

SPARC COMPLIANCE DEFINITION 2.4

**SCD
2.4**

32/64 bits

SPARC International

August 1998

© 1990-1998 SPARC International Inc.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of the copyright owners.

The manual pages for socket functions are

© 1992, 1993 The Regents of the University of California. All rights reserved

Includes material copyrighted by UNIX System Laboratories, Inc., a subsidiary of SCO, Inc. Reprinted with permission.

The SPARC Compliance Definition 2.4 is published and printed by SPARC International.

Any comments relating to the material contained herein may be submitted to:

SPARC International Inc.

3333 Bowers Ave., Suite 280

Santa Clara, CA 95054-2913

TEL: (408) 748-9111 (Ext 228)

FAX: (408) 743-9777

URL: www.sparc.org

ATTN: Ghassan Abbas (abbas@sparc.org)

Trademarks

SPARC® is a registered trademark of SPARC International, Inc.

SPARCstation™ is a trademark of SPARC International, Inc.

Products bearing SPARC® trademarks are based on an architecture developed by Sun Microsystems, Inc.

ONC™ and SunOS™ are trademarks of Sun Microsystems, Inc.

NFS® is a registered trademark of Sun Microsystems, Inc.

UNIX® and OPEN LOOK® are registered trademarks of UNIX System Laboratories, Inc.

The X-Window System™ is a trademark of Massachusetts Institute of Technology.

OSF/Motif™ is a trademark of the TOG (X/Open + Open Software Foundation, Inc).

All other products or services mentioned in this document are identified by the trademarks or service marks of their respective companies or organizations. SPARC International, Inc. disclaims any responsibility for specifying which trademarks are owned by which companies or organizations.

This product contains intellectual property of Sun Microsystems, Inc., and any user of this product will be required to obtain a license from Sun Microsystems, Inc., prior to use.

SPARC COMPLIANCE DEFINITION 2.4

**SCD
2.4**

TABLE OF CONTENTS

Preface

Audience and Purpose	i
Organization and Content	i
Publication Conventions	i
Other Publication Conventions	i

CHAPTER 1: Introduction

The SPARC Compliance Definition	1-1
Introduction Changes	1-2
Definitions of Terms	1-2
32-bit ABI or SPARC 32-bit ABI	1-2
64-bit ABI or SPARC 64-bit ABI	1-2
gABI	1-2
psABI	1-3
Deprecated	1-3
Experimental	1-3
Interface	1-3
Interface Member	1-4
Interface Set	1-4
Optional	1-4
Rationale	1-4
Required	1-4
SPARC V8 Architecture	1-4
SPARC V9 Architecture	1-4
Normative References	1-4
Relationship to other Standards	1-7
Future Direction	1-7
Structure of the SCD	1-7
System Feature Interfaces	1-8
Library Interfaces	1-8
Command Interface	1-8
Definition of SPARC Compliance	1-8
Conforming Implementations	1-8
Conforming Application Programs	1-8
Compliance Testing	1-9
Changes between SCD 2.3 and SCD 2.4	1-9
Upward Compatibility	1-10

CHAPTER 2: Software Installation

Introduction	2-1
Software Installation Changes	2-1
CD-ROM Medium	2-1

CHAPTER 3: Low-Level System Information

Introduction	3-1
Self-Modifying Coding Practices:	3-1
Low-level System Information Changes (32-bit psABI)	3-1
Low-Level System Information (64-bit psABI) -EXPERIMENTAL	3P-1
Machine Interface	3P-1

CHAPTER 4: Object Files

Introduction	4-1
Object Files Changes (32-bit ABI)	4-1
Object Files Changes (64-bit gABI) - EXPERIMENTAL	4G-1
Object Files (64-bit psABI) - EXPERIMENTAL	4P-1
ELF Header	4P-1
Sections	4P-2
Relocation	4P-4

CHAPTER 5: Program Loading and Dynamic Linking

Introduction	5-1
Program Loading and Dynamic Linking Changes (32-bit ABI)	5-1
Program Load. & Dyn. Linking (64-bit gABI) - EXPERIMENTAL	5G-1
Program Loading & Dynamic Linking Changes	5G-1
Program Load. & Dyn. Linking (64-bit psABI) - EXPERIMENTAL	5P-1
Program Loading	5P-1

CHAPTER 6: Libraries

Introduction	6-1
C library Changes	6-3
Network Services Library Changes	6-7
System Data Interface Changes	6-8
Miscellaneous ABI Changes	6-9
Overview of Large Files Support (32bit-ABI)	6-12
Conventions and Techniques for Library Versioning	6-17
Overview of Support for Multi-threaded Applications	6-20
libaio - Asynchronous I/O Library - DEPRECATED	6-24
Overview	6-24
SCD Extensions to the System V ABI	6-24
Library Contents	6-24
Structures and Manifest Constants	6-25
libc - The C Library	6-26
Overview	6-26
The libc ABI Interfaces	6-26
SCD Extensions to the System V ABI	6-26
Long Long Intrinsic Support	6-26
Exported Data	6-26
Library Contents	6-28
Structures and Manifest Constants	6-32
libdl - Dynamic Object File Loading Library	6-45
Overview	6-45
libelf - Executable Linking Format Library	6-46
Overview	6-46
libintl - Internationalization Library - EXPERIMENTAL	6-50

Overview	6-50
SCD Extensions to the System V ABI	6-50
Library Contents	6-50
libm - Math Library	6-51
Overview	6-51
SCD Extensions to the System V ABI	6-51
Library Contents	6-51
Structures and Manifest Constants	6-51
libnisdb - Network Infor. Services Database library - EXPERIMENTAL	6-52
Overview	6-52
SCD Extensions to the System V ABI	6-52
Library Contents	6-52
Structures and Manifest Constants	6-53
libnsl - The Network Services Library	6-54
Overview	6-54
The libnsl ABI Interfaces	6-54
SCD Extensions to the System V ABI	6-54
Exported Data	6-54
Structures and Manifest Constants	6-57
libposix4 - POSIX4 Library - EXPERIMENTAL	6-59
Overview	6-59
SCD Extensions to the System ABI	6-59
Library Contents	6-59
libpthread - POSIX Multithreading Library - EXPERIMENTAL	6-60
Overview	6-60
SCD Extensions to the System V ABI	6-60
Library Contents	6-61
Structures and Manifest Constants	6-62
libresolv - Domain Name Services Library - EXPERIMENTAL	6-64
Overview	6-64
SCD Extensions to the System V ABI	6-64
Library Contents	6-64
Structures and Manifest Constants	6-64
librpcsvc - Remote Procedure Call Services Library - EXPERIMENTAL	6-66
Overview	6-66
SCD Extensions to the System V ABI	6-66
Library Contents	6-66
libsocket - Socket Library	6-67
Overview	6-67
ABI Extensions to the System V ABI	6-67
Library Contents	6-67
Structures and Manifest Constants	6-67
libthread - Multithreading Library	6-77
Overview	6-77
libucb - University of California at Berkeley Compatibility Library	6-80
Overview	6-80
SCD Extension to the System V ABI	6-80
Library Contents	6-80
libw - Wide Character Support Library	6-81
Overview	6-81
SCD Extensions to the System V ABI	6-81
Library Contents	6-81
Structures and Manifest Constants	6-81
Libraries (64-bit gABI) - EXPERIMENTAL	6G-1

Overview	6G-1
Miscellaneous Libraries Changes (64-bit gABI):	6G-1
Libraries (64-bit psABI) - EXPERIMENTAL	6P-1
Overview	6P-1
Library Contents	6P-2
System Data Interfaces	6P-2

CHAPTER 7: Formats and Protocols

Introduction	7-1
Formats and Protocols Changes	7-1
Interconnecting SCD Conforming Systems	7-1
Overview	7-1
Transport Providers	7-1
Additional Interfaces	7-1
Format and Protocols (64-bit gABI) - EXPERIMENTAL	7G-1
Format and Protocols changes:	7G-1

CHAPTER 8: System Commands

Introduction	8-1
System Commands Changes	8-2
System Commands Changes (continued)	8-3

CHAPTER 9: Execution Environment

Introduction	9-1
Execution Environment Changes	9-1

CHAPTER 10: Windowing and Terminal Interfaces

Introduction	10-1
The X Library	10-1
Overview	10-1
Unsafe Macros	10-7
X Library Changes	10-10
X Library Changes (continued)	10-11
X Library Changes (continued)	10-12
The X Extension Library	10-26
Overview	10-26
The Extension Library Interfaces	10-26
The X Toolkit	10-27
Overview	10-27
The libXt Interfaces	10-27
Deprecated X Toolkit Functions	10-30
Subclassing Xt Widgets	10-40
The OPEN LOOK Widget Set	10-48
Overview	10-48
The libXol Interfaces	10-48
Motif 1.2 Widget Set	10-55
Overview	10-55
The Motif Interfaces	10-55
Overview	10P-1

CHAPTER 11: Development Environments

Overview 11-1
Development Environments Changes 11-1

CHAPTER 12: Networking

Overview 12-1
Networking Changes 12-1

Index



SCD

2.4

Preface

Audience and Purpose

The SPARC International *SPARC Compliance Definition* (SCD) is intended for use by anyone who is creating binary compatible SPARC systems or applications.

The intended audience of the SCD documents consists of two groups: system and application developers. For system developers, the SCD provides a reference to those interfaces and features which must be supplied by a SPARC compliant system. For application developers, the SCD provides a reference to interfaces and features that may be relied upon in all SPARC compliant systems.

This publication is intended to fulfill the following purposes:

- *Identify areas beyond the System V Application Binary Interface (gABI) and the System V Application Binary Interface, SPARC Processor Supplement (psABI) that the SPARC community deems important.*
- *Address ambiguous and/or loose specifications in current ABI documents.*

Organization and Content

The SCD 2.4 has been divided into chapters, as follows:

- Chapter 1 Introduction
- Chapter 2 Software Installation
- Chapter 3 Low-Level System Information
- Chapter 4 Object Files
- Chapter 5 Program Loading and Dynamic Linking
- Chapter 6 Libraries
- Chapter 7 Formats and Protocols
- Chapter 8 System Commands
- Chapter 9 Execution Environment
- Chapter 10 Windowing and Terminal Interfaces
- Chapter 11 Development Environments
- Chapter 12 Networking
- Index

This new organization follows the organization of the *System V Application Binary Interface* and *System V Application Binary Interface, SPARC Processor Supplement* documents. Having a parallel organization makes this document easier to use than previous editions of the SCD.

Publication Conventions

This publication uses page format and typographic variances to highlight particular kinds of information. These conventions of usage are generally consistent with publication conventions used by other UNIX publications, such as the *AT&T System V Interface Definition*, Third Edition.

Other Publication Conventions

The following typographical conventions are used within the text of this publication:

- *Filenames, pathnames, and system messages are shown in:*
typewriter font like this.

- Titles of chapters in this publication are shown in plain Roman font, inside quotation marks like this:
"Introduction."
- Document titles are shown in plain, nonbold italic font like this:
System V Interface Definition (Third Edition).

CHAPTER 1: Introduction

SCD
2.4

Introduction

The SPARC Compliance Definition

This document is version 2.4 of the SPARC Compliance Definition.

The SPARC Compliance Definition, or SCD, defines a minimum set of interfaces that all SPARC Compliant systems must provide in their implementations and a maximum set of interfaces that SPARC Compliant applications must use when linking to external system libraries. The SCD provides information for binary-level compatibility, encompassing both the *System V Application Binary Interface* (gABI), and the *System V Application Binary Interface, SPARC Processor Supplement* (psABI) documents.

Primary Intent and Intended Audience

The primary intent of this document is to *define* the SPARC ABI (application binary interface)—that is, to provide a specification for it (which we refer to as the “SPARC Compliance Definition” or “SCD”). The SPARC ABI describes a standard set of runtime interfaces for compiled application programs which will be run on one or more systems (an operating system and hardware platform) based upon the SPARC processor. Complementary to a compiled application program is a system software product. While an application depends on (or *consumes*), this runtime interface, the system software product provides (or *offers*), the interface and all the various functional constituents associated with its provision.

To a first approximation then, the SPARC ABI may be thought of as a contract which describes the relationship between an application program and a system software product *at runtime*. The purpose of the definition is to achieve two specific objectives: First, we wish to ensure that an application binary (i.e. compiled application program) which is “conformant” to a specified version of the SPARC ABI (i.e. depends upon less or equal to the interface and content contained within that specification) may be run on *any* system software product (and hardware platform) which is “conformant” with (offers) that version of the SPARC ABI (or a later one). This ability to run a compiled application program on any of a number of different system software products and/or hardware platforms may be referred to as [binary] “portability”. Second, we wish to ensure that an application binary which is conformant with a specified version of the SPARC ABI will continue to run without change on all system software products (and platforms) offering a later version of the SPARC ABI. This ability to sustain all existing application binaries irrespective of an update in system software or hardware platform may be referred to as [binary] “stability”. The former property is ensured by defining a standard set of features and interface that all conformant system implementations must offer. The latter property is enabled by an upward compatible evolution of the interface (and functional constituent) contained within succeeding revisions of the SPARC ABI (and SCD).

Intended Audience

Since the SPARC ABI is an interface that separates application programs from system software products, this specification is of interest both to the developer of a system software products which offers the interface (and associated application services) and to the developer of an application which is intended to run on such systems. To that extent, this document strives to describe everything that must be contained within a system implementation for it to be considered conformant (and hence be able to run any SCD-conformant application), and the maximum set of dependencies that an application may have if it is to be considered conformant (and hence be able to run on any SCD-conformant system, including all systems conformant to a later revision of the SPARC ABI).

The primary specific purpose to which this document is put is to enable the development of *tests* of conformance for system software products and for application programs. These tests may then be applied by developers of such products to help them determine whether they are conformant.

Depending on the reader’s perspective, this document may be viewed as having some shortcomings. While it tells the application development reader *what* may be relied upon, and what a well-formed application binary looks like, it does not describe *how to construct* such an application, nor all that is needed to do that. And similarly, while it tells a system software product developer all that must be provided, once again it does not describe all that may be needed in order to achieve that.

Finally, it must be admitted that this document has been addressed primarily to developers of system implementations which offer the SPARC ABI, and somewhat secondarily to developers of SPARC-compliant applications. In practice it has been used more as a means of synchronizing and standardizing the set of services and interfaces offered by several different vendors offering SPARC-based system software product implementations than anything else. The objective has been to attempt to identify the required set of application runtime services and then to standardize the way in which

these services are offered by the different system software products in order to unify the SPARC application space—to enable the development of a *single* application binary that can run on any SCD-compliant SPARC system software product, thus increasing possible deployment span for application developers and reducing their testing costs.

SCD 2.4—One Specification, Two ABI Standards

The SCD2.4, as opposed to previous versions of the SCD, describes two distinct and incompatible standards: a 32-bit ABI and an EXPERIMENTAL 64-bit ABI. SCD conforming applications must conform to either the 32-bit ABI or the 64-bit ABI (when a 64-bit ABI becomes REQUIRED). A 32-bit application can only link to 32-bit libraries and 64-bit applications can only link to 64-bit libraries.

Rationale

At this time, the 64-bit ABI is an EXPERIMENTAL interface, due to the fact that two systems vendors have yet to implement these interfaces fully for use by a number of applications. SPARC International systems vendors have agreed at this time to the EXPERIMENTAL 64-bit ABI, and are requested to implement to this EXPERIMENTAL standard as closely as possible.

