Best Available Copy
PCTE Browser Tool
Version Description Document
Version 1.0

Informal Technical Data

This document has been approved for public release and sale; its distribution is unlimited.

94-31243

STARS-UC-05203/002/00
30 November 1992
VERSION DESCRIPTION DOCUMENT

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

PCTE Browser Tool
Version 1.0
SunOS Implementation

STARS-UC-05203/002/00
30 November 1992

Data Type: A005, Informal Technical Data

CONTRACT NO. F19628-88-D-0031
Delivery Order 0011

Prepared for:
Electronic Systems Center
Air Force Systems Command, USAF
Hanscom AFB, MA 01731-5000

Prepared by:
Paramax Systems Corporation
12010 Sunrise Valley Drive
Reston, VA 22091
VERSION DESCRIPTION DOCUMENT
PCTE Browser Tool
Version 1.0
SunOS Implementation

Principal Author(s):

Michael J. Horton, Valley Forge Labs

Approvals:

Task Manager Dr. Paul Orgren

(Signatures on File)
VERSION DESCRIPTION DOCUMENT
PCTE Browser Tool
Version 1.0
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-UC-05203/002/00</td>
<td>Successor volume: Upgrade for software version 1.0</td>
<td>30 November 1992</td>
<td>on file</td>
</tr>
<tr>
<td>STARS-TC-04014/002/00</td>
<td>Original Issue: Describes software version 0.1</td>
<td>12 June 1992</td>
<td>on file</td>
</tr>
</tbody>
</table>
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 Subdirectory: pbt/code ........ 2
      3.1.2 Subdirectory: pbt/bin .......... 2
      3.1.3 Subdirectory: pbt/X-Resources ... 2
         3.1.3.1 Subdirectory: pbt/X-Resources/bitmaps   2
   3.2 Changes Installed .............. 2
      3.2.1 Release 1.0 .................. 2
         3.2.1.1 General Changes ........... 2
         3.2.1.2 Bug Fixes ................. 3
         3.2.1.3 New Features ............. 3
   3.3 Adaptation Data ............... 6
      3.3.1 Operating Environment ....... 6
      3.3.2 Development Environment ..... 6
      3.3.3 Configuration-Unique Data .... 7
   3.4 Interface Compatibility ........ 7
   3.5 Installation Instructions ...... 7
      3.5.1 Build Procedure ............. 8
      3.5.2 Executable Installation Procedure 9
      3.5.3 Installing the X Resource Files 10
      3.5.4 Installing the PBT-Oriented SDS 10
   3.6 Potential Problems ............. 10
   3.7 Enhancements .................. 11
4 USER FEEDBACK

A Appendix: Inventory of Contents

B Appendix: Unix Installation Scripts

<table>
<thead>
<tr>
<th>Appendix</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Inventory of Contents</td>
<td>12</td>
</tr>
<tr>
<td>B</td>
<td>Unix Installation Scripts</td>
<td>14</td>
</tr>
<tr>
<td>B.1</td>
<td>File: Build.PBT.var</td>
<td>14</td>
</tr>
<tr>
<td>B.2</td>
<td>Script: Build.PBT.csh</td>
<td>19</td>
</tr>
</tbody>
</table>
1 SCOPE

1.1 Identification

Version Description Document, PCTE Browser Tool (PBT), Version 1.0, SunOS Implementation

1.2 System Overview

The PCTE Browser Tool (PBT) is designed to graphically display parts of a PCTE object base. Selected graphs of objects in the object base and the relationships amongst these objects are displayed at the PBT user's request. The PBT is intended to complement text-oriented commands such as obj_list_links and obj_list_attr that are included with the Emeraude PCTE V12 release—commands intended to be invoked from the text-oriented esh command shell.

2 RELATED SOFTWARE

The PBT is an instance of the Reusable Graphical Browser (RGB), a generic graphical browser for the display of networks of nodes and arcs. In the case of the PBT, the nodes displayed by the RGB are PCTE objects, and the arcs are PCTE links. PBT version 1.0 was developed using RGB version 1.0.

The PBT is an X Window System application requiring the installation of X11, Release 4 (X11R4). It is a Motif-oriented application making use of the SA-Motif Ada bindings, version 1.0, a commercial product of Systems Engineering Research Corporation (SERC).

