Appendix B
Parameterization Files

In ACVC 2.2, two methods are used to account for the use of implementation-dependent values in the tests.

For legacy tests, a "macro" substitution technique is used. Legacy tests requiring implementation-specific values contain symbols beginning with the '$' character; for example, the symbol $INTEGER_LAST is used where the code expects the implementation-specific integer literal representing the largest integer. For each implementation, these symbols must be systematically replaced with the appropriate values. A data file, "MACRO.DFS", and an Ada program, "Macrosub", are provided to facilitate this substitution.

For tests written since the introduction of Ada 95, a hierearchy of packages is provided that contain constants and functions that provide the desired implementation-specific values. These packages ("ImpDef" and its children) should be modified for each implementation to provide the needed values.

Information regarding the macro substitution technique is presented in Sections B.1 and B.2. Section B.3 describes the ImpDef package hierarchy.

B.1. Macro Substitution File

The support file "MACRO.DFS" provides substitutions for special symbols that appear in certain ACVC 2.2 tests (indicated by the three-letter file type (extension) ".TST" and listed in Section B.2.). The support program "Macrosub" may be used to insert these implementation-specific values in place of the special symbols in the test. The following excerpt from the file describes the file and its use.

```
-- MACRO.DFS
-- THIS FILE CONTAINS THE MACRO DEFINITIONS USED IN THE ACVC TESTS.
-- THESE DEFINITIONS ARE USED BY THE ACVC TEST PRE-PROCESSOR,
-- MACROSUB. MACROSUB WILL CALCULATE VALUES FOR THOSE MACRO SYMBOLS
-- WHOSE DEFINITIONS DEPEND ON THE VALUE OF MAX_IN_LEN (NAMELY, THE
-- VALUES OF THE MACRO SYMBOLS BIG_ID1, BIG_ID2, BIG_ID3, BIG_ID4,
-- BIG_STRING1, BIG_STRING2, MAX_STRING_LITERAL, BIG_INT_LIT, BIG_REAL_LIT,
-- AND BLANKS).  THEREFORE, ANY VALUES GIVEN IN THIS FILE FOR THOSE
-- MACRO SYMBOLS WILL BE IGNORED BY MACROSUB.

-- NOTE: AS REQUIRED BY THE MACROSUB PROGRAM, THE FIRST MACRO DEFINED
-- IN THIS FILE IS $MAX_IN_LEN.  THE NEXT 5 MACRO DEFINITIONS
-- ARE FOR THOSE MACRO SYMBOLS THAT DEPEND ON THE VALUE OF
-- MAX_IN_LEN.  THESE ARE IN ALPHABETIC ORDER.  FOLLOWING THESE
-- ARE 36 MORE DEFINITIONS, ALSO IN ALPHABETIC ORDER.

-- EACH DEFINITION IS ACCORDING TO THE FOLLOWING FORMAT:

-- A. A NUMBER OF LINES PRECEDED BY THE ADA COMMENT DELIMITER, --.
--    THE FIRST OF THESE LINES CONTAINS THE MACRO SYMBOL AS IT APPEARS
--    IN THE TEST FILES (WITH THE DOLLAR SIGN).  THE NEXT FEW "COMMENT"
--    LINES CONTAIN A DESCRIPTION OF THE VALUE TO BE SUBSTITUTED.
--    THE REMAINING "COMMENT" LINES, THE FIRST OF WHICH BEGINS WITH THE
--    WORDS "USED IN:  " (NO QUOTES), CONTAIN A LIST OF THE TEST FILES
--    (WITHOUT THE .TST EXTENSION) IN WHICH THE MACRO SYMBOL APPEARS.
--    EACH TEST FILE NAME IS PRECEDED BY ONE OR MORE BLANKS.
-- B. A LINE, WITHOUT THE COMMENT DELIMITER, CONSISTING OF THE
--    IDENTIFIER (WITHOUT THE DOLLAR SIGN) OF THE MACRO SYMBOL,
--    FOLLOWED BY A SPACE OR TAB, FOLLOWED BY THE VALUE TO BE
--    SUBSTITUTED.  IN THE DISTRIBUTION FILE, A SAMPLE VALUE IS
--    PROVIDED; THIS VALUE MUST BE REPLACED BY A VALUE APPROPRIATE TO
--    THE IMPLEMENTATION.

-- DEFINITIONS ARE SEPARATED BY ONE OR MORE EMPTY LINES.
-- THE LIST OF DEFINITIONS BEGINS AFTER THE FOLLOWING EMPTY LINE.

-- $MAX_IN_LEN
-- AN INTEGER LITERAL GIVING THE MAXIMUM LENGTH PERMITTED BY THE
-- COMPILER FOR A LINE OF ADA SOURCE CODE (NOT INCLUDING AN END-OF-LINE
-- CHARACTER).
-- USED IN:  A26007A
MAX_IN_LEN                60
```

