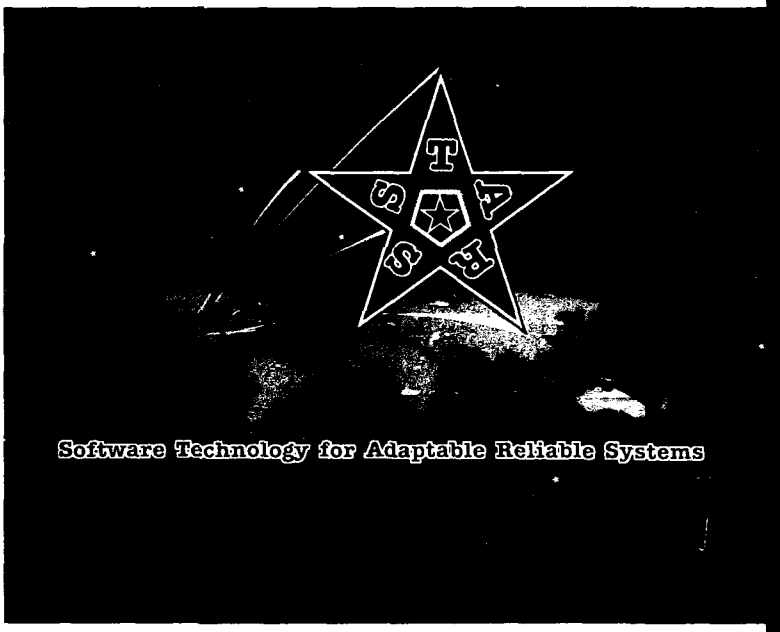
TASK: UR20
CDRL: 01010

(2)

UR20 — Process/Environment Integration Ada/Xt Toolkit, Version 2.2 SunOS Implementation

Version Description Document

UNISYS



STARS–RC–01010/001/00

23 July 1990

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	23 July 1990	Version Description Document

4. TITLE AND SUBTITLE

Ada/Xt Toolkit

5. FUNDING NUMBERS

STARS Contract
F19628-88-D-0031

6. AUTHOR(S)

Kurt C. Wallnau
Robert C. Smith
Timothy M. Schreyer

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Unisys Corporation
12010 Sunrise Valley Drive
Reston, VA 22091

8 PERFORMING ORGANIZATION REPORT NUMBER

GR-7670-1133(NP)

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Department of the Air Force
Headquarters, Electronic Systems Division (AFSC)
Hanscom AFB, MA 01731-5000

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

01010

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release;
distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This software package provides an Ada programmatic interface to a set of reusable user interface abstractions known as widgets. The software provides the full functionality of the M.I.T. X Consortium Version 11 Release 3 X Window System. The software consists of three components: an Ada binding to the Xlib layer, an Ada implementation of the Xt Intrinsics layer, and an Ada widget library. The Ada binding to the Xlib layer is an upgrade of the STARS Foundations Ada Xlib binding, and provides a protocol interface including a set of graphics drawing primitives. The Xt Intrinsics layer provides a policy-free mechanism for creating and managing user interface objects. User interface objects are contained in the widget library which consists of a small set of commonly used user interface abstractions, such as scrollbars and command buttons.

14. SUBJECT TERMS

Ada bindings to Xlib
Xt intrinsics
XT sample widget set

15. NUMBER OF PAGES

51

16. PRICE CODE

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

TASK: UR20
CDRL: 01010

VERSION DESCRIPTION DOCUMENT

For The

SOFTWARE TECHNOLOGY FOR ADAPTABLE, RELIABLE SYSTEMS (STARS)

Ada/Xt Toolkit
Version 2.2
SunOS Implementation

STARS-RC-01010/001/00
Publication No. GR-7670-1133(NP)
23 July 1990

CDRL Type: A005, Informal Technical Data

CONTRACT NO. F19628-88-D-0031
Delivery Order 0002

Prepared for:

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

Prepared by:

Unisys Defense Systems
Tactical Systems Division
12010 Sunrise Valley Drive
Reston, VA 22091

TASK: UR20
CDRL: 01010

VERSION DESCRIPTION DOCUMENT

For The

SOFTWARE TECHNOLOGY FOR ADAPTABLE, RELIABLE SYSTEMS (STARS)

Ada/Xt Toolkit
Version 2.2
SunOS Implementation

STARS-RC-01010/001/00
Publication No. GR-7670-1133(NP)
23 July 1990

CDRL Type: A005, Informal Technical Data

CONTRACT NO. F19628-88-D-0031
Delivery Order 0002

Prepared for:

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

Prepared by:

Unisys Defense Systems
Tactical Systems Division
12010 Sunrise Valley Drive
Reston, VA 22091

PREFACE

This document was prepared by Unisys Defense Systems, Valley Forge Operations, in support of the Unisys STARS Prime contract under the Process/Environment Integration task (UR20). This CDRL, 01010, is type A005 (Informal Technical Data) and is entitled "Ada/Xt Toolkit, Version 2.2 - SunOS Implementation: Version Description Document".

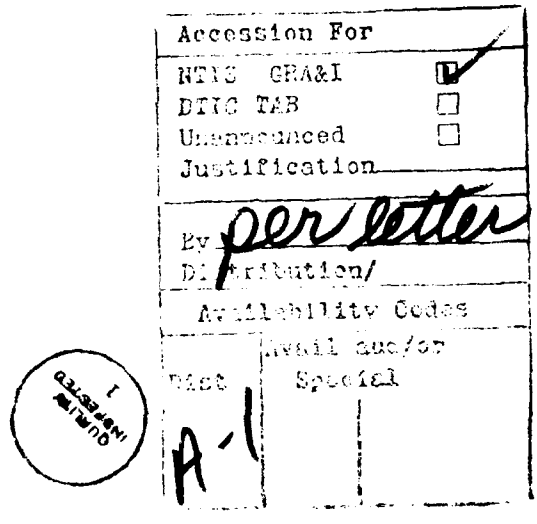
This document has been reviewed and approved by the following Unisys personnel:

Reviewed by: Richard E. Creps

Richard E. Creps, System Architect (Acting)

Approved by: Hans W. Polzer

Hans W. Polzer, Program Manager



Contents

1 SCOPE

1.1 Identification

Version Description Document,
Ada/Xt Toolkit,
Version 2.2,
SunOS Implementation

1.2 System Overview

Ada/Xt is an Ada implementation of the MIT X Toolkit (Xt). Xt consists of a set of intrinsics interfaces, which provide operations for construction, arrangement, and manipulation of windows, and a set of widgets, which are window objects ranging from buttons and scrollbars to text editors. Xt is built on top of Xlib, which is a low-level library of window primitives. The routines in Xlib provide window functionality by communicating with the X window server using the X Window System protocol. This release of Ada/Xt for X Window System version 11, release 3 (X11R3) consists of three major parts. These are:

1. Ada bindings to Xlib - the C language interface to the X protocol;
2. Xt Intrinsics - includes an intrinsics package and several built-in widget classes; and
3. an Xt sample widget set - includes ten MIT Athena widgets and two HP widgets.

The Ada/Xlib bindings are an Ada data structure interface to the underlying C language object code for Xlib. Ada/Xlib uses Ada's *pragma INTERFACE* to "bind" Ada subroutines to their C counterparts. The Ada/Xlib bindings provide a complete mapping to the C Xlib and can be used as a stand-alone product.

The Xt Intrinsics, as noted above, provide a higher-level interface used to construct windowing applications. Ada/Xt is an implementation of Xt in Ada, not a binding like Ada/Xlib. Ada/Xt provides an intrinsics package which provides the functionality of Xt used to manage X resources, events, and hierarchical widget construction. The interfaces of the C Xt are mapped one-to-one in the Ada implementation. Ada/Xt also implements the built-in composite and constraint widgets, in addition to the family of shell widgets.

The sample widget set provided with Ada/Xt, rounds out the toolkit. The widgets are intended as examples for construction of additional widgets and an initial set with which to prototype Ada windowing applications. Ten of the MIT Project Athena widgets have been re-implemented in Ada, along with two of the Hewlett-Packard (HP) widgets. The sample widget set includes the following widgets:

simple a core widget with a sensitive border and variable cursor (Athena).

label a simple widget that can display static text or pixmaps (Athena).

command a label widget that can perform mouse-activated actions (Athena).

scroll a scrollbar used to alter the view in a window (Athena).

form a constraint widget which allows child widgets to be layed-out relative to each other and manages that relationship (Athena).

viewport a form widget which allows a partial view of a larger window using scrollbars (Athena).

manager a constraint widget which provides general geometry and lay-out services to its children (HP).

bboard a bulletin board manager widget which allows children to be positioned at (X,Y) locations (HP).

text the basic text editor widget from which the others are specialized (Athena).

ascii string text a text widget which allows viewing, navigation, and editing of a string (Athena).

ascii disk text a text widget which allows viewing, navigation, and appending to a file (Athena).

dialog a form widget which contains a prompt string, a text entry field, and any number of command buttons (Athena).

This release of Ada/Xt is sufficient to prototype Ada windowing applications of significant complexity. Additional releases of Ada/Xt will enhance and supplement the existing widget set and make the Ada language interface to X an even more viable development tool.

