001D-AB.ADA | C52183R-AB.ADA | P |
| B54A21A-AB.ADA | B58001A-AB.ADA | C52183S-AB.ADA | P |
| B54A2SA-AB.ADA | B58002A-AB.ADA | C52183X-AB.ADA | P |
C52104A-AB.ADA P C54A27A-AB.ADA P C57004B-AB.ADA P
C52104B-AB.ADA P C54A41A.ADA P C57004C-AB.ADA P
C52104C-AB.ADA P C54A42A.ADA P C57005A-B.ADA P
C52104F-AB.ADA P C54A42B.ADA P C58004A-AB.ADA P
C52104G-AB.ADA P C54A42C.ADA P C58004B-AB.ADA P
C52104H-AB.ADA P C54A42D.ADA P C58004C-AB.ADA P
C52104K-AB.ADA P C54A42E.ADA P C58004D-B.ADA P
C52104L-AB.ADA P C54A42F.ADA P C58004E-AB.ADA P
C52104M-AB.ADA P C54A42G.ADA P C58004F-AB.ADA P
C52104P-AB.ADA P C55B03A-AB.ADA P C58005A-AB.ADA P
C52104Q-AB.ADA P C55B04A-AB.ADA P C58005B-AB.ADA P
C52104R-AB.ADA P C55B05A-AB.ADA P C58005E-AB.ADA P
C52104X-B.ADA N/A C55B06A-AB.ADA P C58006A-AB.ADA P
C52104Y-B.ADA N/A C55B06B-AB.ADA P C58006G-AB.ADA P
C53004B-B.ADA P C55B07A-AB.DEP P C59001A-AB.ADA P
C53005A-AB.ADA P C55B07B-AB.DEP P C59002A-AB.ADA P
C53005B-AB.ADA P C55B08A-B.ADA P C59002B-AB.ADA P
C53006A-AB.ADA P C55B09A-AB.ADA P C59002C-AB.ADA P
C53006B-AB.ADA P C55B15A-AB.ADA P D55A83A-AB.ADA P
C53007A-AB.ADA P C55B16A-AB.DEP N/A D55A83B-AB.ADA P
C53008A-AB.ADA P C55C81A-B.ADA P D55A83C-AB.ADA P
C54A83A.ADA P C55C82A-AB.ADA P D55A83D-AB.ADA P
C54A84A-AB.ADA P C55C82B-AB.ADA P D55A83E-AB.ADA P
C54A85A-AB.ADA P C55C83A-AB.ADA P D55A83F-AB.ADA P
C54A87A-AB.ADA P C55C83B-AB.ADA P D55A83G-AB.ADA P
C54A22A-AB.ADA P C55D01A-AB.ADA P D55A83H-AB.ADA P
C54A23A-B.ADA P C56D01A-AB.ADA P D56001G-AB.ADA P
C54A24A-AB.ADA P C57002A-AB.ADA P E52103Y-AB.ADA P
C54A24B.ADA P C57003A-AB.ADA P
C54A26A.ADA P C57004A-AB.ADA P
<table>
<thead>
<tr>
<th>File</th>
<th>Mode</th>
<th>File</th>
<th>Mode</th>
<th>File</th>
<th>Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>D64005EA</td>
<td>C</td>
<td>D64005FF</td>
<td>C</td>
<td>D64005G</td>
<td>C</td>
</tr>
<tr>
<td>D64005EB</td>
<td>C</td>
<td>D64005FG</td>
<td>C</td>
<td>D64005H</td>
<td>C</td>
</tr>
<tr>
<td>D64005EC</td>
<td>C</td>
<td>D64005FH</td>
<td>C</td>
<td>D64005I</td>
<td>C</td>
</tr>
<tr>
<td>D64005ED</td>
<td>C</td>
<td>D64005FJ</td>
<td>C</td>
<td>D64005J</td>
<td>C</td>
</tr>
<tr>
<td>D64005EE</td>
<td>C</td>
<td>D64005K</td>
<td>C</td>
<td>D64005L</td>
<td>C</td>
</tr>
<tr>
<td>D64005EF</td>
<td>C</td>
<td>D64005G</td>
<td>P</td>
<td>D64005GM</td>
<td>C</td>
</tr>
<tr>
<td>D64005F-B.ADA</td>
<td>P</td>
<td>D64005G</td>
<td>C</td>
<td>D64005GM</td>
<td>C</td>
</tr>
<tr>
<td>D64005FM</td>
<td>C</td>
<td>D64005G</td>
<td>C</td>
<td>D64005M</td>
<td>C</td>
</tr>
<tr>
<td>D641005FA</td>
<td>C</td>
<td>D64005GB</td>
<td>C</td>
<td>D64005C</td>
<td>C</td>
</tr>
<tr>
<td>D64005FB</td>
<td>C</td>
<td>D64005GC</td>
<td>C</td>
<td>D64005D</td>
<td>C</td>
</tr>
<tr>
<td>D64005FC</td>
<td>C</td>
<td>D64005GD</td>
<td>C</td>
<td>D64005E</td>
<td>C</td>
</tr>
<tr>
<td>D64005FD</td>
<td>C</td>
<td>D64005GE</td>
<td>C</td>
<td>D64005F</td>
<td>C</td>
</tr>
<tr>
<td>D64005FE</td>
<td>C</td>
<td>D64005GF</td>
<td>C</td>
<td>D64005G</td>
<td>C</td>
</tr>
</tbody>
</table>
## Validation Summary Report

### Chapter 8

<table>
<thead>
<tr>
<th>File Name</th>
<th>Path 1</th>
<th>Path 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>AB3482A.ADA</td>
<td>B8660180-B.ADA</td>
<td>C87405A-B.ADA</td>
</tr>
<tr>
<td>AB3482B.ADA</td>
<td>B8660190-B.ADA</td>
<td>C87405B-B.ADA</td>
</tr>
<tr>
<td>AB3486A-B.ADA</td>
<td>B866019V-B.ADA</td>
<td>C87802A-B.ADA</td>
</tr>
<tr>
<td>AB3501C.ADA</td>
<td>B86601BI-B.ADA</td>
<td>C87902B-B.ADA</td>
</tr>
<tr>
<td>AB3501D.ADA</td>
<td>B86601XK-B.ADA</td>
<td>C87803A-B.ADA</td>
</tr>
<tr>
<td>AB3501E.ADA</td>
<td>B866010AW-AB.DEP</td>
<td>C87804A-B.ADA</td>
</tr>
<tr>
<td>AB3501F.ADA</td>
<td>B86601CP-AB.DEP</td>
<td>N/A</td>
</tr>
<tr>
<td>AB3501G.ADA</td>
<td>B86601CQ-AB.DEP</td>
<td>N/A</td>
</tr>
<tr>
<td>AB3501H.ADA</td>
<td>B86601CR-AB.DEP</td>
<td>C87805A-B.ADA</td>
</tr>
<tr>
<td>AB35041A-A.B.ADA</td>