The PBT is ultimately intended for use in an ECMA PCTE environment and has been implemented using the ECMA-162 Ada programming bindings to PCTE. However, in the absence of a conforming ECMA PCTE implementation, the PBT has been built on top of the Emeraude V12 PCTE implementation, using the subset implementation of the ECMA Ada binding developed by Paramax STARS (version 0.2).

3 VERSION DESCRIPTION

3.1 Inventory of Contents

The PBT distribution is structured as shown below. The top-level directory pbt includes PostScript (VDDpbt.ps) and clear ASCII text (VDDpbt.tty) versions of this document. It contains a complete directory listing of the PBT distribution (Contents.tty, reproduced herein as Appendix A). It also contains a PostScript version of the PBT user manual.
It contains the DDL specification (pbt.sds) of the pbt SDS used by the browser. Finally, it contains the following subdirectories, described below:

- pbt/code
- pbt/bin
- pbt/X-Resources
- pbt/X-Resources/bitmaps

### 3.1.1 Subdirectory: pbt/code

This directory contains the Ada source code for the PBT. It also contains the C shell scripts and associated support files needed to rebuild the PBT.

### 3.1.2 Subdirectory: pbt/bin

This is the directory into which the build process moves the PBT executable after a successful compile and link. This directory contains two PBT executables named PBT12.2 and PBT12.3, which have been linked to execute under Emeraude V12.2 and V12.3, respectively.

### 3.1.3 Subdirectory: pbt/X-Resources

This directory contains the PCTE-Browser.black-n-white and PCTE-Browser.color files describing the X resource values used by the browser for black-and-white and color monitors, respectively. These values specify such characteristics of the PBT as the dimensions to be used for the various windows created by the browser. This directory also contains the bitmaps subdirectory, described below.

#### 3.1.3.1 Subdirectory: pbt/X-Resources/bitmaps

This directory contains X bitmaps for the icons used by the PBT. These icons represent the different types of PCTE objects (e.g., File) and relationships (e.g., Composition and Reference links) recognized by the browser.

### 3.2 Changes Installed

#### 3.2.1 Release 1.0

#### 3.2.1.1 General Changes

- PBT terminology has been changed to be more consistent with PCTE terminology: Overall Views are now called Composite Views.
Files affected:
callbacks.b.a
static_menus.b.a
utilities.b.a

- Clarified the fact that the link name pattern matching refers to link *type* names. This clarification has been made both to the relevant popup menus and in the relevant user manual sections.

Files affected:
static_menus.b.a
USERpbt.ps

- In order to make the PBT output more consistent, PBT commands which display pathnames have been modified to exclude SDS names when the PBT has been customized to exclude such names from View node and arc labels.

Files affected:
utilities.b.a

- Changed the name of the X-Resources/PCTE-bitmaps directory to be called X-Resources/bitmaps. This was done to be consistent with Motif naming conventions for such subdirectories.

Directories affected:
X-Resources/PCTE-bitmaps

- PCTE I/O oriented files used in the 0.1 release of the PBT have been moved into the 0.2 release of the STARS Ada/PCTE bindings, and have, therefore, been removed from this release.

Files deleted:
pcte_object_create.a pcte_support.a
pcte_support.b.a
pcte_text_io.a
pcte_text_io_b.a

3.2.1.2 Bug Fixes

- Fixed bug that prevented the display of non-key link attributes for links without key attributes.

Files affected:
utilities.b.a

3.2.1.3 New Features
The PBT now has a Motif-oriented look-and-feel, primarily because of the changes made in the 1.0 release of the RGB. (The RGB previously used the STARS Ada/Xt implementation, while now uses SERC’s commercial SA-Motif Ada bindings.) These changes includes the new use of control panels, rather than sequences of popup dialog boxes.