Relationship to Product Branding

It is the purview of SPARC International, based on input from its members, to define a branding program, if any, based on mapping(s) into the SCD. The SCD is organized such that brands could be created for:

- o 32-bit applications
- o 64-bit applications
- o systems supporting only 32-bit applications
- o systems supporting only 64-bit applications
- o systems supporting both 32-bit and 64-bit applications

Introduction Changes

The following are changes to the *System V Application Binary Interface* as reported to SPARC International.

#	Facility	Location	Description
1	How to use the System V ABI	gABI	The math routines are also available as a shared resources

Definitions of Terms

32-bit ABI or SPARC 32-bit ABI

In the context of this document, 32-bit ABI refers to a 32-bit version of the SCD for 32-bit SPARC Compliant applications and SPARC Compliant systems that support them.

64-bit ABI or SPARC 64-bit ABI

In the context of this document, 64-bit ABI refers to a 64-bit version of the SCD for 64-bit SPARC Compliant applications and SPARC Compliant systems that support them.

gABI

Depending on its context, the term gABI in this document can mean either:

-
- 1) the “System V Application Binary Interface” document;
 - 2) those sections in the SPARC 32-bit ABI considered to be generic in nature or to be common with other UNIX systems;
or
 - 3) those sections in the SPARC 64-bit ABI considered to be generic in nature or to be common with other UNIX systems.

psABI

Depending on its context, the term psABI in this document can mean either:

- 1) the “System V Application Binary Interface SPARC Processor Supplement” document;
- 2) those sections in the SPARC 32-bit ABI considered to be processor-specific in nature and not in common with other UNIX systems; or
- 3) those sections in the SPARC 64-bit ABI considered to be processor-specific in nature and not in common with other UNIX systems.

Deprecated

The term “deprecated” in this document is a qualifier for the terms “interface set,” “interface member,” and “interface.” When the term “DEPRECATED interface set” is used in this document, programmers are discouraged from using the designated interface set in new applications because the “DEPRECATED interface set” may not be supported in future versions of the SCD. The qualifier “deprecated” is orthogonal to the qualifiers “required” and “optional”. When an “interface set” is designated as “deprecated” the date of deprecation will be stated by the specification. “Interface sets,” marked as “deprecated,” will be kept in the SCD for at least three (3) years from the original deprecation date. The “DEPRECATED interface set” will also include in its specification, the earliest date at which the designated “interface set” may be removed from the specification. No required or optional interface will be removed from the standard without first being deprecated. The terms “DEPRECATED interface member” and “DEPRECATED interface” are defined similarly.

Experimental

The term “experimental” in this document is a qualifier for the terms “interface set,” “interface member,” and “interface.” When the term “EXPERIMENTAL interface set” is used in the document, applications programmers are warned that 1) the designated interface set may not be available on any SPARC conforming systems, and, 2) the specification of the designated interface may change at any time or be deleted from the SCD at the sole discretion of SPARC International; SPARC International makes no commitment of a three-year stable period for any “EXPERIMENTAL interface set.”

Rationale

As an example, this release of the SPARC compliance definition includes the EXPERIMENTAL Large File Support Library. This interface set is completely new. Because the Large File Support Library is new, we have no experience with the correctness of the interface. Field experience may require that certain portions of the interface change to make the interface more useful or practical.

Interface

The unadorned term “interface” means either “interface set” or “interface member” depending on the immediate context of its use. Any REQUIRED or OPTIONAL interface defined in this document will be part of the SCD--including future SCD documents--for at least three year.

Rationale

The SCD is intended as a long term commitment to a set of common SPARC interfaces. As such, interfaces should be implemented consistently on at least two independent SPARC systems and be used by a number of applications before being designated as REQUIRED or OPTIONAL.

Interface Member

The term “interface member” is also used as a generic reference to any single facility that is provided by a platform for use by an application program.

Examples are: the printf function; the errno global data item.

Interface Set

The term “interface set” refers to a named collection of facilities, defined in the SPARC Compliance Definition, that is provided by a platform and can be used by an application. These collections, or “interface sets”, are listed in the section below titled “Structure of the SPARC Compliance Definition”.

An example is: the X11 Library Interface Set.

Optional

The term “optional” in this document is a qualifier for the terms “interface set,” “interface member,” and “interface.” When the term “OPTIONAL interface set” is used in this document, SPARC conforming systems may, but need not, supply the interface set; if a conforming system does supply the interface, the interface set must be present in its entirety, as defined by this document; applications can not rely on the designated interface set being available on any conforming systems, but if the interface set is available on a particular conforming system, a conforming application can rely on the interface set being available in its entirety on that particular conforming system. The terms “OPTIONAL interface member” and “OPTIONAL interface” are similarly defined.

Rationale

Paragraphs labeled “rationale” in this document are non-normative and are for information only. An example of a Rationale paragraph follows below.

Rationale

The SPARC International Compliance and Compatibility Committee agreed that it would be more useful to intersperse rationale comments throughout the document than to confine them to an appendix.

Required

The term “required” in this document is a qualifier for the terms “interface set,” “interface member,” and “interface.” When the term “REQUIRED interface set” is used in this document, SPARC conforming systems must provide the interface set; conforming applications can rely on the designated interface set always being available on any conforming system. The terms “REQUIRED interface member” and “REQUIRED interface” are defined similarly.

SPARC V8 Architecture

All reference to SPARC V8 refers to the Architecture as defined in “*The SPARC Architecture Manual*”, Version 8.

SPARC V9 Architecture

All reference to SPARC V9 refers to the Architecture as defined in “*The SPARC Architecture Manual*”, Version 9.

Normative References

The SCD is based upon all or parts of certain other existing standards.

The definition of each interface in the SPARC Compliance Definition may reference one or more of the following documents. In those cases, the portion of the normative reference that is called out is part of this standard.

The normative references called out in the SPARC Compliance Definition are:

- *SCD 2.4 Interface Semantics*

SPARC International

- *System V Application Binary Interface, Third Edition*

-
- Unix Press (Prentice Hall), ISBN 0-13-100439-5
 - ***System V Application Binary Interface SPARC Processor Supplement, Third Edition***
 Unix Press (Prentice Hall), ISBN 0-13-104696-9
 - ***The SPARC Architecture Manual, Version 8***
 Prentice Hall, ISBN 0-13-825001-4
 - ***The SPARC Architecture Manual, Version 9***
 Prentice Hall, ISBN 0-13-099227-5
 - ***System V Interface Definition, Third Edition, Volumes 1 - 5***
 USL/Novell Select Code 320-136 (Volume 1), 320-137 (Volume 2), 320-138 (Volume 3),
 320-139 (Volume 4),

Volume 1	Addison-Wesley	ISBN 0-201-56652-0
Volume 2	Addison-Wesley	ISBN 0-201-56653-0
Volume 3	Addison-Wesley	ISBN 0-201-56654-0
Volume 4	Addison-Wesley	ISBN 0-201-56655-0
Volume 5	Addison-Wesley	ISBN 0-201-56656-7
 - ***The X Window System (Third Edition)***
 by Robert W. Scheifler and James Gettys
 Digital Press, ISBN 1-55558-088-2
 - ***X Toolkit Intrinsics - C Language Interface***
 by Joel McCormack, Paul Asente, and Ralphe R.Swick
 Distributed by the X consortium with the X Version 11, Release 5
 available through FTP from export.lcs.mit.edu
 - ***X11 Non-rectangular Window Shape Extension***
 by Keith Packard
 Copyright X Consortium
 - ***Large Files Summit document (Adding support for arbitrary file sizes to the Single UNIX Specification)***
 URL: <http://www.sparc.org/standards.html>
 - ***OSF/Motif Programmer's Guide (Rel. 1.2- Revised)***
 Prentice-Hall, ISBN 0-13-643115-1
 - ***OLIT Reference Manual***
 Sun Microsystems, Part No. 800-6055-10, Revision A
 - ***ISO 9660-1988: Volume and file structure of CD-ROM for information interchange***
 1988-09-01
 - ***ISO/IEC 10149: Data Interchange for read-only 120mm optical data disk (CD-ROM)***
 1989-09-01
 - ***Single UNIX Specification Version 2, Commands and Utilities, Issue 5 (C604)***
 Open Group CAE Specification, ISBN 1-85912-191-8
 - ***Single UNIX Specification Version 2, System Interface Definitions, Issue 5 (C605)***
 Open Group CAE Specification, ISBN 1-85912-186-0
-

- **Single UNIX Specification Version 2, System Interfaces and Headers, Issue 5 (Two volumes - C606)**
Open Group CAE Specification, ISBN 1-85912-181-0
- **RFC 1700 (Definition of well-known TCP and UDP port numbers)**
URL: <http://info.internet.isi.edu/1s/in-notes/rfc/files>
- **POSIX 1003.1-1990 (ISO 9945-1) 1990 (E) (ISO/IEC) (IEEE/ANSI Std 1003.1-1990):**
Information Technology - portable operating system interface (POSIX)
Part 1: System Application Program Interface (API)[C Language]
ISBN: 1-55937-061-7
- **POSIX 1003.1b (formally 1003.4) POSIX Asynchronous I/O, and Real-time interfaces)**
The ratified POSIX standards that generally pertain to realtime OS's consist of: 1003.1 (OS, process, filesystem and device API), 1003.2 (utilities), 1003.1b (realtime), and 1003.1c (threads). POSIX 1003.1d (which defines some additional realtime extensions like standardized interrupt handler support) is not yet ratified, although some OS's already support portions of it.
The POSIX 1003.1 standard is ISBN 1-55937-061-0. A good O'Reilly text is "POSIX Programmer's Guide: Writing Portable UNIX Programs". Donald Lewine. ISBN: 0-937175-73-0, <http://cs-www.bu.edu/pub/ieeerts/Home.html>

The definition of each Interface in the SPARC Compliance Definition may list errata to any of the above documents. In each such listed erratum, the definition contained in the erratum supersedes the corresponding portion of the normative reference.

These documents may be acquired from most technical book stores; additionally, SPARC International provides assistance in acquiring these references. If you require assistance in acquiring these references, call SPARC International at:

(408) 748-9111

ISO documents can be ordered from:

*International Organization for Standardization,
1 Rue de Varembe, Case Postale 56, CH-1211 Geneva 20 Switzerland,
(Tel) +41 22 34 12 40
URL: <http://www.iso.ch>*

or

*ANSI (ISO member for the US):
ANSI, 11 W 42nd St. 13th floor, New York, NY 10036,
(Tel) 1-212-642-4900.*

Prentice-Hall documents can be obtained at:

*PTR Prentice Hall, Corporate Sales Department
113 Sylvan Avenue, Englewood Cliffs, New Jersey 07632
(Tel) (201) 592-2863 (bulk copies), (Tel)(515) 284-6761 (single copies).
(Fax) (201) 592-2249*

Open Group documents can be obtained at:

The Open Group, Publications Department
PO Box 96, Witney, Oxon OX8 6PG, England,
(Tel) +44 (0) 1993 708731, (Fax) +44 (0) 1993 708732.
(URL) <http://www.opengroup.org>

Rationale

The SCD represents a proper super-set of the required interfaces and features described in the two ABI documents. One of the purposes of this document is to serve as the conduit through which features may migrate first into the processor specific ABI (SPARC psABI), and finally into the generic ABI (gABI). Consequently, the SCD includes a set of features and their associated interfaces that are beyond the ABI definitions. These features, and their associated interfaces have been included, in some cases to correct deficiencies in the ABI specifications, and in others to standardize functionality already in common use throughout the SPARC community.

Relationship to other Standards

Some of the normative references listed above are themselves based upon other standards. These include at least the following:

- *X/Open Portability Guide, Issue 3 (XPG3)*

X/Open Portability Guide 1988 X/Open Company Limited

Vol1:	XSI Commands and Utilities	ISBN:0-13-685835-X	✓
Vol2:	XSI System Interface and Headers	ISBN:0-13-685843-0	✓
Vol3:	XSI Supplementary Definitions	ISBN:0-13-685850-3	
Vol4:	Programming Languages	ISBN:0-13-685868-6	
Vol5:	Data Management	ISBN:0-13-685876-7	
Vol6:	Window Management	ISBN:0-13-685884-8	
Vol7:	Networking Services	ISBN:0-13-685892-9	
	set of 7 volumes	ISBN:0-13-685819-8	

✓ Referenced indirectly in the SVID.

Future Direction

The future direction of the SCD is to POSIX. Non-POSIX parts of this document are only included for existing or experimental implementation. These parts are designated as DEPRECATED or EXPERIMENTAL.

Structure of the SCD

The Application Binary Interface defined by the SCD consists of a set of System Feature Interfaces, a set of Library Interfaces, and a Command Interface.

Each such named Interface is designated as either Required, Optional, or Experimental.

As necessary, SCD chapters have been split into 32-bit gABI, 32-bit psABI, 64-bit gABI and 64-bit psABI sections. Sections that are errata to the System V gABI or psABI are also marked as such.

System Feature Interfaces

The System Feature Interfaces are:

- Object File Format
- Program Loading and Linking
- Low-level System Information
- Formats and Protocols
- Software Installation

Library Interfaces

Each Library Interface is a collection of facilities that is implemented as one or more shared objects. (Shared objects are defined in the Object File Format section of this specification). The library interfaces are as defined in chapter 6 “Libraries” and chapter 10 “Windowing and Terminal interfaces”.

Each Library Interface consists of

- Function entry points and their names
- Function arguments for each function entry point
- Global data and their names
- Manifest constants used in definitions of function arguments and global data
- Visible data structures used in function arguments and global data
- One or more shared objects, each having a particular name, each accessible through a particular pathname, and each containing the function entry points, function entry point names, global data, and global data names defined for that Library Interface.

Command Interface

The Command Interface is the set of commands available to application programs. The Command Interface is defined in the chapter titled “Commands”.

Definition of SPARC Compliance

The terms “SPARC-compliant” and “conforming” are used interchangeably in this document. Their meaning is:

Conforming Implementations

A conforming implementation is one that provides all of the Required Interfaces, in their entirety.

A conforming implementation may provide one or more of the Optional Interfaces. Each Optional Interface that is provided must be provided in its entirety. The product documentation must state which Optional Interfaces are provided.

A conforming implementation, when provided with standard data formats and values at a named interface, will provide the behavior defined for those values and data formats at that interface. However, a conforming implementation may consist of separately packaged and/or sold components. For example, a vendor of a conforming implementation might sell the hardware, operating system and windowing system as separately packaged items.

A conforming implementation may provide additional interfaces with different names. It may also provide additional behavior corresponding to data values outside the standard ranges, for standard named interfaces. Such additional interfaces, or additional inputs to standard interfaces, are called extensions to the standard. If an implementation provides extensions to the standard, its documentation must clearly identify the extensions as such.

Conforming Application Programs

A conforming application program has the following characteristics:

Its executable files are either Bourne shell scripts or object files in the format defined for the Object File Format System Interface.

Its object files participate in dynamic linking as defined in the Program Loading and Linking System Interface.

It employs only the instructions, traps, and other low-level facilities defined in the Low-Level System Interface as being for use by application programs.

It does not require or use any interface, facility, or implementation-provided extension that is not defined in this standard in order to be installed or to execute successfully.

