

①

AD-A257 430



TASK: UT20
CDRL: 04014
12 June 1992

UT20—Ada PCTE Binding Version Description Document Version 0.1

Informal Technical Data

S DTIC
ELECTE
OCT 28 1992
C **D**

424460



92-28326

20 pgs

Approved for public release
Distribution Unlimited

STARS-TC-04014/001/00
12 June 1992

93 10 27 114

TASK: UT20
CDRL: 04014
12 June 1992

VERSION DESCRIPTION DOCUMENT
For The
SOFTWARE TECHNOLOGY FOR ADAPTABLE, RELIABLE SYSTEMS
(STARS)

Ada PCTE Binding (AdaPCTE)
Version 0.1
SunOS Implementation

STARS-TC-04014/001/00
12 June 1992

Data Type: A005, Informal Technical Data

CONTRACT NO. F19628-88-D-0031
Delivery Order 0008

Prepared for:

Electronic Systems Division
Air Force Systems Command, USAF
Hanscom AFB, MA 01731-5000

Prepared by:

Paramax Systems Corporation
Tactical Systems
12010 Sunrise Valley Drive
Reston, VA 22091

DTIC QUALITY INSPECTED 2

Accession For	
NTIS CRAB	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

TASK: UT20
CDRL: 04014
12 June 1992

VERSION DESCRIPTION DOCUMENT
Ada PCTE Binding (AdaPCTE)
Version 0.1
SunOS Implementation

Principal Author(s):

Robert C. Smith, Paramax, Valley Forge Labs

Date

Michael J. Horton, Paramax, Valley Forge Labs

Date

Approvals:

Thomas E. Shields

6/17/92

Task Manager *Dr. Thomas E. Shields*

Date

(Signatures on File)

TASK: UT20
CDRL: 04014
12 June 1992

VERSION DESCRIPTION DOCUMENT
Ada PCTE Binding (AdaPCTE)
Version 0.1
SunOS Implementation

Change Record:

<i>Data ID</i>	<i>Description of Change</i>	<i>Date</i>	<i>Approval</i>
STARS-TC-04014/001/00	Original Issue	12 June 1992	<i>on file</i>

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE 12 June 1992	3. REPORT TYPE AND DATES COVERED Version Description Document
----------------------------------	--------------------------------	--

4. TITLE AND SUBTITLE Ada PCTE Binding (AdaPCTE) Version 0.1	5. FUNDING NUMBERS F19628-88-D-0031
--	--

6. AUTHOR(S) Paramax Systems Corporation	
---	--

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Paramax System Corporation 12010 Sunrise Valley Drive Reston, VA 22091	8. PERFORMING ORGANIZATION REPORT NUMBER STARS-SC-04014/001/00
--	---

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Department of the Air Force Headquarters Electronic Systems Division Hanscom AFB, MA 01731-5000	10. SPONSORING / MONITORING AGENCY REPORT NUMBER 04014
--	---

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT Distribution "A"	12b. DISTRIBUTION CODE
--	------------------------

13. ABSTRACT (Maximum 200 words)

The Ada Portable Common Tool Environment (PCTE) binding (AdaPCTE) provides Ada applications access to a PCTE object base as defined by the European Computer Manufacturers Association (ECMA) Ada PCTE specification (Standard ECMA-162 Ada Language Binding, December 1991). This "alpha" release provides a minimal set of interfaces to permit Ada developers to experiment with and evaluate PCTE for use in future Ada applications. It is expected that later releases will complete the binding.

14. SUBJECT TERMS PCTE, Ada Bindings	15. NUMBER OF PAGES 14
---	---------------------------

	16. PRICE CODE
--	----------------

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT SAR
---	--	---	-----------------------------------

Contents

1 SCOPE	1
1.1 Identification	1
1.2 System Overview	1
2 RELATED SOFTWARE	1
3 VERSION DESCRIPTION	1
3.1 Inventory of Contents	1
3.1.1 Directory: <i>adapcte/code</i>	1
3.1.2 Sub-directory: <i>adapcte/code/C</i>	2
3.2 Adaptation Data	2
3.2.1 Operating Environment	2
3.2.2 Development Environment	2
3.2.3 Configuration-unique Data	2
3.3 Interface Compatibility	2
3.4 Installation Instructions	3
3.4.1 Build Procedure	3
4 USER FEEDBACK	3
5 NOTES	3
A Appendix: Inventory of Contents	7
B Appendix: Build Scripts	9
B.1 Script: <i>Build_AdaPCTE.var</i>	9
B.2 Script: <i>Build_AdaPCTE.csh</i>	13

1 SCOPE

1.1 Identification

Version Description Document,
Ada PCTE Binding (AdaPCTE),
Version 0.1,
SunOS Implementation

1.2 System Overview

The Ada Portable Common Tool Environment (PCTE) binding (AdaPCTE) provides Ada applications access to a PCTE object base as defined by the European Computer Manufacturers Association (ECMA) Ada PCTE specification (Standard ECMA-162 Ada Language Binding, December 1991). This "alpha" release provides a minimal set of interfaces to permit Ada developers to experiment with and evaluate PCTE for use in future Ada applications. It is expected that later releases will complete the binding.

