ACATS Modification List for ACATS 4.1
Version 4.1L
March 29, 2019

What is this?

This is the ACATS modification list for ACATS 4.1. This document includes test corrections, test removals, and new tests for ACATS 4.1. ACATS 4.1 is the current test suite for Ada, covering the third edition of Ada, ISO/IEC 8652:2012 (usually known as "Ada 2012").

ACATS Modification List version 3.1V for ACATS 3.1 is the current list for that test suite. ACATS 3.1 should be used for testing implementations that support Amendment 1 (ISO/IEC 8652:1995/AMD 1:2007, often known as "Ada 2005"), and will continue to be maintained as necessary for the foreseeable future.

ACATS Modification List version 2.6G for ACATS 2.6 is the current list for that test suite. ACATS 2.6 should be used for testing implementations of the second edition of Ada, ISO/IEC 8652:1995 (usually known as "Ada 95"); it also will continue to be maintained as necessary for the foreseeable future.

Changes from the last list:

A handful of tests changed status on January 1st; see the lists for the details.

New test B452002 checks that the Name Resolution and Legality Rules for the universal access "=" operator are enforced.

New tests C452004, C452005, and C452006 check that in various cases the correct equality operation is called by an individual membership test of a choice expression.

New tests posted between January 1, 2019 and March 31, 2019 will become required for conformity assessments on October 1, 2019. New tests posted between October 1, 2018 and December 31, 2018 will become required for conformity assessments on July 1, 2019. New tests posted between July 1, 2018 and September 30, 2018 will become required for conformity assessments on April 1, 2019. New tests posted before July 1, 2018 are required for conformity assessments. The effective dates of tests are marked in the lists.

Questions about this document? Contact the ACAA Technical Agent at agent@ada-auth.org.

ACATS 4.1 includes optional Test Grading tools

ACATS 4.1 introduces new test grading tools.

When the ACATS was designed (as the ACVC in the early 1980s), the intention always was that running it would give a simple and clear Pass or Fail result. However, grading of tests (particularly of B and L Tests) is somewhat subjective and very time-consuming. (Test grading typically involved poring over compiler listings of the entire ACATS with a large highlighter.) The subjectivity was acceptable in the days of common formal testing, as the formal testers would enforce a common grading standard (even if no one could describe what that was), but in an age of self-testing, it reduces the value of the ACATS.

As such, we're introducing a long-overdue set of automated grading tools. These tools are intended to:

• Greatly reduce the effort to grade an ACATS test run (and, hopefully to reduce the cost of formal testing to make it a more practical option);
• Ensure that all users of the ACATS grade tests in the same way, thus making it more valid to compare ACATS results between different implementations and different testers;
• Give implementers certainty as to whether a particular test result is passing (whether the location of an error is “close enough” to the canonical position);
• Be able to detect process errors (especially errors of omission) – that is, tests improperly run (for instance, files compiled in the wrong order where the ACATS requires an order), or not run at all (if one is checking off lines in a listing, it's hard to notice that some test or portion of a test is missing).

The tools also will optionally give feedback to implementers about quality of error message positioning and of error cascades (does one error generate multiple messages?). (Since “goodness of an implementation” is not something that the ACATS tests for, this is not an official goal for the tools.)

This is an ambitious set of goals. It's unlikely that they'll be completely met by the initial versions of the tools. As such, their use will be completely optional (but encouraged!) for ACATS 4.1. The tools will be updated periodically by the normal process for updating the ACATS – an ACATS Modification List like this one.

Details on the operation of the tools can be found in the ACATS 4.1 documentation (that documentation can be found at http://www.ada-auth.org/acats-files/4.1/docs/UG-6.HTM). As always, if you have questions, contact the ACAA Technical Agent at agent@ada-auth.org.

**Notice:**

The ACAA has created and maintains the Ada Conformity Assessment Test Suite for the purpose of conformity assessments conducted in accordance with the International Standard ISO/IEC 18009 - Ada: Conformity assessment of a language processor. This test suite should not be used to make claims of conformance unless used in accordance with ISO/IEC 18009 and any applicable ACAA procedures.

**List explanation:**

Test numbers given are for the basic test. Categories are “Withdrawn” (the test should not be processed for Ada conformity assessments); “Allowed Modification” (either the original test or the modified one should be processed for Ada conformity assessments); “Modified” (the modified test must be processed for Ada conformity assessments); “Pending New” (the test does not need to be processed for Ada conformity assessments, but will be added in the future); “New” (the test is processed for Ada conformity assessments).