Files affected:
Build_PBT.csh
Build_PBT.var
PCTE-Browser.black.n.white
PCTE-Browser.color
bitmaps/a.rel.xbm
bitmaps/a.rel.inv.xbm
bitmaps/back.r.xbm
bitmaps/c.rel.xbm
bitmaps/c.rel.inv.xbm
bitmaps/f.node.xbm
bitmaps/f.node.inv.xbm
bitmaps/i.rel.xbm
bitmaps/i.rel.inv.xbm
bitmaps/o.node.xbm
bitmaps/o.node.inv.xbm
bitmaps/p.node.xbm
bitmaps/p.node.inv.xbm
bitmaps/p.rel.xbm
bitmaps/p.rel.inv.xbm
bitmaps/r.rel.xbm
bitmaps/r.rel.inv.xbm
bitmaps/s.rel.xbm
bitmaps/s.rel.inv.xbm
bitmaps/v.rel.xbm
bitmaps/v.rel.inv.xbm
browser_instance.a
callbacks.a
callbacks.b.a
main.a
static_cmds.a
static_cmds.b.a
static_menus.a
static_menus.b.a
static_panels.a (new)
static_panels.b.a (new)
utilities.a
utilities.b.a

All global PBT commands are now accessible from every View window.
The PBT is more customizable. For instance, it is now possible to control what types of links to include in new composite views by default.

The PBT now supports the listing of all of the SDSes in the metabase.

Local Views can now be created via the global Display command button, as well as via point-and-click operations on objects in existing Views. This means that it is now just as easy to create Local Views as it is to create Composite Views.

The PBT now supports command line arguments. These arguments can be used to perform such actions as to control customizable PBT features, to set the initial Current
Object, to set the initial working schema, and to create initial Composite and/or Local Views.

Files affected:
- globals.a
- utilities.a
- c_interface.a (new)
- c_interface.b.a (new)
- system_env.a (new)
- system_env.b.a (new)
- utilities.b.a

- The PBT now supports the invoking of arbitrary PCTE processes on behalf of PCTE objects. This new capability allows for specialized viewer programs to be invoked for specialized types of file objects. Previously, only the only way that a file object could be viewed was to display it in a text window—something that could only be done for pure ASCII text files. Now, for instance, a Software Through Picture (StP) diagram, stored in a PCTE file object, can be viewed from within the PBT using the appropriate StP diagram display utility.

Files affected:
- callbacks.a
- callbacks.b.a
- env_int.c (new)
- utilities.a
- utilities.b.a

3.3 Adaptation Data

3.3.1 Operating Environment

Sun-4 Workstations with at least 32 megabytes of main memory

SunOS, Version 4.1.2

X Window System, Version 11, Release 4

Use of any "standard" X window manager (e.g., twm or mwm)

Emeraude PCTE V12.2 or V12.3

3.3.2 Development Environment

Sun-4 workstation with 32 megabytes of main memory

SunOS, Version 4.1.2
3.3.3 Configuration-Unique Data

There are two explicit dependencies in the PBT itself to UNIX, in its use of the “exit” procedure as part of the PBT termination processing, and is access to UNIX environment variables. These features are accessible via the Ada pragma INTERFACE capability in the code files utilities.b.a and system_env.b.a.

The PBT makes extensive use of ECMA PCTE Ada bindings, which, in its current implementation is highly dependent upon the Emeraude V12 PCTE implementation.

3.4 Interface Compatibility

The 1.0 release of the PBT supports all of the functionality of the previous PBT release (0.1). However, the commands and menus used to access many of these functions have been revised in the change to a Motif-oriented user interface.

3.5 Installation Instructions

The sections below describe the steps needed to:

- build the PBT executable (optional—see the note at end of section 3.5.1)
- install the PBT’s X Resource file, PCTE-Browser
- install the PBT executable in the environment
- install the SDS (pbt) needed by the browser in the PCTE metabase

(See the accompanying PBT user manual for details on how to use the browser.)
3.5.1 Build Procedure

This section describes the procedure for compiling and linking the PBT program using the SunAda 1.0 Ada compilation system from Sun Microsystems.

The build process assumes the following. Before proceeding with the build of the PBT, first verify that these assumptions are correct.

- The entire PBT delivery contents have previously been loaded onto the local file system. For purposes of these installation instructions, the top-level directory for the PBT delivery shall be referred to as /local/pbt.

- SERC's SA-Motif, version 1.0, has previously been installed onto the local file system, at a location to be referred to below as /local/serc_samotif.

- The Reusable Graphical Browser, version 1.0, has previously been loaded onto the local file system, at a location to be referred to below as /local/rgb.

