AD-A257 430

TASK: UT20
CDRL: 04014
12 June 1992

UT20—Ada PCTE
Binding Version
Description Document
Version 0.1

Informal Technical Data

STARS-TC-04014/001/00
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
VERSION DESCRIPTION DOCUMENT
Ada PCTE Binding (AdaPCTE)
Version 0.1
SunOS Implementation

Principal Author(s):

Robert C. Smith, Paramaz, Valley Forge Labs

Michael J. Horton, Paramaz, Valley Forge Labs

Approvals:

Task Manager Dr. Thomas E. Shields

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

Change Record:

<table>
<thead>
<tr>
<th>Data ID</th>
<th>Description of Change</th>
<th>Date</th>
<th>Approval</th>
</tr>
</thead>
<tbody>
<tr>
<td>STARS-TC-04014/001/00</td>
<td>Original Issue</td>
<td>12 June 1992</td>
<td>on file</td>
</tr>
</tbody>
</table>
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.
Contents

1 SCOPE
  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: adapcte/code ......................... 1
    3.1.2 Sub-directory: adapcte/code/C ................. 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: Build_AdaPCTE.var ......................... 9
  B.2 Script: Build_AdaPCTE.csh ....................... 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: adapte/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 /adapte/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
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
error_conversion.a
error.a
error_b.a
errors.c.a
pcte_1_5_int.a
pcte_1_5_support.a
pcte_1_5_support_b.a

adapcts/code/C:
util.c
B Appendix: Build Scripts

B.1 Script: Build_AdaPCTE.var

```bash
# Uncomment and edit these lines if you do not want to
# be prompted for the environment variables
#
setenv ADAPCTE /local/adapcte
setenv COMPILERNAME sunada # set to sunada
setenv COMPVERSION SunAda1.0 # e.g. SunAda1.0; not tested on SunAda1.1
setenv COMPILERPATH /local/SunAda
setenv TARGET $ADAPCTE/Build_$COMPVERSION
#
# Define the location of the RGB source code directories.
#
if ( ! $?ADAPCTE ) then
  echo ""
  echo "Specify path to top level Ada PCTE directory "
  echo "(e.g. /local/adapcte )"
  echo ""
  echo -n "ADAPCTE = "
  setenv ADAPCTE "<
  echo ""
endif
if ( ! -e $ADAPCTE ) then
  echo ""
  echo "** $ADAPCTE does not exist **"
  echo "** Script aborted **"
  echo ""
  unsetenv ADAPCTE
  exit -1
endif
#
# Define C Language compilation variable
#
setenv CC " cc -g -c "
```

Page 9
Determine the Ada compilation system to use

Establish a path to the SunAda compilation system

if ( ! $?COMPILERNAME || ! $?COMPERSION || ! $?COMPILERPATH ) then

echo 

echo "Please select your compiler name: [sunada] "

echo ""

echo -n " COMPILERNAME = "

setenv COMPILERNAME $<

echo ""

switch ($COMPILERNAME)

case Vads:

case VADS:

case vads:

  echo -n "Are you building with VADS Version 6.0.3? [y,n](n) "

  set COMPERSION = $<

  echo ""

  switch ($COMPERSION)

    case Y:

    case y:

      set COMPERSION = Vads603

    breaksw

    case N:

    case n:

    default:

      set COMPERSION = Vads

    echo ""

    echo "Warning! Software not tested under your version of the VADS compiler."

    breaksw

    breaksw

  case SunAda:

  case Sunada:

  case sunada:

    echo -n "Which version of SunAda are you using? [0,1](0) "

    set COMPERSION = $<

    echo ""

    switch ($COMPERSION)

    case 1:

      set COMPERSION = SunAda1.1

      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 $COMPILERPATH/bin
120              setenv COMPILER "$COMPILERBIN/ada -v -00 "
121              setenv LINK "$COMPILERBIN/a.1d "
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
# Define the Destination of the ADAPCTE build
# where TARGET = path to build destination (e.g. $ADAPCTE/Build_SunAda1.0)

if ( ! $?TARGET ) then
  echo ""
  echo "Specify the path to the TARGET directory"
  echo "(Defaults to $ADAPCTE/Build_${COMPVERSION})"
  echo ""
  echo "n" TARGET = "
  setenv TEMP $<
  echo ""
  if ( $TEMP == ) then # check for null entry
    setenv TARGET $ADAPCTE/Build_${COMPVERSION}
    unsetenv TEMP
  else
    setenv TARGET $TEMP
    unsetenv TEMP
  endif
endif
endif

# Create the directories for the build

if ( ! -d $TARGET ) mkdir $TARGET
B.2 Script: Build_AdaPCTE.csh

```bash
#!/bin/csh -f

# V

# Defining installation-dependent variables

source Build_AdaPCTE.var

# Building Ada libraries for the Ada Bindings to PCTE

if ! -e $TARGET mkdir $TARGET
if ! -e $TARGET/C mkdir $TARGET/C

cd $TARGET

if ( ($COMPILERNAME == "vads" ) || ($COMPILERNAME == "sunada" ) ) then
    if ( ! -e ada.lib ) then
        $COMPILERBIN/a.mklib -f $TARGET $COMPILERPATH/verdizlib
    $COMPILERBIN/a.info -a WITH1 $TARGET/util.o
    endif
endif

# Creating source code links in $ADAPCTE/code

foreach file ($ADAPCTE/code/*.a)
    if (! -e ${file:t} ) ln -s $file ${file:t}
end

foreach file ($ADAPCTE/code/C/*)
    if (! -e ${file:t} ) ln -s $file ${file:t}
end

rm -rf LOGadapcte

# Compiling the Ada PCTE binding source

$COMPILERE Pcte_error.a
$COMPILERE Pcte.a
$COMPILERE Pcte_contents.a
$COMPILERE Pcte_replicated_object.a
```

Page 13
44 $COMPILE Pcte_message.a
45 $COMPILE Pcte_error_b.a
46 $COMPILE Pcte_notify.a
47 $COMPILE Pcte_discretionary.a
48 $COMPILE Pcte_mandatory.a
49 $COMPILE Pcte_audit.a
50 $COMPILE Pcte_mandatory_b.a
51 $COMPILE Pcte_workstation.a
52 $COMPILE Pcte_discretionary_b.a
53 $COMPILE Pcte_process.a
54 $COMPILE emer_conversion.a
55 $COMPILE Pcte_vol_dev_archi.a
56 $COMPILE errors_c.a
57 $COMPILE error.a
58 $COMPILE error_b.a
59 $COMPILE pchte_1_5_int.a
60 $COMPILE pchte_1_5_support.a
61 $COMPILE pchte_1_5_support_b.a
62 $COMPILE Pcte_process_b.a
63 $COMPILE Pcte_contents_b.a
64 $COMPILE Pcte_b.a
65 $COMPILE Pcte_oms.a
66 $COMPILE Pcte_object_b.a
67 $COMPILE Pcte_oms_b.a
68 $COMPILE Pcte_time.a
69 $COMPILE Pcte_sds.a
70 $COMPILE Pcte_sds_b.a
71 $COMPILE Pcte_queue.a
72 $COMPILE Pcte_accounting.a
73 $COMPILE Pcte_activity.a
74 $COMPILE Pcte_limit.a
75
76
77 echo ""
78 echo "Compiling the C code"
79 echo ""
80 $CC util.c
81
82 echo ""
83 echo "Compilation Complete"