ACATS version labels are provided with all new and modified tests. Use this label to retrieve the test from the ACATS VCS. See “How to get new and modified tests”.

In citations, line numbers are given relative to the start of a test file in the format of '@<line#,line#,...>; if a test program comprises more than one file, then the particular file will be indicated by "fn", where "n" is the test-program name's file-designator numeral. E.g., "B38103E : @f1-31, ... " denotes test file b38103e1.ada. In cases where there are two or more rationales for different errors in a test program, these different rationales with their respective line citations are listed on separate lines. References to the Ada standards are given in the format:

  <section|chapter>.<clause>[.<subclause>]:<paragraph>

These references are to the consolidated Ada standard as available at http://www.ada-auth.org/standards/rm12_w_tc1/html/RM-TOC.html.

---

**Main list:**

===== Core Language =====
Name     Category  Version label  (effective date; old version label)  
Short description.

C350001 [New] VCS Label: A4_1D  
Check that the object form of the Image attribute is implemented.

C352001 [Modified] VCS Label: A4_1D  
Corrected Failed message @267 to reflect intended result.

B354002 [New] VCS Label: A4_1D  
Test basic requirements for the Mod attribute.

C354004 [New] VCS Label: A4_1D  
Test basic operation of the Mod attribute.

B35A001 [New] VCS Label: A4_1D  
Check Legality Rules are enforced when Small is specified by an aspect specification.

C35A001 [New] VCS Label: A4_1D  
Check that Small can be specified by an aspect specification.

B37309B [Modified] VCS Label: A4_1G  
Added additional error locations to directly support common error reporting strategies.

B37310B [Modified] VCS Label: A4_1G  
Added additional error locations to directly support common error reporting strategies.

B37311A [Modified] VCS Label: A4_1G  
Added additional error locations to directly support common error reporting strategies.

B38003D [Modified] VCS Label: A4_1A  
Corrected error tags @70,76,82,88 so “optional” and “error” are on the same line, as required by the test standards.

B392002 [Modified] VCS Label: A4_1G  
Revised to use Possible Error tags, and added error location indicators.

B392003 [Modified] VCS Label: A4_1G  
Added additional error locations to directly support common error reporting strategies.

B392004 [Modified] VCS Label: A4_1G  
Added additional error locations to directly support common error reporting strategies.

B392005 [Modified] VCS Label: A4_1G  
Revised to use Possible Error tags, and added error location indicators.

B393001 [Modified] VCS Label: A4_1G  
Added additional error locations to directly support common error reporting strategies.
reporting strategies.

B393007 [Modified] VCS Label: A4_1G
Added additional error locations to directly support common error reporting strategies. Fixed overlong lines.

B393010 [Modified] VCS Label: A4_1G
Added additional error locations to directly support common error reporting strategies. Fixed overlong lines.

C3A0030 [New] VCS Label: A4_1A
Check that an access discriminant is only null excluding if it includes a null exclusion.

B3A1001 [Modified] VCS Label: A4_1A
Split Pack6 so possible error sets don't intersect; fixed overlong lines.

B3A1002 [Modified] VCS Label: A4_1A
Split Pack6 so possible error sets don't intersect; fixed overlong lines.

B3A2007 [Modified] VCS Label: A4_1G
Added additional error locations to directly support common error reporting strategies.

B3A2008 [Modified] VCS Label: A4_1G
Added additional error locations to directly support common error reporting strategies.

B3A2009 [Modified] VCS Label: A4_1G
Removed tabs to meet test format standards; added error location indicators.

B3A2011 [Modified] VCS Label: A4_1G
Added additional error locations to directly support common error reporting strategies.

B3A2012 [Modified] VCS Label: A4_1G
Added additional error locations to directly support common error reporting strategies.

B3A2016 [Modified] VCS Label: A4_1G
Added additional error locations to directly support common error reporting strategies.

B3A2018 [New] VCS Label: A4_1H
Check that an explicitly aliased parameter has the same accessibility level as a normal parameter unless compared to the return object. Check that master of the function call is properly determined statically.

B413004 [Modified] VCS Label: A4_1J
PPA and PFA trigger the issue of AI12-0289-1; explicit null exclusions were added to the parameters to avoid it.