- The RGB library has previously been built using the SunAda 1.0 Ada compiler. See the VDD for the RGB release for information on how to build this library. This library is assumed to be found in the following UNIX directory: /local/rgb/Build_SunAda1.0/rgb

- The Emeraude PCTE implementation, version V12.3, has been loaded onto the local file system, at a location to be referred to below as /local/emeraude_pcte.

- The PCTE Ada Bindings implementation version 0.2, has been loaded onto the local file system, at a location to be referred to below as /local/adapcpte.

- The PCTE PCTE Ada Bindings has previously been built using the SunAda 1.0 Ada compilation system. See the VDD for the Ada Bindings release for information on how to build this library. This library is assumed to be found in the following UNIX directory: /local/adapcpte02/Build_SunAda1.0

- The Xlib archive file corresponding to the X11R4 delivery has previously been created. Consult with your local system administrator for the exact location of the Xlib archive file on your system. For purposes of this discussion, it is assumed that this file can be found at:
  
  - /usr/lib/X11/libX11.a

To build the PBT, first edit the code/Build_PBT.var file to reflect the actual operating environment. This file (listed in its entirety in Appendix B.1) initializes the environment variables used by the rest of the build process. Variables that must be initialized include the following:

- PBT - the top level directory of the PBT distribution
30 November 1992

- RGB – the directory containing the RGB Ada library built using SunAda 1.0
- LIBX – the pathname of the X11R4 Xlib archive
- COMPILERPATH – the pathname of the top-level directory of the SunAda 1.0 compilation system
- PCTE – the directory containing the SunAda library for the ECMA PCTE Ada bindings (Ada/PCTE)
- PCTE_ROOT – the top level directory of the PCTE delivery

Once the code/BuildPBT.var file has been edited, the rest of the compiling and linking of the PBT is fully automated. Simply cd to the PBT distribution's code directory and execute the code/BuildPBT.csh C shell script (shown in its entirety in Appendix B.2), as in the following example:

```
% cd /local/pbt/code
% BuildPBT.csh >& Build.out &
```

This script creates the directory called Build_SunAda1.0 below the top-level PBT directory in which the PBT's SunAda library will be created and in which the link will take place.

IMPORTANT NOTE: This step is optional and can be skipped because (already compiled and linked) executables for Emeraude V12.2 and Emeraude V12.3 environment have been included in the release. To skip the build step, simply rename the appropriate executable from the bin directory, as in the following UNIX commands:

```
mv /local/pbt/in/PBT12.2 /local/pbt/bin/PBT
```

In this example, /local/pbt would be replaced by the actual full UNIX pathname of the appropriate site-specific directory.

3.5.2 Executable Installation Procedure

Assuming that the build is successful, the executable PBT will be moved into the bin directory beneath the top-level PBT directory—replacing any version of PBT previously in that directory.

The PBT executable could be installed as a static context within the PCTE object base prior to its first use. However, it can also be accessed from within PCTE by placing it in a UNIX directory that is part of the UNIX PATH environment variable. Therefore, it is assumed that the user will either include the PBT's bin directory in the user's path, or will copy the PBT executable to another directory already in the path (e.g., /usr/local/bin).
3.5.3 Installing the X Resource Files

A number of UNIX files associated with the PBT must be on-line at the time that the PBT is executed:

- A set of files describing the bitmaps to be used for the various node and link icons.
- The "X resource file" associated with the PBT, PCTE-Browser, describing such information as which bitmap to use for which type of object, what dimensions to use for the various widgets used by the PBT, etc.

In the case of the PCTE-Browser file, two different versions are supplied in the release:

- PCTE-Browser.color - for use on color monitors
- PCTE-Browser.black.n.white - for use on monochrome monitors

Depending upon the kind of monitor being used, one of these two files should be linked under the name PCTE-Browser, as in:

    ln PCTE-Browser.color PCTE-Browser

It is also important to make sure that the bitmaps subdirectory is located in the same directory as is the PCTE-Browser file. If it is not installed in the same directory, then the PBT will not be able to locate and use the bitmap files found in this subdirectory when the browser is invoked.