<td>C83802A.ADA</td>
<td>C87902B-B.ADA</td>
</tr>
<tr>
<td>AB35082A.B-ADA</td>
<td>C83802B.ADA</td>
<td>C87903A-B.ADA</td>
</tr>
<tr>
<td>BB3489B-A.B.ADA</td>
<td>B866011B-ADA</td>
<td>C87804A-B.ADA</td>
</tr>
<tr>
<td>BB3489B-A.B.ADA</td>
<td>B866011V-ADA</td>
<td>C87805A-B.ADA</td>
</tr>
<tr>
<td>BB3490A-B.ADA</td>
<td>C83802A-ADA</td>
<td>C87902B-B.ADA</td>
</tr>
<tr>
<td>BB3501A-B.ADA</td>
<td>C83802B-ADA</td>
<td>C87903A-B.ADA</td>
</tr>
<tr>
<td>BB3501B.B.ADA</td>
<td>C83802A.B-ADA</td>
<td>C87903A-B.ADA</td>
</tr>
<tr>
<td>BB3501C.B.ADA</td>
<td>C83802B.B-ADA</td>
<td>C87903A-B.ADA</td>
</tr>
<tr>
<td>BB3501B.B.ADA</td>
<td>C83802A-ADA</td>
<td>C87903A-B.ADA</td>
</tr>
<tr>
<td>BB3501C.B.ADA</td>
<td>C83802B-ADA</td>
<td>C87903A-B.ADA</td>
</tr>
</tbody>
</table>

---

D-15
<table>
<thead>
<tr>
<th>File Number</th>
<th>Description</th>
<th>Status</th>
<th>File Number</th>
<th>Description</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>C87B370-B</td>
<td>ADA</td>
<td>P</td>
<td>C87B42A-B</td>
<td>ADA</td>
<td>P</td>
</tr>
<tr>
<td>C87B37E-B</td>
<td>ADA</td>
<td>P</td>
<td>C87B43A-B</td>
<td>ADA</td>
<td>P</td>
</tr>
<tr>
<td>C87B37F-B</td>
<td>ADA</td>
<td>P</td>
<td>C87B44A-B</td>
<td>ADA</td>
<td>P</td>
</tr>
<tr>
<td>C87B38A-B</td>
<td>ADA</td>
<td>P</td>
<td>C87B45A-B</td>
<td>ADA</td>
<td>P</td>
</tr>
<tr>
<td>C87B39A-B</td>
<td>ADA</td>
<td>P</td>
<td>C87B45C-B</td>
<td>ADA</td>
<td>P</td>
</tr>
<tr>
<td>C87B48A-B</td>
<td>ADA</td>
<td>P</td>
<td>C87B47A-B</td>
<td>ADA</td>
<td>P</td>
</tr>
<tr>
<td>C87B49A-B</td>
<td>ADA</td>
<td>P</td>
<td>C87B48B-B</td>
<td>ADA</td>
<td>P</td>
</tr>
</tbody>
</table>

DEP N/A
<table>
<thead>
<tr>
<th>Chapter 9</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS 1002-S. ADA</td>
</tr>
<tr>
<td>091002M-B. ADA</td>
</tr>
<tr>
<td>AS 1003-A. ADA</td>
</tr>
<tr>
<td>091003-A. ADA</td>
</tr>
<tr>
<td>091004-A. ADA</td>
</tr>
<tr>
<td>091005-A. ADA</td>
</tr>
<tr>
<td>091006-A. ADA</td>
</tr>
<tr>
<td>091007-A. ADA</td>
</tr>
<tr>
<td>091008-A. ADA</td>
</tr>
<tr>
<td>091009-A. ADA</td>
</tr>
<tr>
<td>091010-A. ADA</td>
</tr>
<tr>
<td>091011-A. ADA</td>
</tr>
<tr>
<td>091012-A. ADA</td>
</tr>
<tr>
<td>091013-A. ADA</td>
</tr>
<tr>
<td>091014-A. ADA</td>
</tr>
<tr>
<td>091015-A. ADA</td>
</tr>
<tr>
<td>091016-A. ADA</td>
</tr>
<tr>
<td>091017-A. ADA</td>
</tr>
<tr>
<td>091018-A. ADA</td>
</tr>
<tr>
<td>091019-A. ADA</td>
</tr>
<tr>
<td>091020-A. ADA</td>
</tr>
<tr>
<td>091021-A. ADA</td>
</tr>
<tr>
<td>091022-A. ADA</td>
</tr>
<tr>
<td>091023-A. ADA</td>
</tr>
<tr>
<td>091024-A. ADA</td>
</tr>
<tr>
<td>091025-A. ADA</td>
</tr>
<tr>
<td>091026-A. ADA</td>
</tr>
</tbody>
</table>

D-17
<table>
<thead>
<tr>
<th>File Name</th>
<th>Status</th>
<th>File Name</th>
<th>Status</th>
<th>File Name</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>C95681A-B.ADA</td>
<td>P</td>
<td>C96687A-B.ADA</td>
<td>P</td>
<td>C97303A-AB.ADA</td>
<td>P</td>
</tr>
<tr>
<td>C956CA-B.ADA</td>
<td>P</td>
<td>C96688A-B.ADA</td>
<td>P</td>
<td>C97303P-AB.ADA</td>
<td>P</td>
</tr>
<tr>
<td>C956B1B-B.ADA</td>
<td>P</td>
<td>C96688B-B.ADA</td>
<td>P</td>
<td>C97304A-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>C956B1C-B.ADA</td>
<td>P</td>
<td>C97113A-B.ADA</td>
<td>P</td>
<td>C9A093A-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>C956B1D-B.ADA</td>
<td>P</td>
<td>C97114A-B.ADA</td>
<td>P</td>
<td>C9A094A-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>C956B1E-B.ADA</td>
<td>P</td>
<td>C97115A-B.ADA</td>
<td>P</td>
<td>C9A095A-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>C956B1F-B.ADA</td>
<td>P</td>
<td>C97281A-AB.ADA</td>
<td>P</td>
<td>C9A096A-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>C956B1G-B.ADA</td>
<td>P</td>
<td>C97281D-AB.ADA</td>
<td>P</td>
<td>C9A097A-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>C96681A-B.ADA</td>
<td>P</td>