2 RELATED SOFTWARE

The Ada language interface requires a compiled "C" version of X11R3 and its archives. Also required is the library for UNIX system services, used for low-level interaction with the operating system. Routines are used from this library to query environment information, generate temporary file names, probe UNIX sockets, and manipulate memory at the lowest level. (See "Configuration-unique Data".)

An equivalent version of this software is available for operation under the MACH operating system.

3 VERSION DESCRIPTION

3.1 Inventory of Contents

3.1.1 Directory: *code*

The *code* directory is structured as shown below:

```
code
code/C
code/Utils
code/Xlib
code/Xlib/telesoft
code/Xlib/vads
code/Xt
code/Xt/telesoft
code/Xt/vads
code/Xmu
code/Xmu/vads
code/Widgets
code/Widgets/telesoft
code/Widgets/vads
code/Tests
code/Tests/telesoft
code/Tests/vads
code/Demo
code/Demo/telesoft
code/Demo/vads
```

Each higher-level subdirectory (Demo, Tests, Widgets, etc.) contains the source code needed to generate a part of the Ada/Xt Toolkit. Files that can be used with either the VADS or TeleSoft compiler reside in the higher-level sub-directories. The lower-level sub-directories (*vads* and *telesoft*) contain those source files that are dependent upon particular features or syntax of the corresponding compiler. Execution of one of the build scripts listed below will cause the appropriate compiler-specific source files to be copied into the higher-level sub-directories.

In addition to the sub-directories listed above, the *code* directory contains the following files:

```
BuildAdaXt.TG2      -- C-shell script for building Ada/Xt with TeleSoft
BuildAdaXt.VADS     -- C-shell script for building Ada/Xt with VADS
BuildAdaXt.var      -- Defines environment variables for the above scripts
Contents.tty        -- A directory listing (reproduced as Appendix A)
VDDxt.ps            -- PostScript version of this document
VDDxt.tty           -- Clear ASCII text version of this document
```

The two UNIX C-shell scripts, with some minor modification of the *.var* file, can be used to build the Ada/Xt Toolkit from its Ada source code. Similar scripts, in each of the lower-level subdirectories, are invoked for compilation of the individual components. Further information regarding the modification and use of these scripts is provided in section 3.5.

3.1.2 Sub-directory: *C*

This directory contains small C routines used by Ada/Xt. The file *Makefile* can be used to build an archive *lib.a* of C object code, via the UNIX *make* utility. This archive is needed to link programs which depend on the Ada Xlib bindings and/or the Ada/Xt intrinsics or built-in Widgets.

3.1.3 Sub-directory: *Utils*

The toolkit requires that subroutines be "called-back" when a corresponding X event is received. To facilitate this, an Ada procedure type mechanism has been implemented. An object of a procedure type is passed to a dispatch routine, and the Ada subroutine identified by the object is called. Each specific procedure callback type is defined in its own package. This directory contains two UNIX "awk" scripts which can be used to generate the Ada package for an Ada/Xt callback type. Generating new callback packages saves the user from repetitive coding and shields the implementation of the Ada procedure type. Also included in this directory is a sample callback specification, *xt_convert_proc*, which was used to generate the package *xt_converter_procs*, which is one of the pre-existing callback types in Ada/Xt.

The scripts are used in this way:

```
awk -f genSpec.awk spec-file-name > spec-file-name_.a
awk -f genBody.awk spec-file-name > spec-file-name.a
```

These two lines will generate the Ada package specification in *spec-file-name_.a* and the body in *spec-file-name.a* for the callback specified in file *spec-file-name*. The "awk" scripts detail the syntax for how a callback type should be specified, and the file *xt_convert_proc* in this directory is an example.

3.1.4 Sub-directory: *Xlib*

This directory contains the Ada bindings to the MIT X Window System library (Xlib). They have been adapted from bindings produced by SAIC under a STARS Foundations contract. The file *hw.a* contains a test program for the Xlib bindings, which will open a window, display a message at each button click in the window, and exit when "q" is typed on the keyboard.

3.1.5 Sub-directory: *Xt*

This director_ contains the implementation in Ada of the MIT X Toolkit (Xt). Source code in this directory depends on the Ada Xlib bindings and the X miscellaneous utilities (Xmu) package.

3.1.6 Sub-directory: *Xmu*

This directory contains the Ada implementation of the X miscellaneous utilities (Xmu) needed by the X Toolkit (Xt). Also contained in this directory are packages for public resource types defined while developing the sample widget set. (These packages can be identified by their "*_r_.a*" or "*_r.a*" endings.) They are intended as an example of how new public resource types should be created. The code in this directory is dependent on code in the Ada Xlib bindings and in the Xt implementation.

3.1.7 Sub-directory: *Widgets*

This directory contains Ada source for the Widget set accompanying the Ada/Xt toolkit. The code in this directory is dependent on the Ada Xlib bindings, the Ada/Xt intrinsics and built-in Widgets, and the Xmu package.

3.1.8 Sub-directory: *Tests*

This directory contains eleven test programs which test the Ada/Xt intrinsics and exercise the sample widget set hierarchy. These programs depend on the Ada Xlib bindings, the Xmu package, the Ada/Xt Intrinsics, and the Widgets.

test_shell.a	creates and displays an application shell Widget.
test_label.a	creates and displays a label Widget within the application shell Widget.
test_label_callbacks.a	expands *test_label.a* so that it calls some destroy callbacks and exits.
test_label_events.a	expands *test_label.a* so that a button press in the label widget's window will call an event handler which exits.
test_command.a	creates and displays a command button widget. The command button's actions are translated from the file named *trans_file* by Ada/Xt's translation management. *test_command.a* relies on the file *read_trans.a*, which also appears in this directory.

test_scrollbar.a creates and displays a wide vertical scrollbar in a window.

test_bb.a creates and displays a bulletin board widget with three active command buttons positioned in it.

test_vw.a creates and displays a viewport widget wtth two scroll bars which overlays a bulletin board widget which has three active command buttons positioned on it.

test_ascii_string_text.a creates and displays an ascii_string text widget which allows one to view and edit the text in the resource string.

test_ascii_disk_text.a creates and displays an ascii_disk text widget which allows one to view a file and append to it. The file must be specified as a resource.

test_dialog.a creates and displays a dialog widget with two buttons. One can enter text into the ascii_string text window and have it returned with the "ok" button. The "cancel" button will exit the program.

3.1.9 Sub-directory: *Demo*

This directory contains the Ada source files which make up the Ada/Xt demo application (*demo_ada*). The file *DEMOSCRIPT* in this directory defines a manual procedure for conducting a demonstration of the Ada/Xt Toolkit using the demo application. The file *Demo* is an X resource file, and the files *applic*, *folder*, *mensetmanus*, and *woman* are X bitmap files.

3.2 Changes Installed

Version 2.0 is the third delivery of the Ada X Toolkit. Every file's version information in the header has been modified. Changes from previous releases of the toolkit reflect bug fixes and performance enhancements. Also, several files have been added to Ada/Xt. A family of text editor widgets have been added to the widget set for Ada/Xt. In addition, the conventions for adding user-defined widget resource types are now exemplified by several new packages in the Xmu directory (these files end with "*_r_.a*" and "*_r.a*").

The following files have been modified since Version 1.0b:

```
Widgets:
    scroll_public.a      -- bug fix: case sensitivity of continuous scrolling

    form_public.a        -- bug fix: numeric_error when resizing very large
```

```
Tests:
    read_trans.a          -- reduced buffer size to meet TeleSoft constraint

Xlib:
    x_lib_.a              -- type Keycode now 0 .. 255 (was 8 .. 255 causing
                          -- problems in translation management)

    x_keyboard.a          -- bug fixes for Modifiers
                          -- Modifier_Keymap type is incorrect

    os_dependent_.a       -- added Host OS values needed in Xmu

Xmu/vads:
    xt_justify_r.a        -- made return type variable a global object

Xt:
    event_mgt_localSU2.a     -- send_focus_notify bug corrected
                             -- convert_type_to_mask parameter types corrected
                             --    (from "in out" to just "out")
                             -- input and timer event procs protected against
                             --      nullity

    xt_event_managementSU.a -- change parameter types for convert_type_to_mask

    xt_utilitiesSU.a         -- added null checks for pointer to children array

    xt_initializersSU.a      -- initialized device queues in
                             --     xt_create_application_context
                             -- bug fix: get_user_defaults

    device.a                 -- removed heap creep
                             -- modified socket select to return changed
                             --    fd masks

    tm_stateSU2.a            -- work-around VADS code generator bug
                             -- made compute_late_bindings separate

    xt_translation_managementSU.a -- made semi_private routines separate

    parserSU2.a              -- work-around VADS code generator bug
                             -- bug fixes: if conditions typo
```