If it requires any Optional Interface defined in this standard in order to be installed or to execute successfully, the requirement for that Optional Interface is stated in the application's documentation.

It does not use any interface or data format that is not required to be provided by a conforming implementation; unless:

1. if any such interface or data format is used, it is generally available to anyone who wants to purchase or acquire it; and
2. if such an interface or data format is supplied by another program through direct invocation of that program during execution, that program is in turn a SPARC-compliant application; and
3. the use of that interface or data format, as well as its source, is identified in the documentation of the application program.

Rationale

A SPARC-compliant application is expected to have no dependencies on any vendor extensions to the standard. The most common such extensions are additional function entry points and additional libraries other than the ones defined in the SCD. If an application requires such extensions it is not portable, since other SCD-compliant platforms may not provide those extensions.

A SPARC-compliant application is required to use system services on the platform it's running on, rather than importing system routines from some other platform. Thus it must link dynamically to any routines in the platform that perform system traps to kernel services.

It is to be expected that some programs may be companion programs to other programs. For example, a query program may be a companion to a data base program; a pre-processor may be an adjunct to one or more compilers; a data re-formatter may convert data from one document manager to another. In such cases, the program may or may not be SPARC-compliant regardless of whether the other program it's dependent on is SPARC-compliant.

If such an application merely uses data produced by another program, the application's compliance is independent of the other program's compliance.

If such an application actually invokes another program during execution (as, for example, a third-party math library), the invoking program is SPARC-compliant only if it also constitutes a SPARC-compliant application in combination with the invoked program.

Compliance Testing

On authorization by SPARC International of a branding program, test suites will be used in conjunction with this standard to verify the conformance of applications and platforms to this standard. Contact SPARC International for additional test suite information at (408) 748-9111.

The System Compliance Test (SCT) will be used to verify a system's implementation of all the Interfaces defined in either the 32-bit ABI sections or the 64-bit ABI sections of SPARC Compliance Definition.

The SPARC Application Verifier (SAV) will be used to verify an application's adherence to the Interfaces defined in either the 32-bit ABI sections or the 64-bit ABI sections of SPARC Compliance Definition.

Changes between SCD 2.3 and SCD 2.4

1- 64 bit psABI and gABI sections are added to many chapters

2- libsys and libc are merged under libc

- 3- new libraries are added: libw, libucb, libresolv, librpcsvc, libintl, libelf, libnisdb, libposix4, libpthread.
- 4- additional functions are added to libc, libm, libnsl, and libsocket.
- 5- new errata
- 6- the old Large Files Support section is replaced by a new section based on the Large File Summit document.

Upward Compatibility

The interfaces in the 32-bit ABI sections of SCD 2.4 are upwardly compatible with the interfaces in SCD 2.3, which in turn are upwardly compatible with the interfaces in SCD 2.2, SCD 2.1 and SCD 2.0. That is to say, an application written to the interfaces defined in SCD 2.0 will run successfully without change or re-compilation on a system that implements SCD 2.1, SCD 2.2, SCD 2.3 or the 32-bit ABI sections of SCD 2.4.

CHAPTER 2: Software Installation

SCD
2.4

Software Installation

Introduction

This chapter is common to both the 32-bit ABI and 64-bit ABI.

Most information regarding software installation may be found in Chapter 2 of the gABI and Chapter 2 of the psABI. The commands supported are listed in table 2.1 below. This section is an addendum to Chapter 2, page 2-1, of the psABI. This section adds support for using CD-ROM medium for physical distribution of SCD-conforming software. It is an OPTIONAL INTERFACE. If software is distributed on CD-ROM, it must be in one of the formations specified below.

Table 2-1. Software Installation commands

installf	pkgask	pkginfo	pkgrm
pkgadd	pkgchk	pkgparam	removef

Software Installation Changes

The following are changes to the *System V Application Binary Interface*, and the *System V Interface Definition (Third Edition)* as reported to SPARC International.

#	Facility	Location	Description
1	pkginfo(AS_CMD)	SVID, Vol. II	Delete “-r” from the list of supported options for pkginfo.
2	pkgadd(AS_CMD)	gABI	Change page 2-13 of the gABI to specify that the request script, if provided, runs with a uid of root and that standard input is attached to /dev/null.
3	The request script Procedure Scripts	gABI	The description about the execution environment is incorrect. Only uid == root is guaranteed.

CD-ROM Medium

CD-ROM medium recorded in the format specified in *ISO/IEC 10149: Data Interchange for read-only 120mm optical data disk (CD-ROM)* is added to the list of approved media on page 2-1 of the *System V Application Binary Interface, SPARC Processor Supplement*.

The information on the media must be represented either

- serially as the data stream created using dd(AU_CMD) or cpio(BU_CMD) utilities; or
- as file structured data that must be represented as described in *ISO 9660: 1988 - Volume and file structure of CD-ROM for information interchange*.

Rationale

The most common format for CD-ROM's is the ISO 9660 format, which supports MS-DOS filesystem semantics only. The ISO 9660 format is robust and stable, and has a huge installed base. That is why the ISO 9660 format has been included in SCD2.4 as an OPTIONAL standard for SPARC-compliant systems.

Support for ISO 9660 format CD-ROM's is already available from several other operating system vendors.

Since the restrictions placed on a filesystem by the ISO 9660 format are too restrictive for most UNIX users, a POSIX conforming filesystem is needed. The Rock Ridge Interchange Protocol was created to fill this gap. The Rockridge filesystem is actually an extension to (and compliant with) the ISO 9660 specification.

The Rock Ridge filesystem appears to be stable at this time. However there are some issues concerning bootability, security, and sparse files which are still being addressed by the IEEE working group on CD-ROM filesystems. There will be some minor changes made before the Rock Ridge filesystem is adopted as a NIST (National Institute of Science and Technology) standard.

For these reasons, the Rock Ridge filesystem is being excluded from SCD 2.4.

Upon adoption as a standard by NIST, it is expected that the Rock Ridge format will be included in the standard.

CHAPTER 3: Low-Level System Information

SCD
2.4

Low-Level System Information

Introduction

This chapter is split into a 32-bit psABI and a 64-bit psABI section. Low-level system information pertinent to the 32-bit psABI may be found in Chapter 3 of the *System V ABI, SPARC Processor Supplement*. Information such as page size restrictions, as well as stack management, function calling sequence and data representations may be found there. On the other hand, unless explicitly stated, information in the 64-bit psABI section of this chapter is independent of System V ABI documentation.

Self-Modifying Coding Practices:

Self modified (or otherwise changed) code sequences must be the target of the appropriate sequence of FLUSH instructions prior to being executed. A specific example of a problematic code sequence can be seen in the code fragment:

```
(void) read(fd, buf, sizeof (buf));
(*(void(*)())buf)();
```

which treats the contents of "buf" as a function which has just been read in.

Low-level System Information Changes (32-bit psABI)

#	Facility	Location	Description																	
1	Fundamental	psABI	Add the following to Figure 3-1: <table border="1" data-bbox="737 1129 1432 1367"> <thead> <tr> <th>Type</th> <th>C</th> <th>sizeof</th> <th>Alignment (bytes)</th> <th>SPARC</th> </tr> </thead> <tbody> <tr> <td rowspan="3">Integral</td> <td>long long</td> <td>8</td> <td>8</td> <td rowspan="2">signed doubleword</td> </tr> <tr> <td>signed long long</td> <td>8</td> <td>8</td> </tr> <tr> <td>unsigned long long</td> <td>8</td> <td>8</td> <td>unsigned doubleword</td> </tr> </tbody> </table>	Type	C	sizeof	Alignment (bytes)	SPARC	Integral	long long	8	8	signed doubleword	signed long long	8	8	unsigned long long	8	8	unsigned doubleword
Type	C	sizeof	Alignment (bytes)	SPARC																
Integral	long long	8	8	signed doubleword																
	signed long long	8	8																	
	unsigned long long	8	8	unsigned doubleword																
2	Registers and the stack Frame	psABI	Add the following description to the Stack Frame page 3-13: %i0, %i1, %o0 and %o1 64-bit integer return values appear in %i0 and %i1 (most significant word in %i0). A calling function receives values in the coincident out registers, %o0 and %o1.																	
3	Integral and Pointer Arguments	psABI	Add the following description: 64-bit integer argument uses two registers.																	
4	Functions Returning Scalars or No Value	psABI	Add the following description: A function that returns a 64-bit integer value places its result in %i0 and %i1 (most significant word in %i0); the calling function finds that value in %o0 and %o1.																	
5	Bit-Fields	psABI	On page 3-5, replace the statement: "A bit-field must entirely reside in a storage unit appropriate for its declared type." with the statement: "A bit field must entirely reside in a storage unit whose																	

			size and alignment are the size and natural alignment, respectively, of a variable having the field's declared type. This unit of storage is called the field's declared enclosing storage unit".
6	Bit-Fields	psABI	Add the following Bullet on page 3-5: <i>"* If a bit field is accessed by multiple threads, assignments to the bit field, or to any data objects residing in the bit field's declared enclosing storage unit, should be regarded as critical sections, and should be guarded with locks to ensure mutual exclusion."</i>
7	Bit-Fields	psABI	Add the following Bullet on page 3-5: <i>"* A processor executing a bit field assignment should effect the assignment using instructions that read and write units of storage no larger than the field's declared enclosing storage unit."</i>
8	Window Save	psABI	On page 3-10, add to the second bullet: <i>Area %r16(%l0) is saved at %sp+0, %r17(%l1) is saved at %sp+4, and so on. %r31(%i7) is saved at %sp+60.</i>
9	Function	psABI	On page 3-14, remove the phrase <i>"</i> , and the floating-point state register (FSR)" from the sentence that begins <i>"Registers in this category include global, floating-point ..."</i>
10	Function Calling	psABI	<p>On page 3-14, add a paragraph after the line which ends: <i>"application programs must never change the system global registers"</i>:</p> <p>Different fields of the floating-point state register (FSR) are treated thus:</p> <p>Called functions shall preserve the RD, TEM, and NS mode bits, unless changing those mode bits is part of the called function's documented semantics.</p> <p>Called functions shall preserve any aexc status bits set to value 1, unless clearing those bits is part of the called function's documented semantics. Otherwise, called functions should OR into aexc additional bits for un-trapped IEEE exceptions that reflect problems in the called function's numerical results.</p> <p>Called functions should not OR exception bits into aexc corresponding to un-trapped "gratuitous" exceptions that do not indicate any problems with the numerical results returned by those functions. The "should" aexc recommendations reflect that not all functions can economically distinguish significant from gratuitous exceptions. The cxec and fcx bits "belong" to the called function in the same sense as Y and PSR. Even so, functions that do not directly or indirectly execute any floating-point operations should avoid gratuitously altering FSR.</p>

Low-Level System Information (64-bit psABI) -EXPERIMENTAL

Machine Interface

Processor Architecture

The SPARC™ Architecture Manual, Version 9 defines the processor architecture. Programs intended to execute directly on the processor use the instruction set, instruction encoding, and instruction semantics of the architecture. Five points deserve explicit mention.

- A SPARC 64-bit ABI conforming program may not use the **IMPDEP1** and **IMPDEP2** instructions.
- A program may assume all other documented non-privileged instructions exist.
- A program may assume all other documented non-privileged instructions work.
- A program may assume that all documented unrestricted ASI's work.
- A program may use only the non-privileged instructions defined by the architecture, with the exception of **IMPDEP1** and **IMPDEP2**.

In other words, from a program's perspective, the execution environment provides a complete and working implementation of the non-privileged part of the SPARC V9 architecture. Although the **IMPDEP1** and **IMPDEP2** instructions are part of the SPARC V9 architecture, they may not be used by 64-bit ABI conforming programs because their behavior is undefined.

This does not imply that the underlying implementation provides all instructions in hardware, only that the instructions perform the specified operations and produce the specified results. The SPARC 64-bit ABI neither places performance constraints on systems nor specifies what instructions must be implemented in hardware.

Some processors might support the SPARC 64-bit ABI as a subset, providing additional instructions or capabilities. Programs that use those capabilities explicitly do not conform to the SPARC 64-bit ABI. Executing those programs on machines without the additional capabilities gives undefined behavior.

**NOTE**

For performance reasons it is suggested that the **FLUSH** instruction not be used. The routine "**sync_instruction_memory**" is the preferred way to flush instruction memory.

It is suggested that the instructions marked as "deprecated" in "*The SPARC Architecture Manual, Version 9*" not be used. These instructions may exhibit poor performance in some Version 9 implementations of the architecture and may not be available in future versions of the architecture.

Data Representation

Fundamental Types

Figure 3-1 shows the correspondence between ANSI C's scalar types and the SPARC V9 processor's.

Figure 3-1: Scalar Types

Type	C	sizeof	Alignment (bytes)	SPARC V9
Integral	char signed char	1	1	signed byte
	unsigned char	1	1	unsigned byte
	short signed short	2	2	signed halfword
	unsigned short	2	2	unsigned halfword
	int signed int enum	4	4	signed word
	unsigned int	4	4	unsigned word
	long signed long long long signed long long	8	8	signed extended-word
	unsigned long unsigned long long	8	8	unsigned extended-word
Pointer	<i>any-type</i> * <i>any-type</i> (*)()	8	8	unsigned extended-word
Floating-point	float	4	4	single-precision
	double	8	8 (see text)	double-precision
	long double	16	16 (see text)	quad-precision

A null pointer (for all types) has the value zero.

Within this specification, the term halfword refers to a 16-bit object, the term word refers to a 32-bit object, and the terms extended word and doubleword refer to a 64-bit object.

Double and quad-precision floating-point values occupy 1 and 2 extended words, respectively. Their natural alignment is the same, meaning their addresses are multiples of 8 and 16. Compilers should allocate independent data objects with the alignment shown in Figure 3-1; examples include global arrays of double-precision variables, **FORTRAN COMMON** blocks, and unconstrained stack objects. However, some language facilities (such as **FORTRAN EQUIVALENCE** statements) may create objects with only word alignment. Consequently, arbitrary double- and quad-precision addresses, such as pointers or reference parameters, might or might not be properly aligned. Systems should efficiently implement all **LDDF(A)**, **STDF(A)**, **LDQF(A)**, and **STQF(A)** instructions with target addresses that are word aligned, even if they are not aligned as shown in Figure 3-1, so that compilers should emit **LDDF(A)**, **STDF(A)**, **LDQF(A)**, and **STQF(A)** instructions unless it is known at compile time that the target address is not aligned as shown in Figure 3-1.

Aggregates and Unions

Aggregates (structures and arrays) and unions assume the alignment of their most strictly aligned component. The size of any object, including aggregates and unions, always is a multiple of the object's alignment. An array uses the same alignment as its elements. Structure and union objects can require padding to meet size and alignment constraints.

- An entire structure or union object is aligned on the same boundary as its most strictly aligned member.
- Each member is assigned to the lowest available offset with the appropriate alignment. This may require *internal padding*, depending on the previous member.

- A structure's size is increased, if necessary, to make it a multiple of the alignment. This may require *tail padding*, depending on the last member.

In the following examples, members' byte offsets appear in the upper left corners.