<td>C97281E-AB.ADA</td>
<td>P</td>
<td>C9A098A-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>C96682A-B.ADA</td>
<td>P</td>
<td>C97281G-AB.ADA</td>
<td>P</td>
<td>C9A099A-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>C96683A-B.ADA</td>
<td>P</td>
<td>C97281H-AB.ADA</td>
<td>P</td>
<td>C9A099B-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>C96684B-TST</td>
<td>A</td>
<td>C97281X-AB.ADA</td>
<td>A</td>
<td>C9A099C-B.ADA</td>
<td>A</td>
</tr>
<tr>
<td>C96685A-TST</td>
<td>A</td>
<td>C97282A-AB.ADA</td>
<td>A</td>
<td>C9A099D-B.ADA</td>
<td>A</td>
</tr>
<tr>
<td>C96685B-TST</td>
<td>A</td>
<td>C972823A-AB.ADA</td>
<td>A</td>
<td>C9A099E-B.ADA</td>
<td>A</td>
</tr>
<tr>
<td>C96686A-B.ADA</td>
<td>P</td>
<td>C97283E-AB.ADA</td>
<td>P</td>
<td>C9A099F-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>C96686A-B.ADA</td>
<td>P</td>
<td>C97284A-B.ADA</td>
<td>P</td>
<td>C9A099H-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>BA0613A2</td>
<td>C</td>
<td>CA1013A5</td>
<td>C</td>
<td>CA2008A-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>BA0613A3</td>
<td>C</td>
<td>CA1013A6M</td>
<td>C</td>
<td>CA2009A8M</td>
<td>C</td>
</tr>
<tr>
<td>BA0613A4</td>
<td>C</td>
<td>CA1014A-ABADA</td>
<td>P</td>
<td>CA2008A1</td>
<td>C</td>
</tr>
<tr>
<td>BA0613A5</td>
<td>C</td>
<td>CA1014A8M</td>
<td>C</td>
<td>CA2008A2</td>
<td>C</td>
</tr>
<tr>
<td>BA0613A6</td>
<td>C</td>
<td>CA1014A1</td>
<td>C</td>
<td>CA2009A-B.DEP</td>
<td>P</td>
</tr>
<tr>
<td>BA0613A7M</td>
<td>C</td>
<td>CA1014A2</td>
<td>C</td>
<td>CA2009B-B.DEP</td>
<td>W</td>
</tr>
<tr>
<td>CA1002A-B.ADA</td>
<td>P</td>
<td>CA1014A3</td>
<td>C</td>
<td>CA2009C-B.DEP</td>
<td>N/A</td>
</tr>
<tr>
<td>CA1002A</td>
<td>C</td>
<td>CA1022A-B.ADA</td>
<td>P</td>
<td>CA2009C8M</td>
<td>C</td>
</tr>
<tr>
<td>CA1002A1</td>
<td>C</td>
<td>CA1022A8</td>
<td>C</td>
<td>CA2009C1</td>
<td>C</td>
</tr>
<tr>
<td>CA1002A2</td>
<td>C</td>
<td>CA1022A1</td>
<td>C</td>
<td>CA2009D0</td>
<td>C</td>
</tr>
<tr>
<td>CA1002A3M</td>
<td>C</td>
<td>CA1022A2</td>
<td>C</td>
<td>CA2009D-DEP</td>
<td>W</td>
</tr>
<tr>
<td>CA1002A5</td>
<td>C</td>
<td>CA1022A4</td>
<td>C</td>
<td>CA2009F8M</td>
<td>C</td>
</tr>
<tr>
<td>CA1002A6</td>
<td>C</td>
<td>CA1022A5</td>
<td>C</td>
<td>CA2009F1</td>
<td>C</td>
</tr>
<tr>
<td>CA1002A7</td>
<td>C</td>
<td>CA1022A6M</td>
<td>C</td>
<td>CA2009A-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>CA1002A8</td>
<td>C</td>
<td>CA1022A-B.ADA</td>
<td>P</td>
<td>CA2009A0</td>
<td>C</td>
</tr>
<tr>
<td>CA1002A9</td>
<td>C</td>
<td>CA1022A8</td>
<td>C</td>
<td>CA2009A1</td>
<td>C</td>
</tr>
<tr>
<td>CA1002A10</td>
<td>C</td>
<td>CA1022A1</td>
<td>C</td>
<td>CA2009A2M</td>
<td>C</td>
</tr>
<tr>
<td>CA1002A11</td>
<td>C</td>
<td>CA1022A2M</td>
<td>C</td>
<td>CA2009A3</td>
<td>C</td>
</tr>
<tr>
<td>CA1002A12</td>
<td>C</td>
<td>CA1022A8M</td>
<td>C</td>
<td>CA2009A4M</td>
<td>C</td>
</tr>
<tr>
<td>CA1002A13</td>
<td>C</td>
<td>CA1022A1</td>
<td>C</td>
<td>CA2009C0</td>
<td>C</td>
</tr>
<tr>
<td>CA1002A14</td>
<td>C</td>
<td>CA1022A2</td>
<td>C</td>
<td>CA2009C1</td>
<td>C</td>
</tr>
<tr>
<td>CA1002A15</td>
<td>C</td>
<td>CA1022A3</td>
<td>C</td>
<td>CA2009C2</td>
<td>C</td>
</tr>
<tr>
<td>CA1002A16</td>
<td>C</td>
<td>CA1022A4</td>
<td>C</td>
<td>CA2009C3</td>
<td>C</td>
</tr>
<tr>
<td>CA1002A17</td>
<td>C</td>
<td>CA1022A5</td>
<td>C</td>
<td>CA2009C4</td>
<td>C</td>
</tr>
<tr>
<td>CA1002A18</td>
<td>C</td>
<td>CA1022A6M</td>
<td>C</td>
<td>CA2009C5M</td>
<td>C</td>
</tr>
<tr>
<td>CA1002A19</td>
<td>C</td>
<td>CA1022A-B.ADA</td>
<td>W</td>
<td>CA2009D0</td>
<td>C</td>
</tr>
<tr>
<td>CA1103A-B.ADA</td>
<td>P</td>
<td>CA1103A1</td>
<td>C</td>
<td>CA2009D1</td>
<td>C</td>
</tr>
<tr>
<td>CA1103B-AB.ADA</td>
<td>W</td>
<td>CA1103A2M</td>
<td>C</td>
<td>CA2009D2</td>
<td>C</td>
</tr>
<tr>
<td>BA0644-A-AB.ADA</td>
<td>P</td>
<td>CA1103A8M</td>
<td>C</td>
<td>CA2009D3</td>
<td>C</td>