B452001 [Pending New] New VCS Label: A4_1K
Required after 2019-07-01
Check that a membership that contains choice expressions is illegal for a limited type that does not have equality.

B452002 [Pending New] New VCS Label: A4_1L
Required after 2019-10-01
Check that Resolution and Legality Rules for the universal access equality operator are enforced.

C452002 [New] VCS Label: A4_1F
[VCS Label A4_1D] Check that overloading resolution of Ada 2012 membership operations works as intended.
[VCS Label A4_1F] Fixed incorrect test number passed to Report.

C452003 [Pending New] New VCS Label: A4_1K
Required after 2019-07-01
Check that memberships of general access-to-object types work as defined.

C452004 [Pending New] New VCS Label: A4_1L
Required after 2019-10-01
Check that memberships with multiple choices are evaluated as required, and that the correct equality operation is called for untagged record types.

C452005 [Pending New] New VCS Label: A4_1L
Required after 2019-10-01
Check that memberships with multiple choices call the correct equality operation for array types.

C452006 [Pending New] New VCS Label: A4_1L
Required after 2019-10-01
Check that memberships with multiple choices call the correct equality operation for access types.

C452A01 [Pending New] New VCS Label: A4_1K
Required after 2019-07-01
Check that “=” for language-defined private types behaves as if it is predefined. Part 1: core types.

C452A02 [Pending New] New VCS Label: A4_1K
Required after 2019-07-01
Check that “=” for language-defined private types behaves as if it is predefined. Part 2: types of the Ada.Strings subsystem.

C452A03 [Pending New] New VCS Label: A4_1K
Required after 2019-07-01
Check that “=” for language-defined private types behaves as if it is predefined. Part 3: Annex C types.

C453001 [New] VCS Label: A4_1D
Check that modular adding operators wrap around rather than raising Constraint_Error.

B455002 [Modified] VCS Label: A4_1A
Corrected casing of error tags and fixed overlong lines.
B457001 [Modified] VCS Label: A4_1F
Fixed overlong lines. Added location indicators.

B457004 [Modified] VCS Label: A4_1F
There should not be an ERROR tag @282; Single_Highest @290 should be Single_High. Added location indicators.

B458001 [New] VCS Label: A4_1F
Check that a quantified expression has to be surrounded in parentheses if it is not already surrounded by them.

C460014 [New] VCS Label: A4_1C
Check that Constraint_Error is raised appropriately when converting access-to-classwide to another access type.

C460015 [New] VCS Label: A4_1C
Check that Constraint_Error is raised appropriately when converting access-to-limited-classwide to another access type.

B470001 [New] VCS Label: A4_1D
Check that qualified expressions are a constant view of the expression, even if the expression is a variable object.

C470001 [New] VCS Label: A4_1D
Check that qualified expressions are a name, and thus can be used as a prefix and other places a name is required.

C480001 [New] VCS Label: A4_1D
[VCS Label A4_1A] Check that an initialized allocator for a class-wide type does an accessibility check on the tag of the initializing object.
[VCS Label A4_1D] Changed mode to “in” (instead of “in out”) @103 & 144 as those parameters are used in functions, making that an Ada 2012 feature, and this is an Ada 2005 test.

B480004 [New] VCS Label: A4_1G
Check that an allocator is illegal if the access type has a Storage_Size that is statically zero.

B4A010C [Modified] VCS Label: A4_1G
Added additional error locations to directly support common error reporting strategies.

C52101A [Modified] VCS Label: A4_1A
Corrected comment @49 not to have the form of an error tag.

B540001 [Modified] VCS Label: A4_1G
Added additional error locations to directly support common error reporting strategies.

B540002 [Modified] VCS Label: A4_1G
Added additional error locations to directly support common error reporting strategies.

B54B01B [Modified] VCS Label: A4_1G
Added additional error locations to directly support common error reporting strategies.
Added additional error locations to directly support common error reporting strategies.

Fixed an overlong line.

Check that Pre and Post are not allowed on subprograms acting as completions.

Corrected several error location codes.
Check that Pre and Post are not allowed on abstract subprograms and null procedures.

Check that Pre'Class and Post'Class expressions added by AI12-0113-1 are enforced. Clarified error @208; changed error @233 & @247 to OK (as the inherited Proc4 routine is abstract and thus is not subject to the new rule).

Check that Pre'Class and Post'Class are not allowed on entries and protected subprograms of untagged task and protected types. Corrected preconditions @77 and @110 not to violate the new rule added by Binding Interpretation AI12-0166-1.