The following files are new to Version 2.0:

```
Xt:
    compute_late_bindingsSU3.a  -- complex subroutine made separate
    trans_mgt_semi_privateSU2.a -- package body made separate

Xmu/vads:
    mask_r_.a                   -- 32-bit mask widget resource
    mask_r.a
    xt_text_edit_type_r_.a      -- text edit mode widget resource
    xt_text_edit_type_r.a
    selection_array_r_.a        -- selection array widget resource
    xmu_atoms_.a                -- new atoms and atom routines for Xmu
    xmu_atoms.a

Widgets:
    ascii_disk_private_.a                -- new ascii_disk widget
    ascii_disk_public.a
    ascii_disk_public_.a
    ascii_string_private_.a              -- new ascii_string widget
    ascii_string_public.a
    ascii_string_public_.a
    dialog_private_.a
    dialog_public.a                      -- new dialog widget
    dialog_public_.a
    text_ascii_sink.a                    -- sink for ascii data
    text_ascii_sink_.a
    text_check_resize_or_overflowSU2.a   -- separate text edit operation
    text_edit_semi_private_routines.a    -- shared text widget routines
    text_edit_semi_private_routines_.a
    text_insert_charSU2.a                -- separate text edit operation
    text_localSU.a                       -- text edit local routines subunit
    text_private_.a                      -- basic text widget
    text_public.a
    text_public_.a
    text_replace_textSU2.a               -- separate text edit operation
    text_sources.a                       -- disk and ascii data sources
    text_sources_.a
    text_widget_globals_.a               -- global text widget info
    text_widget_support.a                -- text widget support routines
    text_widget_support_.a
```

The following files are now obsolete in Version 2.0:

```
Xlibs:
    x_events.a                           -- incorporated into x_lib.a
```

```
Xt:
    xt_error_managementSU.a                -- no longer used
```

3.3 Adaptation Data

3.3.1 Operating Environment

Sun-3 Workstations, all models

SunOS, Version 3.5 or 4.0.3

MIT X Window System, Version 11, Release 3

3.3.2 Development Environment

Sun-3 Workstations, all models

SunOS, Version 3.5 or 4.0.3

MIT X Window System, Version 11, Release 3

Verdix Ada Development System (VADS), Version 5.5t

TeleSoft TeleGen2 1.4

3.3.3 Configuration-unique Data

This implementation requires access to some UNIX system services. These services are documented in the UNIX "man" pages and include the following routines:

malloc, free	lowest-level memory management routines
select	UNIX socket connection polling routine
getenv, gethostname, getpwnam, getpwuid, getuid, getwd	routines to retrieve UNIX environment
tmpnam	routine to generate temporary file name

Further, the appropriate archive should be located in */usr/lib/libc.a*, and accessed via the standard UNIX link directive, "-lc". These instructions are included in the build scripts which accompany this release. Most often, the Ada compilation system will link this archive in automatically.

3.4 Interface Compatibility

It is possible, although unlikely, that existing Ada applications which use the Ada language bindings to Xlib will be affected by this release. Any programs built with Versions 1.0 or 2.0 of Ada/Xt should be compatible with this release.

3.5 Installation Instructions

3.5.1 VADS

File *code/BuildAdaXt.VADS* is an executable UNIX C-shell script, which can be used to build the Ada/Xt Toolkit from the Ada source code, using Verdix Ada Development System (VADS) version 5.5t. It ensures that VADS library dependencies are established correctly, making it unnecessary for the installer to perform these operations manually. This script invokes similar scripts in each of the *vads* sub-directories, which build individual portions of the software in the proper order.

3.5.2 TeleSoft

File *code/BuildAdaXt.TG2* is an executable UNIX C-shell script, which can be used to build the Ada/Xt Toolkit from the Ada source code, using TeleSoft's TeleGen2 compilation system version 1.4. This script invokes similar scripts in each of the *telesoft* sub-directories, which build individual portions of the software.

3.5.3 Build Procedure

1. Edit the environment variables in file *BuildAdaXt.var* to reflect the actual operating environment. The following environment variables must be modified:

   Code - identifies the full pathname of the directory into which the Ada/Xt source code has been loaded (e.g., /mybase/myuser/adaxt/code);

   XLIB - identifies the full pathname of the directory containing the X11R3 Xlib object archive included in the X11R3 distribution (e.g., /usr/lib/libX11.a);

   Vadspath - (as needed) identifies the full pathname of the directory containing the VADS compilation system (e.g., /mybase/compilers/vads5.5).

   TELEGEN2 - (as needed) identifies the full pathname of the directory containing the TeleSoft compilation system (e.g., /mybase/compilers/telegen2).

   If the *code* directory is structured as described in this document, no further modifications are necessary. If not, the following additional variables in *BuildAdaXt.var* will have to be modified to indicate which host directory contains each of the major code components of this release:

C

Xlib

Xt

Xmu

Widgets

Tests

Demo

CLIB

2. If using the VADS compiler, execute *BuildAdaXt.VADS*. This script will access the appropriate compiler-specific source files prior to compilation.

3. If using the TeleSoft compiler, execute *BuildAdaXt.TG2*. This script will access the appropriate compiler-specific source files prior to compilation.

3.5.4 Build Validation

Several programs are available that can be used to ensure that the software has been built correctly. File *hw.a* in the *Xlib* sub-directory contains a test program (*hello_world*) for the Ada Xlib bindings. When executed, this program opens a window, displays a message at each button click within the window, and exits when "q" is typed on the keyboard. The *demo_ada* program in sub-directory *Demo* and the test programs in sub-directory *Tests/vads* offer more extensive validation capabilities.

3.6 Potential Problems

3.6.1 Subprogram Type Simulation

The subprogram type simulation mechanism relies on knowledge of subprogram calling conventions implemented by particular compilers. The implementation of subprogram types used in this delivery has been tested on the VADS and TeleSoft compilation systems previously identified. However, new releases of these compilation systems may invalidate the current implementation.

3.7 Enhancements

The Ada bindings to Xlib, which were produced by SAIC under a STARS Foundation contract, have been modified significantly in order to conform more closely to their corresponding low-level interfaces.

4 NOTES

1. For application programmers, the following package structure is of interest:

```
package intrinsics
    package xt_ancillary_types
    package xt_type_operations
    package command_line_arguments
    package xt_utilities
    package xt_initializers
    package xt_instance_management
    package xt_composite_management
    package xt_geometry_management
    package xt_popup_management
    package xt_class_management
    package xt_callbacks
    package xt_event_management
    package xt_resource_management
    package xt_translation_management
    package xt_selection_management

    package xt_procedure_types
       package xt_callback_procs
       package xt_input_callback_procs
       package xt_timer_callback_procs
       package xt_event_handler_procs
       package xt_work_procs
       package xt_accept_focus_procs
       package xt_args_funcs
       package xt_args_procs
       package xt_expose_procs
       package xt_init_procs
       package xt_procs
       package xt_realize_procs
       package xt_order_procs
       package xt_create_popup_child_procs
       package xt_set_value_funcs
       package xt_string_procs
       package xt_widget_class_procs
       package xt_widget_procs
       package xt_geometry_handler_procs
       package xt_almost_procs
       package xt_resource_default_procs
       package xt_converter_procs
       package xt_action_procs
```

```
            package xt_key_procs
            package xt_case_procs
            package xt_selection_callback_procs
            package xt_convert_selection_procs
            package xt_selection_done_procs
            package xt_error_msg_handler_procs
            package xt_error_handler_procs
            package xt_lose_selection_procs
            package xt_async_handlers

        package core_public
        package composite_public
        package constraint_public
        package shell_public

        package simple_public
        package label_public
        package command_public
        package scroll_public
        package form_public
        package xw_manager_public
        package xw_bboard_public
        package viewport_public
        package text_public
        package ascii_string_public
        package ascii_disk_public
        package dialog_public
```

2. The following packages are of interest to the widget programmer building more widgets with the toolkit:

```
        package core_private
        package composite_private
        package constraint_private
        package shell_private

        package simple_private
        package label_private
        package command_private
        package scroll_private
        package form_private
        package xw_manager_private
        package xw_bboard_private
        package viewport_private
        package text_private
```

```
package ascii_string_private
package ascii_disk_private
package dialog_private
```

3. Also of interest to application and widget programmers, selection management has been fully implemented, but is untested, and therefore has been "commented-out" in the specification of the intrinsics. These interfaces will be supported sometime in the future.

4. Some of the management sub-packages in the Ada/Xt intrinsics have sub-packages named *semi_private* (e.g., *xt_translation_management.semi_private*) which are not intended for use by the widget or application programmer.