</tr>
<tr>
<td>BA0655-A-AB.ADA</td>
<td>P</td>
<td>CA1103A1</td>
<td>C</td>
<td>CA2009D4</td>
<td>C</td>
</tr>
<tr>
<td>BA0656-A-AB.ADA</td>
<td>P</td>
<td>CA1103A2</td>
<td>C</td>
<td>CA2009D5</td>
<td>C</td>
</tr>
<tr>
<td>BA0657-A-AB.ADA</td>
<td>P</td>
<td>CA1103A3</td>
<td>C</td>
<td>CA2009D6</td>
<td>C</td>
</tr>
<tr>
<td>CA1007A8B</td>
<td>C</td>
<td>CA1103A4</td>
<td>C</td>
<td>CA2009D7</td>
<td>C</td>
</tr>
<tr>
<td>CA1007A1M</td>
<td>C</td>
<td>CA1103A5</td>
<td>C</td>
<td>CA2009D8</td>
<td>C</td>
</tr>
<tr>
<td>CA1007A2</td>
<td>C</td>
<td>CA1103A6</td>
<td>C</td>
<td>CA2009D9</td>
<td>C</td>
</tr>
<tr>
<td>CA1007A3</td>
<td>C</td>
<td>CA1103A7</td>
<td>C</td>
<td>CA2009D0</td>
<td>C</td>
</tr>
<tr>
<td>CA1007A4</td>
<td>C</td>
<td>CA1103A8</td>
<td>C</td>
<td>CA2009D1</td>
<td>C</td>
</tr>
<tr>
<td>CA1007A5</td>
<td>C</td>
<td>CA1103A9</td>
<td>C</td>
<td>CA2009D2</td>
<td>C</td>
</tr>
<tr>
<td>CA1007A6</td>
<td>C</td>
<td>CA1103A10</td>
<td>C</td>
<td>CA2009D3</td>
<td>C</td>
</tr>
<tr>
<td>CA1007A7</td>
<td>C</td>
<td>CA1103A11</td>
<td>C</td>
<td>CA2009D4</td>
<td>C</td>
</tr>
<tr>
<td>CA1007A8</td>
<td>C</td>
<td>CA1103A12</td>
<td>C</td>
<td>CA2009D5</td>
<td>C</td>
</tr>
<tr>
<td>CA1007A9</td>
<td>C</td>
<td>CA1103A13</td>
<td>C</td>
<td>CA2009D6</td>
<td>C</td>
</tr>
<tr>
<td>CA1007A0</td>
<td>C</td>
<td>CA1103A14</td>
<td>C</td>
<td>CA2009D7</td>
<td>C</td>
</tr>
<tr>
<td>CA1007A1</td>
<td>C</td>
<td>CA1103A15</td>
<td>C</td>
<td>CA2009D8</td>
<td>C</td>
</tr>
<tr>
<td>CA1007A2</td>
<td>C</td>
<td>CA1103A16</td>
<td>C</td>
<td>CA2009D9</td>
<td>C</td>
</tr>
<tr>
<td>BA1111A-ADADA</td>
<td>W</td>
<td>CA1103A17</td>
<td>C</td>
<td>CA2009D0</td>
<td>C</td>
</tr>
<tr>
<td>BA1111A8W</td>
<td>W</td>
<td>CA1103A18</td>
<td>C</td>
<td>CA2009D1</td>
<td>C</td>
</tr>
<tr>
<td>BA1111A1W</td>
<td>W</td>
<td>CA1103A19</td>
<td>C</td>
<td>CA2009D2</td>
<td>C</td>
</tr>
<tr>
<td>BA1111A2W</td>
<td>W</td>
<td>CA1103A20</td>
<td>C</td>
<td>CA2009D3</td>
<td>C</td>
</tr>
<tr>
<td>BA1111A3W</td>
<td>W</td>
<td>CA1103A21</td>
<td>C</td>
<td>CA2009D4</td>
<td>C</td>
</tr>
<tr>
<td>BA1111A4W</td>
<td>W</td>
<td>CA1103A22</td>
<td>C</td>
<td>CA2009D5</td>
<td>C</td>
</tr>
<tr>
<td>BA1111A5W</td>
<td>W</td>
<td>CA1103A23</td>
<td>C</td>
<td>CA2009D6</td>
<td>C</td>
</tr>
<tr>
<td>BA1111A6W</td>
<td>W</td>
<td>CA1103A24</td>
<td>C</td>
<td>CA2009D7</td>
<td>C</td>
</tr>
<tr>
<td>BA1112A-B.DEP</td>
<td>P</td>
<td>CA1103A25</td>
<td>C</td>
<td>CA2009D8</td>
<td>C</td>
</tr>
<tr>
<td>BA1112A-B.DEP</td>
<td>P</td>
<td>CA1103A26</td>
<td>C</td>
<td>CA2009D9</td>
<td>C</td>
</tr>
<tr>
<td>BA1112A1</td>
<td>C</td>
<td>CA1103A27</td>
<td>C</td>
<td>CA2009C0</td>
<td>C</td>
</tr>
<tr>
<td>BA1112A2</td>
<td>C</td>
<td>CA1103A28</td>
<td>C</td>
<td>CA2009C1</td>
<td>C</td>
</tr>
<tr>
<td>BA1112A3</td>
<td>C</td>
<td>CA1103A29</td>
<td>C</td>
<td>CA2009C2</td>
<td>C</td>
</tr>
<tr>
<td>BA1112A4M</td>
<td>C</td>
<td>CA1103A30</td>
<td>C</td>
<td>CA2009C3</td>
<td>C</td>
</tr>
<tr>
<td>BA1112A5M</td>
<td>C</td>
<td>CA1103A31</td>
<td>C</td>
<td>CA2009C4</td>
<td>C</td>
</tr>
<tr>
<td>BA1112A6M</td>
<td>C</td>
<td>CA1103A32</td>
<td>C</td>
<td>CA2009C5</td>
<td>C</td>
</tr>
<tr>
<td>BA1112A7M</td>
<td>C</td>
<td>CA1103A33</td>
<td>C</td>
<td>CA2009C6</td>
<td>C</td>
</tr>
<tr>
<td>BA1112A8M</td>
<td>C</td>
<td>CA1103A34</td>
<td>C</td>
<td>CA2009C7</td>
<td>C</td>
</tr>
<tr>
<td>BA1112A9M</td>
<td>C</td>
<td>CA1103A35</td>
<td>C</td>
<td>CA2009C8</td>
<td>C</td>
</tr>
<tr>
<td>BA1113A-B.ADA</td>
<td>P</td>
<td>CA1103A36</td>
<td>C</td>
<td>CA2009C9</td>
<td>C</td>
</tr>
<tr>
<td>BA1113B-A-ADADA</td>
<td>P</td>
<td>CA1103A37</td>
<td>C</td>
<td>CA2009D0</td>
<td>C</td>
</tr>
<tr>
<td>BA1113B1</td>
<td>C</td>
<td>CA1103A38</td>
<td>C</td>
<td>CA2009D1</td>
<td>C</td>
</tr>
<tr>
<td>BA1113B2</td>
<td>C</td>
<td>CA1103A39</td>
<td>C</td>
<td>CA2009D2</td>
