

**Best
Available
Copy**

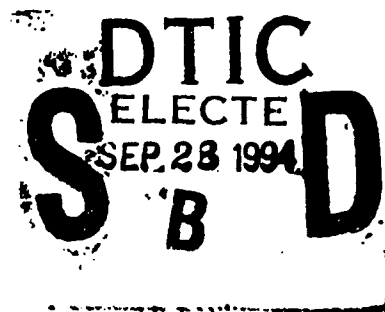
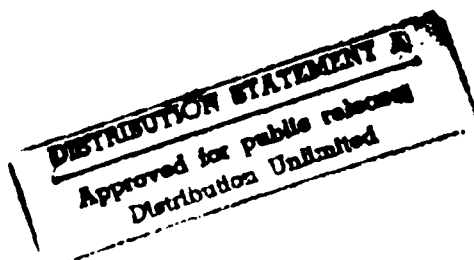
AD-A284 964



TASK: UU03
CDRL: 05156
March 1993

Reuse Library Framework Source Code Release Version 4.1 Version Description Document

Informal Technical Data



STARS-UC-05156/015/00
March 1993

94-30886

94 2 27 06 2

TASK: UU03
CDRL: 05156
March 1993

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

Reuse Library Framework
Source Code Release Version 4.1
SunOS Implementation

STARS-UC-05156/015/00
March 1993

Data Type: A005, Informal Technical Data

CONTRACT NO. F19628-88-D-0031
Delivery Order 0003

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

Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By _____	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

DTIC QUALITY INSPECTED 3

Distribution Statement "A"
per DoD Directive 5230.24
Authorized for public release; Distribution is unlimited.

Data ID: STARS-UC-05156/013/00

Distribution Statement "A"
per DoD Directive 5230.24
Authorized for public release; Distribution is unlimited.

Copyright 1993, Paramax Systems Corporation, Reston, Virginia
Copyright is assigned to the U.S. Government, upon delivery thereto, in accordance with
the DFAR Special Works Clause.

Developed by: Paramax Systems Corporation

This software, developed under the Software Technology for Adaptable, Reliable Systems (STARS) program, is approved for release under Distribution "A" of the Scientific and Technical Information Program Classification Scheme (DoD Directive 5230.24) unless otherwise indicated. Sponsored by the U.S. Defense Advanced Research Projects Agency (DARPA) under contract F19628-88-D-0031, the STARS program is supported by the military services, SEI, and MITRE, with the U.S. Air Force as the executive contracting agent.

Permission to use, copy, modify, and comment on this software and its documentation for purposes stated under Distribution "A" and without fee is hereby granted, provided that this notice appears in each whole or partial copy. This software retains Contractor indemnification to The Government regarding copyrights pursuant to the above referenced STARS contract. The Government disclaims all responsibility against liability, including costs and expenses for violation of proprietary rights, or copyrights arising out of the creation or use of this software.

In addition, the Government, Paramax, and its subcontractors disclaim all warranties with regard to this software, including all implied warranties of merchantability and fitness, and in no event shall the Government, Paramax, or its subcontractor(s) be liable for any special, indirect or consequential damages or any damages whatsoever resulting from the loss of use, data, or profits, whether in action of contract, negligence or other tortious action, arising in connection with the use or performance of this software.

TASK: UU03
CDRL: 05156
March 1993

VERSION DESCRIPTION DOCUMENT
Reuse Library Framework
Source Code Release Version 4.1
SunOS Implementation

Approvals:

Task Manager *Richard E. Creps*

Date

(Signatures on File)

VERSION DESCRIPTION DOCUMENT

Reuse Library Framework

Source Code Release Version 4.1

SunOS Implementation

Change Record:

<i>Data ID</i>	<i>Description of Change</i>	<i>Date</i>	<i>Approval</i>
STARS-UC-05156/013/00	Successor Volume: Upgrade for software version 4.1	25 February 1993	<i>on file</i>
STARS-UC-05156/005/00	Successor Volume: Upgrade for software version 4.0	30 November 1992	<i>on file</i>
STARS-TC-03064/004/00	Successor Volume: Upgrade for software version 3.0	23 January 1992	<i>on file</i>
STARS-SC-03064/003/00	Successor Volume: Upgrade for documentation and procedures	11 October 1991	<i>on file</i>
STARS-SC-03064/002/00	Successor Volume: Upgrade for software version 2.2	06 September 1991	<i>on file</i>
STARS-SC-03064/001/00	Original Issue	26 February 1991	<i>on file</i>

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.					
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE		3. REPORT TYPE AND DATES COVERED Informal Technical Report	
4. TITLE AND SUBTITLE <i>Reuse Library Framework</i>				5. FUNDING NUMBERS F19628-88-D-0031	
6. AUTHOR(S) Paramax Corporation					
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Paramax Corporation 1210 Sunrise Valley Drive Reston, VA 22090				8. PERFORMING ORGANIZATION REPORT NUMBER STARS-UC-05156/015/00	
9. SPONSORING, MONITORING AGENCY NAME(S) AND ADDRESS(ES) Department of the Air Force Headquarter, Electronic Systems Hanscom AFB, MA 01731-5000				10. SPONSORING, MONITORING AGENCY REPORT NUMBER 05156	
11. SUPPLEMENTARY NOTES					
12a. DISTRIBUTION / AVAILABILITY STATEMENT Distribution "A"				12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) The Reuse Library Framework (RLF) is an Ada system designed and implemented to support the production and installation of domain-specific software library systems. The RLF is based on two fundamental subsystems: AdaKNET (Ada Knowledge NETwork) and AdaTAU (TAU is an acronym for Think Ask Update) which are knowledge representation and inferencing systems derived from systems previously developed by Unisys in Prolog. These subsystems are supported by an integrating framework to allow them to be used in combination with each other. AdaKNET and AdaTAU are also equipped with interface specification languages (Library Model Definition Language (LMDL) and Rule Base Definition Language (RBDL) respectively) that are used to initialize domain models that describe the library (or application) domain. In addition to the support of library systems, the RLF was used to develop a prototype Ada unit test assistant during the STARS Foundations period and has been applied to the representation of software and reuse process models which are themselves machine processable.					
14. SUBJECT TERMS				15. NUMBER OF PAGES 86	
				16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT SAR		

Contents

1	SCOPE	1
1.1	Identification	1
1.2	System Overview	1
Δ 2	RELATED SOFTWARE	1
3	VERSION DESCRIPTION	1
3.1	Inventory of Contents	1
3.1.1	Directory: <i>docs</i>	2
3.1.1.1	Sub-directory: <i>manuals</i>	3
3.1.1.1.1	RLF Administrator's Manual.	3
3.1.1.1.2	RLF Installation Guide.	3
3.1.1.1.3	RLF Modeler's Manual.	3
3.1.1.1.4	RLF User's Manual.	3
3.1.1.2	Sub-directory: <i>tutorials</i>	3
3.1.1.2.1	RLF User Tutorial.	3
3.1.1.2.2	RLF Administrator Tutorial.	3
3.1.1.2.3	RLF Modeler Tutorial.	4
3.1.2	Directory: <i>bin</i>	4
3.1.3	Directory: <i>models</i>	4
3.1.3.1	Sub-directory: <i>models/ada_x</i>	4
3.1.3.2	Sub-directory: <i>models/animals</i>	4
3.1.3.3	Sub-directory: <i>models/asw</i>	5
3.1.3.4	Sub-directory: <i>models/common_data_model</i>	5
3.1.3.5	Sub-directory: <i>models/demo_actions</i>	5
3.1.3.6	Sub-directory: <i>models/window_manager</i>	5
3.1.3.7	Sub-directory: <i>models/software_technology</i>	5
3.1.3.8	Sub-directory: <i>models/sort_and_search</i>	5
3.1.4	Directory: <i>code</i>	5
3.1.4.1	Sub-directory: <i>code/Common</i>	5
3.1.4.2	Sub-directory: <i>code/Adaknet</i>	6
3.1.4.3	Sub-directory: <i>code/Adatau</i>	6
3.1.4.4	Sub-directory: <i>code/Ada_Actions</i>	6
3.1.4.5	Sub-directory: <i>code/Hybrid</i>	6
3.1.4.6	Sub-directory: <i>code/Library_Manager</i>	6
3.1.4.7	Sub-directory: <i>code/Rlf_Gb</i>	6
3.1.4.8	Sub-directory: <i>code/Lmdl</i>	6
3.1.4.9	Sub-directory: <i>code/Rbdl</i>	7
Δ 3.2	Changes Installed for Version 4.1	7
Δ 3.2.1	The PCTE version of RLF	7
3.2.2	<i>Library_Manager</i> Application Refinement	9
3.3	Adaptation Data	10
3.3.1	Operating Environment	10

3.3.2	Development Environment	10
3.3.3	Configuration-Unique Data	11
3.3.3.1	Software Dependencies.	11
3.3.3.2	OS Dependencies.	11
3.3.3.3	Defaults.	11
3.4	Interface Compatibility	12
3.4.1	Previously Built RLF Libraries	12
3.4.2	Libraries Built with Different Compilers	12
3.5	Installation and Usage Instructions	12
3.5.1	Invoking RLF Applications	12
3.6	Potential Problems	13
3.6.1	System Memory Limitations	13
3.6.2	<i>Graphical_Browser</i> Known Problems	13
3.7	Enhancements	13
4	USER FEEDBACK	14
5	NOTES	15
A	Appendix: Inventory of Contents	16
B	Appendix: RLF Start-up Files	32
B.1	Sample RLF .rlfrc Start-up File	32
B.1.1	File: .rlfrc	32
B.2	RLF Graphical Browser Start-up Script	33
B.2.1	Script: RLF_GB	33
C	Appendix: Installation Scripts	38
C.1	SunAda Support File	38
C.1.1	File: Build_RLF.var	38
C.2	SunAda Scripts for Installing RLF	47
C.2.1	Script: Install_Rlf_src	47
C.2.2	Script: Build_RLF.csh	52
C.2.3	Script: Build_Ada_Libraries.csh	56
C.2.4	Script: Build_Rlfdir.csh	58
C.2.5	Script: Build_Lmdl.csh	69
C.2.6	Script: Build_Rbdl.csh	72
C.2.7	Script: Build_Library_Manager.csh	75
C.2.8	Script: Build_Rlf_Gb.csh	76
C.3	Scripts for Building Sample Networks	79
C.3.1	Script: Build_Ada_X_Lib.csh	79
C.3.2	Script: Build_Animals_Lib.csh	80
C.3.3	Script: Build_Asw_Lib.csh	81
C.3.4	Script: Build_Common_Data_Model_Lib.csh	82
C.3.5	Script: Build_Demo_Actions_Lib.csh	83
C.3.6	Script: Build_Move_Domain_Lib.csh	84

C.3.7	Script: Build_SW_Tech_Lib.csh	85
C.3.8	Script: Build_Sort_And_Search_Lib.csh	86

1 SCOPE

1.1 Identification

Version Description Document,
Reuse Library Framework (RLF),
Version 4.1,
SunOS Implementation

1.2 System Overview

The Reuse Library Framework (RLF) is an Ada system designed and implemented to support the production and installation of domain-specific software library systems. The RLF is based on two fundamental subsystems: AdaKNET (Ada Knowledge NETwork) and AdaTAU (TAU is an acronym for Think-Ask-Update) which are knowledge representation and inferencing systems derived from systems previously developed by Unisys in Prolog. These subsystems are supported by an integrating framework to allow them to be used in combination with each other. AdaKNET and AdaTAU are also equipped with interface specification languages (Library Model Definition Language (LMDL) and Rule Base Definition Language (RBDL) respectively) that are used to initialize domain models that describe the library (or application) domain. In addition to the support of library systems, the RLF was used to develop a prototype Ada unit test assistant during the STARS Foundations period and has been applied to the representation of software and reuse process models which are themselves machine processable.

Δ 2 RELATED SOFTWARE

In order to build the RLF *Graphical_Browser*, it is necessary to have installed the X Window System Release 4 (X11R4), OSF/Motif version 1.1, SERC Ada/Motif version 1.1, and the STARS Reusable Graphical Browser (RGB) version 1.1. To build the PCTE object management system version of RLF it is also necessary to have installed the Emeraude PCTE implementation version 12.3 and the STARS Ada/PCTE bindings version 0.3.

3 VERSION DESCRIPTION

3.1 Inventory of Contents

This release includes a version of the RLF which can be hosted the UNIX operating system or the Emeraude PCTE implementation. A README file is provided to inform the user of useful places to look for information on RLF documentation and registration. Source code for this version resides in the directory *code*. This directory contains the following subdirectories: *Common*, *Adaknet*, *Adatau*, *Ada_Actions*, *Hybrid*, *Library_Manager*, *Rlf_Gb*.

Lmdl, and *Rbdl*. This version also contains *models*, *bin*, and *docs* directories. The contents of these subdirectories are described in the following sections.

The distribution is organized as follows:

```
bin
bin/bitmaps
code
code/Ada_Actions
code/Adaknet
code/Adatau
code/Common
code/Hybrid
code/Library_Manager
code/Lmdl
code/Rbdl
code/Rlf_Gb
docs
docs/manuals
docs/tutorials
models
models/ada_x
models/ada_x/Text
models/ada_x/Text/Widgets
models/ada_x/Text/Xlib
models/ada_x/Text/Xmu
models/ada_x/Text/Xt
models/animals
models/animals/Text
models/asw
models/asw/Text
models/common_data_model
models/common_data_model/Text
models/demo_actions
models/demo_actions/Text
models/demo_actions/Text/sounds
models/demo_actions/Text/xbm
models/window_manager
models/window_manager/Text
models/software_technology
models/software_technology/Text
models/sort_and_search
models/sort_and_search/Text
```

A complete listing of the files in this distribution is included in Appendix A.

3.1.1 Directory: *docs*

The two sub-directories of *docs* contain the RLF manuals and tutorials currently delivered in this RLF release.

3.1.1.1 Sub-directory: *manuals*

3.1.1.1.1 RLF Administrator's Manual. The *RLF Administrator's Manual* provides the information necessary for an RLF reuse library administrator to install, modify, and maintain a reuse library hosted on RLF.

3.1.1.1.2 RLF Installation Guide. The *RLF Installation Guide* informs the user how to install, build and start up the STARS RLF and its user interface applications, namely the *RLF Graphical_Browser* and the *RLF Library_Manager*.

3.1.1.1.3 RLF Modeler's Manual. The *RLF Modeler's Manual* provides the information necessary for an RLF reuse library domain modeler to model, encode, and build an RLF reuse library specification and the library itself. It also defines how to model, encode, and install the RLF library advice modules called "inferencers."

3.1.1.1.4 RLF User's Manual. The *RLF User's Manual* describes the use and basic customization of the *Graphical_Browser* application. The reader is not expected to be a programmer, but familiarity with the UNIX C shell, and basic X Window System operations using the Motif Window Manager (mwm) or some other window manager is assumed. Some explanation of RLF concepts is provided, but only at an elementary level.

3.1.1.2 Sub-directory: *tutorials* This directory contains three Postscript file representations of the contents of three RLF training packages that will be used as hand-out material in support of the delivery of RLF training sessions. While the documents can be read on their own, and are formatted in an article-style format, they are oriented more for a presentation of the material by a speaker using transparencies. The tutorials are also designed to be supplemented by in-class demonstrations of RLF software and the conducting and monitoring of both in-class and out-of-class student exercises using the software.

3.1.1.2.1 RLF User Tutorial. The *RLF User Tutorial* presents a survey of the usage of the *RLF Graphical_Browser* application which will enable new RLF users to quickly learn the user interface and the various RLF features which it presents.

3.1.1.2.2 RLF Administrator Tutorial. The *RLF Administrator Tutorial* provides an introduction to the installation and maintenance of RLF library systems. This tutorial assumes that the user is familiar with the basic RLF interface (for example, as presented in the *RLF User Tutorial*). A survey of the *Library_Manager* application is also presented in the tutorial.

3.1.1.2.3 RLF Modeler Tutorial. The *RLF Modeler Tutorial* provides a thorough presentation of RLF modeling capabilities so that attendees can begin the construction of RLF models for application domains of interest to them. Familiarity with the material covered in the *RLF User Tutorial* is assumed. Modeling techniques are discussed and the use of the RLF model specification languages is taught through the use of a detailed example.

3.1.2 Directory: *bin*

This directory contains the application resource file *RLF_Browser*, the *Graphical_Browser* start-up script *RLF_GB*, a sample RLF start-up file *.rlfrc*, and the associated bitmaps for the *RLF_Browser* file in the sub-directory *bitmaps*. These items are used with the *Graphical_Browser* application, with the *.rlfrc* also being used for the other RLF applications.

Included in the RLF 4.0 release is a *Sndl_to_Lmdl* translator for the conversion of the old SNDL syntax to the LMDL syntax for models.

This directory also contains public domain executables that are used by the sample models included with this release, which are not part of the standard SunOS or X releases. The executables included are *less* and *xloadimage*. In addition, a script called *view_stp.csh* is provided as an example of an RLF action to view a Software Through Pictures(STP) diagram using STP.

The start-up script *RLF_GB* and the sample start-up file *.rlfrc* are included in this document in Appendix B.

3.1.3 Directory: *models*

Sample libraries and their build scripts are found in the *models* directory, which contains the *ada_x*, *animals*, *asw*, *common_data_model*, *demo_actions*, *window_manager*, *software_technology* and *sort_and_search* subdirectories. The contents of these subdirectories are described in the following sections. This directory also contains the file *library_model_template.lmdl*, which contains an example LMDL specification for a library action sub-tree.

3.1.3.1 Sub-directory: *models/ada_x* ...contains the LMDL and RBDL specifications and associated text files for a sample RLF library describing the STARS Ada/Xt system. The specification files must be processed by the LMDL and RBDL translators to build the *ada_x* knowledge base.

3.1.3.2 Sub-directory: *models/animals* ...contains the LMDL specification and associated text files for a sample knowledge base describing a simple animals taxonomy. The specification files must be processed by the LMDL translator to build the animals knowledge base.