A Appendix A: Inventory of Contents

```
code:
BuildAdaXt.TG2*
BuildAdaXt.VADS*
BuildAdaXt.var*
C/
Contents.tty
Demo/
Tests/
Utils/
VDDxt.ps
VDDxt.tty
Widgets/
Xlib/
Xmu/
Xt/

code/C:
Makefile*
and.c
c_make_string.c
fd.c
funcall.c
get_errno.c
or.c
qfree.c
sys_env.c
xbcopy.c
xbittest.c
xstrlen.c

code/Demo:
DEMOSCRIPT
Demo
applic*
demo.a
demo_callbacks.a
demo_callbacks_.a
demo_except_.a
demo_lines.a
demo_lines_.a
demo_menu.a
demo_menu_.a
folder*
```

```
make_Demo.inv*
mensetmanus*
telesoft/
vads/
woman*

code/Demo/telesoft:
BuildDemo.TG2*
liblst.alb

code/Demo/vads:
BuildDemo.VADS*

code/Tests:
read_trans.a
telesoft/
test_ascii_disk_text.a
test_ascii_string_text.a
test_bb.a
test_command.a
test_dialog.a
test_label.a
test_label_callbacks.a
test_label_events.a
test_scrollbar.a
test_shell.a
test_vw.a
trans_file
vads/

code/Tests/telesoft:
BuildTests.TG2*
liblst.alb

code/Tests/vads:
BuildTests.VADS*

code/Utils:
genBody.awk
genSpec.awk
xt_converter_proc

code/Widgets:
ascii_disk_private_.a
ascii_disk_public.a
```

```
ascii_disk_public_.a
ascii_string_private_.a
ascii_string_public.a
ascii_string_public_.a
command_private_.a
command_public.a
command_public_.a
create_tile.a
create_tile_.a
dialog_private_.a
dialog_public.a
dialog_public_.a
form_private_.a
form_public.a
form_public_.a
label_private_.a
label_public.a
label_public_.a
liblst.alb
pixmaps.a
pixmaps_.a
scroll_private_.a
scroll_public.a
scroll_public_.a
simple_private.a
simple_private_.a
simple_public.a
simple_public_.a
telesoft/
text_ascii_sink.a
text_ascii_sink_.a
text_check_resize_or_overflowSU2.a
text_edit_semi_private_routines.a
text_edit_semi_private_routines_.a
text_insert_charSU2.a
text_localSU.a
text_private_.a
text_public.a
text_public_.a
text_replace_textSU2.a
text_sources_.a
text_widget_globals_.a
text_widget_support.a
text_widget_support_.a
vads/
```

```
viewport_private_.a
viewport_public.a
viewport_public_.a
xt_edit_configuration_r_.a
xw_bboard_private_.a
xw_bboard_public.a
xw_bboard_public_.a
xw_constants_.a
xw_manager_private_.a
xw_manager_public.a
xw_manager_public_.a
xw_res_convert.a
xw_res_convert_.a
xw_traversal_procs.a
xw_traversal_procs_.a

code/Widgets/telesoft:
BuildWidgets.TG2*
liblst.alb
text_sources.a

code/Widgets/vads:
BuildWidgets.VADS*
text_sources.a

code/Xlib:
compiler_dependent_.a
fromcstringSU.a
get_window.a
hw.a
opendispSU.a
os_dependent_.a
string_encaps_.a
system_environment_.a
telesoft/
vads/
x_atoms.a
x_colors.a
x_cursors.a
x_cutpaste.a
x_fonts.a
x_graphic.a
x_keyboard.a
x_keysyms_.a
x_lib_.a
```

```
x_regions.a
x_resmgmt.a
x_win_mgr.a

code/Xlib/telesoft:
BuildXlib.TG2*
compiler_dependent.a
liblst.alb
string_encaps.a
system_environment.a
x_int_.a
x_lib.a

code/Xlib/vads:
BuildXlib.VADS*
compiler_dependent.a
string_encaps.a
system_environment.a
x_int_.a
x_lib.a

code/Xmu:
vads/

code/Xmu/vads:
BuildXmu.VADS*
mask_r.a
mask_r_.a
selection_array_r_.a
xmu.a
xmu_.a
xmu_atoms.a
xmu_atoms_.a
xmu_cvt_string_to_cursorSU.a
xmu_cvt_string_to_widget.a
xmu_cvt_string_to_widget_.a
xmu_paths_.a
xt_edge_r.a
xt_edge_r_.a
xt_float_r_.a
xt_justify_r.a
xt_justify_r_.a
xt_orientation_r.a
xt_orientation_r_.a
xt_text_edit_type_r.a
```

```
xt_text_edit_type_r_.a

code/Xt:
comp_obj_private_.a
comp_obj_public.a
comp_obj_public_.a
composite_private_.a
composite_public.a
composite_public_.a
compute_late_bindingsSU3.a
constraint_private_.a
constraint_public.a
constraint_public_.a
convertSU.a
core_private_.a
core_public.a
core_public_.a
cursor_cache_.a
device_.a
event_mgt_localSU2.a
get_resourcesSU2.a
installation_.a
int_tm_initSU2.a
intrinsics_.a
object_private_.a
object_public.a
object_public_.a
parserSU2.a
realloc_array.a
realloc_array_.a
rect_obj_private_.a
rect_obj_public.a
rect_obj_public_.a
renamed_xlib_types_.a
resourceP.a
resourceP_.a
resource_generic_int.a
resource_generic_int_.a
selection_mgt_localSU2.a
selection_procedure_types.SU2.a
shell_internalsSU.a
shell_private_.a
shell_public.a
shell_public_.a
stringops.a
```

```
stringops_.a
telesoft/
tm_constants_.a
tm_eventsSU2.a
tm_late_bindingsSU2.a
tm_stateSU2.a
trans_mgt_semi_privateSU2.a
type_conversionsSU.a
vads/
window_obj_private_.a
window_obj_public.a
window_obj_public_.a
xrm_get_resources.a
xrm_get_resources_.a
xt_accept_focus_procs.SU2.a
xt_action_procs.SU2.a
xt_almost_procs.SU2.a
xt_args_func.SU2.a
xt_args_procs.SU2.a
xt_async_handler.SU2.a
xt_build_keysym_tableSU2.a
xt_callback_procs.SU2.a
xt_callbacksSU.a
xt_case_procs.SU2.a
xt_class_managementSU.a
xt_composite_managementSU.a
xt_convert_selection_procs.SU2.a
xt_converter_procs.SU2.a
xt_create_popup_child_procs.SU2.a
xt_displaySU.a
xt_error_handler_procs.SU2.a
xt_error_msg_handler_procs.SU2.a
xt_event_handler_procs.SU2.a
xt_event_managementSU.a
xt_expose_procs.SU2.a
xt_geometry_handler_procs.SU2.a
xt_geometry_managementSU.a
xt_init_procs.SU2.a
xt_initializersSU.a
xt_input_callback_procs.SU2.a
xt_instance_managementSU.a
xt_key_procs.SU2.a
xt_keycode_to_keysymSU2.a
xt_lose_selection_procs.SU2.a
xt_order_procs.SU2.a
```

```
xt_popup_managementSU.a
xt_procedure_typesSU.a
xt_procs.SU2.a
xt_realize_procs.SU2.a
xt_resource_default_procs.SU2.a
xt_resource_managementSU.a
xt_resource_type_instances_.a
xt_selection_callback_procs.SU2.a
xt_selection_done_procs.SU2.a
xt_selection_managementSU.a
xt_set_value_func.SU2.a
xt_string_procs.SU2.a
xt_stringdefs_.a
xt_timer_callback_procs.SU2.a
xt_translation_managementSU.a
xt_type_opsSU.a
xt_utilitiesSU.a
xt_widget_class_procs.SU2.a
xt_widget_procs.SU2.a
xt_work_procs.SU2.a

code/Xt/telesoft:
BuildXt.TG2*
bodies.alb
device.a
dispatch_interfaces_.a
external_support_code_.a
intrinsics.a
liblst.alb
seconds.alb

code/Xt/vads:
BuildXt.VADS*
device.a
dispatch_interfaces_.a
external_support_code_.a
intrinsics.a
```

B Appendix B: VADS Build Scripts

B.1 Script: BuildAdaXt.VADS

```
#! /bin/csh -x
#
# Define installation-dependent variables
#
source BuildAdaXt.var
#
# Copy source files into the VADS configuration
#
cp $Xlib/vads/* $Xlib
cp $Xt/vads/* $Xt
cp $Xmu/vads/* $Xmu
cp $Widgets/vads/* $Widgets
cp $Tests/vads/* $Tests
cp $Demo/vads/* $Demo
#
# Build Ada libraries in each sub-directory
#
a.mklib -f $Xlib $Vadspath/verdixlib
a.mklib -f $Xt $Vadspath/verdixlib
a.mklib -f $Xmu $Vadspath/verdixlib
a.mklib -f $Widgets $Vadspath/verdixlib
a.mklib -f $Tests $Vadspath/verdixlib
a.mklib -f $Demo $Vadspath/verdixlib
#
# Establish dependencies
#
cd $Demo
a.path -a $Xlib
a.path -a $Xt
a.path -a $Xmu
a.path -a $Widgets
cd $Tests
a.path -a $Xlib
a.path -a $Xt
a.path -a $Xmu
a.path -a $Widgets
cd $Widgets
a.path -a $Xlib
a.path -a $Xt
a.path -a $Xmu
cd $Xmu
```

```
42 a.path -a $Xlib
43 a.path -a $Xt
44 cd $Xt
45 a.path -a $Xlib
46 a.path -a $Xmu
47 #
48 # Compile the code
49 #
50 cd $C
51 make                >& Makefile.log         # Build the C utilities
52
53 cd $Xlib
54 BuildXlib.VADS      >& BuildXlib.log        # Build the Ada/X bindings
55
56 cd $Xt
57 BuildXt.VADS        >& BuildXt.log          # Build the Xt intrinsics
58                                             # (also builds the Xmu utilities)
59
60 cd $Widgets
61 BuildWidgets.VADS   >& BuildWidgets.log     # Build the Widget set
62
63 cd $Tests
64 BuildTests.VADS     >& BuildTests.log       # Build the test programs
65
66 cd $Demo
67 BuildDemo.VADS      >& BuildDemo.log        # Build the demo application
```