<td>C</td>
</tr>
<tr>
<td>BA1113B3</td>
<td>C</td>
<td>CA1103A40</td>
<td>C</td>
<td>CA2009D3</td>
<td>C</td>
</tr>
<tr>
<td>BA1113B4M</td>
<td>C</td>
<td>CA1103A41</td>
<td>C</td>
<td>CA2009D4</td>
<td>C</td>
</tr>
<tr>
<td>BA1113B5M</td>
<td>C</td>
<td>CA1103A42</td>
<td>C</td>
<td>CA2009D5</td>
<td>C</td>
</tr>
<tr>
<td>BA1113B6M</td>
<td>C</td>
<td>CA1103A43</td>
<td>C</td>
<td>CA2009C0A</td>
<td>C</td>
</tr>
<tr>
<td>BA1113B7M</td>
<td>C</td>
<td>CA1103A44</td>
<td>C</td>
<td>CA2009C1A</td>
<td>C</td>
</tr>
<tr>
<td>BA1113B8M</td>
<td>C</td>
<td>CA1103A45</td>
<td>C</td>
<td>CA2009C2A</td>
<td>C</td>
</tr>
<tr>
<td>BA1113B9M</td>
<td>C</td>
<td>CA1103A46</td>
<td>C</td>
<td>CA2009C3A</td>
<td>C</td>
</tr>
<tr>
<td>BA1113A1</td>
<td>C</td>
<td>CA1103A47</td>
<td>C</td>
<td>CA2009C4A</td>
<td>C</td>
</tr>
<tr>
<td>BA1113A2</td>
<td>C</td>
<td>CA1103A48</td>
<td>C</td>
<td>CA2009C5A</td>
<td>C</td>
</tr>
<tr>
<td>BA1113A3</td>
<td>C</td>
<td>CA1103A49</td>
<td>C</td>
<td>CA2009C6A</td>
<td>C</td>
</tr>
<tr>
<td>BA1113A4</td>
<td>C</td>
<td>CA1103A50</td>
<td>C</td>
<td>CA2009C7A</td>
<td>C</td>
</tr>
</tbody>
</table>

D-20
<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA5003B0</td>
<td>C</td>
<td>LA3004A2</td>
<td>C</td>
</tr>
<tr>
<td>CA5003B1</td>
<td>C</td>
<td>LA3004A3</td>
<td>C</td>
</tr>
<tr>
<td>CA5003B2</td>
<td>C</td>
<td>LA3004A4</td>
<td>C</td>
</tr>
<tr>
<td>CA5003B3</td>
<td>C</td>
<td>LA3004A5</td>
<td>C</td>
</tr>
<tr>
<td>CA5003B4</td>
<td>C</td>
<td>LA3004A5M</td>
<td>C</td>
</tr>
<tr>
<td>CA5003B5M</td>
<td>C</td>
<td>LA3004B-AADA</td>
<td>N/A</td>
</tr>
<tr>
<td>[empty]</td>
<td>[empty]</td>
<td>LA5001A-BADA</td>
<td>N/A</td>
</tr>
<tr>
<td>[empty]</td>
<td>[empty]</td>
<td>LA5001A</td>
<td>C</td>
</tr>
<tr>
<td>[empty]</td>
<td>[empty]</td>
<td>LA5001A5</td>
<td>C</td>
</tr>
<tr>
<td>[empty]</td>
<td>[empty]</td>
<td>LA5001A6</td>
<td>C</td>
</tr>
<tr>
<td>[empty]</td>
<td>[empty]</td>
<td>LA5001A7M</td>
<td>C</td>
</tr>
</tbody>
</table>
Validation Summary Report  01/17/86  AlayCOMP_001, version 1.3

Chapter 11

<table>
<thead>
<tr>
<th>File Reference</th>
<th>File Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>BB2001A-AB.ADA</td>
<td>CB1003A-AB.ADA</td>
</tr>
<tr>
<td>BB2002A-AB.ADA</td>
<td>CB1004A-AB.ADA</td>
</tr>
<tr>
<td>BB2003A-AB.ADA</td>
<td>CB2004A-AB.ADA</td>
</tr>
<tr>
<td>BB2004AB.ADA</td>
<td>CB2005A-AB.ADA</td>
</tr>
<tr>
<td>BB2005AB.ADA</td>
<td>CB2006A-AB.ADA</td>
</tr>
<tr>
<td>BB2006AB.ADA</td>
<td>CB2007A-AB.ADA</td>
</tr>
<tr>
<td>BB2007AB.ADA</td>
<td>CB2008A-AB.ADA</td>
</tr>
<tr>
<td>BB2008AB.ADA</td>
<td>CB2009A-AB.ADA</td>
</tr>
<tr>
<td>BB2009AB.ADA</td>
<td>CB5001A-AB.ADA</td>
</tr>
<tr>
<td>CB1001A-AB.ADA</td>
<td>CB4001A-AB.ADA</td>
</tr>
<tr>
<td>CB1002A-AB.ADA</td>
<td>CB4002A-AB.ADA</td>
</tr>
</tbody>
</table>
Validation Summary Report

Chapter 12

<table>
<thead>
<tr>
<th>BC1001A-B.ADA</th>
<th>BC3002A-AB.ADA</th>
<th>BC3202A-B.ADA</th>
<th>BC3301A-AB.ADA</th>
</tr>
</thead>
<tbody>
<tr>
<td>BC1002A-B.ADA</td>
<td>BC3002B-AB.ADA</td>
<td>BC3301B-AB.ADA</td>
<td>BC3302B-AB.ADA</td>
</tr>
<tr>
<td>BC1005A-AB.ADA</td>
<td>BC3002C-AB.ADA</td>
<td>BC3302A-AB.ADA</td>
<td>BC3302B-AB.ADA</td>
</tr>
<tr>
<td>BC1006C-AB.ADA</td>
<td>BC3002E-AB.ADA</td>
<td>BC3303A-AB.ADA</td>
<td>BC3304A-AB.ADA</td>
</tr>
<tr>
<td>BC1006A-AB.ADA</td>
<td>BC30024-AB.ADA</td>
<td>BC3304B-AB.ADA</td>
<td>BC3305A-AB.ADA</td>
</tr>
<tr>
<td>BC1011A-AB.ADA</td>
<td>BC3003B-AB.ADA</td>
<td>BC3306A-AB.ADA</td>
<td>BC3306B-AB.ADA</td>
</tr>
<tr>
<td>BC1011B-AB.ADA</td>
<td>BC3005A-AB.ADA</td>
<td>BC3307A-AB.ADA</td>
<td>BC3307B-AB.ADA</td>
</tr>
<tr>
<td>BC1012A-AB.ADA</td>
<td>BC3008A-AB.ADA</td>
<td>BC3308A-AB.ADA</td>
<td>BC3309B-AB.ADA</td>
</tr>
<tr>
<td>BC1015A-AB.ADA</td>
<td>BC3009A-AB.ADA</td>
<td>BC3309B-AB.ADA</td>
<td>BC3310A-AB.ADA</td>
</tr>
<tr>
<td>BC1010A-B.ADA</td>