B.2. Macro Substitution Tests

The following test files contain the special symbols used for substituting implementation-specific values, as described in Section B.1. This list also appears in the ACVC 2.2 "support" directory as "TSTTESTS.DAT".

A26007A.TST	BD2C01D.TST	C35503F.TST	CE2102C.TST
AD8011A.TST	BD2C02A.TST	C45231D.TST	CE2102H.TST
B22001A.TST	BD2C03A.TST	C4A007A.TST	CE2103A.TST
B22001B.TST	BD4006A.TST	C87B62D.TST	CE2103B.TST
B22001C.TST	BD8001A.TST	C96005B.TST	CE2120A.TST
B22001D.TST	BD8002A.TST	CC1225A.TST	CE2120B.TST
B22001E.TST	BD8003A.TST	CD1009K.TST	CE2203A.TST
B22001F.TST	BD8004A.TST	CD1009T.TST	CE2403A.TST
B22001G.TST	BD8004B.TST	CD1009U.TST	CE3002B.TST
B22001I.TST	BD8004C.TST	CD1C03E.TST	CE3002C.TST
B22001J.TST	C23003A.TST	CD1C06A.TST	CE3102B.TST
B22001K.TST	C23003B.TST	CD2A83C.TST	CE3107A.TST
B22001L.TST	C23003G.TST	CD2A91C.TST	CE3119A.TST
B22001M.TST	C23003I.TST	CD2C11A.TST	CE3304A.TST
B22001N.TST	C35502D.TST	CD2C11D.TST	SPPRT13S.TST
B54B01B.TST	C35502F.TST	CD4041A.TST	
BD2A02A.TST	C35503D.TST	CD7101G.TST	

B.3. Package ImpDef and Its Children

The package ImpDef (for "Implementation Definitions") provides constants and functions for producing implementation-specific values required by certain test programs. This package resides in the file "ImpDef.a" in the "support" directory. Four child packages are also included in the "support" directory, each providing the means for producing implementation-specific values required by certain test programs for a particular Specialized Needs Annex. These packages have names of the form ImpDef.Annex_X, and reside in files with names of the form "ImpDefX.a", where 'X' is replaced by the letter designating the relevant Annex.

The ImpDef package and each of its children should be modified for each implementation as described in the source code. The following excerpt from the "ImpDef.a" file includes comments showing where such modifications are expected.

```
package ImpDef is

--=====-=====-===========-=====-=====-=====-=====-=====-=====-=====-========--

   -- The following boolean constants indicate whether this validation will
   -- include any of annexes C-H. The values of these booleans affect the
   -- behavior of the test result reporting software.
   --
   --    True  means the associated annex IS included in the validation.
   --    False means the associated annex is NOT included.

   Validating_Annex_C : constant Boolean := False;
   --                                       ^^^^^ --- MODIFY HERE AS NEEDED

   Validating_Annex_D : constant Boolean := False;
   --                                       ^^^^^ --- MODIFY HERE AS NEEDED

   Validating_Annex_E : constant Boolean := False;
   --                                       ^^^^^ --- MODIFY HERE AS NEEDED

   Validating_Annex_F : constant Boolean := False;
   --                                       ^^^^^ --- MODIFY HERE AS NEEDED

   Validating_Annex_G : constant Boolean := False;
   --                                       ^^^^^ --- MODIFY HERE AS NEEDED

   Validating_Annex_H : constant Boolean := False;
   --                                       ^^^^^ --- MODIFY HERE AS NEEDED

--=====-=====-===========-=====-=====-=====-=====-=====-=====-=====-========--

   -- This is the minimum time required to allow another task to get
   -- control.  It is expected that the task is on the Ready queue.
   -- A duration of 0.0 would normally be sufficient but some number
   -- greater than that is expected.

   Minimum_Task_Switch : constant Duration := 0.1;
   --                                         ^^^ --- MODIFY HERE AS NEEDED
```