B.2 Script: BuildAdaXt.var

```
#
# Define the location of the Toolkit source code sub-directories
#
setenv Code <path to the top level code directory>
setenv C $Code/C                    # C utilities source code
setenv Xlib $Code/Xlib              # Ada/Xlib bindings source code
setenv Xt $Code/Xt                  # Ada/Xt Toolkit source code
setenv Xmu $Code/Xmu                # Ada/X Miscellaneous Utilities source code
setenv Widgets $Code/Widgets        # Sample Widgets source code
setenv Tests $Code/Tests            # Test programs source code
setenv Demo $Code/Demo              # Demo program source code
#
# Define the location of the C and Xlib archives
#
setenv CLIB $C/lib.a
setenv XLIB <path to the X11R3 Xlib object archive (e.g. /usr/lib/libX11.a)>
#
# Establish a path to the VADS compilation system
#
setenv Vadspath <path to VADS version 5.5>
set path=($path $Vadspath/bin)
#
# Establish a path to the TeleSoft compilation system
#
setenv TELEGEN2 <path to TeleSoft version 1.4>
setenv TADA $TELEGEN2/bin/ada
```

B.3 Script: Xlib/vads/BuildXlib.VADS

```
1 #! /bin/csh -fx
2
3 #
4 # Compile the system and compiler dependent packages
5 #
6 ada -e -w system_environment_.a
7 ada -e -w compiler_dependent_.a
8 ada -e -w os_dependent_.a
9 ada -e -w compiler_dependent.a
10 ada -e -w string_encaps_.a
11 ada -e -w system_environment.a
12 ada -e -w string_encaps.a
13
14 #
15 # Compile the Xlib bindings specification
16 #
17 ada -e -w  x_lib_.a
18
19 #
20 # Compile the interface bindings to C Xlib code
21 #
22 ada -e -w  x_int_.a
23
24 #
25 # Compile the X key bindings
26 #
27 ada -e -w  x_keysyms_.a
28
29 #
30 # Compile the Xlib bindings body
31 #
32 ada -e -w  x_lib.a
33
34 #
35 # Compile the subunits to Xlib bindings body
36 #
37 ada -e -w  x_fonts.a
38 ada -e -w  x_graphic.a
39 ada -e -w  x_atoms.a
40 ada -e -w  x_colors.a
41 ada -e -w  x_cursors.a
42 ada -e -w  x_cutpaste.a
43 # package body Events is now in x_lib.a, not separate
```

```
44 # ada -e -w  x_events.a
45 ada -e -w  x_keyboard.a
46 ada -e -w  x_regions.a
47 ada -e -w  x_win_mgr.a
48 ada -e -w  x_resmgmt.a
49 ada -e -w  fromcstringSU.a
50 ada -e -w  opendispSU.a
51
52 ada -e -w  get_window.a
53
54 #
55 # Compile and link the Xlib "hello_world" test program
56 #
57 ada -e -w hw.a
58
59 # XLIB and CLIB are installation dependent variables which can be found
60 # in the top level make script of this delivery
61 a.ld hello_world -o hello_world $XLIB $CLIB -lresolv
62
```

B.4 Script: Xt/vads/BuildXt.VADS

```
#! /bin/csh -fx

#
# utility packages
#
ada -w -e installation_.a
ada -w -e realloc_array_.a
ada -w -e realloc_array.a
ada -w -e stringops_.a
ada -w -e stringops.a
ada -w -e cursor_cache_.a

ada -w -e  renamed_xlib_types_.a
ada -w -e  dispatch_interfaces_.a
ada -w -e  device_.a
ada -w -e  device.a
ada -w -e  intrinsics_.a
ada -w -e  xt_stringdefs_.a
ada -w -e  external_support_code_.a
ada -w -e  tm_constants_.a
ada -w -e  resourceP_.a
ada -w -e  resource_generic_int_.a
ada -w -e  resource_generic_int.a
ada -w -e  xt_resource_type_instances_.a
ada -w -e  xrm_get_resources_.a
ada -w -e  xrm_get_resources.a

#
# built-in widget types
#
ada -w -e  object_public_.a
ada -w -e  object_private_.a
ada -w -e  rect_obj_public_.a
ada -w -e  rect_obj_private_.a
ada -w -e  window_obj_public_.a
ada -w -e  window_obj_private_.a
ada -w -e  comp_obj_public_.a
ada -w -e  comp_obj_private_.a
ada -w -e  core_public_.a
ada -w -e  core_private_.a
ada -w -e  composite_public_.a
ada -w -e  composite_private_.a
ada -w -e  constraint_public_.a
```

```
44 ada -w -e  constraint_private_.a
45 ada -w -e  shell_public_.a
46 ada -w -e  shell_private_.a
47
48 ada -w -e  intrinsics.a
49
50 ada -w -e xt_procedure_typesSU.a
51
52 #
53 # compile generic bodies before any possible instantiations
54 #
55 ada -w -e  xt_accept_focus_procs.SU2.a
56 ada -w -e  xt_action_procs.SU2.a
57 ada -w -e  xt_almost_procs.SU2.a
58 ada -w -e  xt_args_func.SU2.a
59 ada -w -e  xt_args_procs.SU2.a
60 ada -w -e  xt_async_handler.SU2.a
61 ada -w -e  xt_callback_procs.SU2.a
62 ada -w -e  xt_case_procs.SU2.a
63 ada -w -e  xt_convert_selection_procs.SU2.a
64 ada -w -e  xt_converter_procs.SU2.a
65 ada -w -e  xt_create_popup_child_procs.SU2.a
66 ada -w -e  xt_error_handler_procs.SU2.a
67 ada -w -e  xt_error_msg_handler_procs.SU2.a
68 ada -w -e  xt_event_handler_procs.SU2.a
69 ada -w -e  xt_expose_procs.SU2.a
70 ada -w -e  xt_geometry_handler_procs.SU2.a
71 ada -w -e  xt_init_procs.SU2.a
72 ada -w -e  xt_input_callback_procs.SU2.a
73 ada -w -e  xt_key_procs.SU2.a
74 ada -w -e  xt_lose_selection_procs.SU2.a
75 ada -w -e  xt_order_procs.SU2.a
76 ada -w -e  xt_procs.SU2.a
77 ada -w -e  xt_realize_procs.SU2.a
78 ada -w -e  xt_resource_default_procs.SU2.a
79 ada -w -e  xt_selection_callback_procs.SU2.a
80 ada -w -e  xt_selection_done_procs.SU2.a
81 ada -w -e  xt_set_value_func.SU2.a
82 ada -w -e  xt_string_procs.SU2.a
83 ada -w -e  xt_timer_callback_procs.SU2.a
84 ada -w -e  xt_widget_class_procs.SU2.a
85 ada -w -e  xt_widget_procs.SU2.a
86 ada -w -e  xt_work_procs.SU2.a
87
88 #
```

```
# intrinsics body subunits level 1
#
ada -w -e xt_utilitiesSU.a
ada -w -e xt_translation_managementSU.a
ada -w -e type_conversionsSU.a
ada -w -e xt_callbacksSU.a
ada -w -e xt_class_managementSU.a
ada -w -e xt_composite_managementSU.a
ada -w -e xt_event_managementSU.a
ada -w -e xt_geometry_managementSU.a
ada -w -e xt_initializersSU.a
ada -w -e xt_instance_managementSU.a
ada -w -e xt_popup_managementSU.a
ada -w -e xt_selection_managementSU.a
ada -w -e xt_type_opsSU.a

#
# resource mgmt and resource conversion body stuff
#
ada -w -e convertSU.a
ada -w -e xt_resource_managementSU.a
ada -w -e resourceP.a

#
# subunits level 2
#
ada -w -e xt_keycode_to_keysymSU2.a
ada -w -e xt_build_keysym_tableSU2.a
ada -w -e parserSU2.a
ada -w -e int_tm_initSU2.a
ada -w -e tm_eventsSU2.a
ada -w -e tm_late_bindingsSU2.a
ada -w -e tm_stateSU2.a
ada -w -e trans_mgt_semi_privateSU2.a
ada -w -e get_resourcesSU2.a
ada -w -e event_mgt_localSU2.a
ada -w -e selection_mgt_localSU2.a
ada -w -e selection_procedure_types.SU2.a

ada -w -e compute_late_bindingsSU3.a

#
# go and compile Xmu stuff since it needs the intrinsics and
# the shell widget needs it
# ($Xmu and $Xt are installation dependent and can be found in
```

```
134 #  the top level make script of this delivery)
135 #
136 cd $Xmu
137 BuildXmu.VADS >& BuildXmu.log &
138 cd $Xt
139
140 #
141 # built-in widget public bodies
142 #
143 ada -w -e  object_public.a
144 ada -w -e  rect_obj_public.a
145 ada -w -e  window_obj_public.a
146 ada -w -e  comp_obj_public.a
147 ada -w -e  core_public.a
148 ada -w -e  composite_public.a
149 ada -w -e  constraint_public.a
150 ada -w -e  shell_public.a
151 ada -w -e  shell_internalsSU.a
```