3.1.3.3 Sub-directory: *models/asw* ...contains the LMDL specification and associated text files for a sample RLF library addressing the anti-submarine warfare(ASW) domain. The specification files must be processed by the LMDL translator to build the *asw* library.

3.1.3.4 Sub-directory: *models/common_data_model* ...contains the LMDL specification and associated text files for a sample RLF library illustrating how the Common Data Model defined in the STARS ALOAF document can be expressed using RLF. The specification files must be processed by the LMDL translator to build the common data model library.

3.1.3.5 Sub-directory: *models/demo_actions* ...contains the LMDL specification and associated text files for a sample RLF library addressing the modeling of LMDL actions. The sound actions contained in this library only work on a Sun workstation that has a sound board. The specification files must be processed by the LMDL translator to build the *demo_actions* library.

3.1.3.6 Sub-directory: *models/window_manager* ...contains the LMDL and RBDL specifications and associated text files for a sample RLF library addressing the SEI's FODA example on move operations in the window manager domain. The specification files must be processed by the LMDL and RBDL translators to build the *window_manager* library.

3.1.3.7 Sub-directory: *models/software_technology* ...contains the LMDL specification and associated text files for a sample RLF library providing both a functional and product oriented view into the domain and defining numerous attributes for describing software engineering components. The specification files must be processed by the LMDL translator to build the *software_technology* library.

3.1.3.8 Sub-directory: *models/sort_and_search* ...contains the LMDL and RBDL specifications and associated text files for a sample RLF library describing a sort and search algorithms domain. The specification files must be processed by the LMDL and RBDL translators to build the *sort_and_search* library.

3.1.4 Directory: *code*

The *code* directory contains the source code directories, build scripts, and support files needed to construct the RLF.

3.1.4.1 Sub-directory: *code/Common* ...contains low-level data abstractions and utilities shared by the other RLF subsystems. For example, basic set and list abstract data types (ADT's) are located in this directory.

3.1.4.2 Sub-directory: *code/Adaknet* ...contains the source code for the AdaKNET subsystem. AdaKNET is a structured-inheritance knowledge representation system, which may be used as a stand-alone system or in conjunction with AdaTAU. The AdaKNET source makes use of the ADTs in the *Common* directory.

3.1.4.3 Sub-directory: *code/Adatau* ...contains the source for the AdaTAU subsystem. AdaTAU is a rule-based inferencer. Like AdaKNET, it may be used as a stand-alone system. The AdaTAU source makes use of the ADTs in the *Common* directory.

3.1.4.4 Sub-directory: *code/Ada_Actions* ...contains the source for Ada Procedures actions. The ada specifications, bodies and separates may be modified to create unique RLF Ada Procedure actions to be used in library modeling.

3.1.4.5 Sub-directory: *code/Hybrid* ...contains the source code for the hybrid knowledge representation system used by RLF library applications. This hybrid system combines AdaKNET and AdaTAU into an integrated knowledge representation system. The hybrid system source makes use of the AdaKNET and AdaTAU subsystems and the ADTs in the *Common* directory.

3.1.4.6 Sub-directory: *code/Library_Manager* ...contains the source code for the *Library_Manager* application. The *Library_Manager* is designed for a library administrator to have access to update and control an RLF library. The *Library_Manager* code makes use of the AdaKNET, AdaTAU and hybrid knowledge representation systems, and the ADTs in the *Common* directory.

3.1.4.7 Sub-directory: *code/Rlf_Gb* ...contains the source code for the RLF *Graphical_Browser* application. This browser uses SERC Ada/Motif interface to the X Window System. It provides a graphical read-only browsing capability for RLF network models. The *Rlf_Gb* code makes use of the AdaKNET, AdaTAU and hybrid knowledge representation systems. It also requires access to the STARS Reusable Graphical Browser interface.

3.1.4.8 Sub-directory: *code/Lmdl* ...contains the source for the Library Model Definition Language (LMDL) translator. This translator transforms a high-level, non-procedural description of a semantic network knowledge base into an AdaKNET knowledge base. The source code in this sub-directory makes use of ADTs defined in the *Common*, *Adaknet*, and *Hybrid* subdirectories.

3.1.4.9 Sub-directory: *code/Rbdl* ...contains the source for the Rule Base Definition Language (RBDL) translator. This translator transforms a high-level, non-procedural description of a rule-based knowledge base into an AdaTAU knowledge base. The source code in this sub-directory makes use of ADTs defined in the *Common* and *Adatau* subdirectories.

Δ 3.2 Changes Installed for Version 4.1

The largest changes in RLF 4.1 from RLF 4.0 is the support for the operation of RLF on top of PCTE in addition to UNIX. Other changes include documentation updates (including UNIX-style *man* pages) and *Library_Manager* application refinement.

Δ 3.2.1 The PCTE version of RLF

The following files were added to the *code/Common* directory to support input/output (IO) to either UNIX file systems or PCTE object bases.

```
code/Common/rlf_direct_io.a
code/Common/rlf_direct_io_pcte..a
code/Common/rlf_direct_io_unix..a
code/Common/rlf_sequential_io.a
code/Common/rlf_sequential_io_pcte..a
code/Common/rlf_sequential_io_unix..a
code/Common/rlf_text_io_pcte..a
code/Common/rlf_text_io_pcte.a
code/Common/rlf_text_io_unix..a
code/Common/rlf_text_io_unix.a
```

These packages replace standard Ada IO packages. Environment variables in the build scripts select either the UNIX or PCTE packages for compilation depending on whether UNIX file system or PCTE is being used. As a result of the IO changes, numerous files in the directories *Rlf*, *Lmdl*, and *Rbdl* were modified.

The method of invoking actions via system calls does not work in PCTE. The file *action_invocation_body.a* in the directory *Adaknet* was modified to invoke *esh* shell scripts. This change required changes to *Lmdl* specifications containing actions. As a result, several

of the sample libraries found in the directory *models* contain two Lmdl specifications to support UNIX and PCTE. The scripts to load the Lmdl specifications will load the appropriate specification.

The package *System_Dependent_Routines* was modified to support PCTE. Separate package bodies for UNIX and PCTE were created. The new files are: *Common/sys_dep_unix.a* and *Common/sys_dep_pcte.a*.

Several files to isolate PCTE and UNIX interface differences were added to the directory *Common*.

```
code/Common/network_constants_pcte.a
code/Common/pcte_invoke_string.a
code/Common/pcte_object_create.a
code/Common/rlf_constants_pcte.a
code/Common/rlf_constants_unix.a
code/Common/rlf_pcte_body.a
code/Common/rlf_pcte_spec.a
code/Common/unix_invoke_string.a
```

Additional scripts for loading sample libraries were added to subdirectories of the *models* directory. These scripts are:

```
models/ada_x/Build_Ada_X_Lib.esh
models/animals/Build_Animals_Lib.esh
models/asw/Build_Asw_Lib.esh
models/common_data_model/Build_Common_Data_Model_Lib.esh
models/demo_actions/Build_Demo_Actions_Lib.esh
models/software_technology/Build_SW_Tech_Lib.esh
models/sort_and_search/Build_SaS_Lib.esh
models/window_manager/Build_Window_Manager_Lib.esh
```

Several files in the *code/Ada_Actions* directory were modified for invoking actions in PCTE. Differences in invocation of actions between PCTE and UNIX are isolated in separate files. The new files in *Ada_Actions* are:

```
code/AdaActions/display_attr_action_ascii_sp_pcte.a
code/AdaActions/display_attr_action_ascii_sp_unix.a
code/AdaActions/display_attr_action_body.a
code/AdaActions/display_attr_action_buf_sp_pcte.a
code/AdaActions/display_attr_action_buf_sp_unix.a
code/AdaActions/display_attr_action_files_sp_pcte.a
code/AdaActions/display_attr_action_files_sp_unix.a
code/AdaActions/extract_file_sp_pcte.a
code/AdaActions/extract_file_sp_unix.a
```

Differences between UNIX and PCTE implementations in the *code/Adaknet* directory are isolated in the files:

```
code/Adaknet/invoke_string_sp_pcte.a
code/Adaknet/invoke_string_sp_unix.a
```

To reduce the amount of PCTE dependent code, the *Gb.Actions* package of the *Graphical_Browser* application was removed and actions invoked from the *Graphical_Browser* now use the action invocation method in the RLF core. The file *Rlf_Gb/gb_actions.a* has been deleted from this release.

A problem compiling generics in the package *GB_CB_Suppress* forced the creation of a separate package containing several generic instantiations. These generic instantiations are now contained in the file *Rlf_Gb/gb_cb_suppress_gen.a*.

Because of the different method for invoking actions in PCTE many of the Lmdl scripts in the *models* directory have been modified. PCTE specific Lmdl specifications have *_pcte* string preceding the *.lmdl* suffix in the UNIX file name. In addition, *esh* scripts used for invoking actions in PCTE have been added to the *Text/* directories, and appear with *.tool* suffix. See the PCTE appendix in the Modeler's Manual for more information on actions in the PCTE version of this release.

3.2.2 *Library_Manager* Application Refinement

The *Library_Manager* application introduced in RLF 4.0 has been refined in version 4.1 to replace some dynamic menus with scrollable list widgets and to desensitize button choices which would lead to the pop-up of an empty menu. A limit on the number of libraries the *Library_Manager* can process was also removed. All of the *Library_Manager* application files changed:

```
code/Library_Manager/callbacks_body.a
code/Library_Manager/callbacks_spec.a
code/Library_Manager/globals_spec.a
code/Library_Manager/library_manager.a
```

3.3 Adaptation Data

3.3.1 Operating Environment

Sun-4 workstations with a minimum of 8 MB of main memory

SunOS, Version 4.1 or later

X Window System, Version 11, Release 4

OSF/Motif version 1.1

For the RLF/PCTE version include the following:

Emeraude PCTE v12.3

STARS Ada/PCTE bindings version 0.3

3.3.2 Development Environment

Sun-4 workstations with a minimum of 8 MB of main memory

SunOS, Version 4.1 or later

X Window System, Version 11, Release 4

OSF/Motif version 1.1

Sun Ada Version 1.1

SERC Ada/Motif, Version 1.1 for Sun Ada version 1.1

Reusable Graphical Browser, Version 1.1 (Graphical_Browser only)

Emeraude PCTE v12.3

STARS Ada/PCTE bindings version 0.3

3.3.3 Configuration-Unique Data

3.3.3.1 Software Dependencies. The *Graphical_Browser* and *Library_Manager* capabilities require the X Window System, OSF/Motif, SERC Ada/Motif, and Reusable Graphical Browser (RGB) as identified in section 3.3.2. The PCTE version of RLF 4.1 also requires Emeraude PCTE version 12.3 and Paramax STARS Ada/PCTE bindings version 0.3. To obtain the Paramax STARS products, please contact Paramax as recommended in Section 4.

3.3.3.2 OS Dependencies. Small parts of the UNIX version of RLF are dependent upon the underlying operating system. Parts of the PCTE version are dependent on the PCTE implementation and the OS it is hosted on. In the case of RLF 4.1, this would be UNIX. Certain UNIX-specific files are replaced by PCTE versions. These files' names end with `_pcte.a` and `_unix.a`. Some RLF files also process environment variables in possibly a system-dependent manner.

In order to access some UNIX facilities, the Ada *pragma INTERFACE* capability is used to access the 'C' language. This capability can be dependent upon the particular development system being used. The details of this interface are contained in the following files:

```
Common/c_interface_body.a
Common/customIO_body.a
Common/extract_file_sp_pcte.a
Common/invoke_string_sp_pcte.a
Common/invoke_string_sp_unix.a
Common/pcte_invoke_string.a
Common/system_dep_pcte.a
Common/system_dep_unix.a
Common/system_environment_tele_body.a
Common/system_environment_vads_body.a
Common/extract_file_sp_unix.a
Common/rlfrc_scanner_io.a
Common/unix_invoke_string.a

Common/support.c          -- 'C' language functions.
```

3.3.3.3 Defaults. The files *Common/network_constants.a* and *Common/network_constants_pcte.a* define defaults for certain file-naming and file-length conventions. It is unlikely that any modifications to these defaults will be required, however, the installer is advised to review the default definitions to ensure compatibility with overall system limits and conventions.

3.4 Interface Compatibility

3.4.1 Previously Built RLF Libraries

This version of the RLF should be fully compatible with RLF 4.0. Of course, libraries built with the UNIX version of RLF will not be readable by the PCTE RLF, and vice versa.

3.4.2 Libraries Built with Different Compilers

Data representations are different between Ada compilers. As a result, RLF libraries created by a version of the RLF built with one compiler may not be interoperable with libraries created by a version of the RLF built with another compiler.

3.5 Installation and Usage Instructions

The file *Install_RLF_src* is an executable UNIX C shell script, which can be used to build the UNIX version of the RLF from the Ada source code, using the Sun Ada compiler. This script invokes subordinate scripts in the *code* directory, which build individual portions of the software in the proper order. The complete installation and verification procedures are located in the *RLF Installation Guide*. This manual also instructs the user how to build the sample libraries provided with this release.

NOTE: Appendix C contains a listing of the UNIX installation scripts provided in this distribution.

Upon completion of these procedures, the directory defined by the *RLFBIN* variable should contain the following newly built executables:

Graphical_Browser
Library_Manager
Lmdl
Rbd1

3.5.1 Invoking RLF Applications

Once the RLF executables have been built, any of the executables can be run by invoking it by name. Information about invoking the RLF *Graphical_Browser* application is located in the *RLF User's Manual*. Additional information about RLF applications and their uses may be found in the *RLF Modeler's Manual* and the *RLF Administrator's Manual*.

3.6 Potential Problems

3.6.1 System Memory Limitations

Depending on your system configuration, you may encounter Ada compilation or LMDL translation problems stemming from an insufficient amount of memory available to the compiler or translator. One possible solution is to remove any artificial data space limitations during the build by uncommenting the *limit datasize unlimited* and *limit stacksize unlimited* directives in each of the *Build...* scripts. If this modification does not improve the situation, consult your system administrator.

3.6.2 Graphical_Browser Known Problems

During the execution of the *Graphical_Browser* a few infrequent errors may occur. The errors listed here are attributed to bugs in Motif version 1.1. It is expected that future versions of Motif will eliminate these errors.

The following is the list of known errors and their descriptions:

1. **Warning: XtRemoveGrab asked to remove a widget not on the list** - This text message, which appears in the originating window, often occurs when a window in the *Graphical_Browser* is exited or canceled.
2. **Menu bar menu relocating to upper left hand corner of the screen** - This event can happen when the Node History menu option, which is in the Navigate View menu bar option, is selected. As the pointer passes over the menu entry the cascading menu may be placed in the upper left hand corner of the screen.
3. **Node menu creation error** - This display alert box randomly appears when a node is selected. If the node is selected again the error usually does not occur. Reselect the node and the correct menu should appear.

3.7 Enhancements

As this is expected to be the last full release of RLF by STARS, not many enhancements are planned for the future under STARS funding. For the basic RLF capabilities, future enhancements might include:

1. An RLF version hosted on a broadcast message server integration back-plane.
2. "Message Pass" type actions used to invoke other tools.
3. Integration with other tools and increased interoperability between the different RLF applications.

4 USER FEEDBACK

This version of RLF is considered an "alpha" release. One of the primary purposes of the release is to encourage experimentation with the software and to solicit feedback from the Ada user community to assist us in improving the product and advancing software reuse. Thus, we would greatly appreciate your comments, suggestions, and criticisms. Although we do not guarantee the applicability of the RLF to particular application needs at this time, we are interested in hearing about successes as well as failures.

We have included three forms in this release which we hope you will use to provide us with needed feedback:

- A registration form (in file *Registration_Form*) that we would like you to fill out and return to us so that we can keep track of our user base and can notify you of product upgrades and other important product news.
- A Program Problem Report (in file *Problem_Report*) that you should use to identify any specific problems you encounter in installing and using the software.
- A New Feature Request (in file *Feature_Request*) that you should use to describe specific enhancements that you believe should be incorporated into the product.

We have established three electronic mailing lists to facilitate RLF usage and feedback:

- **rlf@stars.ballston.paramax.com**
This list provides a public forum for discussing RLF issues. If you ask to be included in this list, you will receive all messages sent to the list and may respond accordingly.
- **rlf-request@stars.ballston.paramax.com**
You should send your completed registration form to this address, as well as requests to be added to or deleted from the rlf list (NOTE: do NOT send add or delete requests to the rlf list itself).
- **rlf-bugs@stars.ballston.paramax.com**
You should send completed Program Problem Reports and New Feature Requests to this address.

If you do not have electronic mail access or wish to send us printed information, please send mail to:

RLF
Paramax STARS Center
12010 Sunrise Valley Drive
Reston, VA 22091

5 NOTES

Both AdaTAU and AdaKNET were designed for independent use by applications requiring knowledge representation and inferencing capabilities. The specification languages provided for these subsystems foster their transfer to diverse application areas and their programmatic interfaces enable their integration into general Ada applications. Additional applications will help determine system shortcomings and lead to their correction.