<td>BC3009B-AB.ADA</td>
<td>BC3310B-AB.ADA</td>
<td>BC3310C-AB.ADA</td>
</tr>
<tr>
<td>BC1010B-B.ADA</td>
<td>BC3009C-AB.ADA</td>
<td>BC3311A-AB.ADA</td>
<td>BC3311B-AB.ADA</td>
</tr>
<tr>
<td>BC1010C-B.ADA</td>
<td>BC3011B-AB.ADA</td>
<td>BC3312A-AB.ADA</td>
<td>BC3312B-AB.ADA</td>
</tr>
<tr>
<td>BC1010D-B.ADA</td>
<td>BC3011C-AB.ADA</td>
<td>BC3312C-AB.ADA</td>
<td>BC3313A-AB.ADA</td>
</tr>
<tr>
<td>BC1010E-B.ADA</td>
<td>BC3013A-AB.ADA</td>
<td>BC3313B-AB.ADA</td>
<td>BC3313C-AB.ADA</td>
</tr>
<tr>
<td>BC1010F-B.ADA</td>
<td>BC3013B-AB.ADA</td>
<td>BC3313D-AB.ADA</td>
<td>BC3314A-AB.ADA</td>
</tr>
<tr>
<td>BC1010G-B.ADA</td>
<td>BC3013C-AB.ADA</td>
<td>BC3314B-AB.ADA</td>
<td>BC3314C-AB.ADA</td>
</tr>
<tr>
<td>BC1010H-B.ADA</td>
<td>BC3013D-AB.ADA</td>
<td>BC3314D-AB.ADA</td>
<td>BC3315A-AB.ADA</td>
</tr>
<tr>
<td>BC1010I-B.ADA</td>
<td>BC3013E-AB.ADA</td>
<td>BC3315B-AB.ADA</td>
<td>BC3315C-AB.ADA</td>
</tr>
<tr>
<td>BC1010J-B.ADA</td>
<td>BC3013F-AB.ADA</td>
<td>BC3315D-AB.ADA</td>
<td>BC3316A-AB.ADA</td>
</tr>
<tr>
<td>BC1010K-B.ADA</td>
<td>BC3013G-AB.ADA</td>
<td>BC3316B-AB.ADA</td>
<td>BC3316C-AB.ADA</td>
</tr>
<tr>
<td>BC1010L-B.ADA</td>
<td>BC3013H-AB.ADA</td>
<td>BC3316D-AB.ADA</td>
<td>BC3316E-AB.ADA</td>
</tr>
<tr>
<td>BC1010M-B.ADA</td>
<td>BC3013I-AB.ADA</td>
<td>BC3316F-AB.ADA</td>
<td>BC3316G-AB.ADA</td>
</tr>
<tr>
<td>BC1010N-B.ADA</td>
<td>BC3013J-AB.ADA</td>
<td>BC3316H-AB.ADA</td>
<td>BC3316I-AB.ADA</td>
</tr>
<tr>
<td>BC1010O-B.ADA</td>
<td>BC3013K-AB.ADA</td>
<td>BC3316J-AB.ADA</td>
<td>BC3316K-AB.ADA</td>
</tr>
<tr>
<td>BC1010P-B.ADA</td>
<td>BC3013L-AB.ADA</td>
<td>BC3316L-AB.ADA</td>
<td>BC3316M-AB.ADA</td>
</tr>
<tr>
<td>BC1010Q-B.ADA</td>
<td>BC3013M-AB.ADA</td>
<td>BC3316N-AB.ADA</td>
<td>BC3316O-AB.ADA</td>
</tr>
<tr>
<td>BC1010R-B.ADA</td>
<td>BC3013N-AB.ADA</td>
<td>BC3316P-AB.ADA</td>
<td>BC3316Q-AB.ADA</td>
</tr>
<tr>
<td>BC1010S-B.ADA</td>
<td>BC3013O-AB.ADA</td>
<td>BC3316R-AB.ADA</td>
<td>BC3316S-AB.ADA</td>
</tr>
<tr>
<td>BC1010T-B.ADA</td>
<td>BC3013P-AB.ADA</td>
<td>BC3316T-AB.ADA</td>
<td>BC3316U-AB.ADA</td>
</tr>
<tr>
<td>BC1010U-B.ADA</td>
<td>BC3013Q-AB.ADA</td>
<td>BC3316U-AB.ADA</td>
<td>BC3316V-AB.ADA</td>
</tr>
<tr>
<td>BC1010V-B.ADA</td>
<td>BC3013R-AB.ADA</td>
<td>BC3316V-AB.ADA</td>
<td>BC3316W-AB.ADA</td>
</tr>
<tr>
<td>BC1010W-B.ADA</td>
<td>BC3013S-AB.ADA</td>
<td>BC3316W-AB.ADA</td>
<td>BC3316X-AB.ADA</td>
</tr>
<tr>
<td>BC1010X-B.ADA</td>
<td>BC3013T-AB.ADA</td>
<td>BC3316X-AB.ADA</td>
<td>BC3316Y-AB.ADA</td>
</tr>
<tr>
<td>BC1010Y-B.ADA</td>
<td>BC3013U-AB.ADA</td>
<td>BC3316Y-AB.ADA</td>
<td>BC3316Z-AB.ADA</td>
</tr>
<tr>
<td>BC1010Z-B.ADA</td>
<td>BC3013V-AB.ADA</td>
<td>BC3316Z-AB.ADA</td>
<td>BC3317A-AB.ADA</td>
</tr>
</tbody>
</table>

D-23
<table>
<thead>
<tr>
<th>Validation Summary Report 01/17/86</th>
<th>AleyCOMF_001. version 1.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>BC3502H-AB.ADA P CC3007A-AB.ADA P</td>
<td>CC3407D-AB.ADA p</td>
</tr>
<tr>
<td>BC3502D-AB.ADA P CC3011A-B.ADA P</td>
<td>CC3407E-AB.ADA p</td>
</tr>
<tr>
<td>BC3503A-B.ADA W CC3011D-B.ADA P</td>
<td>CC3407F-AB.ADA p</td>
</tr>
<tr>
<td>BC3503B-AB.ADA P CC3012A-AB.ADA P</td>
<td>CC3408A-AB.ADA p</td>
</tr>
<tr>
<td>BC3503C-B.ADA P CC3128A-AB.ADA P</td>
<td>CC3408B-AB.ADA p</td>
</tr>
<tr>
<td>BC3503D-B.ADA P CC3129B-AB.ADA P</td>
<td>CC3408C-AB.ADA p</td>
</tr>
<tr>
<td>BC3503F-B.ADA P CC3125A-B.ADA P</td>
<td>CC3408D-AB.ADA p</td>
</tr>
<tr>
<td>BC1504A-AB.ADA P CC3203A-B.ADA P</td>
<td>CC3504A-AB.ADA p</td>
</tr>
<tr>
<td>BC1504B-AB.ADA P CC3208A-AB.ADA P</td>
<td>CC3504B-AB.ADA p</td>
</tr>
<tr>
<td>BC1504C-AB.ADA P CC3209A-AB.ADA P</td>
<td>CC3504C-AB.ADA p</td>
</tr>
<tr>
<td>BC1504D-AB.ADA P CC3205A-AB.ADA P</td>
<td>CC3504D-AB.ADA p</td>
</tr>
<tr>
<td>BC1504E-AB.ADA P CC3205C-AB.ADA P</td>
<td>CC3504E-AB.ADA p</td>
</tr>
<tr>
<td>BC1504F-AB.ADA P CC3205A-AB.ADA P</td>
<td>CC3504F-AB.ADA p</td>