2 RELATED SOFTWARE

Since no conforming implementations of ECMA PCTE exist as defined in Standard ECMA-149, AdaPCTE is implemented on GIE Emeraude's PCTE V12.2 Fix 7. Because only a subset of the ECMA PCTE Ada specification has been implemented for the 0.1 release, and because ECMA PCTE functionality differs somewhat from Emeraude PCTE functionality, the complete functionality of Emeraude PCTE is not available to Ada applications using these bindings.

3 VERSION DESCRIPTION

3.1 Inventory of Contents

The AdaPCTE distribution is structured as shown below. The top-level directory *adapcte* includes PostScript (*VDDadapcte.ps*) and clear ASCII text (*VDDadapcte.tty*) versions of this document, along with a complete directory listing of the distribution (*Contents.tty*, reproduced herein as **Appendix A**).

3.1.1 Directory: *adapcte/code*

The *adapcte/code* directory contains the Ada source files for the Ada binding to PCTE and the UNIX C-shell script *Build_AdaPCTE.csh*. *Build_AdaPCTE.csh* can be used to build the entire AdaPCTE Binding using the SunAda 1.0 Development System. No provisions within the build script have been made for installing the bindings in the PCTE object base.

Applications being developed on these bindings are expected to be developed within a UNIX environment and executable code files may be installed by the user in the PCTE object base (but are not required to be installed in the object base). The build script is reproduced herein as **Appendix B.2**.

3.1.2 Sub-directory: *adapcte/code/C*

This directory contains a small C file, *util.c*, containing utility routines used by AdaPCTE. The build script compiles this file in the target directory, and inserts a link directive in the Ada library, so users need not add *util.o* to link commands for any applications developed on these bindings.

3.2 Adaptation Data

3.2.1 Operating Environment

Sun-4 Workstations

SunOS, Version 4.1.2

Emeraude PCTE V12.2 Fix 7

3.2.2 Development Environment

Sun-4 Workstations

SunOS, Version 4.1.2

Emeraude PCTE V12.2 Fix 7

SunAda 1.0

C compiler

3.2.3 Configuration-unique Data

3.3 Interface Compatibility

AdaPCTE uses the recently adopted standard ECMA-162 for the Ada binding specification. Because no ECMA PCTE implementation is available, AdaPCTE is bound to GIE Emeraude's PCTE 1.5 V12.2 implementation written in "C". As a result, the AdaPCTE specification contains some minor modifications to ECMA-162. The exact specification of the implemented binding can be found in the Ada package specifications located in the directory */adapcte/code* in this delivery.

3.4 Installation Instructions

File *adapcte/code/Build_AdaPCTE.csh* is an executable UNIX C-shell script, which can be used interactively to build the AdaPCTE Binding from the Ada source code, using the SunAda 1.0 system. It ensures that library dependencies are established correctly, making it unnecessary for the installer to perform these operations manually.

3.4.1 Build Procedure

1. (*OPTIONAL*) - To prevent interactive prompting when executing the script, uncomment and edit the environment variables at the beginning of file *code/Build_AdaPCTE.var* (see **Appendix B.1**) to reflect the actual operating environment. The following environment variables must be modified:

AdaPCTE - identifies the full pathname of the directory into which the AdaPCTE distribution has been loaded (e.g., /local/adapcte);

COMPILERNAME - identifies the name of the compiler to be used;

COMPVERSION - identifies the compiler version;

COMPILERPATH - identifies the full pathname of the directory containing the SunAda compilation system (e.g., /local/sunada1.0);

TARGET - identifies a *Build* directory to be used for building the software.

2. Execute *Build_AdaPCTE.csh*, providing configuration information when prompted by the script.

4 USER FEEDBACK

This version of AdaPCTE is considered an "alpha" release. The primary purpose of the release is to encourage experimentation with the software and to solicit feedback from the Ada and PCTE user communities. Thus, we would greatly appreciate your comments, suggestions, and criticisms.