Check that F'Result is not allowed in the body of F.

Check that the Old attribute is not allowed in a subprogram or entry body, or in an accept statement.

Check the Old attribute cannot be used within a precondition expression nor in a generic specification outside of a postcondition expression.

For a discrete X, check that X'Old has the nominal subtype of X.

If an Old attribute reference occurs at a place where it is potentially unevaluated, check that the prefix must statically denote an object. Corrected error location codes @240 and @252.

Check that tagged task and tagged protected types do not allow Pre'Class and Post'Class, nor are they allowed on operations of those types.
Check that Pre'Class cannot be specified for an overriding of a subprogram that does not specify Pre'Class. Declaration of New_Proc @358 is not overriding and thus not subject to the rule. The declaration of Shine @149 never overrides primitives of the actual type by 12.3(18), thus this rule does not apply to instances of Gen2. Added an example using a generic derived type.

Check 6.1.1(10-16/3).

Type T08 @230 and @241 needs to be declared abstract.

Check 6.1.1(17/3).
The declaration of Get_Count @112 never overrides primitives of the actual type by 12.3(18), thus this rule does not apply to instances of Gen2. Added examples using generic derived types.

Check that preconditions and postconditions where both specific and class-wide contracts apply are checked properly.

Check that the correct expressions are evaluated for specific preconditions and postconditions, and that they are evaluated at the correct point.
The main subprogram name was corrected to follow the rules of the ACATS User's Guide, clause 4.3.3.

Check that the correct expressions are evaluated for class-wide preconditions and postconditions, and that they are evaluated at the correct point.
The main subprogram name was corrected to follow the rules of the ACATS User's Guide, clause 4.3.3.

Check that the correct expressions are evaluated for class-wide preconditions and postconditions of an interface type, and that they are evaluated at the correct point.
Fixed overlong lines.

Check that X'Old has the tag of X and not the tag of the nominal subtype of X (for Post).
The main subprogram name was corrected to follow the rules of the ACATS User's Guide, clause 4.3.3.

Check that X'Old has the tag of X and not the tag of the nominal subtype of X (for Post'Class).
The main subprogram name was corrected to follow the rules of the ACATS User's Guide, clause 4.3.3.
C611B03 [New] VCS Label: A4_1D
   [VCS Label A4_1C] Check that X'Old has the tag of X and not the
tag of the nominal subtype of X (for Post'Class of an interface
type).
   [VCS Label A4_1D] Fixed overlong lines.

B62001C [Modified] VCS Label: A4_1A
   Changed error @35 to be optional, as the syntax error @33 is
required (and the comment that only one error need be detected is
bogus).

B62001D [Modified] VCS Label: A4_1A
   Changed error @41 to be optional, as the syntax error @33 is
required (and the comment that only one error need be detected is
bogus).

B65005A [Modified] VCS Label: A4_1D
   Added additional error locations to directly support common error
reporting strategies.

B65005B [Modified] VCS Label: A4_1D
   Added additional error locations to directly support common error
reporting strategies.

C650B04 [New] VCS Label: A4_1A
   Check that the result of a function that returns a class-wide
tagged type does an accessibility check on the tag of the result
object when the result expression is class-wide.

B651001 [New] VCS Label: A4_1D
   Check that a return statement cannot apply to a nonreturning
procedure, and that a renames-as-body of a nonreturning procedure
has to be a nonreturning procedure (pragma No_Return version).

C651001 [New] VCS Label: A4_1D
   Check that a nonreturning procedure can raise an exception and
raises Program_Error otherwise (pragma No_Return version).

B651002 [New] VCS Label: A4_1D
   Check that a return statement cannot apply to a nonreturning
procedure, and that a renames-as-body of a nonreturning procedure
has to be a nonreturning procedure (aspect No_Return version).

C651002 [New] VCS Label: A4_1D
   Check that a nonreturning procedure can raise an exception and
raises Program_Error otherwise (aspect No_Return version).

B651003 [New] VCS Label: A4_1D
   Check that an aspect No_Return can only be given on a procedure or
generic procedure, that the expression has to be nonstatic, and
that the aspect is not given on an instance or null procedure.

B680001 [Modified] VCS Label: A4_1F
   The immediately enclosing expression function @140 does not
declare a parameter A, but A is used @141. Added location
indicators.
Corrected format of error tags; added location indicators.