B.5 Script: Xmu/vads/BuildXmu.VADS

```
#! /bin/csh -fx

#
# hard-coded pathnames for Xmu routines
#
ada -e -w xmu_paths_.a

#
# the Xmu package itself
#
ada -e -w xmu_.a
ada -e -w xmu.a

ada -e -w xt_float_r_.a
ada -e -w xt_orientation_r_.a
ada -e -w xt_edge_r_.a
ada -e -w xt_justify_r_.a
ada -e -w selection_array_r_.a
ada -e -w mask_r_.a
ada -e -w xt_text_edit_type_r_.a
ada -e -w xmu_cvt_string_to_widget_.a

#
# subunits of Xmu
#
ada -e -w xmu_cvt_string_to_cursorSU.a

#
# other bodies
#
ada -e -w xmu_cvt_string_to_widget.a
ada -e -w xt_orientation_r.a
ada -e -w xt_edge_r.a
ada -e -w xt_justify_r.a
ada -e -w mask_r.a
ada -e -w xt_text_edit_type_r.a

#
# new atoms and atom routines for Xmu
#
ada -e -w xmu_atoms_.a
ada -e -w xmu_atoms.a
```

B.6 Script: Widgets/vads/BuildWidgets.VADS

```
1 #! /bin/csh -fx
2
3 #
4 # compile Xaw GrayPixmap.c stuff
5 #
6 ada -e -w pixmaps_.a
7 ada -e -w pixmaps.a
8 ada -e -w create_tile_.a
9
10 #
11 # compile misc specs
12 #
13 ada -e -w xw_constants_.a
14 ada -e -w xw_traversal_procs_.a
15 ada -e -w xw_res_convert_.a
16
17 #
18 # compile public and private specs
19 #
20 ada -e -w simple_public_.a
21 ada -e -w simple_private_.a
22
23 ada -e -w scroll_public_.a
24 ada -e -w scroll_private_.a
25
26 ada -e -w label_public_.a
27 ada -e -w label_private_.a
28
29 ada -e -w command_public_.a
30 ada -e -w command_private_.a
31
32 ada -e -w form_public_.a
33 ada -e -w form_private_.a
34
35 ada -e -w viewport_public_.a
36 ada -e -w viewport_private_.a
37
38 ada -e -w xw_manager_public_.a
39 ada -e -w xw_manager_private_.a
40
41 ada -e -w xw_bboard_public_.a
42 ada -e -w xw_bboard_private_.a
43
```

```
#
# compile widget bodies
#
ada -e -w simple_private.a
ada -e -w simple_public.a
ada -e -w label_public.a
ada -e -w command_public.a
ada -e -w scroll_public.a
ada -e -w form_public.a
ada -e -w viewport_public.a
ada -e -w xw_manager_public.a
ada -e -w xw_bboard_public.a

#
# compile misc. bodies
#
ada -e -w xw_res_convert.a
ada -e -w create_tile.a
ada -e -w xw_traversal_procs.a


#
# compile global and support packages for text widgets
#
ada -w -e text_widget_globals_.a
ada -w -e text_widget_support_.a
ada -w -e text_widget_support.a

#
# compile data sources and sinks for the text widgets
#
ada -w -e text_sources_.a
ada -w -e text_sources.a
ada -w -e text_ascii_sink_.a

#
# compile the public and private specs for all the
# text widgets
#
ada -w -e text_public_.a
ada -w -e text_private_.a

ada -w -e ascii_string_public_.a
ada -w -e ascii_string_private_.a

```

```
ada -w -e ascii_disk_public_.a
ada -w -e ascii_disk_private_.a

ada -w -e dialog_public_.a
ada -w -e dialog_private_.a

#
# some shared routines and other bodies
#
ada -w -e text_edit_semi_private_routines_.a
ada -w -e text_public.a
ada -w -e text_localSU.a
ada -w -e text_check_resize_or_overflowSU2.a
ada -w -e text_replace_textSU2.a
ada -w -e text_insert_charSU2.a
ada -w -e text_edit_semi_private_routines.a
ada -w -e text_ascii_sink.a

#
# local resource type instances
#
ada -w -e xt_edit_configuration_r_.a

#
# widget bodies
#
ada -w -e ascii_string_public.a
ada -w -e ascii_disk_public.a
ada -w -e dialog_public.a

```

B.7 Script: Tests/vads/BuildTests.VADS

```
1 #! /bin/csh -xf
2 #
3 # Compile and link the test programs
4 # (XLIB and CLIB are installation dependent variables which can
5 #  be found in the top level make script of this delivery)
6 #
7
8 ada -e -w test_shell.a
9 a.ld test_shell -o test_shell $XLIB $CLIB -lresolv
10
11 ada -e -w test_label.a
12 a.ld test_label -o test_label $XLIB $CLIB -lresolv
13
14 ada -e -w test_label_callbacks.a
15 a.ld test_label_callbacks -o test_label_callbacks $XLIB $CLIB -lresolv
16
17 ada -e -w test_label_events.a
18 a.ld test_label_events -o test_label_events $XLIB $CLIB -lresolv
19
20 ada -e -w read_trans.a
21 ada -e -w test_command.a
22 a.ld test_command -o test_com $XLIB $CLIB -lresolv
23
24 ada -e -w test_scrollbar.a
25 a.ld test_scroll -o ascroll $XLIB $CLIB -lresolv
26
27 ada -e -w test_bb.a
28 a.ld test_bb -o test_bb $XLIB $CLIB -lresolv
29
30 ada -e -w test_vw.a
31 a.ld test_vw -o test_vw $XLIB $CLIB -lresolv
32
33 ada -e -w test_ascii_string_text.a
34 a.ld test_ascii_string_text -o test_text $XLIB $CLIB -lresolv
35
36 ada -e -w test_ascii_disk_text.a
37 a.ld test_ascii_disk_text -o test_disk_text $XLIB $CLIB -lresolv
38
39 ada -e -w test_dialog.a
40 a.ld test_dialog -o test_dialog $XLIB $CLIB -lresolv
41
```

B.8 Script: Demo/vads/BuildDemo.VADS

```
1 #! /bin/csh -fx
2 #
3 # Demo application
4 #
5
6 ada -e -w demo_except_.a
7 ada -e -w demo_callbacks_.a
8 ada -e -w demo_lines_.a
9 ada -e -w demo_menu_.a
10
11 ada -e -w demo_callbacks.a
12 ada -e -w demo_lines.a
13 ada -e -w demo_menu.a
14
15 ada -e -w demo.a
16
17 a.ld demo -o demo_ada $XLIB $CLIB -lresolv
```

C Appendix C: TeleSoft Build Scripts

C.1 Script: BuildAdaXt.TG2

```
#! /bin/csh -x
#
# Define installation-dependent variables
#
source BuildAdaXt.var
#
# Copy source files into the TeleSoft configuration
#
cp $Xlib/telesoft/* $Xlib
cp $Xt/telesoft/* $Xt
cp $Xmu/vads/* $Xt
cp $Widgets/telesoft/* $Widgets
cp $Tests/telesoft/* $Tests
cp $Demo/telesoft/* $Demo
#
# Build Ada libraries in each sub-directory
#
cd $Xlib
$TELEGEN2/bin/acr -f -m 32000 xlib
cd $Xt
$TELEGEN2/bin/acr -f -m 32000 xt
$TELEGEN2/bin/acr -f -m 32000 bodies
$TELEGEN2/bin/acr -f -m 32000 seconds
cd $Widgets
$TELEGEN2/bin/acr -f -m 32000 widgets
cd $Tests
$TELEGEN2/bin/acr -f -m 32000 test
cd $Demo
$TELEGEN2/bin/acr -f -m 32000 demo
#
# Establish dependencies
#
# Library dependencies are defined in text file "liblst.alb".
# Each code subdirectory contains a "liblst.alb" file that must be edited
# if the dependencies change.
#
# Compile the code
#

cd $C
make            >& Makefile.log     # Build the C utilities
```

```
42
43 cd $Xlib
44 BuildXlib.TG2     >& BuildXlib.log     # Build the Ada/X bindings
45
46 cd $Xt
47 BuildXt.TG2       >& BuildXt.log       # Build the Xt intrinsics
48                                        # (also builds the Xmu utilities)
49
50 cd $Widgets
51 BuildWidgets.TG2 >& BuildWidgets.log # Build the Widget set
52
53 cd $Tests
54 BuildTests.TG2    >& BuildTests.log    # Build the test programs
55
56 cd $Demo
57 BuildDemo.TG2     >& make_Demo.log     # Build the demo application
```