</tr>
<tr>
<td>BC1504G-AB.ADA P CC3205C-AB.ADA P</td>
<td>CC3504G-AB.ADA p</td>
</tr>
<tr>
<td>BC1504H-AB.ADA P CC3205B-AB.ADA P</td>
<td>CC3504H-AB.ADA p</td>
</tr>
<tr>
<td>BC1504I-AB.ADA P CC3205B-AB.ADA P</td>
<td>CC3504I-AB.ADA p</td>
</tr>
<tr>
<td>BC1504J-AB.ADA P CC3205D-AB.ADA P</td>
<td>CC3504J-AB.ADA p</td>
</tr>
<tr>
<td>BC1504K-AB.ADA P CC3205C-AB.ADA P</td>
<td>CC3504K-AB.ADA p</td>
</tr>
<tr>
<td>BC1504L-AB.ADA P CC3205A-AB.ADA P</td>
<td>CC3504L-AB.ADA p</td>
</tr>
<tr>
<td>BC1504M-AB.ADA P CC3205C-AB.ADA P</td>
<td>CC3504M-AB.ADA p</td>
</tr>
<tr>
<td>BC1504N-AB.ADA P CC3205B-AB.ADA P</td>
<td>CC3504N-AB.ADA p</td>
</tr>
<tr>
<td>BC1504O-AB.ADA P CC3205D-AB.ADA P</td>
<td>CC3504O-AB.ADA p</td>
</tr>
<tr>
<td>BC1504P-AB.ADA P CC3205C-AB.ADA P</td>
<td>CC3504P-AB.ADA p</td>
</tr>
<tr>
<td>BC1504Q-AB.ADA P CC3205B-AB.ADA P</td>
<td>CC3504Q-AB.ADA p</td>
</tr>
<tr>
<td>BC1504R-AB.ADA P CC3205D-AB.ADA P</td>
<td>CC3504R-AB.ADA p</td>
</tr>
<tr>
<td>BC1504S-AB.ADA P CC3205C-AB.ADA P</td>
<td>CC3504S-AB.ADA p</td>
</tr>
<tr>
<td>BC1504T-AB.ADA P CC3205B-AB.ADA P</td>
<td>CC3504T-AB.ADA p</td>
</tr>
<tr>
<td>BC1504U-AB.ADA P CC3205D-AB.ADA P</td>
<td>CC3504U-AB.ADA p</td>
</tr>
<tr>
<td>BC1504V-AB.ADA P CC3205C-AB.ADA P</td>
<td>CC3504V-AB.ADA p</td>
</tr>
<tr>
<td>BC1504W-AB.ADA P CC3205B-AB.ADA P</td>
<td>CC3504W-AB.ADA p</td>
</tr>
<tr>
<td>BC1504X-AB.ADA P CC3205D-AB.ADA P</td>
<td>CC3504X-AB.ADA p</td>
</tr>
<tr>
<td>BC1504Y-AB.ADA P CC3205C-AB.ADA P</td>
<td>CC3504Y-AB.ADA p</td>
</tr>
<tr>
<td>BC1504Z-AB.ADA P CC3205B-AB.ADA P</td>
<td>CC3504Z-AB.ADA p</td>
</tr>
<tr>
<td>BC2502A-AB.ADA P CC3407A-AB.ADA P</td>
<td>CC3407B-AB.ADA p</td>
</tr>
<tr>
<td>BC3502A-AB.ADA P CC3407A-AB.ADA P</td>
<td>CC3407C-AB.ADA p</td>
</tr>
<tr>
<td>AE2101A-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2101B-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2101C-B.DEP</td>
<td>P</td>
</tr>
<tr>
<td>AE2101D-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2101E-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2101F-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2102A-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2102B-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2102C-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2102D-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2102E-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2102F-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2102G-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2102H-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2102I-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2102J-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2102K-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2102N-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2102O-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2102P-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2102Q-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2102R-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2102S-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2102T-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2102U-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2102V-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2102W-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2102X-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2102Y-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2102Z-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2103A-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2103B-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2103C-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2103D-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2103E-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2103F-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2103G-