Test various rules about completions of private types and extensions; replaces test B74001A.

Test various rules about completions of private types and extensions in generic units; replaces test B74001B.

Test operations available for a private type; replaces test B74203E.

Added additional error locations to directly support common error reporting strategies.

Added additional error locations to directly support common error reporting strategies.

Added additional error locations to directly support common error reporting strategies.

Added additional error locations to directly support common error reporting strategies.

Added additional error locations to directly support common error reporting strategies.

Added additional error locations to directly support common error reporting strategies.

Added additional error locations to directly support common error reporting strategies.

Added additional error locations to directly support common error reporting strategies.

Modified criteria @75 so that it doesn't look like an error tag to the grading tools; also fixed an overlong line. Modified grading to document that processing the main is optional. Added error location indicators.

Replaced by B730011.
B74001B [Withdrawn] VCS Label: A4_1C
   Replaced by B730012.

B74203E [Withdrawn] VCS Label: A4_1D
   Replaced by B730013.

B750A07 [New] VCS Label: A4_1G
   [VCS Label A4_1A] Check that only legal expressions are allowed
   in the return expression of an expression function that returns a
   limited type.
   [VCS Label A1_1G] Adjusted error indicators to allow the error to
   be reported anywhere in a multi-line expression.

C760A02 [New] VCS Label: A4_1G
   Checks that functions that return immutably limited types are
   built-in-place.

C760A03 [New] VCS Label: A4_1G
   Checks that expression functions that return immutably limited
   types are built-in-place.

B83027B [Modified] VCS Label: A4_1G
   The discriminant @90 needs a default expression as all or none of
   the discriminants of a type need defaults. Also added error
   location indicators.

B83027D [Modified] VCS Label: A4_1G
   Corrected bad error tag, and added error location indicators.

B831006 [Pending New] New VCS Label: A4_1K
   Required after 2019-07-01
   Checks that overriding indicators cannot be given on any
   subprogram declaration in a protected body.

B83E01B [Modified] VCS Label: A4_1G
   Removed possible error cascades, reformatted some error tags, and
   added error location indicators.

B83F02C [Modified] VCS Label: A4_1A
   Updated many optional error markers to match ACATS standards.

B85002A [Withdrawn] VCS Label: A4_1H
   Replaced by B851005.

B85003A [Withdrawn] VCS Label: A4_1H
   Replaced by B851007.

B85003B [Withdrawn] VCS Label: A4_1H
   Replaced by B851007.

B851004 [Modified] VCS Label: A4_1J
   Revised test to eliminate cases whose result would be
   changed by pending AI12-0287-1, and added cases to preserve
   complete testing.

B851005 [New] VCS Label: A4_1H
   Checks that “known to be constrained” is properly enforced for
renames. Part 1: "Normal" cases (no generics or access cases).

B851006 [New] VCS Label: A4_1H
Checks that "known to be constrained" is properly enforced for renames. Part 2: Access cases.

B851007 [Pending New] New VCS Label: A4_1K
Required after 2019-07-01
[VCS Label A4_1H] Checks that "known to be constrained" is properly enforced for renames. Part 3: Generic cases.
[VCS Label A4_1K] Corrected types of declaration @198 and @214, and mode @328.

B853001 [New] VCS Label: A4_1H
Checks that a renamed limited view cannot be used where there is not a with for the original package.

B854002 [Pending New] New VCS Label: A4_1J
Required after 2019-04-01
[VCS Label A4_1H] When parameter or result has a null exclusion for a subprogram rename, check that the subtype of the corresponding parameter or result of the renamed subprogram must exclude null.
[VCS Label A4_1J] Corrected wording of the second test objective.

B854003 [Pending New] New VCS Label: A4_1J
Required after 2019-04-01
Check various rules about renames as body: conformance of both the renaming and renamed entity, and renaming of abstract and intrinsic subprograms.

B854004 [Pending New] New VCS Label: A4_1J
Required after 2019-04-01
Check that renaming a subprogram that requires overriding is illegal.

B854005 [Pending New] New VCS Label: A4_1J
Required after 2019-04-01
Check that a renaming of a prefixed view is illegal if renaming the prefix as an object is illegal.

B91003A [Modified] VCS Label: A4_1C
Added additional error locations to directly support common error reporting strategies.

B91003B [Modified] VCS Label: A4_1C
Added additional error locations to directly support common error reporting strategies.