C.2 Script: Xlib/telesoft/BuildXlib.TG2

```
#! /bin/csh -fx

# XLIB, CLIB, TELEGEN2, and TADA are installation dependent variables
# and their values casn be found in the top level make script for this
# delivery

#
# Compile the system and compiler dependent packages
#
$TADA -v -d system_environment_.a
$TADA -v -d compiler_dependent_.a
$TADA -v -d os_dependent_.a
$TADA -v -d compiler_dependent.a
$TADA -v -d string_encaps_.a
$TADA -v -d system_environment.a
$TADA -v -d string_encaps.a

#
# Compile the Xlib bindings specification
#
$TADA -v -d  x_lib_.a

#
# Compile the interface bindings to C Xlib code
#
$TADA -v -d  x_int_.a

#
# Compile the X key bindings
#
$TADA -v -d  x_keysyms_.a

#
# Compile the Xlib bindings body
#
$TADA -v -d  x_lib.a

#
# Compile the subunits to Xlib bindings body
#
$TADA -v -d  x_resmgmt.a
$TADA -v -d  x_fonts.a
$TADA -v -d  x_graphic.a
```

```
44 $TADA -v -d  x_atoms.a
45 $TADA -v -d  x_colors.a
46 $TADA -v -d  x_cursors.a
47 $TADA -v -d  x_cutpaste.a
48 # package body Events is now a part of x_lib.a
49 # $TADA -v -d  x_events.a
50 $TADA -v -d  x_keyboard.a
51 $TADA -v -d  x_regions.a
52 $TADA -v -d  x_win_mgr.a
53 $TADA -v -d  fromcstringSU.a
54 $TADA -v -d  opendispSU.a
55
56 $TADA -v -d  get_window.a
57
58 #
59 # Compile and link the Xlib "hello_world" test program
60 #
61 $TADA -v -d hw.a
62
63 $TELEGEN2/bin/ald -p "$XLIB $CLIB -lresolv" hello_world
64
```

C.3 Script: Xt/telesoft/BuildXt.TG2

```
#! /bin/csh -fx

# XLIB, CLIB, TELEGEN2, and TADA are installation dependent variables
# and their values can be found in the top level make script for this
# delivery

#
# utility packages
#
$TADA -v -d installation_.a
$TADA -v -d realloc_array_.a
$TADA -v -d realloc_array.a
$TADA -v -d stringops_.a
$TADA -v -d stringops.a
$TADA -v -d cursor_cache_.a

#
# main specifications for the Intrinsics
#
$TADA -v -d  renamed_xlib_types_.a
$TADA -v -d  dispatch_interfaces_.a
$TADA -v -d  device_.a
$TADA -v -d  device.a
$TADA -v -d  intrinsics_.a
$TADA -v -d  xt_stringdefs_.a
$TADA -v -d  external_support_code_.a
$TADA -v -d  tm_constants_.a
$TADA -v -d  resourceP_.a
$TADA -v -d  resource_generic_int_.a
$TADA -v -d  resource_generic_int.a
$TADA -v -d  xt_resource_type_instances_.a
$TADA -v -d  xrm_get_resources_.a
$TADA -v -d  xrm_get_resources.a

#
# built-in widget type specifications
#
$TADA -v -d  object_public_.a
$TADA -v -d  object_private_.a
$TADA -v -d  rect_obj_public_.a
$TADA -v -d  rect_obj_private_.a
$TADA -v -d  window_obj_public_.a
$TADA -v -d  window_obj_private_.a
```

```
44 $TADA -v -d  comp_obj_public_.a
45 $TADA -v -d  comp_obj_private_.a
46 $TADA -v -d  core_public_.a
47 $TADA -v -d  core_private_.a
48 $TADA -v -d  composite_public_.a
49 $TADA -v -d  composite_private_.a
50 $TADA -v -d  constraint_public_.a
51 $TADA -v -d  constraint_private_.a
52 $TADA -v -d  shell_public_.a
53 $TADA -v -d  shell_private_.a
54
55 #
56 # Compile the main Intrinsics body
57 #
58 $TADA -l bodies.alb -v -d  intrinsics.a
59
60 #
61 # intrinsics body subunits level 1
62 #
63 $TADA -l bodies.alb -v -d xt_utilitiesSU.a
64 $TADA -l bodies.alb -v -d xt_translation_managementSU.a
65 $TADA -l bodies.alb -v -d type_conversionsSU.a
66 $TADA -l bodies.alb -v -d xt_callbacksSU.a
67 $TADA -l bodies.alb -v -d xt_class_managementSU.a
68 $TADA -l bodies.alb -v -d xt_composite_managementSU.a
69 $TADA -l bodies.alb -v -d xt_event_managementSU.a
70 $TADA -l bodies.alb -v -d xt_geometry_managementSU.a
71 $TADA -l bodies.alb -v -d xt_initializersSU.a
72 $TADA -l bodies.alb -v -d xt_selection_managementSU.a
73 $TADA -l bodies.alb -v -d convertSU.a
74 $TADA -l bodies.alb -v -d xt_resource_managementSU.a
75 $TADA -l bodies.alb -v -d resourceP.a
76 $TADA -l bodies.alb -v -d xt_instance_managementSU.a
77 $TADA -l bodies.alb -v -d xt_popup_managementSU.a
78 $TADA -l bodies.alb -v -d xt_type_opsSU.a
79
80 #
81 # Compile the Xmu utilities
82 #
83 # hard-coded pathnames for Xmu routines
84 #
85 $TADA -l bodies.alb -v -d xmu_paths_.a
86 #
87 # the Xmu package itself
88 #
```

```
$TADA -l bodies.alb -v -d xmu_.a
$TADA -l bodies.alb -v -d xmu.a
#
# public resource type specifications
#
$TADA -l bodies.alb -v -d xt_float_r_.a
$TADA -l bodies.alb -v -d xt_orientation_r_.a
$TADA -l bodies.alb -v -d xt_edge_r_.a
$TADA -l bodies.alb -v -d xt_justify_r_.a
$TADA -l bodies.alb -v -d selection_array_r_.a
$TADA -l bodies.alb -v -d mask_r_.a
$TADA -l bodies.alb -v -d xt_text_edit_type_r_.a
#
$TADA -l bodies.alb -v -d xmu_cvt_string_to_widget_.a
#
# subunits of Xmu
#
$TADA -l bodies.alb -v -d xmu_cvt_string_to_cursorSU.a
#
# other bodies
#
$TADA -l bodies.alb -v -d xmu_cvt_string_to_widget.a
$TADA -l bodies.alb -v -d xt_orientation_r.a
$TADA -l bodies.alb -v -d xt_edge_r.a
$TADA -l bodies.alb -v -d xt_justify_r.a
$TADA -l bodies.alb -v -d mask_r.a
$TADA -l bodies.alb -v -d xt_text_edit_type_r.a
#
# new atoms and atom routines for Xmu
#
$TADA -l bodies.alb -v -d xmu_atoms_.a
$TADA -l bodies.alb -v -d xmu_atoms.a

#
# Compile the built-in widget public bodies
#
$TADA -l bodies.alb -v -d  object_public.a
$TADA -l bodies.alb -v -d  rect_obj_public.a
$TADA -l bodies.alb -v -d  window_obj_public.a
$TADA -l bodies.alb -v -d  comp_obj_public.a
$TADA -l bodies.alb -v -d  core_public.a
$TADA -l bodies.alb -v -d  composite_public.a
$TADA -l bodies.alb -v -d  constraint_public.a
$TADA -l bodies.alb -v -d  shell_public.a
$TADA -l bodies.alb -v -d  shell_internalsSU.a
```

```
134
135 # compile generic bodies before any possible instantiations
136 # (These would all be separate sub-units except TeleSoft has a bug about
137 #  elaborating generic bodies in sub-units, therefore, they are all
138 #  collapsed into the intrinsics package body :-)
139 # $TADA -l seconds.alb -v -d  xt_accept_focus_procs.SU2.a
140 # $TADA -l seconds.alb -v -d  xt_action_procs.SU2.a
141 # $TADA -l seconds.alb -v -d  xt_almost_procs.SU2.a
142 # $TADA -l seconds.alb -v -d  xt_args_func.SU2.a
143 # $TADA -l seconds.alb -v -d  xt_args_procs.SU2.a
144 # $TADA -l seconds.alb -v -d  xt_async_handler.SU2.a
145 # $TADA -l seconds.alb -v -d  xt_callback_procs.SU2.a
146 # $TADA -l seconds.alb -v -d  xt_case_procs.SU2.a
147 # $TADA -l seconds.alb -v -d  xt_convert_selection_procs.SU2.a
148 # $TADA -l seconds.alb -v -d  xt_converter_procs.SU2.a
149 # $TADA -l seconds.alb -v -d  xt_create_popup_child_procs.SU2.a
150 # $TADA -l seconds.alb -v -d  xt_error_handler_procs.SU2.a
151 # $TADA -l seconds.alb -v -d  xt_error_msg_handler_procs.SU2.a
152 # $TADA -l seconds.alb -v -d  xt_event_handler_procs.SU2.a
153 # $TADA -l seconds.alb -v -d  xt_expose_procs.SU2.a
154 # $TADA -l seconds.alb -v -d  xt_geometry_handler_procs.SU2.a
155 # $TADA -l seconds.alb -v -d  xt_init_procs.SU2.a
156 # $TADA -l seconds.alb -v -d  xt_input_callback_procs.SU2.a
157 # $TADA -l seconds.alb -v -d  xt_key_procs.SU2.a
158 # $TADA -l seconds.alb -v -d  xt_lose_selection_procs.SU2.a
159 # $TADA -l seconds.alb -v -d  xt_order_procs.SU2.a
160 # $TADA -l seconds.alb -v -d  xt_procs.SU2.a
161 # $TADA -l seconds.alb -v -d  xt_realize_procs.SU2.a
162 # $TADA -l seconds.alb -v -d  xt_resource_default_procs.SU2.a
163 # $TADA -l seconds.alb -v -d  xt_selection_callback_procs.SU2.a
164 # $TADA -l seconds.alb -v -d  xt_selection_done_procs.SU2.a
165 # $TADA -l seconds.alb -v -d  xt_set_value_func.SU2.a
166 # $TADA -l seconds.alb -v -d  xt_string_procs.SU2.a
167 # $TADA -l seconds.alb -v -d  xt_timer_callback_procs.SU2.a
168 # $TADA -l seconds.alb -v -d  xt_widget_class_procs.SU2.a
169 # $TADA -l seconds.alb -v -d  xt_widget_procs.SU2.a
170 # $TADA -l seconds.alb -v -d  xt_work_procs.SU2.a
171
172 # -- subunits level 2
173 $TADA -l seconds.alb -v -d xt_keycode_to_keysymSU2.a
174 $TADA -l seconds.alb -v -d xt_build_keysym_tableSU2.a
175 $TADA -l seconds.alb -v -d parserSU2.a
176 $TADA -l seconds.alb -v -d int_tm_initSU2.a
177 $TADA -l seconds.alb -v -d tm_eventsSU2.a
178 $TADA -l seconds.alb -v -d tm_late_bindingsSU2.a
```

```
179 $TADA -l seconds.alb -v -d tm_stateSU2.a
180 $TADA -l seconds.alb -v -d get_resourcesSU2.a
181 $TADA -l seconds.alb -v -d event_mgt_localSU2.a
182 $TADA -l seconds.alb -v -d selection_mgt_localSU2.a
183 $TADA -l seconds.alb -v -d selection_procedure_types.SU2.a
184
185 $TADA -l seconds.alb -v -d trans_mgt_semi_privateSU2.a
186 $TADA -l seconds.alb -v -d compute_late_bindingsSU3.a
```