Figure 3-2: Structure Smaller Than a Word

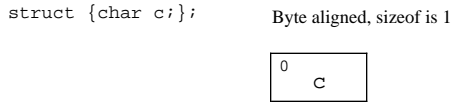


Figure 3-3: No Padding

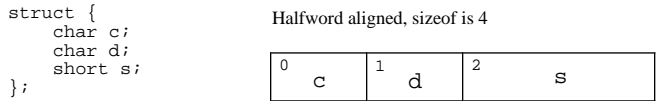


Figure 3-4: Internal Padding

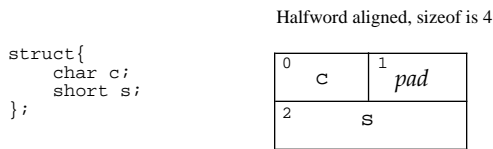


Figure 3-5: Internal and Tail Padding

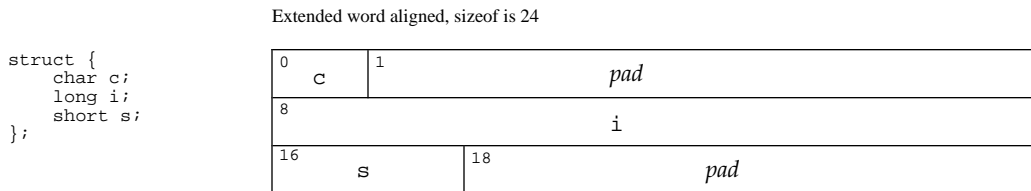
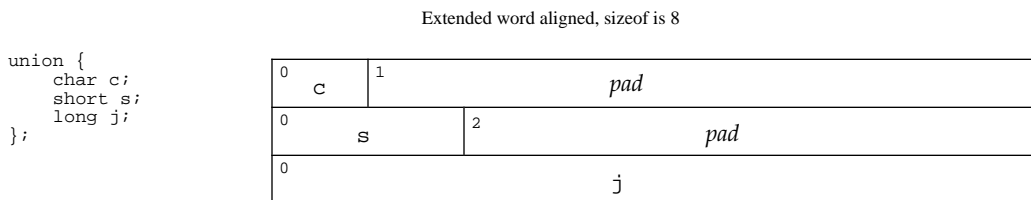


Figure 3-6: Union Allocation



Bit-Fields

C struct and union definitions may have *bit-fields*, defining integral objects with a specified number of bits.

Figure 3-7: Bit-Field Ranges

Bit-field Type	Width w	Range
signed char char unsigned char	1 to 8	-2^{w-1} to $2^{w-1}-1$ 0 to 2^w-1 0 to 2^w-1
signed short short unsigned short	1 to 16	-2^{w-1} to $2^{w-1}-1$ 0 to 2^w-1 0 to 2^w-1
signed int int unsigned int enum	1 to 32	-2^{w-1} to $2^{w-1}-1$ 0 to 2^w-1 0 to 2^w-1 0 to 2^w-1
signed long long unsigned long	1 to 64	-2^{w-1} to $2^{w-1}-1$ 0 to 2^w-1 0 to 2^w-1

“Plain” bit-fields always have non-negative values. Although they may have type char, short, int, long, or enum (which can have negative values), these bit-fields are extracted into an extended word with zero fill. Bit-fields obey the same size and alignment rules as other structure and union members, with the following additions.

- Bit-fields are allocated from left to right (most to least significant).
- A bit-field must entirely reside in a storage unit appropriate for its declared type. Thus a bit-field never crosses a unit boundary. Note: changing a bit-field may involve a non-atomic read-modify-write operation affecting the entire containing storage unit. Programs which thus might concurrently access the same storage unit must take the appropriate precautions.
- Bit-fields may share a storage unit with other struct/union members, including members that are not bit-fields. Of course, struct members occupy different parts of the storage unit. (A normal member sharing a bit-field’s storage unit is subject to potential conflicting updates as described above).
- Unnamed bit-field’s types do not affect the alignment of a structure or union, although individual bit-fields’ member offsets obey the alignment constraints.

The following examples show struct and union members’ byte offsets in the upper left corners; bit numbers appear in the lower corners.

Figure 3-8: Bit Numbering

0x8899AABBCCDDEEFF	0	1	2	3	4	5	6	7
	88	99	AA	BB	CC	DD	EE	FF
	63	55	47	39	31	23	15	7
								0

Figure 3-9: Left to Right Allocation

Extended word aligned, sizeof is 8								
0	j	k	m	<i>pad</i>				0
63	58	52	45					

```

struct {
    long j:5;
    long k:6;
    long m:7;
};
    
```


Figure 3-10: Boundary Alignment

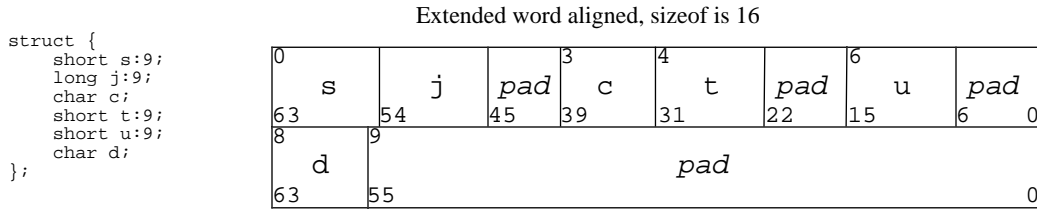


Figure 3-11: Storage Unit Sharing

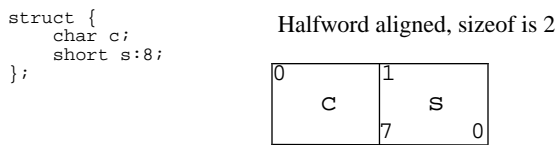


Figure 3-12: union Allocation

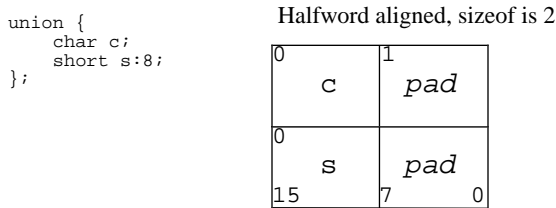
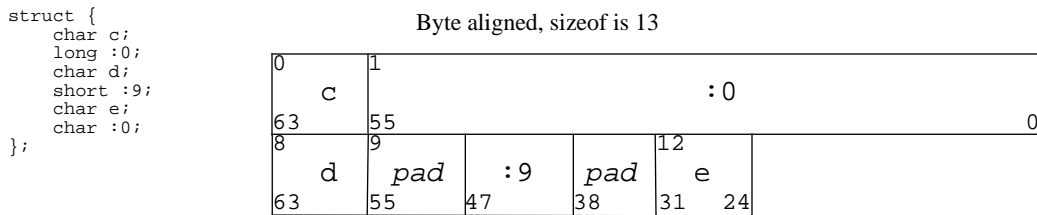


Figure 3-13: Unnamed Bit-fields



As the examples show, long bit-fields (including signed and unsigned) pack more densely than smaller base types. One can use char, short, and int bit-fields to force particular alignments, but long generally works better.

Function Calling Sequence

This section discusses the standard function calling sequence, including stack frame layout, register usage, parameter passing, etc. The system libraries described in Chapter 6 require this calling sequence.

NOTE

C programs follow the conventions given here. For specific information on the implementation of C, see "Coding Examples" in this chapter.

Registers and the Stack Frame

In the SPARC 64-bit ABI all floating-point registers and 8 integer registers are global to a running program, as the `save` and `restore` instructions do not affect them. All remaining integer registers are windowed: 24 are visible at any time, and sets of 24 overlap by 8 registers each. The `save` and `restore` instructions manipulate the windows as part of the normal function prologue and epilogue, making the caller's 8 *out* registers coincide with the callee's 8 *in* registers. Each window set also has 8 un-shared *local* registers. Generally, each new frame on the dynamic call stack uses a new register window. Brief register descriptions appear in Figures 3-14 and 3-15; more complete information appears later.

Figure 3-14: A Functions Window Register

Type	Name		Usage
in	%i7	%r31	return address - 8 †
	%fp	%i6 %r30	frame pointer †
	%i5	%r29	incoming param †
	%i4	%r28	incoming param †
	%i3	%r27	incoming param, † (outgoing return value)
	%i2	%r26	incoming param, † (outgoing return value)
	%i1	%r25	incoming param, † (outgoing return value)
	%i0	%r24	incoming param, † (outgoing return value)
local	%l7	%r23	local †
	%l6	%r22	local †
	%l5	%r21	local †
	%l4	%r20	local †
	%l3	%r19	local †
	%l2	%r18	local †
	%l1	%r17	local †
	%l0	%r16	local †
out	%o7	%r15	address of call instruction, ‡ temporary value
	%sp	%o6 %r14	stack pointer †
	%o5	%r13	outgoing param ‡
	%o4	%r12	outgoing param ‡
	%o3	%r11	outgoing param, ‡ (incoming return value)
	%o2	%r10	outgoing param, ‡ (incoming return value)
	%o1	%r9	outgoing param, ‡ (incoming return value)
	%o0	%r8	outgoing param, ‡ (incoming return value)

Figure 3-15: A Function's Global Registers

Type	Name		Usage	
global	%g7	%r7	global (reserved for system)	
	%g6	%r6	global (reserved for system)	
	%g5	%r5	global †	
	%g4	%r4	global †	
	%g3	%r3	global (reserved for application)	
	%g2	%r2	global (reserved for application)	
	%g1	%r1	global †	
	%g0	%r0	0	
floating-point	%q60	%d60,d62	floating-point †	
	%q56	%d56,d58	floating-point †	
	%q52	%d52,d54	floating-point †	
	%q48	%d48,d50	floating-point †	
	%q44	%d44,d46	floating-point †	
	%q40	%d40,d42	floating-point †	
	%q36	%d36,d38	floating-point †	
	%q32	%d32,d34	floating-point †	
	%q28	%d28,d30	%f28-f31	parameter †
	%q24	%d24,d26	%f24-f27	parameter †
	%q20	%d20,d22	%f20-f23	parameter †
	%q16	%d16,d18	%f16-f19	parameter †
	%q12	%d12,d14	%f12-f15	parameter †
	%q8	%d8,d10	%f8-f11	parameter †
	%q4	%d4,d6	%f4-f7	parameter, † (return value)
	%q0	%d0,d2	%f0-f3	parameter, † (return value)
special		%y	Y register †	
		%ccr	condition code register †	
		%asi	(see below)	
		%fpsr	(see below)	
		%fsr	(see below)	

NOTE Registers marked † above are assumed to be preserved across a function call. Registers marked ‡ above are assumed to be destroyed (volatile) across a function call.

In addition to a register window, each function has a frame on the run-time stack. This grows downward from high addresses, moving in parallel with the current register window. Figure 3-16 shows the stack frame organization.

Figure 3-16: Standard Stack Frame

Base	Offset	Contents	Frame
%fp+BIAS	>+176	unspecified ... variable size	Previous
%fp+BIAS	+176	(if present) additional incoming argument slots	
%fp+BIAS	+128	6 extended word argument slots	
%fp+BIAS	0	16 extended word save area (see below)	
%fp+BIAS	-1	unspecified ... variable size	Current
%sp+BIAS	>+176	(if needed) additional outgoing argument slots	
%sp+BIAS	+128	6 extended word argument slots	
	+120	save area for %i7	
	+112	save area for %i6	
	+104	save area for %i5	
	+96	save area for %i4	
	+88	save area for %i3	
	+80	save area for %i2	
	+72	save area for %i1	
	+64	save area for %i0	
	+56	save area for %i7	
	+48	save area for %i6	
	+40	save area for %i5	
	+32	save area for %i4	
	+24	save area for %i3	
	+16	save area for %i2	
	+8	save area for %i1	
%sp+BIAS	0	save area for %i0	
%sp+BIAS	-1	volatile memory	Low Address
%sp	0	(do not use)	

BIAS = 2047

Several key points about the stack frame deserve mention.

- Every stack frame must be 16-byte aligned.
- Every stack frame must have a 16-extended-word save area for the *in* and *local* registers, in case of window overflow or under-flow. This save area always must exist at %sp plus a **BIAS** of 2047 (0x7ff).
- Arguments that do not fit in the argument registers are passed on the stack.
- Other areas depend on the compiler and the code being compiled. The standard calling sequence does not restrict how a language system uses the “unspecified” areas of the standard stack frame.

NOTE The stack pointer is offset from the stack frame by a BIAS of 2047 (0x7ff). This BIAS permits stack frame references in the range of %fp+BIAS-6143 to %fp+BIAS+2048 and %sp+BIAS to %sp+BIAS+2048 to be made with only immediate offset addressing. By making the BIAS an odd number, the least significant bit of the stack pointer will be set and the register overflow and under-flow handlers can easily distinguish a 64-bit register window from a 32-bit register window.

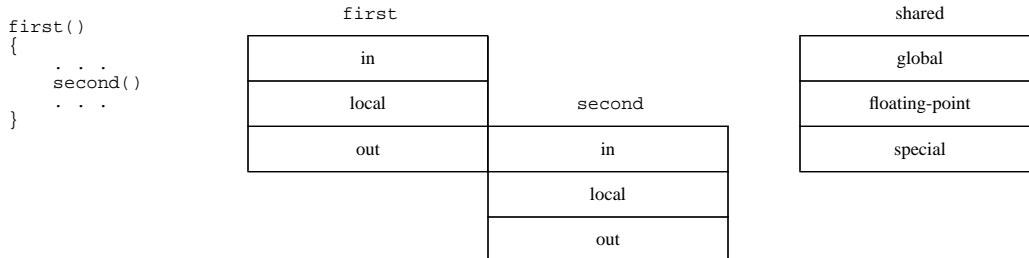
Across function boundaries, the standard function prologue shifts the register window, making the calling function’s *out* registers the called function’s *in* registers. It also allocates stack space, including the required areas of figure 3-16 and any private space it needs. The lowest 16 extended-words in the stack must—at all times—be reserved as the register save area. The example below illustrates this and allocates 176 bytes for the stack frame.

Figure 3-17: Function Prologue

```
second:
    save %sp, -176, %sp
```

For demonstration, assume a function named `first` calls `second`. The register windows for the two functions appear below.

Figure 3-18: Register Windows



As explained later, the function epilogue executes a restore instruction to unwind the stack and restore the register windows to their original condition.

NOTE Strictly speaking a function does not need the save and restore instructions if it preserves the registers as described below. Although some functions can be optimized to eliminate the save and restore, the general case uses the standard prologue and epilogue.

Some registers have assigned roles.