B91003C [Modified] VCS Label: A4_1C
Added additional error locations to directly support common error reporting strategies.

B91003D [Modified] VCS Label: A4_1C
Added additional error locations to directly support common error reporting strategies.
B940003 [Modified] VCS Label: A4_1C
   Added additional error locations to directly support common error reporting strategies.

B940004 [Modified] VCS Label: A4_1C
   Split into 8 files; added location indicators.

B95007B [Modified] VCS Label: A4_1A
   Fixed error markers @62,66,70 to match ACATS standards.

BA1001D [Modified] VCS Label: A4_1G
   Revised to mark the main subprogram as optional, as it is not necessary to check the test objective. Added error location indicators.

BA1020F [Modified] VCS Label: A4_1G
   Revised to mark the main subprogram as optional, as it is not necessary to check the test objective. Added error location indicators.

BA11007 [Modified] VCS Label: A4_1G
   Split into six test files to avoid requiring error recovery.

BA11013 [Modified] VCS Label: A4_1G
   Revised to make processing the last compilation unit optional, as it is not necessary to check the test objective. Added error location indicators.

BA1101E [Withdrawn] Org VCS Label: A4_1G
   Replaced by test BA14001.

BA12010 [Modified] VCS Label: A4_1G
   Revised to use Possible Error tags, and added error location indicators.

BA12018 [New] VCS Label: A4_1H
   Check that names mentioned but not named in a private with are also treated as privately withed.

BA14001 [New] VCS Label: A4_1G
   Replaces test BA1101E with modern format and additional test cases for Ada 95 compilation units.

BA14002 [New] VCS Label: A4_1G
   Similar to BA14001, testing private with.

BA14003 [New] VCS Label: A4_1G
   Checks that a unit is illegal if it depends on two versions of the same unit.

BA21005 [New] VCS Label: A4_1D
   Test that Pure and Preelaborate aspects cannot be specified late, with a nonstatic expression, or an expression of a type other than Boolean.

BB42001 [New] VCS Label: A4_1C
   [VCS Label A4_1A] Check that the arguments of a pragma Assert
have to have the correct types.

[VCS Label A4_1C] Corrected two incorrect error location codes.

CB42001 [New] VCS Label: A4_1A
Check the usual operation of pragma Assert, including the optional message.

BB42002 [New] New VCS Label: A4_1A
Check that the arguments of a pragma Assert cannot be reordered even if named notation is used.

CB42002 [New] New VCS Label: A4_1A
Check that if the assertion policy is Ignore, none of the expressions of a pragma Assert are evaluated and no exception is raised by the pragma.

BB42003 [New] VCS Label: A4_1A
Check that a pragma Assertion_Policy is illegal if the policy is not recognized.

CB42003 [New] VCS Label: A4_1A
Check that procedure Ada.Assertions.Assert evaluates its arguments and raises Assertion_Error regardless of the assertion policy.

CC3601C [Modified] VCS Label: A4_1E
Moved "=" for type Rec in order that it not violate 4.5.2(9.8/4).

BC40003 [Pending New] New VCS Label: A4_1J
Required after 2019-04-01
[VCS A4_1H] Checks that the subtype of the actual for a generic formal object with a null exclusion must exclude null.
[VCS A4_1J] Revised test to eliminate cases whose result would be changed by pending AI12-0287-1, and to preserve complete testing.

CC51016 [Modified] VCS Label: A4_1G
Corrected bad error tag, and added error location indicators.

BC54004 [New] VCS Label: A4_1H
Checks that the null exclusion of the formal type and of the actual subtype for a generic formal access type must match.

BC60005 [Pending New] New VCS Label: A4_1J
Required after 2019-04-01
[VCS Label A4_1H] Checks that the subtype of a parameter or result of the actual subprogram for a generic formal subprogram where the corresponding parameter or result has a null exclusion must exclude null.
[VCS Label A4_1J] Corrected wording of the second test objective.

CD1009A [Withdrawn] VCS Label: A4_1D
Has one of the objectives of, and thus redundant with, test CD2A31C (now CD30012).
Has one of the objectives of, and thus redundant with, test CD2A21C (now CD30011).

Virtually identical to, and thus redundant with, test CD2D13A.

Virtually identical to, and thus redundant with, test CD3014C.

Check that appropriate entities are visible in an aspect specification.

Replaced by CD30011.

Replaced by CD30012.