A Appendix: Inventory of Contents

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

```

.:
Contents.tty
Install_RLF_src*
README
VDDr1f.ps
VDDr1f.tty
bin/
code/
docs/
man/
models/

bin:
.rlfrc
RLF_GB*

code:
Ada_Actions/
Adaknet/
Adatau/
Build_Ada_Libraries.csh*
Build_Library_Manager.csh*
Build_Lmdl.csh*
Build_RLF.csh*
Build_RLF.var
Build_Rbdl.csh*
Build_Rlf_Gb.csh*
Build_Rlfdir.csh*
Common/
Hybrid/
Library_Manager/
Lmdl/
Rbdl/
Rlf_Gb/

code/Ada_Actions:
action_routines_body.a
action_routines_spec.a
display_attr_action_ascii_sp_pcte.a
display_attr_action_ascii_sp_pcte.a.last
display_attr_action_ascii_sp_unix.a
display_attr_action_ascii_sp_unix.a.last
display_attr_action_body.a
display_attr_action_buf_sp_pcte.a
display_attr_action_buf_sp_pcte.a.last
display_attr_action_buf_sp_unix.a
display_attr_action_buf_sp_unix.a.last
display_attr_action_files_sp_pcte.a
display_attr_action_files_sp_pcte.a.last
display_attr_action_files_sp_unix.a

```

display_attr_action_files_sp_unix.a.last
display_attr_action_spec.a
display_attributes_sp.a
export_sp.a
extract_action_body.a
extract_action_spec.a
extract_file_sp_pcte.a
extract_file_sp_pcte.a.last
extract_file_sp_unix.a
extract_file_sp_unix.a.last
extract_sp.a
import_sp.a

code/Adaknet:

action_invocation_body.a
action_invocation_body.a.orig
action_invocation_spec.a
action_invocation_spec.a.orig
action_mod_ops_sp.a
action_operations_body.a
action_operations_spec.a
action_predicates_sp.a
action_query_ops_sp.a
actions_body.a
actions_spec.a
adaket_name_types.a
adanet_body.a
adanet_composites_body.a
adanet_composites_spec.a
adanet_constr_destr_ops_sp.a
adanet_mod_ops_sp.a
adanet_predicates_sp.a
adanet_query_ops_sp.a
adanet_spec.a
adanet_state_body.a
adanet_state_spec.a
agg_browser_body.a
agg_browser_spec.a
app_utils_body.a
app_utils_spec.a
changes_sp.a
composites_body.a
composites_spec.a
concept_mod_ops_sp.a
concept_predicates_sp.a
concept_query_ops_sp.a
examine_network_body.a
examine_network_spec.a
generic_concepts_body.a
generic_concepts_spec.a
individual_concepts_body.a
individual_concepts_spec.a
invoke_ada_proc_sp.a
invoke_string_sp_pcte.a

invoke_string_sp_pcte.a.last
invoke_string_sp_unix.a
invoke_string_sp_unix.a.last
invoke_sys_string_sp.a
invoke_sys_string_sp.a.last
invoke_sys_string_sp.a.orig
isa_browser_body.a
isa_browser_spec.a
misc_ops_sp.a
network_composites_body.a
network_composites_spec.a
networks_body.a
networks_spec.a
ranges_body.a
ranges_spec.a
restrictions_sp.a
role_mod_ops_sp.a
role_predicates_sp.a
role_query_ops_sp.a
roles_body.a
roles_spec.a
roleset_mod_ops_sp.a
roleset_predicates_sp.a
roleset_query_ops_sp.a
roleset_spec_ops_sp.a
rolesets_body.a
rolesets_spec.a
schema_body.a
schema_spec.a
set_conversions_sp.a
sndl_dump_body.a
sndl_dump_spec.a
states_sp.a
subroles_sp.a

code/Adatau:
agendas_body.a
agendas_spec.a
basic_config_functions_body.a
basic_config_functions_spec.a
debug_body.a
debug_spec.a
dist_rddl_test.a
dump_rddl_body.a
dump_rddl_spec.a
fact_base_schemas_body.a
fact_base_schemas_spec.a
fact_bases_body.a
fact_bases_spec.a
fact_depend_functions_body.a
fact_depend_functions_spec.a
fact_depend_spec.a
fact_lists_body.a
fact_lists_spec.a

fact_parameter_body.a
fact_parameter_list_body.a
fact_parameter_list_spec.a
fact_parameter_spec.a
fact_schemas_body.a
fact_schemas_spec.a
fact_value_lists_body.a
fact_value_lists_spec.a
facts_body.a
facts_spec.a
frules_body.a
frules_spec.a
irule_bases_spec.a
irules_body.a
irules_spec.a
label_table_body.a
label_table_spec.a
lib_static_persistence_body.a
lib_static_persistence_spec.a
librarian_configuration.a
message_DTAU_components_body.a
message_DTAU_components_spec.a
message_DTAU_main.a
message_TAU_components_body.a
message_TAU_components_spec.a
message_adv_config_body.a
message_adv_config_spec.a
message_config_body.a
message_config_spec.a
message_get_response_sp.a
message_get_response_sp.a.last
non_monotonic_support_body.a
non_monotonic_support_spec.a
persistence_body.a
persistence_spec.a
q_agendas_spec.a
qrule_bases_spec.a
qrules_body.a
qrules_spec.a
question_bases_spec.a
questions_body.a
questions_spec.a
response_schemas_body.a
response_schemas_spec.a
rule_bases_body.a
rule_bases_spec.a
static_persistence_body.a
static_persistence_spec.a
truth_maintenance_body.a
truth_maintenance_spec.a

code/Common:
c_interface_body.a
c_interface_spec.a

commonIO_body.a
commonIO_spec.a
customIO_body.a
customIO_spec.a
filenames_body.a
filenames_spec.a
fixed_strings_body.a
fixed_strings_spec.a
fstring_body.a
fstring_spec.a
generic_hash_body.a
generic_hash_spec.a
generic_sequences_body.a
generic_sequences_spec.a
hashmap_body.a
hashmap_spec.a
hybrid_tables_body.a
hybrid_tables_spec.a
interrupts.c*
labels_body.a
labels_spec.a
lists_body.a
lists_spec.a
message_io_body.a
message_io_spec.a
mono_lock_manager_body.a
mono_lock_manager_spec.a
network_constants.a
network_constants_pcte.a
objects_spec.a
pcte_invoke_string.a
pcte_invoke_string.a.last
pcte_object_create.a
pcte_object_create.a.last
rlf_constants_pcte.a
rlf_constants_pcte.a.last
rlf_constants_unix.a
rlf_constants_unix.a.last
rlf_direct_io.a
rlf_direct_io.a.last
rlf_direct_io_pcte.a
rlf_direct_io_pcte.a.last
rlf_direct_io_unix.a
rlf_direct_io_unix.a.last
rlf_globals_body.a
rlf_globals_spec.a
rlf_pcte_body.a
rlf_pcte_body.a.last
rlf_pcte_spec.a
rlf_pcte_spec.a.last
rlf_sequential_io.a
rlf_sequential_io.a.last
rlf_sequential_io_pcte.a
rlf_sequential_io_pcte.a.last

```

rlf_sequential_io_unix.a
rlf_sequential_io_unix.a.last
rlf_text_io_pcte.a
rlf_text_io_pcte.a.last
rlf_text_io_pcte_.a
rlf_text_io_pcte_.a.last
rlf_text_io_unix.a
rlf_text_io_unix.a.last
rlf_text_io_unix_.a
rlf_text_io_unix_.a.last
rlf_univ_types.a
rlfrc_parser.a
rlfrc_parser_goto.a
rlfrc_parser_shift_reduce.a
rlfrc_parser_spec.a
rlfrc_parser_support_body.a
rlfrc_parser_support_spec.a
rlfrc_parser_tokens.a
rlfrc_scanner.a
rlfrc_scanner_dfa.a
rlfrc_scanner_io.a
sets_body.a
sets_spec.a
sorting_body.a
sorting_spec.a
stacks_body.a
stacks_spec.a
strings_body.a
strings_spec.a
support.c*
system_dep_pcte.a
system_dep_spec.a
system_dep_unix.a
system_environment_spec.a
system_environment_tele_body.a
system_environment_vads_body.a
tables_body.a
tables_spec.a
tau_lists_body.a
tau_lists_spec.a
unique_identifiers_body.a
unique_identifiers_spec.a
unix_invoke_string.a
unix_invoke_string.a.last

```

code/Hybrid:

```

fbase_ops_body.a
fbase_ops_spec.a
inf_hybrid_states_body.a
inf_hybrid_states_spec.a
inf_state_ops_body.a
inf_state_ops_spec.a
integer_hybrid_states_body.a
integer_hybrid_states_spec.a

```

integer_state_ops_body.a
integer_state_ops_spec.a
library_hybrid_networks.a
library_hybrid_state_ops_body.a
library_hybrid_state_ops_spec.a
library_hybrid_states_body.a
library_hybrid_states_spec.a
text_hybrid_states_body.a
text_hybrid_states_spec.a
text_state_file_ops_body.a
text_state_file_ops_spec.a
text_state_ops_body.a
text_state_ops_spec.a

code/Library_Manager:

callbacks_body.a
callbacks_spec.a
globals_spec.a
library_manager.a

code/Lmdl:

adaknet_ops_body.a
adaknet_ops_spec.a
backend_body.a
backend_interface.a
backend_spec.a
globals_body.a
globals_spec.a
hybrid_ops_body.a
hybrid_ops_spec.a
lexacts_body.a
lexacts_spec.a
lmdl_attributed_tree_body.a
lmdl_attributed_tree_spec.a
lmdl_dgts_body.a
lmdl_dgts_spec.a
lmdl_display.a
lmdl_evaluator_body.a
lmdl_evaluator_spec.a
lmdl_gets.a
lmdl_lex_pkg.a
lmdl_lexdef.a
lmdl_makes.a
lmdl_parser.a
lmdl_parserdefs.a
lmdl_puts.a
lmdl_system_types_spec.a
operations_list_def_body.a
operations_list_def_spec.a
std_boot.a
std_lex_spec.a
std_magic_body.a
std_magic_spec.a
std_output_spec.a

```

std_parser_spec.a
std_predefined_tree.a
std_report_body.a
std_report_spec.a
std_user_pkg.a
std_yyval_error_body.a
std_yyval_error_spec.a
support_body.a
support_spec.a
topsort_body.a
topsort_spec.a
user_types_body.a
user_types_spec.a

code/Rbdl:
backend_interface.a
be1.a
be1_spec.a
besl_support_body.a
besl_support_spec.a
besl_vl_string_handler_body.a
besl_vl_string_handler_spec.a
create_inferencer_support_body.a
create_inferencer_support_spec.a
globals_body.a
globals_spec.a
lexacts_body.a
lexacts_spec.a
nonstd_boot.a
nonstd_lex_spec.a
nonstd_magic_spec.a
question_hash_types_body.a
question_hash_types_spec.a
question_hashes.a
rbdl_attributed_tree_body.a
rbdl_attributed_tree_spec.a
rbdl_dgts_body.a
rbdl_dgts_spec.a
rbdl_display.a
rbdl_evaluator_body.a
rbdl_evaluator_spec.a
rbdl_gets.a
rbdl_lex_pkg.a
rbdl_lexdef.a
rbdl_makes.a
rbdl_parser.a
rbdl_parserdefs.a
rbdl_puts.a
rbdl_system_types_spec.a
rbdl_treedump_body.a
rbdl_treedump_spec.a
report_body.a
report_spec.a
std_magic_body.a

```

std_output_spec.a
std_parser_spec.a
std_predefined_tree.a
std_user_pkg.a
std_yyval_error_body.a
std_yyval_error_spec.a
support_body.a
support_spec.a
user_types_body.a
user_types_spec.a

code/Rlf_Gb:

gb_agg_browser_body.a
gb_agg_browser_spec.a
gb_callbacks_body.a
gb_callbacks_spec.a
gb_cb_suppress_body.a
gb_cb_suppress_gen.a
gb_cb_suppress_gen2.a
gb_cb_suppress_spec.a
gb_dyn_node_menus_body.a
gb_dyn_node_menus_spec.a
gb_dynamic_menus_body.a
gb_dynamic_menus_spec.a
gb_globals.a
gb_hash.a
gb_history.a
gb_infer.a
gb_infer_dtau_body.a
gb_infer_dtau_spec.a
gb_instance.a
gb_main.a
gb_make_view.a
gb_params_body.a
gb_params_spec.a
gb_static_cmds_body.a
gb_static_cmds_spec.a
gb_static_menus_body.a
gb_static_menus_spec.a
gb_utils_body.a
gb_utils_spec.a

docs:

manuals/
tutorials/

docs/manuals:

AdministratorsManual.ps
AdministratorsManual.tty
InstallationGuide_Binary.ps
InstallationGuide_Binary.tty
InstallationGuide_Source.ps
InstallationGuide_Source.tty
ModelersManual.ps

ModelersManual.tty
UsersManual.ps
UsersManual.tty

docs/tutorials:
RLF_UM
admin-art.ps
admin-training.tex
model-art.ps
model-training.tex
user-art.ps
user-training.tex

man:
cat1/
man1/
whatis

man/cat1:
Graphical_Browser.1
Library_Manager.1
Lmdl.1
RLF.1
RLF_GB.1
Rbdl.1
rlf.1

man/man1:
Graphical_Browser.1
Library_Manager.1
Lmdl.1
RLF.1
RLF_GB.1
Rbdl.1
rlf.1

models:
ada_x/
animals/
asw/
common_data_model/
demo_actions/
library_model_template.lmdl
library_model_template_pcte.lmdl
software_technology/
sort_and_search/
window_manager/

models/ada_x:
Build_Ada_X_Lib.csh*
Build_Ada_X_Lib.esh*
ada_x.lmdl*
ada_x_pcte.lmdl*

ada_xt_widget_package.rbd1*
ada_xt_widget_pkg.rbd1*
application_shell_widget_package.rbd1*
application_shell_widget_pkg.rbd1*
ascii_disk.rbd1*
ascii_disk_widget_package.rbd1*
ascii_disk_widget_pkg.rbd1*
ascii_string.rbd1*
ascii_string_widget_package.rbd1*
ascii_string_widget_pkg.rbd1*
bboard.rbd1*
bboard_widget_package.rbd1*
bboard_widget_pkg.rbd1*
command.rbd1*
command_widget_package.rbd1*
command_widget_pkg.rbd1*
composite_object_widget_package.rbd1*
composite_object_widget_pkg.rbd1*
composite_widget_package.rbd1*
composite_widget_pkg.rbd1*
constraint_widget_package.rbd1*
constraint_widget_pkg.rbd1*
core.rbd1*
core_widget_package.rbd1*
core_widget_pkg.rbd1*
dialog.rbd1*
dialog_widget_package.rbd1*
dialog_widget_pkg.rbd1*
form.rbd1*
form_widget_package.rbd1*
form_widget_pkg.rbd1*
label.rbd1*
label_widget_package.rbd1*
label_widget_pkg.rbd1*
manager.rbd1*
manager_widget_package.rbd1*
manager_widget_pkg.rbd1*
object_widget_package.rbd1*
object_widget_pkg.rbd1*
override_shell_widget_package.rbd1*
override_shell_widget_pkg.rbd1*
rect_object_widget_package.rbd1*
rect_object_widget_pkg.rbd1*
scroll.rbd1*
scroll_widget_package.rbd1*
scroll_widget_pkg.rbd1*
shell_widget_package.rbd1*
shell_widget_pkg.rbd1*
simple.rbd1*
simple_widget_package.rbd1*
simple_widget_pkg.rbd1*
text.rbd1*
text_widget_package.rbd1*
text_widget_pkg.rbd1*

top_level_shell_widget_package.rbd1*
top_level_shell_widget_pkg.rbd1*
transient_shell_widget_package.rbd1*
transient_shell_widget_pkg.rbd1*
vendor_shell_widget_package.rbd1*
vendor_shell_widget_pkg.rbd1*
viewport.rbd1*
viewport_widget_package.rbd1*
viewport_widget_pkg.rbd1*
widget.rbd1*
window_object_widget_package.rbd1*
window_object_widget_pkg.rbd1*
wm_shell_widget_package.rbd1*
wm_shell_widget_pkg.rbd1*

models/animals:
Build_Animals_Lib.csh*
Build_Animals_Lib.esh*
Text/
animals.lmdl*
animals_pcte.lmdl*

models/animals/Text:
del*
dick*
snoopy*
teri*
tim*
xterm_pager.tool*

models/asw:
Build_Asw_Lib.csh*
Build_Asw_Lib.esh*
Text/
asw.lmdl*
asw_pcte.lmdl*

models/asw/Text:
AGP_CommandsSada*
AGP_InputBada*
AGP_InputSada*
AGP_Memory_ManagerSada*
AGP_OutputBada*
AGP_OutputSada*
confirm_panel_package*
dialog_public_a*
dialog_publica*
dialog_publica2*
form_public_a*
form_publica*
viewport_public_a*
viewport_publica*
xterm_int.tool*
xterm_less.tool*

```
models/common_data_model:
Build_Common_Data_Model_Lib.csh*
Build_Common_Data_Model_Lib.esh*
Text/
common_data_model.lmdl*
common_data_model_pcte.lmdl*
```

```
models/common_data_model/Text:
astronomical_constants_s.a*
desc_astronomical_constants_s*
desc_math_interface_sb*
desc_overpass*
desc_sat_comp_sb*
desc_sat_io_b*
desc_sat_io_s*
desc_units_s*
math_interface_sb.a*
overpass.a*
restr_as_is_warranty*
sat_comp_sb.a*
sat_io_b.a*
sat_io_s.a*
units_s.a*
```

```
models/demo_actions:
Build_Demo_Actions_Lib.csh*
Build_Demo_Actions_Lib.esh*
Text/
demo_actions.lmdl*
demo_actions_pcte.lmdl*
```

```
models/demo_actions/Text:
building*
general_floorplan*
imprint.tool*
lpr.tool*
message*
my_floorplan*
play.tool*
sounds/
xbm/
xloadimage.tool*
xterm_less.tool*
xterm_vi.tool*
```

```
models/demo_actions/Text/sounds:
clint_eastwood.au*
completely_op.au*
goodcoffee.au*
kirk_spock_boundary.au*
klaxton.au*
mccoy_all.au*
phasars_3.au*
```

photons_3.au*

models/demo_actions/Text/xbm:

Jerry_Bob.xbm*

eye.xbm*

fist.xbm*

full_owl.xbm*

launch.xbm*

lips.xbm*

mandelbrot.xbm*

mandelbrot_seahorses.xbm*

nebula.xbm*

owl_head.xbm*

planet_miranda.xbm*

small_galaxy.xbm*

spock.xbm*

models/software_technology:

Build_SW_Tech_Lib.csh*

Build_SW_Tech_Lib.esh*

SW_Tech_Model_Description.txt*

Text/

sw_tech.lmdl*

sw_tech_pcte.lmdl*

models/software_technology/Text:

bob_pollack*

darpa_isto*

jack_chapman*

karen_roth*

payton_ssags_paper*

pollack_and_loftus*

pollack_mfpl_paper*

pollack_tree_transformation_paper*

q13_tools_clc.a*

q13_tools_clc.abs*

q13_tools_clc.con*

q13_tools_clc_build.csh*

q13_tools_clc_test.a*

q9-c300.abs*

q9-c300.con*

q9-c300.doc*

q9-c300.ref*

q9-c300.tem*

q9-c340.abs*

q9-c340.con*

q9-c340.doc*

q9-c340.ref*

q9-c340.tem*

q9-c350.abs*

q9-c350.con*

q9-c350.doc*

q9-c350.ref*

q9-c350.tem*

q9-c360.abs*
q9-c360.con*
q9-c360.doc*
q9-c360.ref*
q9-c360.tem*
software_a_and_e*
software_technology_inc*
ssags.abs*
ssags.con*
ssags.tem*
vfl_history*

models/sort_and_search:
Build_SaS_Lib.esh*
Build_Sort_And_Search_Lib.csh*
Text/
algorithms.rbd1*
binary_ins.rbd1*
diminishing_inc.rbd1*
exchange_sorts.rbd1*
heapsort.rbd1*
insertion_sorts.rbd1*
internal_sorts.rbd1*
quicksort.rbd1*
selection_sorts.rbd1*
shakersort.rbd1*
shellsort.rbd1*
sort_algorithms.rbd1*
sort_and_search.lmdl*
sort_and_search_pcte.lmdl*
straight_ins.rbd1*
straight_sel.rbd1*

models/sort_and_search/Text:
exchange_sort_desc*
heap_spec_.a*
insertion_sort_desc*
quick_sort_.a*
selection_sort_desc*
shaker_sort_.a*
xterm_less.tool*
xterm_less_int.tool*

models/window_manager:
Build_Window_Manager_Lib.csh*
Build_Window_Manager_Lib.esh*
Text/
move_domain.lmdl*
move_domain.rbd1*
move_domain_pcte.lmdl*
option_move_resize.rbd1*
sunview_move.rbd1*
x10_move.rbd1*

```
models/window_manager/Text:  
abort_move.att*  
constrained_move.att*  
expose_after_move.att*  
move_domain_concept.help*  
move_icon.att*  
partially_off_screen.att*  
tiled_layout.descr*  
xterm_less_12.tool*  
xterm_less_40.tool*
```

B Appendix: RLF Start-up Files

B.1 Sample RLF .rlfrc Start-up File

B.1.1 File: .rlfrc

```
1  --|
2  --| Sample startup file for the Reuse Library Framework version 4.1
3  --|
4
5  --|
6  --| Library directory or name specifications
7  --|
8  --library directory : /path/Libraries
9  --library : "Sort and Search Algorithms"
10
11 --|
12 --| Parameters for the RLF Graphical Browser
13 --|
14 topology : off
15 cardinality : off
16 layout offset : x : 20
17 layout offset : y : 5
18 history length : 50
19 view type : specialization
20 view depth : relationship : 2
21
22 --|
23 --| AdaTau inferencing settings
24 --|
25 advice : explanations : all
26 advice : automatic move : false
27
28 --|
29 --| Bitmaps for nodes
30 --|
31 --node bitmap : category : /path/box_m.xbm
32 --node bitmap : category : inferencer : /path/box_I_m.xbm
33 --node bitmap : category : actions : /path/box_A_m.xbm
34 --node bitmap : category : inferencer actions : /path/box_AI_m.xbm
35 --node bitmap : object : /path/cube_m.xbm
36 --node bitmap : object : inferencer : /path/cube_I_m.xbm
37 --node bitmap : object : actions : /path/cube_A_m.xbm
38 --node bitmap : object : inferencer actions : /path/cube_AI_m.xbm
39
40 --|
41 --| Specification translator settings
42 --|
43 translator: Lmdl: quiet: no
44 translator: Rbd1: quiet: no
```

B.2 RLF Graphical Browser Start-up Script

B.2.1 Script: RLF_GB

```

1  #!/bin/csh -f
2  #
3  #-----
4  # RLF_GB - Startup script for the RLF Graphical Browser, v.4.1
5  #
6  # 1.) Check that an X environment is present and running.
7  #
8  # 2.) Ensure the environment variables (RLF_LIBRARIES, DISPLAY, and possibly
9  #      XAPPLRESDIR) are properly set.
10 #
11 # 3.) Invoke the Graphical Browser with all command line arguments specified
12 #      by the user.
13 #
14 # If either an environment variable is not set or incorrectly set or X is not
15 #      running, then abort the script and notify the user of the problem.
16 #
17 #-----
18
19 echo ""
20 echo " ====="
21 echo "  RLF v.4.1 Graphical Browser Startup Script "
22 echo " ====="
23 echo ""
24
25 #-----
26 # Determine if the DISPLAY environment variable is set;
27 # if it is set, then proceed;
28 # if it is not set, attempt to set it to a meaningful value.
29 #-----
30 if ( ! $?DISPLAY ) then
31     set host_name = 'hostname'
32     setenv DISPLAY ${host_name}:0
33 endif
34
35
36 echo ""
37 echo "  Ensure the DISPLAY environment variable is"
38 echo "  set correctly; the correct format is <host_name>:0,"
39 echo "  where the host_name indicates what CPU your X server "
40 echo "  is running on."
41 echo ""
42 echo "Currently, DISPLAY = "
43 echo "  $DISPLAY"
44 echo ""
45 set local_host = `echo $DISPLAY | sed 's/.*$//'`
46 echo "This means the graphical output will be sent to host: "
47 echo "  $local_host"
48 echo ""
49
50 #-----

```

```

51 # Query the X resource database to determine whether $DISPLAY is valid.
52 #-----
53 xrdp -query >& /dev/null
54
55 #-----
56 # The DISPLAY environment variable was set incorrectly
57 # if the status is not 0.  Notify the user.
58 #-----
59 if ( ! $status == 0 ) then
60     unsetenv DISPLAY
61     echo ""
62     echo "    There's a problem with your X server."
63     echo "    There's probably no X server running on host 'hostname'."
64     echo "    Determine where your X server is running,"
65     echo "    then issue the following command: "
66     echo ""
67     echo "        setenv DISPLAY <hostname>:0 "
68     echo ""
69     echo "    where <hostname> is the host where your "
70     echo "    X server is running."
71     echo ""
72 endif
73
74 #-----
75 # If RLF_LIBRARIES environment variable not already set, or
76 # incorrectly set exit the script and notify the user.
77 #-----
78 if ( ! $?RLF_LIBRARIES ) then
79
80     #
81     # Check the command line options to see if the user
82     # specified a library
83     #
84     if ( $#argv >= 2 ) then
85
86         @ index = 1
87
88         while ( $#argv >= $index + 1)
89
90             @ index2 = $index + 1
91
92             if ( $argv[$index] == "-I" ) then
93                 if ( ( -d $argv[$index2]/Text ) && \
94                     ( -d $argv[$index2]/Taustuff ) ) then
95                     echo "Library directory to be used is $argv[$index2]"
96                     echo ""
97                     goto Library_Found
98                 else
99                     echo ""
100                    echo "FATAL ERROR:"
101                    echo "    The RLF library ($argv[$index2]) you"
102                    echo "    indicated from the command line is invalid."
103                    echo "    You must set it to a proper RLF library location."
104                    echo ""

```

```

105
106             exit(-1)
107
108             endif
109         endif
110
111         @ index++
112
113     end
114
115 endif
116
117 echo ""
118 echo "FATAL ERROR:"
119 echo "    RLF_LIBRARIES is currently unset."
120 echo "    You must set it to the proper location"
121 echo "    or specify a library directory with "
122 echo "    a command line option."
123 echo ""
124
125 exit(-1)
126
127 else if ( ( ! -d $RLF_LIBRARIES/Text) || ( ! -d $RLF_LIBRARIES/Taustuff) ) then
128
129     echo ""
130     echo "FATAL ERROR:"
131     echo "    RLF_LIBRARIES is incorrectly set."
132     echo "    There are missing elements in the libraries."
133     echo "    You must set it to the proper location."
134     echo ""
135
136     exit(-1)
137
138 endif
139
140 echo "Currently, RLF_LIBRARIES = "
141 echo " $RLF_LIBRARIES"
142 echo ""
143
144 Library_Found:
145
146 #-----
147 # Set other X Window System environment variables (besides DISPLAY).
148 #
149 # Make a couple of guesses as to where the RLF_Browser file resides.
150 # If the RLF_Browser is not found, then alert the user.
151 #-----
152 if ( ! $?XAPPLRESDIR ) then
153
154     No_Browser_File:
155
156     if ( -e RLF_Browser ) then
157
158         setenv XAPPLRESDIR '/bin/pwd'

```

```
159
160     else
161         if ( ! -e /usr/lib/X11/app-defaults/RLF_Browser ) then
162
163             echo ""
164             echo "WARNING: "
165             echo "     Environment variable XAPPLRESDIR is undefined."
166             echo "     You need to find the pathname to the RLF_Browser file."
167             echo "     Then issue the following command:"
168             echo "         setenv XAPPLRESDIR <pathname>"
169             echo ""
170             echo ""
171
172         else
173
174             echo ""
175             echo "You will be using the following RLF_Browser resource file"
176             echo "  /usr/lib/X11/app-defaults/RLF_Browser"
177             echo ""
178             setenv XAPPLRESDIR /usr/lib/X11/app-defaults
179
180         endif
181
182     endif
183
184 else
185     if ( ! -e $XAPPLRESDIR/RLF_Browser ) then
186
187         goto No_Browser_File
188
189     endif
190
191 endif
192
193 #-----
194 # Check if a "bitmaps" directory resides beneath $XAPPLRESDIR.
195 #-----
196 if ( $?XAPPLRESDIR ) then
197     echo ""
198     echo "Currently, XAPPLRESDIR = "
199     echo "  $XAPPLRESDIR"
200     echo ""
201
202     if ( ! -d $XAPPLRESDIR/bitmaps ) then
203         echo ""
204         echo "WARNING: "
205         echo "     Bitmaps directory not found:"
206         echo "     $XAPPLRESDIR/bitmaps was not found.""
207         echo ""
208         echo "     The RLF Graphical Browser will not be able to display"
209         echo "     its bitmaps for the graph nodes. This may make the"
210         echo "     graph display less aesthetically pleasing."
211         echo ""
212         echo "     The 'bitmaps' directory should exist as a subdirectory"
```

```
213         echo "      from the location of the 'RLF_Browser' file."
214         echo "      (This is a Motif limitation.)"
215         echo ""
216     endif
217 endif
218
219 #-----
220 # If the user has not already defined the environment variables
221 # RLF_PAGER and RLF_EDITOR, the script will default the to be
222 # "more" and "vi", respectively.
223 #-----
224 if ( ! $?RLF_PAGER ) then
225     setenv RLF_PAGER    more
226 endif
227
228 if ( ! $?RLF_EDITOR ) then
229     setenv RLF_EDITOR   vi
230 endif
231
232 echo ""
233 echo "RLF_PAGER = $RLF_PAGER"
234 echo "RLF_EDITOR = $RLF_EDITOR"
235 if ( ! $?RLF_WORKING_DIR ) then
236     echo ""
237     echo "RLF_WORKING_DIR undefined, so default to current working directory."
238     echo ""
239     setenv RLF_WORKING_DIR 'pwd'
240 endif
241 echo "RLF_WORKING_DIR = $RLF_WORKING_DIR"
242 echo ""
243
244 #-----
245 # Invoke the RLF Graphical_Browser with any command line arguments
246 # entered by the user.
247 #-----
248 echo ""
249 echo "Starting the RLF Graphical Browser..."
250 echo ""
251 Graphical_Browser $argv
```

C Appendix: Installation Scripts

C.1 SunAda Support File

C.1.1 File: Build_RLF.var

```

1  #-----
2  #
3  # Build_RLF.var - RLF software installation configuration file.
4  #
5  #-----
6
7  setenv COMPILERNAME SunAda
8
9  #
10 # Directory for installation of the RLF Graphical Browser resource file
11 # and the bitmaps sub-directory.
12 # Note: You usually need root privilege to write in this directory,
13 # Installation of the resource file and the bitmaps sub-directory will fail
14 # if write permission is denied.
15 #
16 setenv APPDEFAULTS /afs/stars.reston.unisys.com/work/rlf/app-defaults
17
18 #
19 # Uncomment and edit these lines if you do not want to
20 # be prompted for the environment variables
21 #
22
23 setenv RLFHOME /afs/stars.reston.unisys.com/test/rlf/4.1
24
25 setenv COMPILERPATH /compiler/sunada_1.1
26
27 setenv RGB /afs/reston/test/rgb/1.1/Build_Sun4/SunAda1.1/rgb
28
29 setenv ADANOTIFHOME /testbed_4/sercx11a
30
31 setenv BUILD_SAMPLE_LIB Y
32
33 setenv LIBX /usr/lib/libX11.a
34 setenv CC /bin/cc
35
36 #
37 # Uncomment and edit these lines for building on PCTE
38 #
39
40 setenv LIBEMER /testbed/emer/bin/lib/libemer.a
41 setenv PCTE_ADA /afs/reston/test/for_PCTE/adapcte/0.3/Build_Sun4/SunAda1.1
42
43 setenv PCTE_UTIL $PCTE_ADA/util.o
44 if ( ! $PCTE ) setenv PCTE N
45
46 #
47 # If you really don't want to build the browser, set to N.

```

```

48 #
49 setenv BUILD_RLFGB Y
50
51 #
52 # Uncomment (but do not edit) these lines.
53 #
54 setenv RLFBIN $RLFHOME/bin
55
56 if ( $PCTE == Y ) then
57     setenv RLFBINDEST $RLFHOME/pcte/bin
58     if ( ! -d $RLFHOME/pcte ) mkdir $RLFHOME/pcte
59     if ( ! -d $RLFHOME/pcte ) mkdir $RLFHOME/pcte/bin
60 else
61     setenv RLFBINDEST $RLFHOME/unix/bin
62     if ( ! -d $RLFHOME/unix ) mkdir $RLFHOME/unix
63     if ( ! -d $RLFHOME/unix/bin ) mkdir $RLFHOME/unix/bin
64 endif
65
66 setenv RLFCODE $RLFHOME/src_rel/code
67
68 if ( $PCTE == Y ) then
69     setenv TARGET $RLFHOME/pcte/build_sunada1.1
70     if ( ! -d $RLFHOME/pcte ) mkdir $RLFHOME/pcte
71 else
72     setenv TARGET $RLFHOME/unix/build_sunada1.1
73     if ( ! -d $RLFHOME/unix ) mkdir $RLFHOME/unix
74 endif
75 setenv RLF_LIBRARIES $RLFHOME/unix/Libraries
76
77
78 #-----
79 # END OF REQUIRED EDITING FOR BATCH MODE
80 #-----
81
82 #
83 # Define the location of RLFHOME
84 #
85 setRLFHOME:
86 if ( $?RLFHOME == 0 ) then      # if NOT set
87     echo ""
88     echo "Specify path to top-level RLFHOME directory "
89     echo "----- "
90     echo " Examples: "
91     echo "      /mybase/RLF "
92     echo "      /afs/myhome/sec/rlf "
93     echo "      /usr/tools/rlf "
94     echo "      etc. "
95     echo ""
96     echo ""
97     echo -n " RLFHOME = "
98     set noglob
99     setenv RLFHOME $<
100     echo ""
101 endif

```

```

102
103 if ( $RLFHOME == "" ) then
104     unsetenv RLFHOME
105     goto setRLFHOME
106 endif
107
108 if ( ! -e $RLFHOME ) then
109     echo ""
110     echo "T*** $RLFHOME does not exist ***"
111     echo "*** Please try again. ***"
112     echo ""
113     unsetenv RLFHOME
114     goto setRLFHOME
115 else
116     if ( $?RLFBIN == 0 ) setenv RLFBIN $RLFHOME/bin
117     if ( $?RLFCODE == 0 ) setenv RLFCODE $RLFHOME/code
118 endif
119
120 echo ""
121 echo "          RLFHOME = $RLFHOME"
122 echo "          RLFBIN = $RLFBIN"
123 echo "          RLFBINDEST = $RLFBINDEST"
124 echo "          RLFCODE = $RLFCODE"
125 echo ""
126
127 #
128 # Establish a path to the compilation system.
129 #
130 setCOMPILERPATH:
131
132 if ( $?COMPILERPATH == 0 ) then
133     echo ""
134     echo "Specify path to the SunAda compiler "
135     echo "-----"
136     echo "  Examples: "
137     echo "          /compilers/sunada_1.1"
138     echo "          /usr/bin/sunada"
139     echo "          etc."
140     echo ""
141     echo -n "  COMPILERPATH = "
142     setenv COMPILERPATH $<
143     echo ""
144 endif
145 if ( ( $COMPILERPATH == ) || ( ! -e $COMPILERPATH/bin/ada ) ) then
146     echo ""
147     echo "*** Cannot find Ada compiler in $COMPILERPATH/bin ***"
148     echo "*** Please try again. ***"
149     unsetenv COMPILERPATH
150     goto setCOMPILERPATH
151 else
152     setenv COMPILERBIN $COMPILERPATH/bin
153     setenv COMPILE "$COMPILERBIN/ada -v "
154     setenv LINK "$COMPILERBIN/a.ld -v "
155     setenv LINK_FLAGS ""

```

```

156 endif
157
158
159 #
160 # Define the destination of the RLF build.
161 #
162 if ( $?TARGET == 0 ) then
163     echo ""
164     echo "Specify directory where the RLF Ada libraries will be built"
165     echo "-----"
166     echo ""
167     echo "      1. $RLFFHOME/Build_$COMPILERNAME "
168     echo "      2. Let me specify the target build directory."
169     echo "      3. Exit. "
170     echo ""
171
172 setTARGETDIR:
173 #
174 # Read input from user.
175 #
176 set answer = 0
177 echo -n "Please enter 1, 2, or 3 > "
178 set answer = ( $< )
179 set answer = ( $answer )
180 set answer = $answer[1]
181
182 Get_Valid_Input:
183 while ( $answer[1] != 1 && $answer[1] != 2 && $answer[1] != 3 )
184     echo ""
185     echo "I*** Invalid input. Please try again. ***"
186     echo ""
187     echo -n "Please enter A NUMBER: 1, 2, or 3 > "
188     set answer = ( $< )
189     set answer = ( $answer )
190     set answer = $answer[1]
191
192 end
193 while ( $answer < 1 || $answer > 3 )
194     echo ""
195     echo "I*** Invalid input. Please try again. ***"
196     echo ""
197     echo -n "Please enter 1, 2, or 3 > "
198     set answer = ( $< )
199     set answer = ( $answer )
200     set answer = $answer[1]
201 end
202
203 echo ""
204 echo "You chose: $answer"
205
206 switch ( $answer )
207     case 1:
208         setenv TARGET $RLFFHOME/Build_$COMPILERNAME
209         breaksw