3.5.4 Installing the PBT-Oriented SDS

Some of the new PBT features depend upon the presence of the pbt SDS. In order to install this SDS into the PCTE metabase, the DDL specification of this SDS must be compiled. A file containing this DDL specification is included in this delivery, under the name /local/pbt/pbt.sds. To compile this file, it is necessary to first log into PCTE and then execute the sds.compile command, as in:

    sds.compile /local/pbt/pbt.sds

In this example, /local/pbt would be replaced by the actual full UNIX pathname of the appropriate site-specific directory.

3.6 Potential Problems

1. When the Initial Browser window is used to create the first Composite or Local View, the PBT is designed to transform the Initial Browser window into the View window.
This transformation involves expanding the size of the window, after which the second (view-specific) menu bar is added, along with the graph of the new View at the bottom of the window. However, occasionally, the window will be expanded in size, but the second menu bar and the graph will not be displayed. If and when this happens, simply use the window manager to resize the window: Even a very small change in window size—as small as one pixel—will cause the View window to be displayed properly.

2. The PCTE-Browser is required to be in the directory identified by the XAPPLRESDIR environment variable. If it is not found in this directory, or if the XAPPLRESDIR variable is not properly set, then the PBT will terminate at the first attempt to create a Composite or Local View window, with the following error message:

   Error in kernel:: exception_handler: below bottom of user stack

3. If the bitmaps subdirectory is not found in the directory identified by XAPPLRESDIR, then the browser will not be able to display the proper node and link icons in View windows.

4. The PBT will occasionally crash when deleting a View window if a number of View windows windows had previously been deleted.

5. Warning messages will occasionally be written to the PBT's standard output device when selected windows (primarily control panels) are closed down. These warning messages are harmless and can all be ignored. The following are examples of such warnings:

   Warning: XtRemoveGrab asked to remove a widget not on the list
   Warning: Attempt to remove non-existant passive grab

3.7 Enhancements

Possible future enhancements to the PBT include:

- Addition of a PCTE object base editing capability.
- Addition of a graphical SDS display and editing capability.
- Improved graph layout algorithms.
- Migration to a conforming ECMA PCTE environment.

4 USER FEEDBACK

We encourage experimentation with the PBT, and we solicit feedback from the PCTE community to assist us in improving the product. Thus, we would greatly appreciate your comments, suggestions, and criticisms.
A Appendix: Inventory of Contents

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

pbt:
Contents.tty
USERpbt.ps
VDDpbt.ps
VDDpbt.tty
X-Resources/
bin/
code/
pbt.sds

X-Resources:
PCTE-Browser.black_n_white
PCTE-Browser.color

bin:
PBT12.2*
PBT12.3*
code:
Build_PBT.csh*
Build_PBT.var
browser_instance.a
browser_params.a
browser_params.b.a
c_interface.a
c_interface_b.a
callbacks.a
callbacks_b.a
env_int.c
globals.a
main.a
pcte_layout.a
pcte_layout_b.a
pipe_int.c
static_cmds.a
static_cmds.b.a
static_menus.a
static_menus_b.a
static_panels.a
static_panels_b.a
system_env.a
system_env_b.a
utilities.a
utilities_b.a
B Appendix: Unix Installation Scripts

B.1 File: Build.PBT.var

# Edit these lines and leave them uncommented if you do not want to
# be prompted for the environment variables
#
setenv PBT /local/pbt
setenv RGB /local/rgb/Build_SunAda1.0/rgb
setenv LIBX /usr/lib/libX11.a
setenv COMPILERPATH /local/SunAda
setenv SAMOTIFHOME /local/serc.samotif
setenv PCTE /local/adapcte/Build_SunAda1.0
setenv PCTEROOT /local/emeraude_pcte

# Variables that need not be modified:
setenv OS 4.1
setenv Sun 4
setenv COMPILERNAME sunada
setenv COMPVERSION SunAda1.0
setenv TARGET $PBT/Build_$COMPVERSION
#
# Define the location of the PBT source code directories.
#
if ( ! $?PBT ) then
  echo ""
  echo "Specify path to top level PBT directory "
  echo "(e.g. /local/pbto1) "
  echo ""
  echo "n " PBT = 
  setenv PBT <$
  echo ""
endif
if ( ! -e $PBT ) then
  echo ""
  echo "* $PBT does not exist *"
  echo "* Script aborted **"
  echo ""
  unsetenv PBT
  exit -1
endif
#
# Define the location of the RGB source code directories.