5 NOTES

The full set of PCTE path names as described in the ECMA PCTE Abstract Specification (149) has not been implemented for this release. The following characters "_", ".", "~", and "/" plus alphanumeric characters are valid characters in AdaPCTE path names. The following are examples of valid AdaPCTE path names:

_/sun4.tools

~/history.e

AdaPCTE Version 0.1 has not implemented all the interfaces defined in ECMA-162. The following describes which interfaces are implemented in Version 0.1 including any limitations.

Package Pcte

Package Sequence

- function get
- procedure put
- procedure delete
- procedure copy
- function length_of
- function index_of
- function equal
- procedure normalize

Package Reference

- These procedures use a limited form of path names as defined
- in the abstract spec. You can use ~, _, ., .., / plus ascii
- characters
- function get_path
- procedure set_absolute
- procedure set_relative
- procedure unset
- New operations added by VFL
- function get_reference_id
- procedure set_reference_id

Package Pcte_contents

- This package is only implemented for files; no pipes or devices
- procedure close
- function get_position
- procedure open
- function read
- procedure seek
- procedure set_position
- procedure set_properties
- procedure write
- New operations added by VFL
- procedure standard_input
- procedure standard_output
- procedure standard_error
- function end_of_contents
- procedure write_s
- (writes a string)
- procedure read_s

-- (reads a string)

Package Pcte_error

procedure set
procedure unset
procedure set_will_raise
procedure set_will_record
function will_raise
function will_record
function last_error

Package Pcte_object

procedure create
-- can not specify another volume
procedure delete
procedure get_attribute
-- for boolean, integer, natural and string types only
procedure get_several_attributes
-- for boolean, integer, natural and string types only
function get_type
procedure list_all_links
-- does not support EXTERNAL extents
-- does not support COMPOSITE scopes
-- ignores links parameter
-- none of the other 8 procedure variations of
-- object_list_links is supported

Package Pcte_process

procedure create_and_start
-- no process objects created; just fire up a process
-- local execution site only
procedure set_working_schema
-- for current process only
procedure wait_for_any_child
procedure wait_for_child

package Pcte_sds

procedure get_link_type_properties
procedure get_object_type_properties
function get_type_name
-- ignores any sds param value other than IN_WORKING_SCHEMA

package Pcte_link

procedure get_attribute
-- for boolean, integer, natural and string types only

12 June 1992

STARS-TC-04014/001/00

procedure get_several_attributes

-- for boolean, integer, natural and string types only

A Appendix: Inventory of Contents

NOTE: "*" identifies executables; "/" identifies directories.

adapcte:

**Contents.tty
VDDadapcte.ps
VDDadapcte.tty
code/**

adapcte/code:

**Build_AdaPCTE.csh*
Build_AdaPCTE.var
C/
Pcte.a
Pcte_accounting.a
Pcte_activity.a
Pcte_audit.a
Pcte_b.a
Pcte_contents.a
Pcte_contents_b.a
Pcte_discretionary.a
Pcte_discretionary_b.a
Pcte_error.a
Pcte_error_b.a
Pcte_limit.a
Pcte_mandatory.a
Pcte_mandatory_b.a
Pcte_message.a
Pcte_notify.a
Pcte_object_b.a
Pcte_oms.a
Pcte_oms_b.a
Pcte_process.a
Pcte_process_b.a
Pcte_queue.a
Pcte_replicated_object.a
Pcte_sds.a
Pcte_sds_b.a
Pcte_time.a
Pcte_vol_dev_archi.a
Pcte_workstation.a
emer_conversion.a
error.a
error_b.a**