```

```

210         case 2:
211             echo -n " TARGET = "
212             setenv TARGET $<
213             echo ""
214             breaksw
215         case 3:
216             unsetenv TARGET
217             echo ""
218             echo \
219             "*** Sorry, a target build directory must be specified ***"
220             echo ""
221             exit 0
222             breaksw
223         default:
224             echo "T*** Invalid input value ***"
225             echo "*** Please try again ***"
226             echo ""
227             goto setTARGETDIR
228     endsw
229
230     #
231     # Check for null entry
232     #
233     if ( $TARGET == "" ) then
234         setenv TARGET $RLHOME/Build_$COMPILENAME
235     endif
236 endif
237
238 #
239 # By default, use /bin/cc for the C compiler.
240 #
241 if ( $?CC == 0 ) then
242     if ( -e /bin/cc ) then
243         setenv CC /bin/cc
244     else
245         unsetenv CC
246     endif
247 endif
248
249 #
250 # Establish a path to the C compiler.
251 #
252 setCC:
253
254 if ( $?CC == 0 ) then
255     echo ""
256     echo "Specify path to the C compiler (e.g., /bin/cc)"
257     echo ""
258     echo -n " CC = "
259     setenv CC $<
260     echo ""
261 endif
262 if ( ! -e $CC ) then
263     echo ""

```

```

264     echo "*** Cannot find C compiler $CC ***"
265     echo "*** Please try again. ***"
266     unsetenv CC
267     goto setCC
268 endif
269 echo ""
270 echo "Using the following C compiler: $CC"
271 echo "----- "
272
273
274 #
275 # The following environment variables identify the directories
276 # required to build the RLF Graphical Browser. This requires the
277 # SERC Ada/Motif and Reusable Graphical Browser software.
278 #
279
280 setLIBX:
281     if ( $?LIBX == 0 ) then
282         echo ""
283         echo "Specify pathname of the X11 Xlib object archive "
284         echo "----- "
285         echo "  Examples: "
286         echo "    /usr/lib/libX11.a"
287         echo ""
288         echo -n "  LIBX = "
289         setenv LIBX $<
290         echo ""
291     endif
292     if ( $LIBX == ) goto setLIBX
293     if ( ! -e $LIBX ) then
294         echo ""
295         echo "I*** $LIBX does not exist ***"
296         echo "*** Please try again. ***"
297         echo ""
298         unsetenv LIBX
299         goto setLIBX
300     endif
301
302 setADAMOTIFHOME:
303     if ( $?ADAMOTIFHOME == 0 ) then
304         echo ""
305         echo "Specify path to the SERC Ada/Motif software home directory "
306         echo "----- "
307         echo "  Examples: "
308         echo "    /mybase/adamotif1.0/sercx11 "
309         echo "    /sercx11 "
310         echo "    etc."
311         echo ""
312         echo -n "  ADAMOTIFHOME = "
313         setenv ADAMOTIFHOME $<
314         echo ""
315     endif
316     if ( $ADAMOTIFHOME == ) goto setADAMOTIFHOME
317     if ( ! -e $ADAMOTIFHOME ) then

```

```

318     echo ""
319     echo "I*** $ADAMOTIFHOME does not exist ***"
320     echo "*** Please try again. ***"
321     echo ""
322     unsetenv ADAMOTIFHOME
323     goto setADAMOTIFHOME
324 endif
325
326 setRGB:
327     if ( $?RGB == 0 ) then
328         echo ""
329         echo "Specify path to the RGB Ada libraries "
330         echo "----- "
331         echo "  Examples: "
332         echo "    /mybase/browser/Build_Sun4/rgb "
333         echo "    /afs/myhome/see/rgb/1.0/Sun4/rgb "
334         echo "    etc. "
335         echo ""
336         echo -n "  RGB = "
337         setenv RGB $<
338         echo ""
339     endif
340     if ( $RGB == ) goto setRGB
341     if ( ! -e $RGB ) then
342         echo ""
343         echo "I*** $RGB does not exist ***"
344         echo "*** Please try again. ***"
345         echo ""
346         unsetenv RGB
347         goto setRGB
348     else
349         if ( -e $RGB/call_ada.o ) then
350             setenv RGB_OBJ $RGB/call_ada.o
351         else
352             echo ""
353             echo "I*** Cannot find $RGB/call_ada.o ***"
354             echo "*** Please try again. ***"
355             unsetenv RGB
356             goto setRGB
357         endif
358     endif
359
360 #
361 # The following environment variable determines whether the sample animals
362 # model will be built and if so sets up the RLF_LIBRARIES environment
363 # variable.
364 #
365 if ( ! $?BUILD_SAMPLE_LIB ) then
366     echo ""
367     Get_Valid_Build_Lib_Input:
368     echo ""
369     echo "Specify sample library build decision "
370     echo "----- "
371     echo ""

```

```
372 echo " Do you wish to translate the sample RLF library Animals "
373 echo -n " after the source code build? (y/n) > "
374 #
375 # Read input from user.
376 #
377 set answer = 0
378 set answer = ( $< )
379 set answer = ( $answer )
380 set answer = $answer[1]
381
382 if ( $answer[1] != y && $answer[1] != Y && \
383     $answer[1] != n && $answer[1] != N ) then
384     echo ""
385     echo "I*** Invalid input. Please try again. ***"
386     echo ""
387     echo -n "Please enter y or n "
388     echo ""
389     goto Get_Valid_Build_Lib_Input
390 else
391     setenv BUILD_SAMPLE_LIB $answer
392 endif
393 endif
394
395 if ( $BUILD_SAMPLE_LIB == y || $BUILD_SAMPLE_LIB == Y ) then
396     setenv BUILD_SAMPLE_LIB Y
397     if ( ! $?RLF_LIBRARIES ) then
398         setenv RLF_LIBRARIES $RLFHOMELibraries
399     endif
400     echo ""
401     echo " BUILD_SAMPLE_LIB = $BUILD_SAMPLE_LIB"
402     echo " RLF_LIBRARIES = $RLF_LIBRARIES"
403     echo ""
404 else
405     setenv BUILD_SAMPLE_LIB N
406     echo ""
407     echo " BUILD_SAMPLE_LIB = $BUILD_SAMPLE_LIB"
408     echo ""
409 endif
410
411
412 echo ""
413 echo " TARGET = $TARGET"
414 echo ""
415 echo " COMPILERNAME = $COMPILERNAME"
416 echo " COMPILERPATH = $COMPILERPATH"
417 echo " COMPILERBIN = $COMPILERBIN"
418 echo " COMPILE = $COMPILE"
419 echo " LINK = $LINK"
420 echo " LINK_FLAGS = $LINK_FLAGS"
421 echo " CC = $CC"
422 echo ""
423 echo " RGB = $RGB"
424 echo " RGB_OBJ = $RGB_OBJ"
425 echo ""
```

March 1993

STARS-UC-05156/013/00

```
426 echo "          LIBX = $LIBX"  
427 echo "      ADAMOTIFHOME = $ADAMOTIFHOME"  
428 echo ""
```

C.2 SunAda Scripts for Installing RLF

C.2.1 Script: Install_Rlf_src

```

1  #! /bin/csh -f
2  #
3  # Install_RLF - C Shell script to install RLF software.
4  #
5  # Usage: Install_RLF
6  #
7  set config_file = "code/Build_RLF.var"          # name of installation
8                                                    # configuration file
9  set interactv_install = "code/Build_RLF.csh"    # name of interactive
10                                                    # installation file
11
12
13  stty ignbrk          # ignore break on input
14  stty -brkint         # don't signal SIGINT on break
15
16  set cmdname = $0
17  if ( $#argv != 0 ) then      # check cmd line usage
18      echo "Usage: $cmdname:t"  # print only tail of cmd name
19      exit
20  end.f
21
22  clear
23
24  cat << X_SCREEN_X
25
26  +-----+
27  |                                     |
28  |               RLF 4.1 Installation Script               |
29  |               Source Code Release                       |
30  |                                     |
31  +-----+
32
33  You must choose one of the following installation options:
34
35
36  1) Interactive installation
37
38      * You are prompted for all necessary
39      configuration values (i.e., pathnames).
40
41
42  2) Edit the file that contains the configuration values
43
44      * You edit the file "Build_RLF.var" and
45      set the configuration values appropriately
46      for your site.
47
48
49  3) EXIT this script.
50

```

```

51
52
53     (If you do not edit the "Build_RLF.var" file, or specify
54     invalid values, you will be prompted for the configuration
55     values anyway.)
56
57
58     Which installation option do you prefer?
59
60 X_SCREEN_X
61
62 #
63 # Read input from user.
64 #
65 set answer = 0
66 echo -n "Please enter 1, 2, or 3 > "
67 set noglob
68 set answer = ( $< )
69 set answer = ( $answer )
70 set answer = $answer[1]
71 unset noglob
72
73 Get_Valid_Input:
74     while ( $answer[1] != 1 && $answer[1] != 2 && $answer[1] != 3 )
75         echo ""
76         echo "I*** Invalid input. Please try again. ***"
77         echo ""
78         echo -n "Please enter A NUMBER: 1, 2, or 3 > "
79         set answer = ( $< )
80         set answer = ( $answer )
81         set answer = $answer[1]
82
83     end
84     while ( $answer[1] < 1 || $answer[1] > 3 )
85         echo ""
86         echo "I*** Invalid input. Please try again. ***"
87         echo ""
88         echo -n "Please enter 1, 2, or 3 > "
89         set answer = ( $< )
90         set answer = ( $answer )
91         set answer = $answer[1]
92     end
93
94     echo ""
95     echo "You chose: $answer[1]"
96
97 #
98 # Process input, execute appropriate procedure.
99 #
100 switch ( "$answer[1]" )      # look at char
101     case [1]:                # Interactive
102         echo ""
103         echo "+-----+
104         echo "| Executing interactive installation script. |"
```

```

105      echo "+-----+"
106      echo ""
107      echo "Build the UNIX or the PCTE version of RLF?"
108      echo " (The default is UNIX.)"
109      echo ""
110      echo "  1. UNIX"
111      echo "  2. PCTE"
112      echo ""
113
114      #
115      # Read input from user.
116      #
117      set reply = 0
118      echo -n "Please enter 1 or 2 > "
119      set noglob
120      set reply = ( $< )
121      set reply = ( $reply )
122      set reply = $reply[1]
123      unset noglob
124
125      Get_Valid_Input:
126          while ( $reply[1] != 1 && $reply[1] != 2 )
127              echo ""
128              echo "I*** Invalid input. Please try again. ***"
129              echo ""
130              echo -n "Please enter A NUMBER: 1 or 2 > "
131              set reply = ( $< )
132              set reply = ( $reply )
133              set reply = $reply[1]
134
135          end
136          while ( $reply[1] < 1 || $reply[1] > 3 )
137              echo ""
138              echo "I*** Invalid input. Please try again. ***"
139              echo ""
140              echo -n "Please enter 1 or 2 > "
141              set reply = ( $< )
142              set reply = ( $reply )
143              set reply = $reply[1]
144          end
145
146      echo ""
147      echo "You chose: $reply[1]"
148
149      pushd code > /dev/null
150
151      #
152      # Invoke the build script with the appropriate argument.
153      #
154      unset noglob
155      if ( $reply[1] == 1 ) then
156          setenv PCTE N
157          echo "Building the UNIX version."
158

```

```

159         $interactv_install
160
161     else
162         setenv PCTE Y
163         echo "Building the PCTE version."
164
165         $interactv_install -pcte
166
167     endif
168
169     popd > /dev/null
170     breaksw
171
172 case [2]:                # Edit the 'var' file
173     #
174     # Calculate string lengths for proper display.
175     #
176     set beginning = "      |                $config_file"
177
178     @ line = 'expr length "      +-----+"'
179     @ remainder = $line - 'expr length "$beginning"'
180
181     echo ""
182     echo "      +-----+"
183     echo "      | To build the Reuse Library Framework in batch mode,|"
184     echo "      | you must edit the installation configuration file:  |"
185     echo "      |                                                        |"
186
187     set ctr = 1
188
189     set line = "${beginning}"
190     while ( $ctr < $remainder )
191         set line = "${line} "
192         @ ctr = $ctr + 1
193     end
194     echo -n "$line"
195     echo "|"
196     echo "      | Then execute the command:                               |"
197     echo "      |                                                                |"
198     echo "      | Build_RLF.csh >& LOG &                                       |"
199     echo "      |                                                                |"
200     echo "      | Once the job is finished, check the LOG file for          |"
201     echo "      | errors.                                                       |"
202     echo "      |                                                                |"
203     echo "      +-----+"
204     breaksw
205
206 case [3]:                # Exit
207     echo ""
208     echo "Exiting installation script."
209     breaksw
210
211 case [!%]:

```

```
212      echo ""
213      echo "Pathological input."
214      echo "Of course C shell scripts are breakable, please be kind."
215      echo "I"
216      exit -1
217      breaksw
218
219      default:
220          # if here, something's wrong
221          echo "*** Invalid input. ***"
222          goto Get_Valid_Input
223          breaksw
224      endsw
225
226
227      echo ""
228      exit 1
229
230
```

C.2.2 Script: Build_RLF.csh

```

1  #!/bin/csh -f
2  #-----
3  #
4  # Build_RLF.csh - C Shell script to build the source code release
5  #                 of the RLF software.
6  #
7  #
8  #      Usage:
9  #      Build_RLF.csh [-pcte] [-link_only]"
10 #
11 #-----
12 #
13 # Uncomment the following two lines if you need to increase the
14 # system resources on your host; else ignore.
15 #
16 ###limit stacksize unlimited
17 ###limit datasize unlimited
18 #
19 #
20 # RLF v.4.2 -- argument processing.
21 #
22 if ( $#argv > 2 ) then
23     echo ""
24     echo "${prog:t}: Only 2 arguments maximum are allowed."
25     echo ""
26     goto USAGE
27 endif
28 #
29 #
30 # These are the default settings -- build for UNIX; LINK_ONLY is false.
31 #
32 setenv PCTE N
33 setenv LINK_ONLY N
34 #
35 #
36 # v.4.1
37 #
38 # Process the command-line arguments.
39 #
40 #
41 if ( $#argv > 0 ) then
42     if ( ( "$argv[1]" != "-pcte" ) && ( "$argv[1]" != "-link_only" ) ) then
43         echo ""
44         echo "Error in arguments: $argv[1] was not understood"
45         goto USAGE
46     endif
47     if ( "$argv[1]" == "-pcte" ) setenv PCTE Y
48     if ( "$argv[1]" == "-link_only" ) setenv LINK_ONLY Y
49 endif
50 if ( $#argv > 1 ) then
51     if ( ( "$argv[2]" != "-pcte" ) && ( "$argv[2]" != "-link_only" ) ) then
52         echo ""

```

```
53         echo "Error in arguments: $argv[2] was not understood"
54     goto USAGE
55 endif
56     if ( "$argv[2]" == "-pcte" ) setenv PCTE Y
57     if ( "$argv[2]" == "-link_only" ) setenv LINK_ONLY Y
58 endif
59
60 #
61 # Read in the site-dependent data from the 'var' file.
62 #
63 echo ""
64 echo "Define the site-dependent environment variables."
65 echo "-----"
66 echo ""
67 source Build_RLF.var
68
69 if ( $LINK_ONLY == Y ) goto LINK_IT
70
71
72
73 foreach dir ( Rbd1 Rlf Rlf_Gb Lmdl Library_Manager )
74     set echo
75     if ( ! -d $TARGET/$dir ) mkdir -p $TARGET/$dir
76     unset echo >& /dev/null
77 end
78
79
80 #
81 # Build Ada libraries
82 #
83 Build_Ada_Libraries.csh
84 if ($status != 0) exit $status
85
86 #
87 # Perform the build.
88 #
89 echo ""
90 echo "Building RLF Executables"
91 echo ""
92 Build_Rlfdir.csh
93 if ($status != 0) exit $status
94
95 LINK_IT:
96 echo ""
97 echo "Building Lmdl Translator Executable"
98 echo ""
99 Build_Lmdl.csh
100 if ($status != 0) exit $status
101
102 echo ""
103 echo "Building Rbd1 Translator Executable"
104 echo ""
105 Build_Rbd1.csh
106 if ($status != 0) exit $status
```

```
107
108 echo ""
109 echo "Building RLF Library_Manager Executable"
110 echo ""
111 Build_Library_Manager.csh
112 if ($status != 0) exit $status
113
114 echo ""
115 echo "Building Graphical_Browser Executable"
116 echo ""
117 Build_Rlf_Gb.csh
118 if ($status != 0) exit $status
119
120 echo ""
121 echo "RLF Source Code Build Complete"
122 echo ""
123
124
125 echo ""
126 echo "Moving executables to the directory:"
127 echo "    $RLFBINDEST"
128 echo ""
129 if ( ! -d $RLFBINDEST ) mkdir $RLFBINDEST
130 mv -f $TARGET/Library_Manager/Library_Manager $RLFBINDEST
131 mv -f $TARGET/Lmdl/Lmdl $RLFBINDEST
132 mv -f $TARGET/Rbdl/Rbdl $RLFBINDEST
133 mv -f $TARGET/Rlf_Gb/Graphical_Browser $RLFBINDEST
134
135
136 echo ""
137 echo "Moving the RLF GB resource file (RLF_Browser) to directory: "
138 echo "    $APPDEFAULTS"
139 echo ""
140 if ( ! -d $APPDEFAULTS ) mkdir $APPDEFAULTS
141 cp $RLFBIN/RLF_Browser $APPDEFAULTS/RLF_Browser
142
143 echo ""
144 echo "Moving the RLF GB bitmap files to directory: "
145 echo "    $APPDEFAULTS/bitmaps"
146 echo ""
147 cp -r $RLFBIN/bitmaps $APPDEFAULTS
148
149 echo ""
150 echo "Setting permissions for bitmap files in directory: "
151 echo "    $APPDEFAULTS/bitmaps"
152 echo ""
153 chmod 744 $APPDEFAULTS/bitmaps/*
154
155 if ( $BUILD_SAMPLE_LIB == Y || $BUILD_SAMPLE_LIB == y ) then
156     if ( $PCTE == Y || $PCTE == y ) then
157         echo "Not translating the Animals sample library."
158         echo "Run the appropriate 'esh' script after starting PCTE."
159     else
160         echo ""
```

```
161     echo "Translating the Animals sample library."
162     echo ""
163     pushd $RLFSHOME/models/animals > /dev/null
164     set path = ( $RLFBIN $path )
165     Build_Animals_Lib.csh
166     popd > /dev/null
167     if ($status != 0) exit $status
168 endif
169
170 echo ""
171 echo "Installation Complete"
172 echo ""
173 exit 0
174
175 USAGE:
176     echo ""
177     echo "Usage:"
178     echo "  ${prog:t}: [-pcte] [-link_only]"
179     echo ""
180     exit 1
181
```