if (! $?RGB ) then
  echo ""
  echo "Specify path to directory containing RGB Ada library"
  echo "(e.g. /local/rgbOS/Build_SunAdal.0/rgb)"
  echo ""
  echo -n " RGB="
  setenv RGB $<
  echo ""
endif

if ( ! -e $RGB ) then
  echo ""
  echo "** $RGB does not exist **"
  echo "** Script aborted **"
  echo ""
  unsetenv RGB
  exit -1
endif

# Define the location of the dependencies.

# Define the location of the X11R3/R4 Xlib archives

# where XLIB = path to the X11 Xlib object archive (e.g./usr/lib/libX11.a)

if ( ! $?LIBX ) then
  echo ""
  echo "Specify the path to the X11 Xlib object archive"
  echo "(e.g. /usr/lib/libX11.a)"
  echo ""
  echo -n " LIBX="
  setenv LIBX $<
  echo ""
endif

if ( ! -e $LIBX ) then
  echo ""
  echo "** $LIBX does not exist **"
  echo "** Script aborted **"
  echo ""
  unsetenv LIBX

exit -1
endif
#

Define C Language compilation variable
#
setenv CC " cc -g -c "
#
# Determine the Ada compilation system to use
#
# Establish a path to the SunAda compilation system
#
if ( ! $?COMPILERNAME || ! $?COMPVERSION || ! $?COMPILERPATH ) then
  echo ""
echo "Please select your compiler name: [sunada] "
echo ""
echo -n " COMPILERNAME = "
setenv COMPILERNAME $<
echo ""
switch ($COMPILERNAME)
case SunAda:
case SUNADA:
case sunada:
  echo -n "Are you building with SunAda1.0? [y,n](n) "
  set COMPVERSION = $<
  echo ""
  switch ($COMPVERSION)
    case Y:
    case y:
      set COMPVERSION = SunAda1.0
      breaksw
    case N:
    case n:
    default:
      set COMPVERSION = SunAda
      echo ""
      echo "Warning! Software has only been tested using SunAda 1.0."
      breaksw
    endsw
  breaksw
endif
breaksw
default:
  echo ""
  echo "You must specify a compiler name."
echo ""
unsetenv COMPVERSION
exit -1
breaksw
echo ""

if ( ! -e $COMPILERPATH/bin/ada ) then
    echo "" "" Cannot find Ada compiler in $COMPILERPATH/bin ""
    echo "" Script aborted ""
    echo ""
    unsetenv COMPILERPATH
    exit -1
endif

if ( -e $COMPILERPATH/bin/ada ) then
    if ( $COMPILERNAME == "sunada" ) then
        setenv COMPILERBIN $COMPILERPATH/bin
        setenv COMPILE "$COMPILERBIN/ada -v -e -00 "
        setenv LINK "$COMPILERBIN/a.ld "
    else
        echo "" "" Cannot find $COMPILERPATH/bin/ada ""
        echo "" Script aborted ""
        echo ""
        unsetenv COMPILERPATH
        exit -1
    endif
else
    echo ""
    echo "" "" Cannot find $COMPILERPATH/bin/ada ""
    echo "" Script aborted ""
    echo ""
    unsetenv COMPILERPATH
    exit -1
endif

# Define the Destination of the PBT build
# where TARGET = path to build destination (e.g. $PBT/Build_SunAda1.0)
if ( ! ??TARGET ) then
echo ""
echo "Specify the path to the TARGET directory "
echo "(Defaults to $PBT/Build_${COMPVERSION}) "
echo ""
177    echo -n "TARGET = "
178    setenv TEMP $<
179    echo ""
180    if ( $TEMP == ) then    # check for null entry
181        setenv TARGET $PBT/Build_$(COMPVERSION)
182        unsetenv TEMP
183    else
184        setenv TARGET $TEMP
185        unsetenv TEMP
186    endif
187    endif
188
189    echo ""
190    echo "TARGET = $TARGET"
191    echo ""
192    echo "RGB = $RGB"
193    echo "PBT = $PBT"
194    echo "PCTEROOT = $PCTEROOT"
195    echo ""
196    echo "LIBX = $LIBX"
197    echo "COMPILERNAME = $COMPILERNAME"
198    echo "COMPERSION = $COMPERSION"
199    echo "COMPILERPATH = $COMPILERPATH"
200    echo "COMPILE = $COMPILE"
201    echo "LINK = $LINK"
202    echo "OS = $OS"
203    echo "Sun = $Sun"
204
205    #
206    # Create the directories for the build
207    #
208    if ( ! -d $TARGET ) mkdir $TARGET
209    if ( ! -d $TARGET/rgb ) mkdir $TARGET/rgb
210    if ( ! -d $TARGET/application ) mkdir $TARGET/application
B.2 Script: Build_PBT.csh