12 June 1992

STARS-TC-04014/001/00

errors_c.a
pcte_1_5_int.a
pcte_1_5_support.a
pcte_1_5_support_b.a

adapcte/code/C:
util.c

B Appendix: Build Scripts**B.1 Script: *Build_AdaPCTE.var***

```
1 #
2 # Uncomment and edit these lines if you do not want to
3 # be prompted for the environment variables
4 #
5 setenv ADAPCTE      /local/adapcte
6 setenv COMPILERNAME sunada      # set to sunada
7 setenv COMPVERSION SunAda1.0  # e.g. SunAda1.0; not tested on SunAda1.1
8 setenv COMPILERPATH /local/SunAda
9 setenv TARGET      $ADAPCTE/Build_$COMPVERSION
10
11 #
12 # Define the location of the RGB source code directories.
13 #
14
15 if ( ! $?ADAPCTE ) then
16     echo ""
17     echo "Specify path to top level Ada PCTE directory "
18     echo "(e.g. /local/adapcte ) "
19     echo ""
20     echo -n "          ADAPCTE = "
21     setenv ADAPCTE $<
22     echo ""
23 endif
24 if ( ! -e $ADAPCTE ) then
25     echo ""
26     echo "*** $ADAPCTE does not exist ***"
27     echo "*** Script aborted ***"
28     echo ""
29     unsetenv ADAPCTE
30     exit -1
31 endif
32
33
34
35 #
36 # Define C Language compilation variable
37 #
38 setenv CC          " cc -g -c "
39
40
41
```

```
42 #
43 # Determine the Ada compilation system to use
44 #
45 #
46 # Establish a path to the SunAda compilation system
47 #
48 if ( ! $?COMPILERNAME || ! $?COMPVERSION || ! $?COMPILERPATH ) then
49     echo ""
50     echo "Please select your compiler name: [sunada] "
51     echo ""
52     echo -n " COMPILERNAME = "
53     setenv COMPILERNAME $<
54     echo ""
55     switch ($COMPILERNAME)
56     case Vads:
57     case VADS:
58     case vads:
59         echo -n "Are you building with VADS Version 6.0.3? [y,n](n) "
60         set COMPVERSION = $<
61         echo ""
62         switch ($COMPVERSION)
63         case Y:
64         case y:
65             set COMPVERSION = Vads603
66             breaksw
67         case N:
68         case n:
69         default:
70             set COMPVERSION = Vads
71             echo ""
72             echo "Warning! Software not tested under your version of the VADS com
73 piler."
74             breaksw
75         endsw
76     breaksw
77     case SunAda:
78     case Sunada:
79     case sunada:
80         echo -n "Which version of SunAda are you using? [0,1](0) "
81         set COMPVERSION = $<
82         echo ""
83         switch ($COMPVERSION)
84         case 1:
85             set COMPVERSION = SunAda1.1
86             echo "Warning! Software not tested under your version of the SunAda compiler."
```



```
87         breaksw
88     case 0:
89     default:
90         set COMPVERSION = SunAda1.0
91         breaksw
92     endsw
93 breaksw
94 default:
95     echo ""
96     echo "You must specify a compiler name."
97     echo ""
98     unsetenv COMPVERSION
99     exit -1
100    breaksw
101    endsw
102
103    echo ""
104    echo "Specify path to the compiler (e.g. /local/SunAda)"
105    echo ""
106    echo -n "    COMPILERPATH = "
107    setenv COMPILERPATH $<
108    if ( ( $COMPILERPATH == ) || ( ! -e $COMPILERPATH/bin/ada ) ) then
109        echo ""
110        echo "*** Cannot find Ada compiler in $COMPILERPATH/bin ***"
111        echo "*** Script aborted ***"
112        echo ""
113        unsetenv COMPILERPATH
114        exit -1
115    endif
116 endif
117 if ( -e $COMPILERPATH/bin/ada ) then
118     if ( $COMPILERNAME == "sunada" || $COMPILERNAME == "vads" ) then
119         setenv COMPILERBIN $COMPILERPATH/bin
120         setenv COMPILE "$COMPILERBIN/ada -v -00 "
121         setenv LINK "$COMPILERBIN/a.ld "
122     endif
123 else
124     echo ""
125     echo "*** Cannot find $COMPILERPATH/bin/ada ***"
126     echo "*** Script aborted ***"
127     echo ""
128     unsetenv COMPILERPATH
129     exit -1
130 endif
131
```

```
132
133 #
134 # Define the Destination of the ADAPCTE build
135 #   where TARGET = path to build destination (e.g. $ADAPCTE/Build_SunAda1.0)
136 #
137 if ( ! $?TARGET ) then
138     echo ""
139     echo "Specify the path to the TARGET directory "
140     echo "(Defaults to $ADAPCTE/Build_${COMPVERSION}) "
141     echo ""
142     echo -n "          TARGET = "
143     setenv TEMP $<
144     echo ""
145     if ( $TEMP == ) then # check for null entry
146         setenv TARGET $ADAPCTE/Build_${COMPVERSION}
147         unsetenv TEMP
148     else
149         setenv TARGET $TEMP
150         unsetenv TEMP
151     endif
152 endif
153
154 echo ""
155 echo "          TARGET = $TARGET"
156 echo ""
157 echo "          ADAPCTE = $ADAPCTE"
158 echo ""
159 echo "  COMPILERNAME = $COMPILERNAME"
160 echo "  COMPVERSION  = $COMPVERSION"
161 echo "  COMPILERPATH = $COMPILERPATH"
162 echo "          COMPILE = $COMPILE"
163 echo "          LINK    = $LINK"
164
165 #
166 # Create the directories for the build
167 #
168 if ( ! -d $TARGET ) mkdir $TARGET
169
170
171
```