Check Legality Rules are enforced when Size is specified by an aspect specification for a type.

Check Legality Rules are enforced when Alignment is specified by an aspect specification for a type.

Check Legality Rules are enforced when Component_Size is specified by an aspect specification.

Check that Component_Size can be specified via an aspect specification.

Merges legacy tests CD1009B and CD2A21C to eliminate duplication (CD1009B has the same objective as one of the objectives of CD2A21C) while adding modern formatting and applicability criteria.

Merges legacy tests CD1009A and CD2A31C to eliminate duplication (CD1009A has the same objective as one of the objectives of CD2A31C) while adding modern formatting and applicability criteria.

Moved error marker from @66 to @67 so it is on the incorrect construct.

Revised to use Possible Error tags, and added error location indicators.
Check that a call on an instance of Unchecked_Deallocation is illegal if the access type has a Storage_Size that is statically zero.

BDD2003 [Modified] VCS Label: A4_1G
Added additional error locations to directly support common error reporting strategies.

CDE0002 [Modified] VCS Label: A4_1E
Changed to use Storage_Size rather than Size so that the static expression rule of 13.14(7.2/3) is not triggered.

CXAC009 [New] VCS Label: A4_1D
Check that Stream_IO.File_Type has Preelaborable_Initialization.

CXAG003 [New] VCS Label: A4_1H
[VCS Label A4_1G] Check that function Ada.Directories.Name_Case_Equivalence exists and that it reflects the handling of file names.
[VCS Label A4_1H] Added “return;” @171; if the file can't be opened, the Read_File trial is complete.

CXB3023 [New] VCS Label: A4_1A
Check that a value of an unchecked union type (defined with pragma Unchecked_Union) can be passed to a C function.

CXB3024 [New] VCS Label: A4_1A
Check that a value of an unchecked union type (defined with aspect Unchecked_Union) can be passed to a C function.

===== Annex C =====

BXC6A03 [Modified] VCS Label: A4_1G
Changed subtest result to require the changes of AI05-0218-1.

===== Annex D =====

CXD2007 [Modified] VCS Label: A4_1C
Added missing N/A => Error tag.

LXD7001 [Modified] VCS Label: A4_1C
Added missing error tag.

LXD7003 [Modified] VCS Label: A4_1C
Added missing error tag.

LXD7004 [Modified] VCS Label: A4_1C
Added missing error tag.

LXD7005 [Modified] VCS Label: A4_1C
Added missing error tag.

LXD7006 [Modified] VCS Label: A4_1C
Added missing error tag.

LXD7007 [Modified] VCS Label: A4_1C
Added missing error tag.

LXD7008 [Modified] VCS Label: A4_1C
   Added missing error tag.

LXD7009 [Modified] VCS Label: A4_1C
   Added location indicators.

CXDB001 [Modified] VCS Label: A4_1C
   Added missing N/A => Error tag.

CXDB002 [Modified] VCS Label: A4_1C
   Added missing N/A => Error tag.

CXDB003 [Modified] VCS Label: A4_1C
   Added missing N/A => Error tag.

CXDB004 [Modified] VCS Label: A4_1C
   Added missing N/A => Error tag.

===== Annex E =====
[None.]

===== Annex F =====
[None.]

===== Annex G =====
[None.]

===== Annex H =====

LXH4001 [Modified] VCS Label: A4_1C
   Corrected format of error tag; added location indicators.

LXH4003 [Modified] VCS Label: A4_1C
   Corrected format of error tag; added location indicators.

LXH4006 [Modified] VCS Label: A4_1C
   Added missing error tags.

LXH4009 [Modified] VCS Label: A4_1C
   Added missing error tags.

LXH4013 [Modified] VCS Label: A4_1C
   Added missing error tags.

LXH4014 [Modified] VCS Label: A4_1C
   Added missing error tags.

===== Support =====

F611A00 [New] VCS Label: A4_1A
   [VCS Label A4_1A] Foundation for tests C611A01, C611A02, and C611A03.
[VCS Label A4_1C] Added additional data for test C611A04. The changes are only needed for that test.

F611B00 [New] VCS Label: A4_1A
Foundation for tests C611B01, C611B02, and C611B03.