%sp or %o6	The <i>stack pointer</i> plus the stack BIAS determines the limit of the current stack frame, which is the address of the stack's bottommost, valid word. At all times the stack pointer plus the stack BIAS must point to a 16-byte aligned, 16 extended words window save area.
%fp or %i6	The <i>frame pointer</i> plus the stack BIAS is the address of the previous stack frame, which coincides with the word immediately above the current frame. Consequently, a function has registers with which it can access both ends of its frame. Incoming overflow arguments reside in the previous frame, referenced as positive offsets from the frame pointer plus the stack BIAS.
%i0 and %o0	<i>Integral and pointer return values</i> appear in %i0. A calling function receives values in the coincident <i>out</i> register %o0.
%i0,%i1,%i2,%i3 (%o0,%o1,%o2,%o3)	<i>The integral fields of structure and all of the fields of union return values with a total size 32 bytes or less</i> appear in registers %i0, %i1, %i2 and %i3. A calling function receives values in the coincident <i>out</i> registers
%i7 and %o7	The <i>return address</i> is the location to which a function should return control. Because a calling function's <i>out</i> registers coincide with the called function's <i>in</i> registers, the calling function puts a return address in its own %o7, while the called function finds the address in %i7 (if it has established its own stack frame). Actually, the return address register holds the call instruction's address, normally making the return address %i7+8 for the called function. (Every call instruction has a delay instruction.) Between function calls, %o7 serves as a scratch register.
%f0,%f1,%f2,%f3 (%d0,%d2) (%q0)	<i>Floating-point return values</i> appear in the floating-point registers. Single-precision values occupy %f0; double-precision values occupy %d0; quad-precision values occupy %q0. (Refer to the <i>SPARC™ Architecture Manual, Version 9</i> for details on the register numbering scheme). Otherwise, these are scratch registers.
%f0 through %f7 (%d0 through %d6) (%q0 and %q4)	<i>Floating-point fields from structure return values with a total size of 32 bytes or less</i> appear in the floating-point registers.
%i0 through %i5	<i>Incoming non-floating-point parameter slots</i> use up to 6 corresponding <i>in</i> registers. Arguments beyond the sixth extended-word appear on the stack.
%o0 through %o5	<i>Outgoing non-floating-point parameters slots</i> use up to 6 corresponding <i>out</i> registers. Arguments beyond the sixth extended-word appear on the stack.
%f1,%f3 through %f29,%f31 (%d0 through %d30) (%q0 through %q28)	<i>Floating-point arguments</i> are passed in the floating-point registers. Unpromoted single-precision arguments are passed in the first 16 odd-numbered %f registers. Double-precision arguments are passed in registers %d0 through %d30. Quad-precision arguments are passed in registers %q0 through %q28. These registers are assumed volatile across the call.
%l0 through %l7	<i>Local registers</i> have no specified role in the standard calling sequence.
%d32 through %d62 (%q32 through %q60)	These <i>floating-point registers</i> have no specified role in the standard calling sequence. They are assumed volatile across function calls.
%g0	<i>Global register 0</i> has no specified role in the standard calling sequence.
%g1,%g4,%g5	<i>Global registers 1, 4, and 5</i> have no specified role in the standard calling sequence. They are assumed volatile across function calls. In addition, registers %g1 and %g5 are volatile between caller and callee if the call is to an external function (goes through PLT).
%g2,%g3	<i>Global registers 2 and 3</i> are reserved for the application software. Software outside the control of the application (including the libraries described in Chapter 6) should not modify these registers at any time. Software that uses these registers must indicate how it has used them; see "Symbol Table" in Chapter 4 for details. (Note: in a multi-threaded environment, these registers are thread-local. System libraries that implement thread-switching must save and restore these registers on a thread switch.)
%g6 and %g7	<i>Global registers 6 and 7</i> are reserved for system software.
%ccr,%y	These <i>special registers</i> are volatile across function calls.
%asi	The <i>address space identifier register</i> by default holds the value ASI_PRIMARY_NOFAULT. If modified, it must be restored to the default value before calling another function or returning.
%fsr	The RD, TEM and NS fields are preserved across function calls; the other fields are volatile . The AEXC bits may be set by a callee, but may not be cleared.
%fprs	The <i>floating point register state register</i> has no specified role in the calling sequence. The DU and DL bits are volatile across function calls. For treatment of the FEF see "Special Registers" in Section 3.4.1.

With some exceptions given below, all registers visible to both a calling and a called function 'belong' to the called function. In other words, a called function may use all visible registers without saving their values before it changes them and without restoring their values before it returns. Registers in this category include *global* registers 1, 4, and 5,

floating-point registers, out registers (for the calling function), in registers (for the called function), the Y register, CCR, and the volatile bits of the FSR. Correspondingly, if a calling function wants to preserve such a register value across a function call, it must save the value and restore it explicitly. The exceptions are the stack pointer, %sp, %asi, global registers 2, 3, 6 and 7. A called function is obligated to preserve the stack pointer for its caller. Application programs must never change global registers 6 and 7. Local registers in each window are private. A called function should not change its calling function's local or in registers, even though the registers may be visible temporarily.

Signals can interrupt processes [see *signal(BA_OS)*]. Functions called during signal handling have no unusual restrictions on their use of registers. Moreover, if a signal handling function returns, the process resumes its original execution path with registers restored to their original values. Thus programs and compilers may freely use all non-reserved registers, even *global* and *floating-point* registers, without the danger of signal handlers changing their values. The *address space identifier* register will be set to **ASI_PRIMARY_NOFAULT** on entry to the signal handler

NOTE

There are some routines, like *setjmp()*, *sigsetjmp()*, and *vfork()*, that require the caller assume the registers %i0 through %i7, and %i0 through %i5 are **volatile** across the call.

Function Argument Passing

It is convenient to describe parameter linkage in terms of an array. Conceptually, parameters are assigned into an array of extended words, left-to-right, with an occasional "hole" to satisfy alignment restrictions. Typically, most parameter values will be "promoted" from their memory locations into registers, and most calls are expected to execute this way with less overhead.

The following diagram shows the correspondence between parameter registers and the "parameter array."

Memory	Integral	Float	Float from structure	Double	Quad
%sp+BIAS+248		%f31	%f30 and %f31	%d30	
%sp+BIAS+240		%f29	%f28 and %f29	%d28	%q28
%sp+BIAS+232		%f27	%f26 and %f27	%d26	
%sp+BIAS+224		%f25	%f24 and %f25	%d24	%q24
%sp+BIAS+216		%f23	%f22 and %f23	%d22	
%sp+BIAS+208		%f21	%f20 and %f21	%d20	%q20
%sp+BIAS+200		%f19	%f18 and %f19	%d18	
%sp+BIAS+192		%f17	%f16 and %f17	%d16	%q16
%sp+BIAS+184		%f15	%f14 and %f15	%d14	
%sp+BIAS+176		%f13	%f12 and %f13	%d12	%q12
%sp+BIAS+168	%o5	%f11	%f10 and %f11	%d10	
%sp+BIAS+160	%o4	%f9	%f8 and %f9	%d8	%q8
%sp+BIAS+152	%o3	%f7	%f6 and %f7	%d6	
%sp+BIAS+144	%o2	%f5	%f4 and %f5	%d4	%q4
%sp+BIAS+136	%o1	%f3	%f2 and %f3	%d2	
%sp+BIAS+128	%o0	%f1	%f0 and %f1	%d0	%q0

An "integral type" is any eight-bit char, sixteen-bit short, thirty-two bit int, sixty-four bit long, sixty-four bit long long, or a sixty-four bit pointer to any type.

A "floating type" is any thirty-two bit float, sixty-four bit double, or a one-hundred-twenty-eight bit long double.

Structures and unions up to sixteen bytes in size are passed more efficiently than structures and unions larger than sixteen bytes in size.

To call a function with parameters, a calling function allocates a “parameter array” in its stack frame (see Figure 3-16), sufficiently large to pass all parameters in memory. However, some values are not stored in this array, but are passed only in registers; see below. In the description below, *%i* and *%o* register names are used according to context. See the descriptions of the **SAVE** and **RESTORE** instructions for the relationship between these.

Every register used to pass parameter values has a corresponding location, at a fixed offset, in the parameter array.

Integral and Pointer Arguments

Argument values of all integral types will be assigned to one extended word in the parameter array. Each argument value of integral type smaller than an extended word will be widened by the caller to an extended word according to the signed-ness of the argument type. Any integral or pointer parameters assigned to locations $\%sp+BIAS+128$ through $\%sp+BIAS+168$ in the parameter array will be passed in registers *%o0..%o5*. The corresponding locations in the parameter array will have undefined values. The corresponding *%f*, *%d*, and *%q* register(s) will also be undefined.

NOTE	The type of an argument depends on whether the called function has a prototype declaration in scope at the call site. In the case where no prototype is present, the type of an argument of integral type is determined by the “integral promotions” rule of ANSI C and K&R C . The use of functions without prototypes is deprecated in ANSI C and may be removed from a future version of the C standard. When using a 64-bit ABI load, the address of any unpromoted integral parameter is the address of the slot in the parameter array.
-------------	--

Floating Arguments

Each single-precision parameter value will be assigned to one extended word in the parameter array, and right-justified within that word; the left half (even float register) is undefined. Each double-precision parameter value will be assigned to one extended word in the parameter array. Each quad-precision parameter value will be assigned to two extended words in the parameter array. Long doubles must be quad-aligned, and thus a “hole” might be introduced into the parameter array to force alignment. When a callee prototype exists, and does not indicate variable arguments, floating-point values assigned to locations $\%sp+BIAS+128$ through $\%sp+BIAS+248$ will be promoted into floating-point registers, as shown above.

When a callee prototype exists and a particular floating argument matches the “...” of a function with variable arguments, floating values assigned to locations $\%sp+BIAS+128$ through $\%sp+BIAS+168$ will be promoted to *%i0..%i5*.

When no prototype exists for a callee:

Floating values assigned to locations $\%sp+BIAS+128$ through $\%sp+BIAS+168$ will be passed simultaneously in *%i0..%i5* and *%d0..%d10* (or *%q0..%q8*).

Floating values assigned to locations $\%sp+BIAS+176$ through $\%sp+BIAS+248$ will be passed simultaneously in memory and in *%d12..%d30*.

Structure and Union arguments

Structure or union types up to eight bytes in size are assigned to one parameter array word, and align to eight-byte boundaries.

Structure or union types larger than eight bytes, and up to sixteen bytes in size are assigned to two consecutive parameter array words, and align according to the alignment requirements of the structure or at least to an eight-byte boundary.

Structure or union types are always left-justified, whether stored in registers or memory. The individual fields of a structure (or containing storage unit in the case of bit fields) are subject to promotion into registers based on their type using the same rules as apply to scalar values (with the addition that a single-precision floating-point number assigned to the left half of an argument slot will be promoted into the corresponding even-numbered float register.). Any union type being passed directly is subject to promotion into the appropriate integer register(s).

Note that a sixteen-byte structure with all integral fields assigned to locations $\%sp+BIAS+168$ and $\%sp+BIAS+176$ will be “split,” as the contents of location $\%sp+BIAS+168$ will be promoted to *%o5*.

Structures or unions larger than sixteen bytes are copied by the caller and passed indirectly; the caller will pass the address of a correctly aligned structure value. This sixty-four bit address will occupy one word in the parameter array, and may be promoted to an `%o` register like any other pointer value. The callee may modify the addressed structure.

The caller can omit the copy if such omission cannot be detected. That requires (at least) that:

- the original aggregate is already properly aligned,
- the original aggregate is not aliased,
- the original aggregate is not used after the call, and
- no language-specific semantics require the copy.

Variable Argument Lists

A function that expects a variable argument list typically uses the `stdarg.h` mechanism to process the list. That mechanism defines a `va_list` type that can be passed to another function. Due to the use of the parameter array described above `va_list` is of type `void *`.

Function Result Passing

Functions declared to return the void type do not return a value. All other functions return their values according to the following rules.

Integral and pointer return values

Integral and pointer return types are returned in integer register `%o0`. Functions returning integral and pointer return values always return an extended-word, expanding signed and unsigned bytes, halfwords, and words as needed.

Floating return values

A return value of a floating-point type is passed in `%f0`, `%d0`, or `%q0` as appropriate.

Structure or Union return values

Structure and union return types up to thirty-two bytes in size are returned in registers. The registers are assigned as if the value was being passed as the first argument to a function with a known prototype.

For types with a larger size the caller allocates an area large enough and aligned properly to hold the return value, and passes a pointer to that area as an implicit first argument (of type pointer-to-data) to the callee. This implicit argument logically precedes the first actual argument, and is allocated according to normal argument passing rules (i.e. into `%o0`). The callee must store the function return value in the result area before control is returned to the caller and after the last use or definition of any variable that might overlap with the result area. If the callee is terminated through any means other than a normal function return (e.g., through a call to the `longjmp` function), the contents of the result area are undefined.

In the common case that the caller immediately assigns the returned value to a program variable, the caller may substitute the address of the assigned program variable in place of the allocated result area and omit the code to do the assignment, as long as this substitution does not change the program's externally visible behavior.

Note also that the caller is required to provide the implicit argument and a properly sized and aligned receiving area even if it does not wish to use the callee's function result. In that case, the caller may simply pass a pointer to a scratch area.

So that compilers are not forced to emit in-line code for structure copy, Section 6.2 defines a set of routines optimized for this purpose. In the case of a routine which had kept its first argument in `%i0` and was returning a value pointed to by `%i1`, epilogue code would take the form:

```

mov      %i0, %o0
mov      %i1, %o1
call     __align_cpy_n
mov      size, %o2
ret
restore  %o0, %g0, %o0

```

Examples of Argument Passing

All the following examples assume the caller sees a prototype for the callee.

Integral and Pointer Arguments

As mentioned, a function receives up to the first six parameter array words (those that contain integral and pointer arguments) through the *in* registers, *%i0* through *%i5*. Functions pass all integral arguments as extended-words, expanding signed or unsigned bytes, halfwords and words as needed. If a function call has more than 6 integral and pointer arguments the others go on the stack.

Figure 3-19: Integral and Pointer Arguments

Argument	Call	Caller	Callee
1	<code>g(char,</code>	<code>%o0</code>	<code>%i0</code>
2	<code>char,</code>	<code>%o1</code>	<code>%i1</code>
3	<code>short,</code>	<code>%o2</code>	<code>%i2</code>
4	<code>int,</code>	<code>%o3</code>	<code>%i3</code>
5	<code>char *,</code>	<code>%o4</code>	<code>%i4</code>
6	<code>int,</code>	<code>%o5</code>	<code>%i5</code>
7	<code>int,</code>	<code>[%sp+BIAS+176]</code>	<code>[%fp+BIAS+176]</code>
8	<code>void *);</code>	<code>[%sp+BIAS+184]</code>	<code>[%fp+BIAS+184]</code>

Floating-Point Arguments

Up to the first sixteen parameter array words (those that contain floating-point arguments) are passed in floating-point registers.

Figure 3-20: Floating-Point Arguments

Argument	Call	Caller	Callee
1	<code>h(float,</code>	<code>%f1</code>	<code>%f1</code>
2	<code>float,</code>	<code>%f3</code>	<code>%f3</code>
3	<code>double,</code>	<code>%d4</code>	<code>%d4</code>
4	<code>float,</code>	<code>%f7</code>	<code>%f7</code>
5	<code>double,</code>	<code>%d8</code>	<code>%d8</code>
6	<code>float,</code>	<code>%f11</code>	<code>%f11</code>
7	<code>float,</code>	<code>%f13</code>	<code>%f13</code>
8	<code>long double,</code>	<code>%q16</code>	<code>%q16</code>
9	<code>double</code>	<code>%d20</code>	<code>%d20</code>
10	<code>long double);</code>	<code>%q24</code>	<code>%q24</code>

An Example of Mixed Arguments

Figure 3-20a: Mixed Arguments

Argument	Call	Caller	Callee
1	f(char,	%o0	%i0
2	float,	%f3	%f3
3	short,	%o2	%i2
4	double,	%d6	%d6
5	int,	%o4	%i4
6	float,	%f11	%f11
7	long,	[%sp+BIAS+176]	[%fp+BIAS+176]
8	long,	[%sp+BIAS+184]	[%fp+BIAS+200]
9	double);	%d16	%d16

Examples of Result Passing

Functions Returning Scalars or No Value

A function that returns an integral or pointer value places its result in `%i0`; the calling function finds that value in `%o0`. A floating-point return value appears in the floating-point registers for both the calling and the called function. Single-precision uses `%f0`; double-precision uses `%d0`; quad-precision uses `%q0`. Functions that return no value (also called procedures or `void` functions) put no particular value in any return register. Those registers may be used as scratch registers, however. A call instruction writes its own address into *out* register `%o7`. As usual for a control transfer instruction, the call instruction takes a delay instruction that is executed before the instruction of the called function. Because every instruction is one word long, the return address is the address of the call instruction plus 8. The value is `%i7+8` for the called function and `%o7+8` for the calling function. The following example returns the value contained in *local* register `%l4`.