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2103H-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2103I-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2103J-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2103K-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2103L-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2103M-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2103N-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2103O-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2103P-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2103Q-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2103R-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2103S-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2103T-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2103U-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2103V-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2103W-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2103X-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2103Y-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>AE2103Z-B.ADA</td>
<td>P</td>
</tr>
</tbody>
</table>

Chapter 14
<table>
<thead>
<tr>
<th>CE3602A-B.ADA</th>
<th>P</th>
<th>CE3704N-B.ADA</th>
<th>P</th>
<th>CE3804A-B.ADA</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>CE3602B-B.ADA</td>
<td>P</td>
<td>CE3704O-B.ADA</td>
<td>P</td>
<td>CE3804C-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>CE3602C-B.ADA</td>
<td>P</td>
<td>CE3706C-B.ADA</td>
<td>P</td>
<td>CE38060-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>CE3602D-B.ADA</td>
<td>P</td>
<td>CE3706D-B.ADA</td>
<td>P</td>
<td>CE3806E-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>CE3603A-B.ADA</td>
<td>W</td>
<td>CE3706F-B.ADA</td>
<td>P</td>
<td>CE3808A-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>CE3604A-B.ADA</td>
<td>W</td>
<td>CE3706G-B.ADA</td>
<td>P</td>
<td>CE3808B-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>CE3605A-B.ADA</td>
<td>P</td>
<td>CE3707A-B.ADA</td>
<td>P</td>
<td>CE3810A-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>CE3605B-B.ADA</td>
<td>P</td>
<td>CE3708A-B.ADA</td>
<td>P</td>
<td>CE3801A-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>CE3605C-B.ADA</td>
<td>P</td>
<td>CE3801A-B.ADA</td>
<td>P</td>
<td>CE3905A-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>CE3605D-B.ADA</td>
<td>P</td>
<td>CE3804A-B.ADA</td>
<td>P</td>
<td>CE3905B-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>CE3605E-B.ADA</td>
<td>P</td>
<td>CE3804B-B.ADA</td>
<td>P</td>
<td>CE3905C-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>CE3606A-B.ADA</td>
<td>P</td>
<td>CE3804C-B.ADA</td>
<td>P</td>
<td>CE3905L-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>CE3606B-B.ADA</td>
<td>P</td>
<td>CE3804D-B.ADA</td>
<td>P</td>
<td>CE3906A-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>CE3701A-B.ADA</td>
<td>P</td>
<td>CE3804E-B.ADA</td>
<td>P</td>
<td>CE3906B-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>CE3701A-B.ADA</td>
<td>P</td>
<td>CE3804F-B.ADA</td>
<td>P</td>
<td>CE3906C-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>CE3709B-B.ADA</td>
<td>P</td>
<td>CE3804G-B.ADA</td>
<td>P</td>
<td>CE3906D-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>CE3709C-B.ADA</td>
<td>P</td>
<td>CE3804J-B.ADA</td>
<td>P</td>
<td>CE3906E-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>CE3709D-B.ADA</td>
<td>P</td>
<td>CE3804K-B.ADA</td>
<td>P</td>
<td>CE3906F-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>CE3709E-B.ADA</td>
<td>P</td>
<td>CE3804M-B.ADA</td>
<td>P</td>
<td>CE3907A-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>CE3709F-B.ADA</td>
<td>P</td>
<td>CE3805A-B.ADA</td>
<td>P</td>
<td>CE3908A-B.ADA</td>
<td>P</td>
</tr>
<tr>
<td>CE3709M-B.ADA</td>
<td>W</td>
<td>CE3805B-B.ADA</td>
<td>P</td>
<td>CE3102C-B.ADA</td>
<td>P</td>
</tr>
</tbody>
</table>