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

# Defining installation-dependent variables
source Build_PBT.var

if ! -e $TARGET mkdir $TARGET

cd $TARGET

echo "Building Ada libraries for the PCTE Browser Tool (PBT)"

if ( $COMPILERNAME == "sunada" ) then
    if (! -e $RGB/ada.lib) then
        echo "Sorry. RGB must be built first. Script aborted."
        exit -1
    endif
else
    echo "Sorry. Only SunAda is currently supported. Script aborted."
    exit -1
endif

if (-e ada.lib ) a.rmlib -f # clean out old library

$SAMOTIFHOME/sup/xii.mklib

echo "Establishing dependencies"

$COMPILERBII/a.path -i $PCTE
$COMPILERBII/a.path -i $RGB
else
    echo "Sorry. Only SunAda is currently supported. Script aborted."
    exit -1
endif

echo "Building TARGET directory with symbolic links to source code"

foreach file ($PBT/code/*.a $PBT/code/*.c)
    if ( ! -e $(file:t) ) ln -s $file $(file:t)
end

cd $TARGET
```

Page 19
44 echo ""
45 echo "Compiling the C source"
46 echo ""
47 cc -c pipe_int.c
48 if ( $status != 0 ) exit $status
49 cc -c env_int.c
50 if ( $status != 0 ) exit $status
51
52 echo ""
53 echo "Compiling the PBT source"
54 echo ""
55 $COMPILE c_interface.a
56 if ( $status != 0 ) exit $status
57 $COMPILE c_interface_b.a
58 if ( $status != 0 ) exit $status
59 $COMPILE browser_params.a
60 if ( $status != 0 ) exit $status
61 $COMPILE browser_params_b.a
62 if ( $status != 0 ) exit $status
63 $COMPILE browser_instance.a
64 if ( $status != 0 ) exit $status
65 $COMPILE system_env.a
66 if ( $status != 0 ) exit $status
67 $COMPILE system_env_b.a
68 if ( $status != 0 ) exit $status
69 $COMPILE pcte_layout.a
70 if ( $status != 0 ) exit $status
71 $COMPILE pcte_layout_b.a
72 if ( $status != 0 ) exit $status
73 $COMPILE globals.a
74 if ( $status != 0 ) exit $status
75 $COMPILE static_menus.a
76 if ( $status != 0 ) exit $status
77 $COMPILE static_cmds.a
78 if ( $status != 0 ) exit $status
79 $COMPILE callbacks.a
80 if ( $status != 0 ) exit $status
81 $COMPILE static_panels.a
82 if ( $status != 0 ) exit $status
83 $COMPILE static_menus_b.a
84 if ( $status != 0 ) exit $status
85 $COMPILE static_cmds_b.a
86 if ( $status != 0 ) exit $status
87 $COMPILE utilities.a
88 if ( $status != 0 ) exit $status
30 November 1992

89 $COMPILE static-pauels.b.a
90 if ( $status != 0 ) exit $status
91 $COMPILE utilities.b.a
92 if ( $status != 0 ) exit $status
93 $COMPILE callbacks_b.a
94 if ( $status != 0 ) exit $status
95 $COMPILE main.a
96 if ( $status != 0 ) exit $status
97
98 echo ""
99 echo "Linking the objects"
100 echo ""
101 set objects = ($RGB/call_ada.o $PCTE/util.o pipe_int.o env_int.o)
102 setlibs = ($LIBX $PCTEROOT/lib/libemer.a)
103 $LINK -v main $objects -o PBT $libs
104 if ( $status != 0 ) exit $status
105 mv PBT $PBT/bin/PBT
106
107 echo ""
108 echo "Build Complete"