Figure 3-23: Function Epilogue

```

jmp1 %i7 + 8, %g0
restore %l4,0,%o0

```

If a function returns no value or if the return register already contains the desired value, the next epilogue would suffice.

Figure 3-24: Alternative Function Epilogue

```

jmp1 %i7 + 8, %g0
restore %g0,0,%g0

```

Operating System Interface

Virtual Address Space

Processes execute in a 64-bit ABI virtual address space; this is mapped as the primary address space. Memory management hardware translates virtual addresses to physical addresses, hiding physical addressing and letting a process run anywhere in the system's real memory. Processes typically begin with three logical segments, commonly called text, data and stack. As Chapter 5 describes, dynamic linking creates more segments during execution, and a process can create additional segments for itself with system services.

NOTE

The effects of using load and store alternate instructions with address space identifiers other than **ASI_PRIMARY**, **ASI_PRIMARY_NOFAULT**, **ASI_PRIMARY_LITTLE**, and **ASI_PRIMARY_NOFAULT_LITTLE** are undefined and/or implementation-dependent. Programs that use address space identifiers other than these four do not conform to the 64-bit ABI.

Page Size

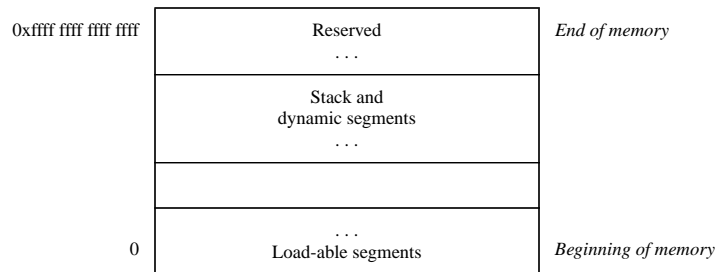
Memory is organized by pages, which are the system’s smallest units of memory allocation. Page size can vary from one system to another, depending on the processor, memory management unit and system configuration. Processes may call `sysconf(BA_OS)` to determine the system’s current page size. The maximum page size for SPARC 64-bit ABI is 1 MB.

Virtual Address Assignments

Conceptually, processes have the full 64-bit address space available. In practice, however, several factors may limit the size of a process.

- The system may reserve a configuration-dependent amount of virtual space.
- A tunable configuration parameter may limit process size.
- A process whose size exceeds the system’s available, combined physical memory and secondary storage cannot run. Although some physical memory must be present to run any process, the system can execute processes that are bigger than physical memory, paging them to and from secondary storage. Nonetheless, both physical memory and secondary storage are shared resources. System load, which can vary from one process execution to the next, affects the available amounts .

Figure 3-26: Virtual Address Configuration



Load-able segments: Processes’ load-able segments may begin at 0. The exact addresses depend on the executable file format [see Chapters 4 and 5].

Stack and dynamic segments: A process’s stack and dynamic segments reside below the reserved area. Processes can control the amount of virtual memory allocated for stack space, as described below.

Reserved A reserved area may reside at the top of virtual memory.

NOTE

Although application programs may begin at virtual address 0, they conventionally begin at or above `0x100000` (1 MB), leaving the initial 1 MB with an invalid address mapping. Processes that reference this invalid memory (for example by dereferencing a null pointer) generate an access exception trap, as described in the “Trap Interface” section of this chapter. A process may, however, establish a valid mapping for this area using the `mmap(KE_OS)` facilities.

As the figure shows, the system reserves the high end of virtual space with a process’s stack and dynamic segments below that. Although the exact boundary between the reserved area and a process depends on the system’s configuration, the reserved area shall not consume more than 8 exabytes (EB) from the virtual address space. Thus the user virtual address range has a minimum upper bound of `0x7ff ffff ffff`. Individual systems may reserve less space, increasing

processes' virtual memory range. More information follows in the section "Managing the Process Stack".

NOTE

Unlike 32-bit ABI, for 64-bit ABI the stack is mapped with read and write but not execute permissions. This change is intended to make programs more secure against buffer overflow attacks.

Although applications may control their memory assignments, the typical arrangement follows the diagram above. Loadable segments reside at low addresses; dynamic segments occupy the higher range. When applications let the system choose addresses for dynamic segments (including shared object segments), it will choose high addresses. This leaves the "middle" of the address spectrum available for dynamic memory allocation with facilities such as `malloc(BA_OS)`.

Managing the Process Stack

The section "Process Initialization" in this chapter describes the initial stack contents. Stack addresses can change from one system to the next—even from one process execution to the next on the same system. Processes, therefore, should not depend on finding their stack at particular virtual address. A tunable configuration parameter controls the system maximum stack size. A process also can use `setrlimit(BA_OS)`, to set its own maximum stack size, up to the system limit. On 64-bit ABI, the stack segment has read and write permissions.

Coding Guidelines

Operating system facilities, such as `mmap(KE_OS)`, allow a process to establish address mappings in two ways. First, the program can let the system choose an address. Second, the program can force the system to use an address the program supplies. This second alternative can cause application portability problems, because the requested address might not always be available. Differences in virtual address space can be particularly troublesome between different architectures, but the same problems can arise within a single architectures.

Processes' address spaces typically have three segment areas that can change size from one execution to the next: the stack [through `setrlimit(BA_OS)`], the data segment [through `malloc(BA_OS)`], and the dynamic segment area [through `mmap(KE_OS)`]. Changes in one area may affect the virtual addresses available for another. Consequently, an address that is available in one process execution might not be available in the next. A program that used `mmap(KE_OS)` to request a mapping at a specific address thus could appear to work in some environments and fail in others. For this reason, programs that wish to establish a mapping in their address space should let the system choose the address.

Despite these warnings about requesting specific addresses, the facility can be used properly. For example, a multiprocess application might map several files into the address space of each process and build relative pointers among the files' data. This could be done by having each process ask for a certain amount of memory at an address chosen by the system. After each process receives its own private address from the system, it would map the desired files into memory, at specific addresses within the original area. This collection of mappings could be at different addresses in each process but their relative positions would be fixed. Without the ability to ask for specific addresses, the application could not build shared data structures, because the relative positions for files in each process would be unpredictable.

Trap Interface

Hardware Trap Types

The operating system defines the following correspondence between hardware traps and the signals specified by *signal(BA_OS)*.

Figure 3-27: Hardware Traps and signals

Trap Name	Signal
instruction_access_exception	SIGSEGV, SIGBUS
instruction_access_MMU_miss	SIGSEGV
instruction_access_error	SIGBUS
illegal_instruction	SIGILL
privileged_opcode	SIGILL
fp_disabled	SIGILL
fp_exception_ieee_754	SIGFPE
fp_exception_other	SIGFPE
tag_overflow	SIGEMT
division_by_zero	SIGFPE
data_access_exception	SIGSEGV, SIGBUS
data_access_MMU_miss	SIGSEGV
data_access_error	SIGBUS
data_access_protection	SIGSEGV
mem_address_not_aligned	SIGBUS
privileged_action	SIGILL
async_data_error	SIGBUS
trap_instruction	see next table

The signal is sent only if no user trap handler is provided. See User Traps.

Two trap types, *instruction_access_exception* and *data_access_exception*, can generate two signals. In both cases, the “normal” signal is SIGSEGV. Nonetheless, if the access also causes some external memory error (such as parity error), the system generates SIGBUS.

Floating point instructions exist in the architecture, but they may be implemented either in hardware or software. If the *fp_disabled* or *fp_exception_other* trap occurs because of an unimplemented, valid instruction, the process receives no signal. Instead the system intercepts the trap, emulates the instruction, and returns control to the process. A process receives **SIGILL** for the *fp_disabled* trap only when the indicated floating-point instruction is illegal (invalid encoding, etc.).

Software Trap Types

The operating system defines the following correspondence between software traps and the signals specified by *signal(BA_OS)*.

Figure 3-28: Software Trap Types

Trap Number	Signal	Purpose
0	unspecified	Reserved to OS vendor (SunOS 4.x syscall)
1	SIGTRAP	Breakpoint
2	SIGFPE	Division by zero
3	unspecified	Reserved to OS vendor (old flush windows trap)
4	none	clean windows trap
5	SIGILL	Range checking

Figure 3-28: Software Trap Types

Trap Number	Signal	Purpose
6	none	Fix alignment
7	SIGFPE	Integer overflow
8	unspecified	Reserved to the OS vendor (32-bit ABI syscall)
9-15	unspecified	Reserved to the OS vendor
16-31	SIGILL	Reserved for user applications
32	unspecified	Reserved (old get condition codes)
33	unspecified	Reserved (old set condition codes)
34-47	unspecified	Reserved to the OS vendor
48-63	unspecified	Reserved to the OS vendor
64-66	SIGSYS	system calls (reserved for OS vendor)
67	SIGSYS	system calls (reserved for OS source licensee)
68	SIGILL	Return from deferred trap
69-79	unspecified	Reserved to SPARC International
80-95	unspecified	Reserved to SPARC International
96-111	unspecified	Reserved to the OS vendor
112-255	unspecified	Reserved to the OS vendor

- 0 and 8 Trap types 0 and 8 were used in some 32-bit ABI systems to implement operating system service routines. In the SPARC 64-bit ABI they are reserved.
- 1 A debugger can set a breakpoint by inserting a trap instruction whose type is 1.
- 2 A process can explicitly signal division by zero with this trap.
- 3 Trap type 3 was used in 32-bit ABI systems to ask the system to flush all its register windows to the stack. In the SPARC 64-bit ABI the `flushw` instruction can be used instead. The trap is reserved.
- 4 Normally during process execution, **SAVE** instructions allocate new register windows with local and out registers that either all contain zeros or all contain the corresponding register contents from a currently-live function that is higher on the dynamic call stack (earlier in the call chain) than the current function. (A **SAVE** instruction can also allocate an old register window; this window will contain data from a previously-called, no-longer-active function in this address space.) This differs from the default 32-bit ABI behavior, where new register windows could have contained kernel data or data from another process. The default 64-bit behavior is adequate for most programs that need predictable register window contents. However, some programs need even cleaner windows than SPARC 64-bit ABI provides by default. Executing a type 4 trap causes the system to initialize local and out registers in all existing old windows and all subsequent new windows to zero. This behavior continues until the process terminates. Re-issuing the trap causes existing old windows to be cleaned.
- 5 A process can explicitly signal a range checking error with this trap.
- 6 Executing a type 6 trap makes the operating system “fix” subsequent unaligned data references. Although the references still generate *memory_address_not_aligned* traps, the operating system handles the trap, emulates the data reference, and returns control to the process without generating a signal. In this context a “data reference” is a load or store operation. Implicit memory references, such as control transfers, must always be aligned properly, and the stack must always be aligned as described elsewhere.

This trap is provided to ease porting of existing code. Its use in new code is deprecated. A user trap handler should be used instead. If a user trap handler for **UT_MEM_ADDRESS_NOT_ALIGNED** is installed, it takes precedence.
- 7 A process can explicitly signal integer overflow with this trap. Either a positive or a negative value can cause overflow.
- 9-15 The operating system reserves these traps for its own use. Programs that use them do not conform to the 64-bit ABI.
- 16-31 These traps are reserved for user applications and are subject to being handled by user trap handlers (see following section).

- 32 Trap type 32 was used in 32-bit ABI systems to copy the *icc* integer condition codes from the PSR register to global register %g1. In the 64-bit ABI the CCR register is not privileged and can be accessed directly. The trap is reserved.
- 33 Trap type 33 was used in 32-bit ABI systems to copy the rightmost four bits from global register %g1 to the **PSR** *icc* integer condition codes. In 64-bit ABI the CCR register is not privileged and can be accessed directly. The trap is reserved.
- 34-63 The operating system reserves these traps for its own use. Programs that use them do not conform to the 64-bit ABI.
- 64-66 Operating system service routines defined by an OS vendor are implemented using these trap types.
- 67 OS source licensee specific operating system service routines are implemented using this trap type.

NOTE

The 64-bit ABI does not define the implementation of individual system calls. Instead, programs should use the system libraries that chapter 6 describes. Programs with embedded system call trap instructions do not conform to the 64-bit ABI.

- 68 Trap 68 is used to return control to the system from a deferred user trap handler.
- 69-95 Reserved for future allocation by SPARC International.
- 96-127 The operating system reserves these trap types for its own use. Programs that use them do not conform to the 64-bit ABI.

User Traps

The operating system can redirect certain traps from non-privileged code back to user trap handlers. The interface for this functionality is declared in the new include file `<sys/utrap.h>` as shown in Figure 3-28a. See Figure 6-143 for the header file.

Figure 3-28a: Hardware Traps and User Traps

Trap Name	User Trap
illegal_instruction	UT_ILLTRAP_INSTRUCTION UT_ILLEGAL_INSTRUCTION †*
fp_disabled	UT_FP_DISABLED †*
fp_exception_ieee_754	UT_FP_EXCEPTION_IEEE_754 †
fp_exception_other	UT_FP_EXCEPTION_OTHER
tag_overflow	UT_TAG_OVERFLOW †*
division_by_zero	UT_DIVISION_BY_ZERO †
mem_address_not_aligned	UT_MEM_ADDRESS_NOT_ALIGNED †
privileged_action	UT_PRIVILEGED_ACTION †
privileged_opcode	UT_PRIVILEGED_OPCODE
async_data_error	UT_ASYNC_DATA_ERROR
trap_instruction	UT_TRAP_INSTRUCTION_16 through UT_TRAP_INSTRUCTION_31 †* †*
instruction_access_exception instruction_access_MMU_miss instruction_access_error	UT_INSTRUCTION_EXCEPTION or UT_INSTRUCTION_PROTECTION or UT_INSTRUCTION_ERROR
data_access_exception data_access_MMU_miss data_access_error data_access_protection	UT_DATA_EXCEPTION or UT_DATA_PROTECTION or UT_DATA_ERROR

User trap types marked with † above are required and must be provided by all 64-bit ABI-conforming implementations. The others may not be present on every implementation; an attempt to install a user trap handler for a type not present will return **EINVAL**. User trap types marked with * above are implemented as precise traps only. Most user trap types are self-explanatory; a few require a few more words.

UT_ILLTRAP_INSTRUCTION

This trap is raised by user execution of the **ILLTRAP** instruction. It is always precise.

UT_ILLEGAL_INSTRUCTION

This trap will be raised by execution of otherwise undefined opcodes. It is implementation-dependent as to what opcodes raise this trap; the 64-bit ABI only specifies the interface. The trap may be precise or deferred.