C.4 Script: Widgets/telesoft/BuildWidgets.TG2

```
1 #! /bin/csh -fx
2
3 # CLIB, XLIB, TELEGEN2, and TADA are installation specific variables
4 # and their values can be found in the top level make script of
5 # this delivery
6
7 #
8 # compile Xaw GrayPixmap.c stuff
9 #
10 $TADA -v -d pixmaps_.a
11 $TADA -v -d pixmaps.a
12 $TADA -v -d create_tile_.a
13
14 #
15 # compile misc specs
16 #
17 $TADA -v -d xw_constants_.a
18 $TADA -v -d xw_traversal_procs_.a
19 $TADA -v -d xw_traversal_procs.a
20 $TADA -v -d xw_res_convert_.a
21
22
23 #
24 # compile public and private specs
25 #
26 $TADA -v -d simple_public_.a
27 $TADA -v -d simple_private_.a
28
29 $TADA -v -d scroll_public_.a
30 $TADA -v -d scroll_private_.a
31
32 $TADA -v -d label_public_.a
33 $TADA -v -d label_private_.a
34
35 $TADA -v -d command_public_.a
36 $TADA -v -d command_private_.a
37
38 $TADA -v -d form_public_.a
39 $TADA -v -d form_private_.a
40
41 $TADA -v -d viewport_public_.a
42 $TADA -v -d viewport_private_.a
43
```

```
44 $TADA -v -d xw_manager_public_.a
45 $TADA -v -d xw_manager_private_.a
46
47 $TADA -v -d xw_bboard_public_.a
48 $TADA -v -d xw_bboard_private_.a
49
50 #
51 # compile widget bodies
52 #
53 $TADA -v -d simple_private.a
54 $TADA -v -d simple_public.a
55 $TADA -v -d label_public.a
56 $TADA -v -d command_public.a
57 $TADA -v -d scroll_public.a
58 $TADA -v -d form_public.a
59 $TADA -v -d viewport_public.a
60 $TADA -v -d xw_manager_public.a
61 $TADA -v -d xw_bboard_public.a
62
63 #
64 # compile misc. bodies
65 #
66 $TADA -v -d xw_res_conv rt.a
67 $TADA -v -d create_tile.a
68 #$TADA -v -d xw_traversal_procs.a MOVED UP TO BELOW SPEC COMPILE
69
70 #
71 # compile global and support packages for text widgets
72 #
73 $TADA -v -d text_widget_globals_.a
74 $TADA -v -d text_widget_support_.a
75 $TADA -v -d text_widget_support.a
76
77 #
78 # compile data sources and sinks for the text widgets
79 #
80 $TADA -v -d text_sources_.a
81 $TADA -v -d text_sources.a
82 $TADA -v -d text_ascii_sink_.a
83
84 #
85 # compile the public and private specs for all the
86 # text widgets
87 #
88 $TADA -v -d text_public_.a
```

```
89 $TADA -v -d text_private_.a
90
91 $TADA -v -d ascii_string_public_.a
92 $TADA -v -d ascii_string_private_.a
93
94 $TADA -v -d ascii_disk_public_.a
95 $TADA -v -d ascii_disk_private_.a
96
97 $TADA -v -d dialog_public_.a
98 $TADA -v -d dialog_private_.a
99
100 #
101 # some shared routines and other bodies
102 #
103 $TADA -v -d text_edit_semi_private_routines_.a
104 $TADA -v -d text_public.a
105 $TADA -v -d text_localSU.a
106 $TADA -v -d text_check_resize_or_overflowSU2.a
107 $TADA -v -d text_replace_textSU2.a
108 $TADA -v -d text_insert_charSU2.a
109 $TADA -v -d text_edit_semi_private_routines.a
110 $TADA -v -d text_ascii_sink.a
111
112 #
113 # local resource type instances
114 #
115 $TADA -v -d xt_edit_configuration_r_.a
116
117 #
118 # widget bodies
119 #
120 $TADA -v -d ascii_string_public.a
121 $TADA -v -d ascii_disk_public.a
122 $TADA -v -d dialog_public.a
123
124
```

C.5 Script: Tests/telesoft/BuildTests.TG2

```
1 #! /bin/csh -xf
2
3 # XLIB, CLIB, TELEGEN2, and TADA are installation dependent variables
4 # and their values can be found in the top level make script of
5 # this delivery.
6
7 $TADA -v -d test_shell.a
8 $TELEGEN2/bin/ald -o test_shell -p "$XLIB $CLIB -lresolv" test_shell
9
10 $TADA -v -d test_label.a
11 $TELEGEN2/bin/ald -o test_label -p "$XLIB $CLIB -lresolv" test_label
12
13 $TADA -v -d test_label_callbacks.a
14 $TELEGEN2/bin/ald -o test_label_callbacks -p "$XLIB $CLIB -lresolv" test_label_callbacks
15
16 $TADA -v -d test_label_events.a
17 $TELEGEN2/bin/ald -o test_label_events -p "$XLIB $CLIB -lresolv" test_label_events
18
19 $TADA -v -d read_trans.a
20 $TADA -v -d test_command.a
21 $TELEGEN2/bin/ald -o test_command -p "$XLIB $CLIB -lresolv" test_command
22
23 $TADA -v -d test_scrollbar.a
24 $TELEGEN2/bin/ald -o test_scroll -p "$XLIB $CLIB -lresolv" test_scroll
25
26 $TADA -v -d test_bb.a
27 $TELEGEN2/bin/ald -o test_bb -p "$XLIB $CLIB -lresolv" test_bb
28
29 $TADA -v -d test_vw.a
30 $TELEGEN2/bin/ald -o test_vw -p "$XLIB $CLIB -lresolv" test_vw
31
32 $TADA -v -d test_ascii_string_text.a
33 $TELEGEN2/bin/ald -o test_text -p "$XLIB $CLIB -lresolv" test_ascii_string_text
34
35 $TADA -v -d test_ascii_disk_text.a
36 $TELEGEN2/bin/ald -o test_disk_text -p "$XLIB $CLIB -lresolv" test_ascii_disk_text
37
38 $TADA -v -d test_dialog.a
39 $TELEGEN2/bin/ald -o test_dialog -p "$XLIB $CLIB -lresolv" test_dialog
```

C.6 Script: Demo/telesoft/BuildDemo.TG2

```
1 #! /bin/csh -fx
2
3 # XLIB, CLIB, TELEGEN2, and TADA are installation dependent variables and
4 # their values are in the top level make script of this delivery.
5
6 $TADA -vd demo_except_.a
7 $TADA -vd demo_callbacks_.a
8 $TADA -vd demo_lines_.a
9 $TADA -vd demo_menu_.a
10
11 $TADA -vd demo_callbacks.a
12 $TADA -vd demo_lines.a
13 $TADA -vd demo_menu.a
14
15 $TADA -vd demo.a
16
17 $TELEGEN2/bin/ald -o demo_ada -p "$XLIB $CLIB -lresolv" demo
18
19
```