C.2.3 Script: Build_Ada_Libraries.csh

```

1  #!/bin/csh -f
2  #-----
3  #
4  # Build_Ada_Libraries.csh - RLF installation file for building Ada libraries
5  #
6  #-----
7
8  ###limit stacksize unlimited
9  ###limit datasize unlimited
10
11  echo ""
12  echo "Making Ada libraries"
13  echo ""
14
15  #
16  # Add the compiler binaries to the path for SERC Ada/Motif
17  # to make the Motif Ada libraries.
18  #
19  set path = ( $COMPILERBIN $path )
20
21  #
22  # Core RLF.
23  #
24  cd $TARGET/Rlf
25  if ( ! -e ada.lib ) then
26      $COMPILERBIN/a.mklib -f . $COMPILERPATH/verdirxlib
27      if ($PCTE == y || $PCTE == Y) then
28          $COMPILERBIN/a.path -i $PCTE_ADA
29      endif
30  else
31      echo "Ada library exists in directory:"
32      echo -n " "
33      pwd
34  endif
35
36  #
37  # Modeling language translators.
38  #
39  foreach dir ( Lmdl Rbdl )
40      cd $TARGET/$dir
41      if ( ! -e ada.lib ) then
42          $COMPILERBIN/a.mklib -f . $COMPILERPATH/verdirxlib
43          $COMPILERBIN/a.path -i $TARGET/Rlf
44          if ($PCTE == y || $PCTE == Y) then
45              $COMPILERBIN/a.path -i $PCTE_ADA
46          endif
47      else
48          echo "Ada library exists in directory:"
49          echo -n " "
50          pwd
51      endif
52  end

```

```
53
54 #
55 # RLF Library Manager.
56 #
57 cd $TARGET/Library_Manager
58 if ( ! -e ada.lib ) then
59     $ADAMOTIFHOME/sup/AdaMotif.mklib
60     $COMPILERBIN/a.path -i $TARGET/Rlf
61     if ($PCTE == y || $PCTE == Y) then
62         $COMPILERBIN/a.path -i $PCTE_ADA
63     endif
64 else
65     echo "Ada library exists in directory:"
66     echo -n " "
67     pwd
68 endif
69
70 #
71 # RLF Graphical Browser.
72 #
73 cd $TARGET/Rlf_Gb
74 if ( ! -e ada.lib ) then
75     $ADAMOTIFHOME/sup/AdaMotif.mklib
76     $COMPILERBIN/a.path -i $TARGET/Rlf
77     $COMPILERBIN/a.path -i $RGB
78     if ($PCTE == y || $PCTE == Y) then
79         $COMPILERBIN/a.path -i $PCTE_ADA
80     endif
81 else
82     echo "Ada library exists in directory:"
83     echo -n " "
84     pwd
85 endif
86
```

C.2.4 Script: Build_Rlfdir.csh

```

1  #!/bin/csh -f
2  ###limit stacksize unlimited
3  ###limit datasize unlimited
4
5  pushd $TARGET/Rlf > /dev/null
6
7  echo ""
8  echo "Establishing soft links to the Rlf source files"
9  echo ""
10 foreach dir ( Common Adaknet Adatau Hybrid Ada_Actions )
11     foreach file ( $RLFCODE/$dir/*.a )
12         set echo
13         if ( ! -e ${file:t} ) ln -s $file ${file:t}
14         unset echo >& /dev/null
15     end
16 end
17 #
18 # ...including the 'C' source files
19 #
20 if ( ! -e support.c ) ln -s $RLFCODE/Common/support.c .
21
22 echo ""
23 echo "Compiling Rlf code"
24 echo ""
25 $CC -c support.c
26
27 #
28 # compiling IO specs for Unix, Pcte or Cais
29 #
30
31 $COMPILE c_interface_spec.a
32 if ($status != 0) exit $status
33
34 if ($PCTE == y || $PCTE == Y) then
35     $COMPILE rlf_text_io_pcte.a
36     if ($status != 0) exit $status
37     $COMPILE rlf_sequential_io_pcte.a
38     if ($status != 0) exit $status
39     $COMPILE rlf_direct_io_pcte.a
40     if ($status != 0) exit $status
41     $COMPILE rlf_pcte_spec.a
42     if ($status != 0) exit $status
43     $COMPILE rlf_pcte_body.a
44     if ($status != 0) exit $status
45     $COMPILE pcte_object_create.a
46     if ($status != 0) exit $status
47     $COMPILE rlf_text_io_pcte.a
48     if ($status != 0) exit $status
49     $COMPILE rlf_sequential_io.a
50     if ($status != 0) exit $status
51     $COMPILE rlf_direct_io.a
52     if ($status != 0) exit $status

```

```
53     $COMPILE pcte_invoke_string.a
54     if ($status != 0) exit $status
55 else
56     $COMPILE rlf_text_io_unix_.a
57     if ($status != 0) exit $status
58     $COMPILE rlf_sequential_io_unix_.a
59     if ($status != 0) exit $status
60     $COMPILE rlf_direct_io_unix_.a
61     if ($status != 0) exit $status
62     $COMPILE rlf_text_io_unix.a
63     if ($status != 0) exit $status
64     $COMPILE rlf_sequential_io.a
65     if ($status != 0) exit $status
66     $COMPILE rlf_direct_io.a
67     if ($status != 0) exit $status
68     $COMPILE unix_invoke_string.a
69     if ($status != 0) exit $status
70 endif
71
72
73     $COMPILE generic_sequences_spec.a
74     if ($status != 0) exit $status
75     $COMPILE generic_sequences_body.a
76     if ($status != 0) exit $status
77     $COMPILE objects_spec.a
78     if ($status != 0) exit $status
79     $COMPILE generic_hash_spec.a
80     if ($status != 0) exit $status
81     $COMPILE generic_hash_body.a
82     if ($status != 0) exit $status
83     $COMPILE system_environment_spec.a
84     if ($status != 0) exit $status
85     $COMPILE system_environment_vads_body.a
86     if ($status != 0) exit $status
87     $COMPILE rlf_globals_spec.a
88     if ($status != 0) exit
89     $COMPILE rlfrc_parser_spec.a
90     if ($status != 0) exit
91     $COMPILE rlfrc_parser_support_spec.a
92     if ($status != 0) exit
93
94
95 if ($PCTE == y || $PCTE == Y) then
96     $COMPILE rlf_constants_pcte.a
97     if ($status != 0) exit
98 else
99     $COMPILE rlf_constants_unix.a
100     if ($status != 0) exit
101 endif
102
103
104     $COMPILE rlf_globals_body.a
105     if ($status != 0) exit
106     $COMPILE rlfrc_scanner_dfa.a
```

```
107     if ($status != 0) exit
108     $COMPILE rlfrc_scanner_io.a
109     if ($status != 0) exit
110     $COMPILE rlfrc_parser_tokens.a
111     if ($status != 0) exit
112     $COMPILE rlfrc_scanner.a
113     if ($status != 0) exit
114     $COMPILE rlfrc_parser_support_body.a
115     if ($status != 0) exit
116     $COMPILE rlfrc_parser_goto.a
117     if ($status != 0) exit
118     $COMPILE rlfrc_parser_shift_reduce.a
119     if ($status != 0) exit
120     $COMPILE rlfrc_parser.a
121     if ($status != 0) exit
122     $COMPILE adaknet_name_types.a
123     if ($status != 0) exit $status
124     $COMPILE fstring_spec.a
125     if ($status != 0) exit $status
126     $COMPILE fstring_body.a
127     if ($status != 0) exit $status
128     $COMPILE labels_spec.a
129     if ($status != 0) exit $status
130     $COMPILE lists_spec.a
131     if ($status != 0) exit $status
132     $COMPILE lists_body.a
133     if ($status != 0) exit $status
134
135
136 if ($PCTE == y || $PCTE == Y) then
137     $COMPILE network_constants_pcte.a
138     if ($status != 0) exit $status
139 else
140     $COMPILE network_constants.a
141     if ($status != 0) exit $status
142 endif
143
144
145     $COMPILE rlf_univ_types.a
146     if ($status != 0) exit $status
147     $COMPILE ranges_spec.a
148     if ($status != 0) exit $status
149     $COMPILE sets_spec.a
150     if ($status != 0) exit $status
151     $COMPILE sets_body.a
152     if ($status != 0) exit $status
153     $COMPILE sorting_spec.a
154     if ($status != 0) exit $status
155     $COMPILE sorting_body.a
156     if ($status != 0) exit $status
157     $COMPILE stacks_spec.a
158     if ($status != 0) exit $status
159     $COMPILE stacks_body.a
160     if ($status != 0) exit $status
```

```
161     $COMPILE strings_spec.a
162     if ($status != 0) exit $status
163     $COMPILE system_dep_spec.a
164     if ($status != 0) exit $status
165     $COMPILE tables_spec.a
166     if ($status != 0) exit $status
167     $COMPILE tables_body.a
168     if ($status != 0) exit $status
169     $COMPILE tau_lists_spec.a
170     if ($status != 0) exit $status
171     $COMPILE tau_lists_body.a
172     if ($status != 0) exit $status
173     $COMPILE agendas_spec.a
174     if ($status != 0) exit $status
175     $COMPILE agendas_body.a
176     if ($status != 0) exit $status
177     $COMPILE customIO_spec.a
178     if ($status != 0) exit $status
179     $COMPILE commonIO_spec.a
180     if ($status != 0) exit $status
181     $COMPILE message_io_spec.a
182     if ($status != 0) exit $status
183     $COMPILE hashmap_spec.a
184     if ($status != 0) exit $status
185     $COMPILE hashmap_body.a
186     if ($status != 0) exit $status
187     $COMPILE hybrid_tables_spec.a
188     if ($status != 0) exit $status
189     $COMPILE hybrid_tables_body.a
190     if ($status != 0) exit $status
191     $COMPILE integer_hybrid_states_spec.a
192     if ($status != 0) exit $status
193     $COMPILE mono_lock_manager_spec.a
194     if ($status != 0) exit $status
195     $COMPILE unique_identifiers_spec.a
196     if ($status != 0) exit $status
197     $COMPILE unique_identifiers_body.a
198     if ($status != 0) exit $status
199     $COMPILE fixed_strings_spec.a
200     if ($status != 0) exit $status
201     $COMPILE fixed_strings_body.a
202     if ($status != 0) exit $status
203     $COMPILE message_io_body.a
204     if ($status != 0) exit $status
205     $COMPILE facts_spec.a
206     if ($status != 0) exit $status
207     $COMPILE filenames_spec.a
208     if ($status != 0) exit $status
209     $COMPILE generic_concepts_spec.a
210     if ($status != 0) exit $status
211     $COMPILE individual_concepts_spec.a
212     if ($status != 0) exit $status
213     $COMPILE label_table_spec.a
214     if ($status != 0) exit $status
```

```
215     $COMPILE label_table_body.a
216     if ($status != 0) exit $status
217     $COMPILE roles_spec.a
218     if ($status != 0) exit $status
219     $COMPILE rolesets_spec.a
220     if ($status != 0) exit $status
221     $COMPILE rolesets_body.a
222     if ($status != 0) exit $status
223     $COMPILE rule_bases_spec.a
224     if ($status != 0) exit $status
225     $COMPILE rule_bases_body.a
226     if ($status != 0) exit $status
227     $COMPILE actions_spec.a
228     if ($status != 0) exit $status
229     $COMPILE actions_body.a
230     if ($status != 0) exit $status
231     $COMPILE schema_spec.a
232     if ($status != 0) exit $status
233     $COMPILE text_hybrid_states_spec.a
234     if ($status != 0) exit $status
235     $COMPILE fact_depend_functions_spec.a
236     if ($status != 0) exit $status
237     $COMPILE fact_depend_spec.a
238     if ($status != 0) exit $status
239     $COMPILE fact_lists_spec.a
240     if ($status != 0) exit $status
241     $COMPILE fact_lists_body.a
242     if ($status != 0) exit $status
243     $COMPILE fact_value_lists_spec.a
244     if ($status != 0) exit $status
245     $COMPILE fact_value_lists_body.a
246     if ($status != 0) exit $status
247     $COMPILE frules_spec.a
248     if ($status != 0) exit $status
249     $COMPILE frules_body.a
250     if ($status != 0) exit $status
251     $COMPILE networks_spec.a
252     if ($status != 0) exit $status
253     $COMPILE networks_body.a
254     if ($status != 0) exit $status
255     $COMPILE non_monotonic_support_spec.a
256     if ($status != 0) exit $status
257     $COMPILE response_schemas_spec.a
258     if ($status != 0) exit $status
259     $COMPILE fact_schemas_spec.a
260     if ($status != 0) exit $status
261     $COMPILE irules_spec.a
262     if ($status != 0) exit $status
263     $COMPILE network_composites_spec.a
264     if ($status != 0) exit $status
265     $COMPILE questions_spec.a
266     if ($status != 0) exit $status
267     $COMPILE adanet_spec.a
268     if ($status != 0) exit $status
```

```
269     $COMPILE composites_spec.a
270     if ($status != 0) exit $status
271     $COMPILE fact_base_schemas_spec.a
272     if ($status != 0) exit $status
273     $COMPILE fact_bases_spec.a
274     if ($status != 0) exit $status
275     $COMPILE fact_parameter_spec.a
276     if ($status != 0) exit $status
277     $COMPILE irule_bases_spec.a
278     if ($status != 0) exit $status
279     $COMPILE qrules_spec.a
280     if ($status != 0) exit $status
281     $COMPILE question_bases_spec.a
282     if ($status != 0) exit $status
283     $COMPILE adanet_composites_spec.a
284     if ($status != 0) exit $status
285     $COMPILE adanet_state_spec.a
286     if ($status != 0) exit $status
287     $COMPILE adanet_state_body.a
288     if ($status != 0) exit $status
289     $COMPILE app_utils_spec.a
290     if ($status != 0) exit $status
291     $COMPILE fact_parameter_list_spec.a
292     if ($status != 0) exit $status
293     $COMPILE fact_parameter_list_body.a
294     if ($status != 0) exit $status
295     $COMPILE fbase_ops_spec.a
296     if ($status != 0) exit $status
297     $COMPILE isa_browser_spec.a
298     if ($status != 0) exit $status
299     $COMPILE qrule_bases_spec.a
300     if ($status != 0) exit $status
301     $COMPILE agg_browser_spec.a
302     if ($status != 0) exit $status
303     $COMPILE basic_config_functions_spec.a
304     if ($status != 0) exit $status
305     $COMPILE q_agendas_spec.a
306     if ($status != 0) exit $status
307     $COMPILE message_config_spec.a
308     if ($status != 0) exit $status
309     $COMPILE debug_spec.a
310     if ($status != 0) exit $status
311     $COMPILE dump_rddl_spec.a
312     if ($status != 0) exit $status
313     $COMPILE message_TAU_components_spec.a
314     if ($status != 0) exit $status
315     $COMPILE truth_maintenance_spec.a
316     if ($status != 0) exit $status
317     $COMPILE message_adv_config_spec.a
318     if ($status != 0) exit $status
319     $COMPILE message_adv_config_body.a
320     if ($status != 0) exit $status
321     $COMPILE librarian_configuration.a
322     if ($status != 0) exit $status
```

```
323     $COMPILE persistence_spec.a
324     if ($status != 0) exit $status
325     $COMPILE static_persistence_spec.a
326     if ($status != 0) exit $status
327     $COMPILE lib_static_persistence_spec.a
328     if ($status != 0) exit $status
329     $COMPILE message_DTAU_components_spec.a
330     if ($status != 0) exit $status
331     $COMPILE inf_hybrid_states_spec.a
332     if ($status != 0) exit $status
333     $COMPILE library_hybrid_states_spec.a
334     if ($status != 0) exit $status
335     $COMPILE library_hybrid_networks.a
336     if ($status != 0) exit $status
337     $COMPILE library_hybrid_state_ops_spec.a
338     if ($status != 0) exit $status
339     $COMPILE sndl_dump_spec.a
340     if ($status != 0) exit $status
341     $COMPILE examine_network_spec.a
342     if ($status != 0) exit $status
343     $COMPILE examine_network_body.a
344     if ($status != 0) exit $status
345     $COMPILE text_state_file_ops_spec.a
346     if ($status != 0) exit $status
347     $COMPILE text_state_ops_spec.a
348     if ($status != 0) exit $status
349     $COMPILE inf_state_ops_spec.a
350     if ($status != 0) exit $status
351     $COMPILE integer_state_ops_spec.a
352     if ($status != 0) exit $status
353     $COMPILE c_interface_body.a
354     if ($status != 0) exit $status
355     $COMPILE labels_body.a
356     if ($status != 0) exit $status
357     $COMPILE strings_body.a
358     if ($status != 0) exit $status
359
360
361     if ($PCTE == y || $PCTE == Y) then
362         $COMPILE system_dep_pcte.a
363         if ($status != 0) exit $status
364     else
365         $COMPILE system_dep_unix.a
366         if ($status != 0) exit $status
367     endif
368
369
370     $COMPILE customIO_body.a
371     if ($status != 0) exit $status
372     $COMPILE commonIO_body.a
373     if ($status != 0) exit $status
374     $COMPILE mono_lock_manager_body.a
375     if ($status != 0) exit $status
376     $COMPILE filenames_body.a
```

```
377     if ($status != 0) exit $status
378 $COMPILE ranges_body.a
379     if ($status != 0) exit $status
380 $COMPILE generic_concepts_body.a
381     if ($status != 0) exit $status
382 $COMPILE individual_concepts_body.a
383     if ($status != 0) exit $status
384 $COMPILE roles_body.a
385     if ($status != 0) exit $status
386 $COMPILE schema_body.a
387     if ($status != 0) exit $status
388 $COMPILE network_composites_body.a
389     if ($status != 0) exit $status
390 $COMPILE adanet_body.a
391     if ($status != 0) exit $status
392 $COMPILE adanet_constr_destr_ops_sp.a
393     if ($status != 0) exit $status
394 $COMPILE adanet_mod_ops_sp.a
395     if ($status != 0) exit $status
396 $COMPILE adanet_predicates_sp.a
397     if ($status != 0) exit $status
398 $COMPILE adanet_query_ops_sp.a
399     if ($status != 0) exit $status
400 $COMPILE changes_sp.a
401     if ($status != 0) exit $status
402 $COMPILE concept_mod_ops_sp.a
403     if ($status != 0) exit $status
404 $COMPILE concept_predicates_sp.a
405     if ($status != 0) exit $status
406 $COMPILE concept_query_ops_sp.a
407     if ($status != 0) exit $status
408 $COMPILE misc_ops_sp.a
409     if ($status != 0) exit $status
410 $COMPILE restrictions_sp.a
411     if ($status != 0) exit $status
412 $COMPILE role_mod_ops_sp.a
413     if ($status != 0) exit $status
414 $COMPILE role_predicates_sp.a
415     if ($status != 0) exit $status
416 $COMPILE role_query_ops_sp.a
417     if ($status != 0) exit $status
418 $COMPILE roleset_mod_ops_sp.a
419     if ($status != 0) exit $status
420 $COMPILE roleset_predicates_sp.a
421     if ($status != 0) exit $status
422 $COMPILE roleset_query_ops_sp.a
423     if ($status != 0) exit $status
424 $COMPILE roleset_spec_ops_sp.a
425     if ($status != 0) exit $status
426 $COMPILE action_mod_ops_sp.a
427     if ($status != 0) exit $status
428 $COMPILE action_predicates_sp.a
429     if ($status != 0) exit $status
430 $COMPILE action_query_ops_sp.a
```

```
431     if ($status != 0) exit $status
432 $COMPILE set_conversions_sp.a
433     if ($status != 0) exit $status
434 $COMPILE states_sp.a
435     if ($status != 0) exit $status
436 $COMPILE subroles_sp.a
437     if ($status != 0) exit $status
438 $COMPILE composites_body.a
439     if ($status != 0) exit $status
440 $COMPILE adanet_composites_body.a
441     if ($status != 0) exit $status
442 $COMPILE app_utils_body.a
443     if ($status != 0) exit $status
444 $COMPILE isa_browser_body.a
445     if ($status != 0) exit $status
446 $COMPILE sndl_dump_body.a
447     if ($status != 0) exit $status
448 $COMPILE agg_browser_body.a
449     if ($status != 0) exit $status
450 $COMPILE facts_body.a
451     if ($status != 0) exit $status
452 $COMPILE fact_depend_functions_body.a
453     if ($status != 0) exit $status
454 $COMPILE non_monotonic_support_body.a
455     if ($status != 0) exit $status
456 $COMPILE response_schemas_body.a
457     if ($status != 0) exit $status
458 $COMPILE fact_schemas_body.a
459     if ($status != 0) exit $status
460 $COMPILE irules_body.a
461     if ($status != 0) exit $status
462 $COMPILE questions_body.a
463     if ($status != 0) exit $status
464 $COMPILE fact_base_schemas_body.a
465     if ($status != 0) exit $status
466 $COMPILE fact_bases_body.a
467     if ($status != 0) exit $status
468 $COMPILE fact_parameter_body.a
469     if ($status != 0) exit $status
470 $COMPILE qrules_body.a
471     if ($status != 0) exit $status
472 $COMPILE basic_config_functions_body.a
473     if ($status != 0) exit $status
474 $COMPILE debug_body.a
475     if ($status != 0) exit $status
476 $COMPILE dump_rddl_body.a
477     if ($status != 0) exit $status
478 $COMPILE message_TAU_components_body.a
479     if ($status != 0) exit $status
480 $COMPILE truth_maintenance_body.a
481     if ($status != 0) exit $status
482 $COMPILE message_config_body.a
483     if ($status != 0) exit $status
484 $COMPILE persistence_body.a
```

```
485     if ($status != 0) exit $status
486     $COMPILE static_persistence_body.a
487     if ($status != 0) exit $status
488     $COMPILE lib_static_persistence_body.a
489     if ($status != 0) exit $status
490     $COMPILE message_DTAU_components_body.a
491     if ($status != 0) exit $status
492     $COMPILE integer_hybrid_states_body.a
493     if ($status != 0) exit $status
494     $COMPILE text_hybrid_states_body.a
495     if ($status != 0) exit $status
496     $COMPILE fbase_ops_body.a
497     if ($status != 0) exit $status
498     $COMPILE inf_hybrid_states_body.a
499     if ($status != 0) exit $status
500     $COMPILE library_hybrid_states_body.a
501     if ($status != 0) exit $status
502     $COMPILE library_hybrid_state_ops_body.a
503     if ($status != 0) exit $status
504     $COMPILE text_state_file_ops_body.a
505     if ($status != 0) exit $status
506     $COMPILE text_state_ops_body.a
507     if ($status != 0) exit $status
508     $COMPILE inf_state_ops_body.a
509     if ($status != 0) exit $status
510     $COMPILE integer_state_ops_body.a
511     if ($status != 0) exit $status
512     $COMPILE action_operations_spec.a
513     if ($status != 0) exit $status
514     $COMPILE action_operations_body.a
515     if ($status != 0) exit $status
516     $COMPILE action_invocation_spec.a
517     if ($status != 0) exit $status
518     $COMPILE action_routines_spec.a
519     if ($status != 0) exit $status
520     $COMPILE action_routines_body.a
521     if ($status != 0) exit $status
522     $COMPILE display_attr_action_spec.a
523     if ($status != 0) exit $status
524     $COMPILE display_attr_action_body.a
525     if ($status != 0) exit $status
526
527
528 if ($PCTE == y || $PCTE == Y) then
529     $COMPILE display_attr_action_ascii_sp_pcte.a
530     if ($status != 0) exit $status
531     $COMPILE display_attr_action_files_sp_pcte.a
532     if ($status != 0) exit $status
533     $COMPILE display_attr_action_buf_sp_pcte.a
534     if ($status != 0) exit $status
535 else
536     $COMPILE display_attr_action_ascii_sp_unix.a
537     if ($status != 0) exit $status
538     $COMPILE display_attr_action_files_sp_unix.a
```

```
539         if ($status != 0) exit $status
540     $COMPILE display_attr_action_buf_sp_unix.a
541         if ($status != 0) exit $status
542 endif
543
544
545     $COMPILE display_attributes_sp.a
546         if ($status != 0) exit $status
547     $COMPILE export_sp.a
548         if ($status != 0) exit $status
549     $COMPILE extract_action_spec.a
550         if ($status != 0) exit $status
551     $COMPILE extract_action_body.a
552         if ($status != 0) exit $status
553
554
555 if ($PCTE == y || $PCTE == Y) then
556     $COMPILE extract_file_sp_pcte.a
557         if ($status != 0) exit $status
558 else
559     $COMPILE extract_file_sp_unix.a
560         if ($status != 0) exit $status
561 endif
562
563
564     $COMPILE extract_sp.a
565         if ($status != 0) exit $status
566     $COMPILE import_sp.a
567         if ($status != 0) exit $status
568     $COMPILE action_invocation_body.a
569         if ($status != 0) exit $status
570     $COMPILE invoke_sys_string_sp.a
571         if ($status != 0) exit $status
572
573
574 if ($PCTE == y || $PCTE == Y) then
575     $COMPILE invoke_string_sp_pcte.a
576         if ($status != 0) exit $status
577 else
578     $COMPILE invoke_string_sp_unix.a
579         if ($status != 0) exit $status
580 endif
581
582     $COMPILE invoke_ada_proc_sp.a
583         if ($status != 0) exit $status
584
585 popd > /dev/null
```