GRADE.A [Modified] VCS Label: A4_1G
[VCS Label A4_1C] Allowed manual grading for tests that are missing execution or linking (with no earlier errors). Corrected the result for binder crashes. Added category totals to the summary.
[VCS Label A4_1D] Corrected so that Annex C RQMT failures are properly counted in their own category.
[VCS Label A4_1G] Corrected call @1578 so that the special handling flag rather than the failed flag is used with a manual grading message for a missing error set. (The actual grade was correct, only the message was wrong.)

GRD_DATA.A [Modified] VCS Label: A4_1D
Increased table sizes as the new tests have caused grading the entire ACATS to exceed the old limits.

SPECIAL.A [Modified] VCS Label: A4_1J
[VCS Label A4_1C] Added test BA21005 to the list of optional main subprograms.
[VCS Label A4_1H] Added seven tests to the list of optional main subprograms.
[VCS Label A4_1J] Added more tests to the list of optional main subprograms.

SUMMARY.A [Modified] VCS Label: A4_1C
Corrected identification of main subprograms in modern tests.
Changed the unit name for subunits to be the full expanded name including the parent name.

TCTOUCH.ADA [Modified] VCS Label: A4_1H
Added code so that Touch only reports buffer overflow once. Fixed the Id string.

---

**How to get new and modified tests:**

The official home of the ACATS test suite is the ACAA web site. The site can be accessed at www.ada-auth.org.

The complete set of official new and modified tests are available in the Mod_4_1L.Zip and Tar files on the ACAA web site. As always, when a multi-file test is modified, we provide all of the files, including the unmodified files.

Individual new and modified tests can be retrieved from the ACATS VCS. Simply navigate to the appropriate directory (folder), and select the test you want. Select the revision you want (usually the most recent) and it will be retrieved for you. Tests are stored in directories that correspond to the test’s prefix. For instance, the test CD92001.A will be found in the CD directory. You can also browse the differences between versions of a test. [In case of a conflict, the files in the ACATS VCS should be considered the 'official' versions.]
If you cannot get the files from the official site or you cannot get the files in a format appropriate for your processor, please contact your ACAL.

The baseline ACATS 4.1 is available in various formats on the ACAA web site.

**Summary lists:**

**Withdrawn Tests**

(Refer to the main list for details)

B74001A  
B74001B  
B74203E  
B85002A  
B85003A  
B85003B  
BA1101E  
CD1009A  
CD1009B  
CD1009L  
CD1009M  
CD2A21C  
CD2A31C

**Allowed Modification Tests**

(Refer to the main list for details)

[None in this list.]

**Modified Tests**

(Refer to the main list for details)

B37309B  
B37310B  
B37311A  
B38003D  
B392002  
B392003  
B392004  
B392005  
B393001  
B393007  
B393010  
B3A1001  
B3A1002  
B3A2007  
B3A2008  
B3A2009  
B3A2011  
B3A2012
New Tests

(Refer to the main list for details)

B354002
B35A001
B3A2018
B458001
B480004
B611001
B611002
B611003
B611004
B611005
B611006
B611007
B611008
B611009
B611010
B611011
B611012
B611013
B611014
B611015
## Pending New Tests

(Refer to the main list for details)

<table>
<thead>
<tr>
<th>Code</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>B452001</td>
<td>2019-07-01</td>
</tr>
<tr>
<td>B452002</td>
<td>2019-10-01</td>
</tr>
<tr>
<td>B831006</td>
<td>2019-07-01</td>
</tr>
<tr>
<td>B851007</td>
<td>2019-07-01</td>
</tr>
<tr>
<td>B854002</td>
<td>2019-04-01</td>
</tr>
<tr>
<td>B854003</td>
<td>2019-04-01</td>
</tr>
<tr>
<td>B854004</td>
<td>2019-04-01</td>
</tr>
<tr>
<td>BC40003</td>
<td>2019-04-01</td>
</tr>
<tr>
<td>BC60005</td>
<td>2019-04-01</td>
</tr>
<tr>
<td>C452003</td>
<td>2019-07-01</td>
</tr>
<tr>
<td>C452004</td>
<td>2019-10-01</td>
</tr>
<tr>
<td>C452005</td>
<td>2019-10-01</td>
</tr>
<tr>
<td>C452006</td>
<td>2019-10-01</td>
</tr>
<tr>
<td>C452A01</td>
<td>2019-07-01</td>
</tr>
<tr>
<td>C452A02</td>
<td>2019-07-01</td>
</tr>
<tr>
<td>C452A03</td>
<td>2019-07-01</td>
</tr>
</tbody>
</table>