B.2 Script: *Build_AdaPCTE.csh*

```
1  #! /bin/csh -f
2  echo ""
3  echo "Defining installation-dependent variables"
4  echo ""
5  source Build_AdaPCTE.var
6
7  echo ""
8  echo "Building Ada libraries for the Ada Bindings to PCTE"
9  echo ""
10
11 if ! -e $TARGET mkdir $TARGET
12 if ! -e $TARGET/C mkdir $TARGET/C
13
14 cd $TARGET
15
16 if ( ( $COMPILERNAME == "vads" ) || ( $COMPILERNAME == "sunada" ) ) then
17     if ( ! -e ada.lib ) then
18         $COMPILERBIN/a.mklib -f $TARGET $COMPILERPATH/verdirlib
19         $COMPILERBIN/a.info -a WITH1 $TARGET/util.o
20     endif
21 endif
22
23 echo ""
24 echo "Creating source code links in $ADAPCTE/code"
25 echo ""
26 foreach file ($ADAPCTE/code/*.a)
27     if ( ! -e ${file:t} ) ln -s $file ${file:t}
28 end
29
30 foreach file ($ADAPCTE/code/C/*)
31     if ( ! -e ${file:t} ) ln -s $file ${file:t}
32 end
33
34 rm -rf LOGadapcte
35
36 echo ""
37 echo "Compiling the Ada PCTE binding source"
38 echo ""
39
40 $COMPILE Pcte_error.a           >>& LOGadapcte
41 $COMPILE Pcte.a                 >>& LOGadapcte
42 $COMPILE Pcte_contents.a       >>& LOGadapcte
43 $COMPILE Pcte_replicated_object.a >>& LOGadapcte
```

```
44 $COMPILE Pcte_message.a          >>& LOGadapcte
45 $COMPILE Pcte_error_b.a          >>& LOGadapcte
46 $COMPILE Pcte_notify.a           >>& LOGadapcte
47 $COMPILE Pcte_discretionary.a     >>& LOGadapcte
48 $COMPILE Pcte_mandatory.a         >>& LOGadapcte
49 $COMPILE Pcte_audit.a             >>& LOGadapcte
50 $COMPILE Pcte_mandatory_b.a       >>& LOGadapcte
51 $COMPILE Pcte_workstation.a       >>& LOGadapcte
52 $COMPILE Pcte_discretionary_b.a   >>& LOGadapcte
53 $COMPILE Pcte_process.a           >>& LOGadapcte
54 $COMPILE emer_conversion.a        >>& LOGadapcte
55 $COMPILE Pcte_vol_dev_archi.a     >>& LOGadapcte
56 $COMPILE errors_c.a              >>& LOGadapcte
57 $COMPILE error.a                 >>& LOGadapcte
58 $COMPILE error_b.a               >>& LOGadapcte
59 $COMPILE pcte_1_5_int.a           >>& LOGadapcte
60 $COMPILE pcte_1_5_support.a       >>& LOGadapcte
61 $COMPILE pcte_1_5_support_b.a     >>& LOGadapcte
62 $COMPILE Pcte_process_b.a         >>& LOGadapcte
63 $COMPILE Pcte_contents_b.a       >>& LOGadapcte
64 $COMPILE Pcte_b.a                 >>& LOGadapcte
65 $COMPILE Pcte_oms.a               >>& LOGadapcte
66 $COMPILE Pcte_object_b.a          >>& LOGadapcte
67 $COMPILE Pcte_oms_b.a             >>& LOGadapcte
68 $COMPILE Pcte_time.a              >>& LOGadapcte
69 $COMPILE Pcte_sds.a               >>& LOGadapcte
70 $COMPILE Pcte_sds_b.a             >>& LOGadapcte
71 $COMPILE Pcte_queue.a             >>& LOGadapcte
72 $COMPILE Pcte_accounting.a        >>& LOGadapcte
73 $COMPILE Pcte_activity.a          >>& LOGadapcte
74 $COMPILE Pcte_limit.a             >>& LOGadapcte
75
76
77 echo ""
78 echo "Compiling the C code"
79 echo ""
80 $CC util.c                        >>& LOGadapcte
81
82 echo ""
83 echo "Compilation Complete"
```