C.2.5 Script: Build_Lmdl.csh

```
1  #!/bin/csh -f
2  ###limit stacksize unlimited
3  ###limit datasize unlimited
4
5  pushd $TARGET/Lmdl > /dev/null
6
7  if ( $LINK_ONLY == Y ) goto LINK_IT
8
9  echo ""
10 echo "Establishing soft links to the LMDL source files"
11 echo ""
12 foreach file ($RLFCODE/Lmdl/*.a)
13     if ( ! -e ${file:t} ) ln -s $file ${file:t}
14 end
15
16 echo ""
17 echo "Compiling Lmdl"
18 echo ""
19
20 $COMPILE std_report_spec.a
21     if ($status != 0) exit $status
22 $COMPILE topsort_spec.a
23     if ($status != 0) exit $status
24 $COMPILE topsort_body.a
25     if ($status != 0) exit $status
26 $COMPILE std_yyval_error_spec.a
27     if ($status != 0) exit $status
28 $COMPILE std_magic_spec.a
29     if ($status != 0) exit $status
30 $COMPILE std_magic_body.a
31     if ($status != 0) exit $status
32 $COMPILE std_output_spec.a
33     if ($status != 0) exit $status
34 $COMPILE std_yyval_error_body.a
35     if ($status != 0) exit $status
36 $COMPILE std_predefined_tree.a
37     if ($status != 0) exit $status
38 $COMPILE lmdl_system_types_spec.a
39     if ($status != 0) exit $status
40 $COMPILE user_types_spec.a
41     if ($status != 0) exit $status
42 $COMPILE user_types_body.a
43     if ($status != 0) exit $status
44 $COMPILE lmdl_attributed_tree_spec.a
45     if ($status != 0) exit $status
46 $COMPILE lmdl_gets.a
47     if ($status != 0) exit $status
48 $COMPILE lmdl_attributed_tree_body.a
49     if ($status != 0) exit $status
50 $COMPILE globals_spec.a
51     if ($status != 0) exit $status
52 $COMPILE support_spec.a
```

```
53     if ($status != 0) exit $status
54     $COMPILE lmdl_puts.a
55     if ($status != 0) exit $status
56     $COMPILE lmdl_makes.a
57     if ($status != 0) exit $status
58     $COMPILE support_body.a
59     if ($status != 0) exit $status
60     $COMPILE lmdl_evaluator_spec.a
61     if ($status != 0) exit $status
62     $COMPILE lmdl_dgts_spec.a
63     if ($status != 0) exit $status
64     $COMPILE std_lex_spec.a
65     if ($status != 0) exit $status
66     $COMPILE std_user_pkg.a
67     if ($status != 0) exit $status
68     $COMPILE lmdl_lexdef.a
69     if ($status != 0) exit $status
70     $COMPILE lexacts_spec.a
71     if ($status != 0) exit $status
72     $COMPILE globals_body.a
73     if ($status != 0) exit $status
74     $COMPILE lexacts_body.a
75     if ($status != 0) exit $status
76     $COMPILE lmdl_lex_pkg.a
77     if ($status != 0) exit $status
78     $COMPILE lmdl_dgts_body.a
79     if ($status != 0) exit $status
80     $COMPILE lmdl_evaluator_body.a
81     if ($status != 0) exit $status
82     $COMPILE lmdl_parserdefs.a
83     if ($status != 0) exit $status
84     $COMPILE backend_spec.a
85     if ($status != 0) exit $status
86     $COMPILE backend_interface.a
87     if ($status != 0) exit $status
88     $COMPILE std_parser_spec.a
89     if ($status != 0) exit $status
90     $COMPILE lmdl_parser.a
91     if ($status != 0) exit $status
92     $COMPILE std_boot.a
93     if ($status != 0) exit $status
94     $COMPILE std_report_body.a
95     if ($status != 0) exit $status
96     $COMPILE lmdl_display.a
97     if ($status != 0) exit $status
98
99     $COMPILE hybrid_ops_spec.a
100     if ($status != 0) exit $status
101     $COMPILE hybrid_ops_body.a
102     if ($status != 0) exit $status
103     $COMPILE adaknet_ops_spec.a
104     if ($status != 0) exit $status
105     $COMPILE adaknet_ops_body.a
106     if ($status != 0) exit $status
```

```
107     $COMPILE operations_list_def_spec.a
108     if ($status != 0) exit $status
109     $COMPILE operations_list_def_body.a
110     if ($status != 0) exit $status
111
112     $COMPILE backend_body.a
113     if ($status != 0) exit $status
114
115
116 LINK_IT:
117 echo ""
118 echo "Linking Lmdl"
119 echo ""
120
121 if ($PCTE == y || $PCTE == Y) then
122     $LINK boot -o Lmdl $TARGET/Rlf/support.o \
123         $LIBEMER $PCTE_ADA/util.o \
124         $LINK_FLAGS
125     if ($status != 0) exit $status
126 else
127     $LINK boot -o Lmdl $TARGET/Rlf/support.o \
128         $LINK_FLAGS
129     if ($status != 0) exit $status
130 endif
131 popd > /dev/null
132 exit 0
```

C.2.6 Script: Build_Rbdl.csh

```
1  #!/bin/csh -f
2  ###limit stacksize unlimited
3  ###limit datasize unlimited
4
5  pushd $TARGET/Rbdl > /dev/null
6
7  if ( $LINK_ONLY == Y ) goto LINK_IT
8
9  echo ""
10 echo "Establishing soft links to the RBDL source files"
11 echo ""
12 foreach file ($RLFCODE/Rbdl/*.a)
13     if ( ! -e ${file:t} ) ln -s $file ${file:t}
14 end
15
16 echo ""
17 echo "Compiling Rbdl"
18 echo ""
19
20
21 $COMPILE std_yyval_error_spec.a
22     if ($status != 0) exit $status
23 $COMPILE nonstd_magic_spec.a
24     if ($status != 0) exit $status
25 $COMPILE std_magic_body.a
26     if ($status != 0) exit $status
27 $COMPILE std_output_spec.a
28     if ($status != 0) exit $status
29 $COMPILE std_yyval_error_body.a
30     if ($status != 0) exit $status
31 $COMPILE std_predefined_tree.a
32     if ($status != 0) exit $status
33 $COMPILE rbd1_system_types_spec.a
34     if ($status != 0) exit $status
35 $COMPILE user_types_spec.a
36     if ($status != 0) exit $status
37 $COMPILE user_types_body.a
38     if ($status != 0) exit $status
39 $COMPILE rbd1_attributed_tree_spec.a
40     if ($status != 0) exit $status
41 $COMPILE rbd1_gets.a
42     if ($status != 0) exit $status
43 $COMPILE rbd1_attributed_tree_body.a
44     if ($status != 0) exit $status
45 $COMPILE support_spec.a
46     if ($status != 0) exit $status
47 $COMPILE rbd1_puts.a
48     if ($status != 0) exit $status
49 $COMPILE rbd1_makes.a
50     if ($status != 0) exit $status
51 $COMPILE globals_spec.a
52     if ($status != 0) exit $status
```

```
53      $COMPILE globals_body.a
54      if ($status != 0) exit $status
55      $COMPILE support_body.a
56      if ($status != 0) exit $status
57      $COMPILE report_spec.a
58      if ($status != 0) exit $status
59      $COMPILE report_body.a
60      if ($status != 0) exit $status
61      $COMPILE rbd1_evaluator_spec.a
62      if ($status != 0) exit $status
63      $COMPILE rbd1_dgts_spec.a
64      if ($status != 0) exit $status
65      $COMPILE nonstd_lex_spec.a
66      if ($status != 0) exit $status
67      $COMPILE std_user_pkg.a
68      if ($status != 0) exit $status
69      $COMPILE rbd1_lexdef.a
70      if ($status != 0) exit $status
71      $COMPILE lexacts_spec.a
72      if ($status != 0) exit $status
73      $COMPILE lexacts_body.a
74      if ($status != 0) exit $status
75      $COMPILE rbd1_display.a
76      if ($status != 0) exit $status
77      $COMPILE rbd1_lex_pkg.a
78      if ($status != 0) exit $status
79      $COMPILE rbd1_dgts_body.a
80      if ($status != 0) exit $status
81      $COMPILE rbd1_evaluator_body.a
82      if ($status != 0) exit $status
83      $COMPILE rbd1_parserdefs.a
84      if ($status != 0) exit $status
85      $COMPILE std_parser_spec.a
86      if ($status != 0) exit $status
87      $COMPILE rbd1_parser.a
88      if ($status != 0) exit $status
89      $COMPILE bes1_v1_string_handler_spec.a
90      if ($status != 0) exit $status
91      $COMPILE bes1_v1_string_handler_body.a
92      if ($status != 0) exit $status
93      $COMPILE bes1_support_spec.a
94      if ($status != 0) exit $status
95      $COMPILE bes1_support_body.a
96      if ($status != 0) exit $status
97      $COMPILE question_hash_types_spec.a
98      if ($status != 0) exit $status
99      $COMPILE question_hash_types_body.a
100     if ($status != 0) exit $status
101     $COMPILE question_hashes.a
102     if ($status != 0) exit $status
103     $COMPILE create_inferencer_support_spec.a
104     if ($status != 0) exit $status
105     $COMPILE create_inferencer_support_body.a
106     if ($status != 0) exit $status
```

```
107     $COMPILE be1.a
108     if ($status != 0) exit $status
109     $COMPILE backend_interface.a
110     if ($status != 0) exit $status
111     $COMPILE nonstd_boot.o
112     if ($status != 0) exit $status
113
114 LINK_IT:
115 echo ""
116 echo "Linking Rbd1"
117 echo ""
118
119 if ($PCTE == y || $PCTE == Y) then
120     $LINK boot -o Rbd1 $TARGET/Rlf/support.o \
121         $LIBEMER $PCTE_UTIL \
122         $LINK_FLAGS
123     if ($status != 0) exit $status
124 else
125     $LINK boot -o Rbd1 $TARGET/Rlf/support.o \
126         $LINK_FLAGS
127     if ($status != 0) exit $status
128 endif
129 popd > /dev/null
130 exit 0
```