UT_PRIVILEGED_OPCODE

All the opcodes declared to be privileged in the SPARC V9 architecture will raise this trap. It is implementation-dependent whether other opcodes will raise it as well; the 64-bit ABI only specifies the interface.

UT_DATA_EXCEPTION, UT_INSTRUCTION_EXCEPTION

No valid user mapping can be made to this address, for a data or instruction access, respectively.

UT_DATA_PROTECTION, UT_INSTRUCTION_PROTECTION

A valid mapping exists, and user privilege to it exists, but the type of access (read, write, or execute) is denied, for a data or instruction access, respectively.

UT_DATA_ERROR, UT_INSTRUCTION_ERROR

A valid mapping exists, and both user privilege and the type of access are allowed, but an unrecoverable error occurred in attempting the access, for a data or instruction access, respectively. *%l1* will contain either **BUS_ADDRERR** or **BUS_OBJERR**.

UT_FP_DISABLED

This trap is raised when an application issues a floating point instruction (including load or store) and **FEF** is zero. If a user handler is installed for this trap, it will be given control. Otherwise, the system will set **FEF** to one and retry the instruction.

A functional interface is provided to establish the user trap handlers and is defined Section 6.2.

For all traps, the handler executes in a new window, where the *in* registers are the *out* registers of the previous frame and have the value they contained at the time of the trap. Similarly the global registers (including the special registers *%ccr*, *%asi*, and *%y*) and the floating-point registers have their values at the time of the trap. The stack pointer register *%sp* plus the **BIAS** will point to a properly-aligned 128-byte register save area; if the handler needs scratch space, it should decrement the stack pointer to obtain it. If the handler needs access to the previous frame's *in* registers or *local* registers, it should execute a **FLUSHW** instruction, and then access them off of the frame pointer. If the handler calls an 64-bit ABI-conforming function, it must set the *%asi* register to **ASI_PRIMARY_NOFAULT** before the call.

Precise Traps

On entry to a precise user trap handler *%l6* contains the *%pc* and *%l7* contains the *%npc* at the time of the trap. To return from a handler and reexecute the trapped instruction, the handler would execute:

```

    jmp1 %l6, %g0
    return %l7

```

To return from a handler and skip the trapped instruction, the handler would execute:

```

    jmp1 %l7, %g0
    return %l7+4

```

Deferred Traps

On entry to a deferred user trap handler *%o0* contains the address of the instruction that caused the trap and *%o1* contains the actual instruction (right-justified, zero extended), if the information is available. Otherwise *%o0* contains the value -1 and *%o1* is undefined. For certain cases additional information may be made available as indicated in the following table.

Instructions	Additional Information
LD-type LDSTUB	<i>%o2</i> contains the effective address (<i>rs1 + rs2 simm13</i>).
ST-type CAS SWAP	<i>%o2</i> contains the effective address (<i>rs1 + rs2 simm13</i>). <i>%o3</i> contains the data to be stored if available.

Integer arithmetic	%o2 contains the <i>rs1</i> value. %o3 contains the <i>rs2 simm13</i> value. %o4 contains the contents of %y register.
Floating-point arithmetic	%o2 contains the address of <i>rs1</i> value. %o3 contains the address of <i>rs2</i> value.
Control-transfer	%o2 contains the target address (<i>rs1 + rs2 simm13</i>).
Asynchronous data errors	%o2 contains the address that caused the error. %o3 contains the effective ASI, if available, else -1

To return from a deferred trap, the trap handler issues:

```
ta      68      !ST_RETURN_FROM_DEFERRED_TRAP
```

The instruction that causes the trap will NOT be retried.

Dispatching Traps

The following pseudo-code explains how the operating system dispatches traps.

```
if (precise_trap) {
    if (precise_handler) {
        invoke(precise_handler);
        /* not reached */
    } else {
        convert_to_signal(precise_trap);
    }
} else if (deferred_trap) {
    if (deferred_handler) {
        invoke(deferred_handler);
        /* not reached */
    } else {
        convert_to_signal(deferred_trap);
    }
}
if (signal)
    send(signal);
```

User trap handlers must preserve all registers except the *locals* (%l0-7) and *outs* (%o0-7), i.e. %i0-7, %g1-7, %d0-62, %asi, %fsr, %fprs, %ccr, and %y, except to the extent that modifying the registers is part of the desired functionality of the handler. For example, the handler for **UT_FP_DISABLED** may load floating-point registers.

Process Initialization

This section describes the machine state that *exec(BA_OS)* creates for “infant” processes, including argument passing, register usage, stack frame layout, and so on. Programming language systems use this initial program state to establish a standard environment for their application programs. As an example, a C program begins executing at a function named *main*, conventionally declared in the following way.

Figure 3-39: Declaration for main

```
extern int main (int argc, char *argv[], char *envp[]);
```

Briefly, *argc* is a non-negative argument count; *argv* is an array of argument strings, with *argv[argc] == 0*; and *envp* is an array of environment strings, also terminated by a null pointer. Although this section does not describe C program initialization, it gives the information necessary to implement the call to *main* or the entry point for a program in any other language.

Special Registers

The architecture defines three main non-privileged state registers and one privileged register to control and monitor the processor. They are the condition code register (**CCR**), the floating-point register state register (**FPRS**), the floating-point

state register (**FSR**), and the processor state register (**PSTATE**). The tables below give the initial state of these registers.

Figure 3-30a: Condition Code Register (CCR) Fields

Field	Value	Note
xcc	unspecified	Extended integer condition codes unspecified
icc	unspecified	Integer condition codes unspecified

The architecture defines floating point instructions, and those instructions work whether the processor has a hardware floating-point unit or not. (A system may provide hardware or software floating point facilities.) In either case, however, the processor presents a working floating-point implementation, including an **FPRS** and an **FSR** with the following initial values.

Figure 3-30b: Floating-point Registers State (FPRS) Fields

Field	Value	Note
FEF	0	Floating-point unit disabled
DL	0	Lower half of floating point registers are not dirty
DU	0	Upper half of floating-point registers are not dirty

If an application issues a floating-point instruction (including load or store) when **FEF** is zero, and a user-trap handler is installed for **UT_FP_DISABLED**, that trap handler is invoked. If no such user-trap handler is installed, the system sets **FEF** to one and retries the instruction.

An application may set **FEF** to 1 at any time, or may let the system set **FEF** to 1 for it. When **FEF** is 1, the system preserves floating-point state for the application over context switches.

An application may set **FEF** to 0 at any time. The system will never set **FEF** to 0 on its own, except that a multi-threading library may set **FEF** to 0 when dispatching a thread with a saved **FEF** of 0. When **FEF** is 0, the system does not preserve floating-point state for the application over context switches. The user-visible contents of the floating-point registers, including the **FSR**, become undefined when **FEF** is set to 0. An application may find it useful to set **FEF** to 0 when it no longer needs to preserve the contents of the floating-point registers, as the system will not need to save and restore them over context switches.

Figure 3-31: Floating-point State (FSR) Register Fields

Field	Value	Note
fcc3	unspecified	Floating-point condition codes unspecified
fcc2	unspecified	Floating-point condition codes unspecified
fcc1	unspecified	Floating-point condition codes unspecified
RD	0	Round to nearest
TEM	0	Floating-point traps not enabled
NS	0	Nonstandard mode off
ver	read only	Implementation version number
ftt	unspecified	Floating-point trap type unspecified
qne	0	Floating-point queue (if any) is empty
fcc0	unspecified	Floating-point condition codes unspecified
aexc	0	No accrued exceptions
cexc	0	No current exceptions

Application programs cannot access the **PSTATE** register directly; they run in the processor’s non-privileged mode, and the instructions to read and write **PSTATE** are privileged. Nonetheless, a program “sees” a processor that behaves as if **PSTATE** had the following values. **PSTATE** fields not in the table either have unspecified values or do not affect user program behavior.

Figure 3-31a: Processor State Register Fields

Field	Value	Note
CLE	0	Data reads and writes using an implicit ASI are big-Indian. Set according to the rules in Chapter 4, ELF Header.
MM	unspecified	
RED	0	Processes run in normal mode.
AM	0	Processes run in a 64-bit address space.
PRIV	0	Processes run in non-privileged mode.
IE	1	Interrupts enabled.

Other non-privileged registers and their initial states are listed in the table below.

Figure 3-31b: Other Non-privileged Registers

Register	Value	Note
%asi	ASI_PRIMARY_NOFAULT	Address space identifier default (see below)
%tick	positive	
%pc	--	The current program counter
%y	unspecified	Y register unspecified

The %tick register is a per-cpu register that is incremented on every cpu clock cycle. The %tick registers in an MP system may or may not be synchronized. While the %tick register is monotonically increasing on each cpu, it may appear to increase by huge amounts if an interrupt or signal occurs between instructions, and it may appear to increase or decrease by huge amounts if a context-switch (to another cpu) occurs between instructions.

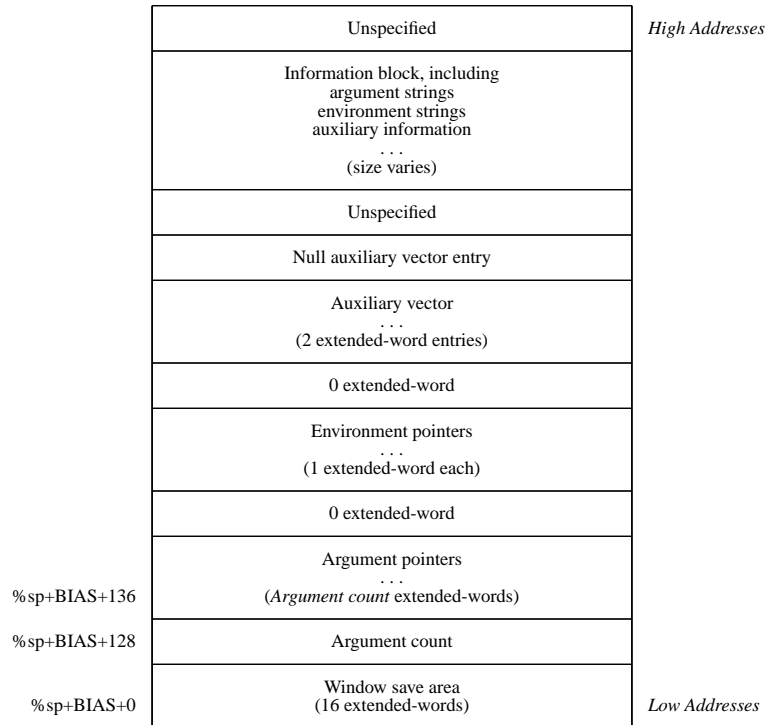
Therefore, programmers may use %tick directly to measure short sequences of code that have a high probability of completing execution before an interrupt (such as end of time slice) occurs. Repeated measurements should be taken, with outlier data points discarded. System vendors are encouraged to provide low-latency real and virtual timing facilities that may be used to measure longer sequences of code.

The ancillary state registers besides the Y, CCR, ASI, TICK, PC, and FPRS registers either are privileged or unspecified by the architecture. Applications thus may not execute the RDASR and WRASR instructions, with the exception of RDY, RDCCR, RDASI, RDTICK, RDPC, RDFPRS, STBAR, MEMBAR, WRY, WRCCR, WRASI, WRFPRS, and SIR.

Process Stack and Registers

When a process receives control, its stack holds the arguments and environment from `exec(BA_OS)`.

Figure 3-32: Initial Process Stack



Argument strings, environment strings, and the auxiliary information appear in no specific order within the information block; the system makes no guarantees about their arrangement. The system also may leave an unspecified amount of memory between the null auxiliary vector entry and the beginning of the information block.

Except as shown below, global, floating point, and window registers have unspecified values at process entry. Consequently, a program that requires registers to have specific values must set them explicitly during process initialization. It should *not* rely on the system to set all registers to zero.

- `%g1` A non-zero value gives a function pointer that the application should register with `atexit(BA_OS)`. If `%g1` contains zero, no action is required.
- `%fp` The system marks the deepest stack frame by setting the frame pointer to zero. No other frame's `%fp` has a zero value.
- `%sp` Performing its usual job, the stack pointer plus the stack **BIAS** gives the address of the bottom of the stack, which is guaranteed to be 16-byte aligned.

Every process has a stack, but the system defines *no* fixed stack address. Furthermore, a program's stack address can change from one system to another - even from one process invocation to another. Thus the process initialization code must use the stack address in `%sp`. Data in the stack segment at addresses below the bottom of the stack contain undefined values.

Whereas the argument and environment vectors transmit information from one application program to another, the auxiliary vector conveys information from the operating system to the program. This vector is an array of the following structures, interpreted according to the `a_type` member.

Figure 3-33:Auxiliary Vector

```
typedef struct
{
    int    a_type;
    union {
        long    a_val;
        void    *a_ptr;
        void    (*a_fcn)();
    } a_un;
} auxv_t;
```

In the following example, the stack resides below 0x8000 0000 0000 0000, growing toward lower addresses. The process receives three arguments.

- cp
- src
- dst

It also inherits two environment strings (this example is not intended to show a fully configured execution environment).

- HOME=*/home/dir*
- PATH=*/home/dir/bin:/usr/bin:*

Its auxiliary vector holds one non-null entry, a file descriptor for the executable file.

- 13

The initialization sequence preserves the stack pointer's extended-word alignment.

Figure 3-35: Example Process Stack

0x7fff ffff ffff fff8	pad	pad	pad	pad	pad	pad	pad	pad	<i>High addresses</i>
	r	/	b	i	n	:	\0	pad	
0x7fff ffff ffff ffe8	/	b	i	n	:	/	u	s	
	h	o	m	e	/	d	i	r	
0x7fff ffff ffff ffd8	r	\0	P	A	T	H	=	/	
	/	h	o	m	e	/	d	i	
0x7fff ffff ffff ffc8	s	t	\0	H	O	M	E	=	
	c	p	\0	s	r	c	\0	d	
0x7fff ffff ffff ffb8	0								
	0				Uninitialized				
0x7fff ffff ffff ffa8	13								<i>Auxiliary vector</i>
	2				Uninitialized				
0x7fff ffff ffff ff98	0								
	0x7fff ffff ffff ffda								
0x7fff ffff ffff ff88	0x7fff ffff ffff ffcb								<i>Environment vector</i>
	0								
0x7fff ffff ffff ff78	0x7fff ffff ffff ffc7								
	0x7fff ffff ffff ffc3								
0x7fff ffff ffff ff68	0x7fff ffff ffff ffc0								<i>Argument vector</i>
0x7fff ffff ffff ff60	3								<i>Argument count</i>
%sp+BIAS 0x7fff ffff ffff fee0	Window save area (16 extended-words)								<i>Low addresses</i>

Coding Examples

This section discusses example code sequences for fundamental operations such as calling functions, accessing static objects, and transferring control from one part of a program to another. Previous sections discuss how a program may use the machine or the operating system, and they specify what a program may and may not assume about the execution environment. Unlike previous material, the information here illustrates how operations *may* be done, not how they *must* be done. As before, examples use the **ANSI C** language. Other programming languages may use the same conventions displayed below, but failure to do so does *not* prevent a program from conforming to the 64-bit ABI. In the sections that follow several new relocation types will be used to demonstrate features of the SPARC V9 architecture. From a reference of %hh or %lm, for example, one will infer correspondence to relocations named **R_SPARC_HH**, **R_SPARC_LM**, respectively. See the section “Relocation Types” in Chapter 4 for all the details.