C.2.7 Script: Build_Library_Manager.csh

```
1  #!/bin/csh -f
2  ###limit stacksize unlimited
3  ###limit datasize unlimited
4
5  pushd $TARGET/Library_Manager > /dev/null
6
7  if ( $LINK_ONLY == Y ) goto LINK_IT
8
9  echo ""
10 echo "Establishing soft links to the Library Manager source files"
11 echo ""
12 foreach file ( $RLFCODE/Library_Manager/*.a )
13     if ( ! -e ${file:t} ) ln -s $file ${file:t}
14 end
15
16 echo ""
17 echo "Compiling Library Manager"
18 echo ""
19
20     $COMPILE globals_spec.a
21     if ($status != 0) exit $status
22     $COMPILE callbacks_spec.a
23     if ($status != 0) exit $status
24     $COMPILE callbacks_body.a
25     if ($status != 0) exit $status
26     $COMPILE library_manager.a
27     if ($status != 0) exit $status
28
29 LINK_IT:
30 echo ""
31 echo "Linking Library Manager"
32 echo ""
33 if ( $PCTE == y || $PCTE == Y ) then
34     $LINK library_manager -o Library_Manager $TARGET/Rlf/support.o \
35                                     $LIBEMER $PCTE_ADA/util.o \
36                                     $LINK_FLAGS
37     if ($status != 0) exit $status
38 else
39     $LINK library_manager -o Library_Manager $TARGET/Rlf/support.o \
40
41     if ($status != 0) exit $status
42 endif
43 popd > /dev/null
44 exit 0
```

C.2.8 Script: Build_Rlf_Gb.csh

```
1  #!/bin/csh -f
2  ###limit stacksize unlimited
3  ###limit datasize unlimited
4
5
6
7  pushd $TARGET/Rlf_Gb > /dev/null
8
9  if ( $LINK_ONLY == Y ) goto LINK_IT
10
11  echo ""
12  echo "Establishing soft links to the Rlf_Gb source files"
13  echo ""
14
15  set echo
16  foreach file ($RLFCODE/Rlf_Gb/*.a)
17      if ( ! -e ${file:t} ) ln -s $file ${file:t}
18  end
19  unset echo
20
21  echo ""
22  echo "Compiling Rlf_Gb"
23  echo ""
24
25
26  $COMPILE gb_params_spec.a
27      if ($status != 0) exit $status
28  $COMPILE gb_params_body.a
29      if ($status != 0) exit $status
30  $COMPILE gb_instance.a
31      if ($status != 0) exit $status
32  $COMPILE gb_hash.a
33      if ($status != 0) exit $status
34  $COMPILE gb_globals.a
35      if ($status != 0) exit $status
36  $COMPILE gb_dynamic_menus_spec.a
37      if ($status != 0) exit $status
38  $COMPILE gb_agg_browser_spec.a
39      if ($status != 0) exit $status
40  $COMPILE gb_callbacks_spec.a
41      if ($status != 0) exit $status
42  $COMPILE gb_static_menus_spec.a
43      if ($status != 0) exit $status
44  $COMPILE gb_static_cmds_spec.a
45      if ($status != 0) exit $status
46  $COMPILE gb_dyn_node_menus_spec.a
47      if ($status != 0) exit $status
48  $COMPILE gb_utils_spec.a
49      if ($status != 0) exit $status
50  $COMPILE gb_history.a
51      if ($status != 0) exit $status
52  $COMPILE gb_make_view.a
```

```

53     if ($status != 0) exit $status
54     $COMPILE gb_infer_dtau_spec.a
55     if ($status != 0) exit $status
56     $COMPILE gb_infer.a
57     if ($status != 0) exit $status
58     $COMPILE gb_infer_dtau_body.a
59     if ($status != 0) exit $status
60
61     $COMPILE gb_cb_suppress_spec.a
62     if ($status != 0) exit $status
63     $COMPILE gb_cb_suppress_gen2.a
64     if ($status != 0) exit $status
65     $COMPILE gb_cb_suppress_gen.a
66     if ($status != 0) exit $status
67
68     $COMPILE gb_cb_suppress_body.a
69     if ($status != 0) exit $status
70
71     $COMPILE gb_agg_browser_body.a
72     if ($status != 0) exit $status
73     $COMPILE gb_callbacks_body.a
74     if ($status != 0) exit $status
75     $COMPILE gb_dynamic_menus_body.a
76     if ($status != 0) exit $status
77     $COMPILE gb_static_menus_body.a
78     if ($status != 0) exit $status
79     $COMPILE gb_static_cmds_body.a
80     if ($status != 0) exit $status
81     $COMPILE gb_dyn_node_menus_body.a
82     if ($status != 0) exit $status
83     $COMPILE gb_utils_body.a
84     if ($status != 0) exit $status
85     $COMPILE gb_main.a
86     if ($status != 0) exit $status
87
88 LINK_IT:
89 echo ""
90 echo "Linking Rlf_Gb"
91 echo ""
92 if ($PCTE == y || $PCTE == Y) then
93     $LINK Gb_Main -o Graphical_Browser $TARGET/Rlf/support.o \
94                                     $RGB_OBJ \
95                                     $LIBX \
96                                     $LIBEMER $PCTE_UTIL \
97                                     $LINK_FLAGS
98     if ($status != 0) exit $status
99 else
100    $LINK Gb_Main -o Graphical_Browser $TARGET/Rlf/support.o \
101                                    $RGB_OBJ \
102                                    $LIBX \
103                                    $LINK_FLAGS
104    if ($status != 0) exit $status
105 endif
106

```

March 1993

STARS-UC-05156/013/00

```
107 popd > /dev/null
108 exit 0
```

C.3 Scripts for Building Sample Networks

C.3.1 Script: Build_Ada_X_Lib.csh

```
1  #!/bin/csh -f
2  ###limit stacksize unlimited
3  ###limit datasize unlimited
4
5  #
6  # This script builds the "Paramax STARS Ada/X" library for the RLF.
7  #
8
9  #
10 # Locate the RLF Libraries
11 #
12 setRLF_LIBRARIES:
13 if (! $?RLF_LIBRARIES) then
14     echo ""
15     echo "Specify path to the RLF libraries"
16     echo "(e.g. /afs/reston/see/rlf/4.0/Libraries)"
17     echo ""
18     echo -n "   RLF_LIBRARIES = "
19     setenv RLF_LIBRARIES $<
20     echo ""
21 endif
22
23 echo ""
24 echo "Creating required sub-directories"
25 echo ""
26 if ( ! -d $RLF_LIBRARIES/Text/ada_x ) mkdir -p $RLF_LIBRARIES/Text/ada_x
27 if ( ! -d $RLF_LIBRARIES/Taustuff ) mkdir $RLF_LIBRARIES/Taustuff
28
29 echo ""
30 echo "Initializing text files"
31 echo ""
32 cp -r Text/* $RLF_LIBRARIES/Text/ada_x
33
34 echo ""
35 echo "Building library model from ada_x.lmdl"
36 echo ""
37 Lmdl ada_x.lmdl
38
39 foreach i (*rbd1)
40     echo ""
41     echo "Creating Inferencer from $i"
42     echo ""
43     Rbd1 < $i
44 end
```

C.3.2 Script: Build_Animals_Lib.csh

```
1  #!/bin/csh -f
2  ###limit stacksize unlimited
3  ###limit datasize unlimited
4
5  #
6  # This script builds a demonstration animals library for the RLF.
7  #
8
9  #
10 # Locate the RLF Libraries
11 #
12 setRLF_LIBRARIES:
13 if (! $?RLF_LIBRARIES) then
14     echo ""
15     echo "Specify path to the RLF libraries"
16     echo "(e.g. /afs/reston/see/rlf/4.0/Libraries)"
17     echo ""
18     echo -n " RLF_LIBRARIES = "
19     setenv RLF_LIBRARIES $<
20     echo ""
21 endif
22
23 echo ""
24 echo "Creating required sub-directories"
25 echo ""
26 if ( ! -d $RLF_LIBRARIES ) mkdir $RLF_LIBRARIES
27 if ( ! -d $RLF_LIBRARIES/Taustuff ) mkdir $RLF_LIBRARIES/Taustuff
28 if ( ! -d $RLF_LIBRARIES/Text ) mkdir $RLF_LIBRARIES/Text
29 if ( ! -d $RLF_LIBRARIES/Text/animals ) mkdir -p $RLF_LIBRARIES/Text/animals
30
31
32
33 echo ""
34 echo "Initializing text files"
35 echo ""
36 cp Text/* $RLF_LIBRARIES/Text/animals
37
38 echo ""
39 echo "Building Lmdl Network from animals.lmdl"
40 echo ""
41 Lmdl animals.lmdl
```

C.3.3 Script: Build_Asw_Lib.csh

```
1  #!/bin/csh -f
2  ###limit stacksize unlimited
3  ###limit datasize unlimited
4
5  #
6  # This script builds a sample Anti-Submarine Warfare library for the RLF.
7  #
8
9  #
10 # Locate the RLF Libraries
11 #
12 setRLF_LIBRARIES:
13 if (! $?RLF_LIBRARIES) then
14     echo ""
15     echo "Specify path to the RLF libraries"
16     echo "(e.g. /afs/reston/see/rlf/4.0/Libraries)"
17     echo ""
18     echo -n " RLF_LIBRARIES = "
19     setenv RLF_LIBRARIES $<
20     echo ""
21 endif
22
23 echo ""
24 echo "Creating required sub-directories"
25 echo ""
26 if ( ! -d $RLF_LIBRARIES/Text/asw ) mkdir -p $RLF_LIBRARIES/Text/asw
27 if ( ! -d $RLF_LIBRARIES/Taustuff ) mkdir $RLF_LIBRARIES/Taustuff
28
29 echo ""
30 echo "Initializing text files"
31 echo ""
32 cp Text/* $RLF_LIBRARIES/Text/asw
33
34 echo ""
35 echo "Building library model from asw.lmdl"
36 echo ""
37 Lmdl asw.lmdl
```

C.3.4 Script: Build_Common_Data_Model_Lib.csh

```
1  #!/bin/csh -f
2  ###limit stacksize unlimited
3  ###limit datasize unlimited
4
5  #
6  # This script builds a demo Common Data Model library for the RLF.
7  #
8
9  #
10 # Locate the RLF Libraries
11 #
12 setRLF_LIBRARIES:
13 if ( ! $?RLF_LIBRARIES ) then
14     echo ""
15     echo "Specify path to the RLF libraries"
16     echo "(e.g. /afs/reston/see/rlf/4.0/Libraries)"
17     echo ""
18     echo -n " RLF_LIBRARIES = "
19     setenv RLF_LIBRARIES $<
20     echo ""
21 endif
22
23 echo ""
24 echo "Creating required sub-directories"
25 echo ""
26 if ( ! -d $RLF_LIBRARIES/Text/satText ) mkdir -p $RLF_LIBRARIES/Text/satText
27 if ( ! -d $RLF_LIBRARIES/Taustuff ) mkdir $RLF_LIBRARIES/Taustuff
28
29 echo ""
30 echo "Initializing text files"
31 echo ""
32 cp Text/* $RLF_LIBRARIES/Text/satText
33
34 echo ""
35 echo "Building LMDL Network from cdm.lmdl"
36 echo ""
37 Lmdl < common_data_model.lmdl
```

C.3.5 Script: Build_Demo_Actions_Lib.csh

```
1  #!/bin/csh -f
2  ###limit stacksize unlimited
3  ###limit datasize unlimited
4
5  #
6  # This script builds a demonstration actions library for the RLF.
7  #
8
9  #
10 # Locate the RLF Libraries
11 #
12 setRLF_LIBRARIES:
13 if (! $?RLF_LIBRARIES) then
14     echo ""
15     echo "Specify path to the RLF libraries"
16     echo "(e.g. /afs/reston/see/rlf/4.0/Libraries)"
17     echo ""
18     echo -n " RLF_LIBRARIES = "
19     setenv RLF_LIBRARIES $<
20     echo ""
21 endif
22
23 echo ""
24 echo "Creating required sub-directories"
25 echo ""
26 if ( ! -d $RLF_LIBRARIES/Text/demo_actions ) then
27     mkdir -p $RLF_LIBRARIES/Text/demo_actions
28 endif
29 if ( ! -d $RLF_LIBRARIES/Taustuff ) mkdir $RLF_LIBRARIES/Taustuff
30
31 echo ""
32 echo "Initializing text files"
33 echo ""
34 cp -r Text/* $RLF_LIBRARIES/Text/demo_actions
35
36 echo ""
37 echo "Building LMDL Network from demo_actions.lmdl"
38 echo ""
39 Lmdl demo_actions.lmdl
```

C.3.6 Script: Build_Move_Domain_Lib.csh

```
1  #!/bin/csh -f
2  ###limit stacksize unlimited
3  ###limit datasize unlimited
4
5  #
6  # This script builds the Cathy Lin's Window Manager library for the RLF.
7  #
8
9  #
10 # Locate the RLF Libraries
11 #
12 set RLF_LIBRARIES:
13 if (! $?RLF_LIBRARIES) then
14     echo ""
15     echo "Specify path to the RLF libraries"
16     echo "(Defaults to $RLF/Libraries)"
17     echo ""
18     echo -n "  RLF_LIBRARIES = "
19     setenv RLF_LIBRARIES $<
20     echo ""
21     if ($RLF_LIBRARIES ==) setenv RLF_LIBRARIES $RLF/Libraries
22 endif
23
24 echo ""
25 echo "Creating required sub-directories"
26 echo ""
27 if ( ! -d $RLF_LIBRARIES/Text/wm_move ) mkdir -p $RLF_LIBRARIES/Text/wm_move
28 if ( ! -d $RLF_LIBRARIES/Taustuff ) mkdir $RLF_LIBRARIES/Taustuff
29
30 echo ""
31 echo "Initializing text files"
32 echo ""
33 cp -r Text/* $RLF_LIBRARIES/Text/wm_move
34
35 echo ""
36 echo "Building LMDL Network from move_domain.lmdl"
37 echo ""
38 Lmdl move_domain.lmdl
39
40 foreach i (*rbd1)
41     echo ""
42     echo "Creating Inferencer from $i"
43     echo ""
44     Rbd1 < $i
45 end
```

C.3.7 Script: Build_SW_Tech_Lib.csh

```
1  #!/bin/csh -f
2  ###limit stacksize unlimited
3  ###limit datasize unlimited
4
5  #
6  # This script builds the Software Technology library for the RLF.
7  #
8
9  #
10 # Locate the RLF Libraries
11 #
12 set RLF_LIBRARIES:
13 if (! $?RLF_LIBRARIES) then
14     echo ""
15     echo "Specify path to the RLF libraries"
16     echo "(Defaults to $RLF/Libraries)"
17     echo ""
18     echo -n " RLF_LIBRARIES = "
19     setenv RLF_LIBRARIES $<
20     echo ""
21     if ($RLF_LIBRARIES ==) setenv RLF_LIBRARIES $RLF/Libraries
22 endif
23
24 echo ""
25 echo "Creating required sub-directories"
26 echo ""
27 if ( ! -d $RLF_LIBRARIES/Text/sw_tech ) mkdir -p $RLF_LIBRARIES/Text/sw_tech
28 if ( ! -d $RLF_LIBRARIES/Taustuff ) mkdir $RLF_LIBRARIES/Taustuff
29
30 echo ""
31 echo "Initializing text files"
32 echo ""
33 cp -r Text/* $RLF_LIBRARIES/Text/sw_tech
34
35 echo ""
36 echo "Building LMDL Network from sw_tech.lmdl"
37 echo ""
38 Lmdl sw_tech.lmdl
```

C.3.8 Script: Build_Sort_And_Search_Lib.csh

```
1  #!/bin/csh -f
2  ###limit stacksize unlimited
3  ###limit datasize unlimited
4
5  #
6  # This script builds the "Sort and Search Algorithms" library for the RLF.
7  #
8
9  #
10 # Locate the RLF Libraries
11 #
12 set RLF_LIBRARIES:
13 if (! $?RLF_LIBRARIES) then
14     echo ""
15     echo "Specify path to the RLF libraries"
16     echo "(e.g. /afs/reston/see/rlf/4.0/Libraries)"
17     echo ""
18     echo -n "  RLF_LIBRARIES = "
19     setenv RLF_LIBRARIES $<
20     echo ""
21 endif
22
23 echo ""
24 echo "Creating required sub-directories"
25 echo ""
26 if ( ! -d $RLF_LIBRARIES/Text/sort_and_search ) \
27     mkdir -p $RLF_LIBRARIES/Text/sort_and_search
28 if ( ! -d $RLF_LIBRARIES/Taustuff ) mkdir $RLF_LIBRARIES/Taustuff
29
30 echo ""
31 echo "Initializing text files"
32 echo ""
33 cp -r Text/* $RLF_LIBRARIES/Text/sort_and_search
34
35 echo ""
36 echo "Building library model from sort_and_search.lmdl"
37 echo ""
38 Lmdl sort_and_search.lmdl
39
40 foreach i (*rbd1)
41     echo ""
42     echo "Creating Inferencer from $i"
43     echo ""
44     Rbd1 < $i
45 end
```