Architectural Constraints

The SPARC V9 architecture has a number of constraints that make it desirable to use several different code models for

different purposes, in order to improve performance and reduce code size. The relevant constraints are:

- a) The *call* instruction has a 30 bit signed immediate value. The target address of a *call* instruction may thus be at most 2^{29} instructions (2^{31} bytes) before it or $2^{29} - 1$ instructions ($2^{31} - 4$ bytes) after it.
- b) Memory access instructions (e.g., *ldx* and *stx*) and arithmetic and logical instructions (e.g., *add* and *or*) have a 13-bit signed immediate value.
- c) The *sethi* instruction has a 22 bit unsigned immediate value that is placed in register bits 31..10. The other register bits are cleared.

Code Positionability

There are two code positionability models of interest:

- absolute The virtual addresses of instructions and static data are known at static link time. To execute properly, the object must be loaded at a specific virtual address, making the program's absolute addresses correspond with the process's virtual addresses.
- position-independent (PIC) The virtual addresses of instructions and static data are not known until dynamic link time. PIC uses PC-relative addresses, not absolute addresses. Consequently, the code is not tied to a specific load address, allowing it to execute properly at various positions in virtual memory.

Typically, executables have absolute code and shared objects such as dynamically linked libraries have PIC.

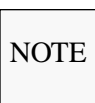
Code Size

Because of constraint (a) and (c), there are two code size models of interest:

- medium The address range spanned by all instructions in the object is less than 2^{31} bytes (2 GB) and the distance from any instruction to the location of the label `_GLOBAL_OFFSET_TABLE_` is less than 2^{31} bytes (2 GB)
- full The only limit on address range spanned is the available virtual address space.

One limiting case is a **CALL** instruction at the beginning of the code whose target address is at the end of the procedure linkage table. A single **CALL** instruction can be used for all subroutine calls within a medium code size object, more code is needed for full code size programs.

The second limiting factor is the ability of a simple code sequence to materialize a pointer to the Global Offset Table (applies to position independent code only).



Whether or not a program has medium code size may depend on the relative ordering of sections in the program. In particular it may be important to place the procedure linkage table before the global offset table.

Location

Because of constraints (b) and (c), there are four location models of interest:

- low The executable must be in the low 4 GB of the virtual address space.
- middle The executable must be in the low 16 TB of the virtual address space.
- high The executable must be in the high 4 GB of the virtual address space.
- anywhere The executable or shared object can be placed anywhere in the virtual address space.

All except the anywhere model apply only to absolute code. The low model generates the most efficient code for accessing static objects: two instructions and one register always suffice.

External Object References

A shared object that references an object external to itself must use indirect addressing. For example, the libc function *localtime()* references the external variable *daylight*. At the time the libc shared library is created, the address of *daylight* is not known, so references to it from libc go through a global offset table. Each shared object has its own global offset table, which is just a vector of addresses. Each object, e.g. *daylight*, is associated with an index into the global offset table. At dynamic link time, the dynamic linker fills in *daylight's* element in the global offset table with the absolute address of *daylight*.

Because of the effects of constraints (b) and (c) on addressing elements in global offset tables, there are four external object reference models. However, only the first three are of practical interest.

none	no global offset table is used.
small	The executable or shared object references at most 1024 external objects.
large	The executable or shared object references at most 2^{29} external objects.
huge	The size of the global offset table is limited only by the available virtual address space.

The limiting factor is the 13-bit signed immediate in load instructions. Assuming the address of the middle of the global offset table is already in some register, the small model can load any element with one LDX instruction, whereas the large model requires three instructions.

Combinations of Practical Interest

The following combinations of models might be of practical use. All models use dynamic linking.

Figure 3.35a: Code Models

Positionability	Code Size	Location	External Object Reference Model
absolute	medium	low	none
absolute	medium	middle	none
absolute	medium	anywhere	none
PIC	medium	anywhere	small
PIC	medium	anywhere	large

Integer Constant Loading

There are a number of ways to load an integer constant, *c*, into a register. The examples in the following table assume:

- %hi(*c*) is bits 31..10 of *c*,
- %lo(*c*) is the low-order 10 bits of *c*,
- %hix(*c*) is the ones complement of bits 31..10 of *c* (treated as a 64-bit vector),
- %lox(*c*) is the binary value 111 followed by the low-order 10 bits of *c*,
- %h44(*c*) is bits 43.. 22 of *c*,
- %m44(*c*) is bits 21..12 of *c*,
- %l44(*c*) is bits 11..0 of *c*,
- %hh(*c*) is bits 63..42 of *c*,
- %hm(*c*) is bits 41..32 of *c*,
- %lm(*c*) is bits 31..10 of *c*.

The table is not exhaustive. In the case where “*c*” is a symbolic address constant, each operator, e.g. %hi(), generates a specific relocation. See Section 4.3.1: Relocation Types, Figure 4-4 for a complete list of relocation types.

Figure 3.35b: Loading Integer Constants

Range	Code
$-2^{12} .. 2^{12} - 1$	or %g0, c, %o0
$0 .. 2^{32} - 1$	sethi %hi(c), %o0 or %o0, %lo(c), %o0
$-2^{32} .. -1$	sethi %hix(c), %o0 xor %o0, %lox(c), %o0
$0 .. 2^{44} - 1$	sethi %h44(c), %o0 or %o0, %m44(c), %o0 sllx %o0, 12, %o0 or %o0, %l44(c), %o0
$-2^{63} .. 2^{63} - 1$	sethi %hh(c), %o1 sethi %lm(c), %o0 or %o1, %hm(c), %o1 or %o0, %lo(c), %o0 sllx %o1, 32, %o1 or %o0, %o1, %o0

NOTE

Since the general case costs 6 instructions and two scratch registers, loading from a constant table may be more efficient in some cases.

Addressing Global Offset Tables

A subroutine in a shared object must obtain the address of the shared object’s global offset table before the subroutine can access the table. Typically, this is done in a prologue. The offset between the subroutine’s address and the middle of the global offset table must be known when the shared object is created. The following code examples place the address of the middle of the global offset table in %17; other registers can also be used. *offset* is the offset in bytes from the *rd* instruction to the middle of the global offset table. In the medium size case it is assumed to be positive. Implementation of the 64-bit ABI may use different, more efficient code sequences.

Figure 3.35c: Addressing Global Offset Tables

Medium Size Code		Full Size Code	
rd	%pc, %17	rd	%pc, %17
sethi	%hi(<i>offset</i>), %o0	sethi	%hh(<i>offset</i>), %o1
or	%o0, %lo(<i>offset</i>), %o0	sethi	%lm(<i>offset</i>), %o0
		or	%o1, %hm(<i>offset</i>), %o1
		or	%o0, %lo(<i>offset</i>), %o0
		sllx	%o1, 32, %o1
		or	%o0, %o1, %o0
add	%17, %o0, %17	add	%17, %o0, %17

Static Data References from Absolute Code

Figure 3.35d: Static Data References from Absolute Code

ANSI C	low	middle	anywhere
long s; long d; long *p;	.global s .global d .global p	.global s .global d .global p	.global s .global d .global p
p = &d;	sethi %hi(d),%o0 or %o0,%lo(d),%o0 sethi %hi(p),%o1 stx %o0,[%o1+%lo(p)]	sethi %h44(d),%o0 or %o0,%m44(d),%o0 sllx %o0,12,%o0 add %o0,%l44(d),%o0 sethi %h44(p),%o1 or %o1,%m44(p),%o1 sllx %o1,12,%o1 stx%o0,[%o1+%l44(p)]	sethi %hh(d),%o5 sethi %lm(d),%o0 or %o5,%hm(d),%o5 or %o0,%lo(d),%o0 sllx %o5,32,%o5 or %o0,%o5,%o0 sethi %hh(p),%o5 sethi %lm(p),%o1 or %o5,%hm(p),%o5 or %o1,%lo(p),%o1 sllx %o5,32,%o5 stx %o0,[%o1+%o5]
*p = s;	sethi %hi(s),%o0 ldx [%o0+%lo(s)],%o0 sethi %hi(p),%o1 ldx [%o1+%lo(p)],%o1 stx %o0,[%o1]	sethi %h44(s),%o0 or %o0,%m44(s),%o0 sllx %o0,12,%o0 ldx [%o0+l44(s)],%o0 sethi %h44(p),%o1 or %o1,%m44(p),%o1 sllx %o1,12,%o1 ldx[%o1+%l44(p)],%o1 stx %o0,[%o1]	sethi %hh(s),%o5 sethi %lm(s),%o0 or %o5,%hm(s),%o5 or %o0,%lo(s),%o0 sllx %o5,32,%o5 ldx [%o0+%o5],%o0 sethi %hh(p),%o5 sethi %lm(p),%o1 or %o5,%hm(p),%o5 or %o1,%lo(p),%o1 sllx %o5,32,%o5 ldx [%o1+%o5],%o1 stx %o0,[%o1]

Static Data References from PIC

The code sequences given below assume that %l7 points to the Global Offset Table.

Figure 3.35e: Static Data References from Position Independent Code

ANSI C	Small Model	Large Model
extern long s; extern long d; extern long *p;	.global s .global d .global p	.global s .global d .global p
p = &d;	ldx [%l7+d],%o0 ldx [%l7+p],%o1 stx %o0,[%o1]	sethi %hi(d),%o0 or %o0,%lo(d),%o0 ldx [%l7+%o0],%o0 sethi %hi(p),%o1 or %o1,%lo(p),%o1 ldx [%l7+%o1],%o1 stx %o0,[%o1]
*p = s;	ldx [%l7+s],%o0 ldx [%o0],%o0 ldx [%l7+p],%o1 ldx [%o1],%o1 stx %o0,[%o1]	sethi %hi(s),%o0 or %o0,%lo(s),%o0 ldx [%l7+%o0],%o0 ldx [%o0],%o0 sethi %hi(p),%o1 or %o1,%lo(p),%o1 ldx [%l7+%o1],%o1 ldx [%o1],%o1 stx %o0,[%o1]

Function Calls

Direct function calls are those where the name of the called function is known at compile time. The following code shows the cases of interest. The call instruction can be used in all medium size executables and shared objects. The PIC example assumes use of the Global Offset Table for addressing either procedures or Procedure Linkage Table slots.

Figure 3.43a: Function Calls

ANSI C	medium	absolute/full	PIC/full
<code>extern void f();</code>	<code>.global f</code>	<code>.global f</code>	<code>.global f</code>
<code>f();</code>	<code>call f</code> <code>nop</code>	<code>sethi %hh(f),%g5</code> <code>sethi %lm(f),%g1</code> <code>or %g5,%hm(f),%g5</code> <code>or %g1,%lo(f),%g1</code> <code>sllx %g5,32,%g5</code> <code>jmp1 %g1+%g5,%o7</code>	<code>sethi %hi(f),%g1</code> <code>or %g1,%lo(f),%g1</code> <code>ldx [%l7+%g1],%g1</code> <code>jmp1 %g1,%o7</code> <code>nop</code>

For indirect function calls, the address of the function is in a pointer. Appropriate code is used to load the value of the pointer into a register, just as with static data. A `jmp1` instruction is then used.

Branching

Programs use branch instructions to control their execution flow. As defined by the architecture, branch instructions hold a PC-relative value with up to a 2 MB range, allowing a branch to locations up to 1 MB away in either direction.

C switch statements provide multiway selection. The best implementation of a switch statement depends on the distribution of the case label values. When they are dense, as in the C example below then the computed-jump approach shown may generate good code. The example uses several simplifying conventions to hide irrelevant details:

- The selection expression resides in local register `%l0`.
- case label constants begin at zero.
- case labels and default use assembly names `.Lcasei` and `.Ldef`, respectively.

The following example is position-independent, and can also be used in absolute code.

Figure 3-46: Position-Independent switch Code

ANSI C	Assembly
<pre>switch (j) { case 0: ... case 2: ... case 3: ... default: ... }</pre>	<pre>subcc %l0, 4, %g0 movgu %xcc, 1, %l0 1: rd %pc, %l1 sllx %l0, 5, %l0 add %l0, (.Lcase0 - 1b), %l0 jmp1 %l0 + %l1, %g0 nop .Lcase0: instruction 1 instruction 2 instruction 3 instruction 4 instruction 5 instruction 6 ba .Lcase0_continued instruction 8 .Ldef: instruction 1 instruction 2 instruction 3 instruction 4 instruction 5 instruction 6 instruction 6 ba .Lcase_end instruction 8 .Lcase2:Lcase0_continued:Lcase_end:</pre>

The number of instructions in the legs can be varied. If there is not enough space in a leg, a branch to additional code

can be used.

C Stack Frame

The figure below shows the C stack frame organization. It conforms to the standard stack frame with designated roles

Figure 3-47: C Stack Frame

Base	Offset	Contents	Address
$\%fp+BIAS$	-1	y extended words local space: automatic variables ...	<i>High addresses</i>
	-8y	other address-able objects	
$\%sp+BIAS$	$+176 + 8x - 1$	x extended-words compiler scratch temporaries, register save area, and extra outgoing argument slots	
$\%sp+BIAS$	+176		
$\%sp+BIAS$	+128	required outgoing argument slots	
$\%sp+BIAS$	0	16 extended word window save area	<i>Low addresses</i>

for unspecified areas in the standard frame. A C stack frame doesn't normally change size during execution. The exception is dynamically allocated stack memory, discussed below. By convention, a function allocates automatic (local) variables in the top of its frame and references them as negative offsets from $\%fp+BIAS$. Its incoming overflow arguments reside in the previous frame, referenced as positive offsets from $\%fp+BIAS$.

Variable Argument List

Previous sections describe the rules for passing arguments. Unfortunately, some otherwise portable C programs depend on the argument passing scheme, implicitly assuming that 1) all arguments reside on the stack, and 2) arguments appear in increasing order on the stack. Programs that make these assumptions never have been portable, but they have worked on many machines. They do *not* work on 64-bit ABI because some of the arguments reside in integer and/or floating point registers. Portable C programs should use the facilities defined in the header files `<stdarg.h>` or `<varargs.h>` to deal with variable argument lists.

Allocating Stack Space Dynamically

Unlike some other languages, C does not need dynamic stack allocation *within* a stack frame. Frames are allocated dynamically on the program stack, depending on program execution, but individual stack frames can have static sizes. Nonetheless, the architecture supports dynamic allocation for those languages that require it, and the standard calling sequence and stack frame support it as well. Thus languages that need dynamic stack frame sizes can call C functions, and vice versa.

Figure 3-47 shows the layout of the C stack frame. The double line divides the area referenced with the frame pointer from the area referenced with the stack pointer. Dynamic space is allocated above the line as a downward growing heap whose size changes as required. Typical C functions have no space in the heap. All areas below the double line in the current frame have a known size to the compiler. Dynamic stack allocation thus takes the following steps.

1. Stack frames are 16-byte aligned; dynamic allocation should preserve this property. Thus the program rounds (up) the desired byte count to a multiple of 16.
2. The program decreases the stack pointer by the rounded byte count, increasing its frame size. At this point, the "new" space resides just above the register save area at the bottom of the stack.
3. The program copies the "bottom half" of the stack frame down into the new space, opening the middle of the frame.

Even in the presence of signals, dynamic allocation is "safe". If a signal interrupts allocation, one of the three things can

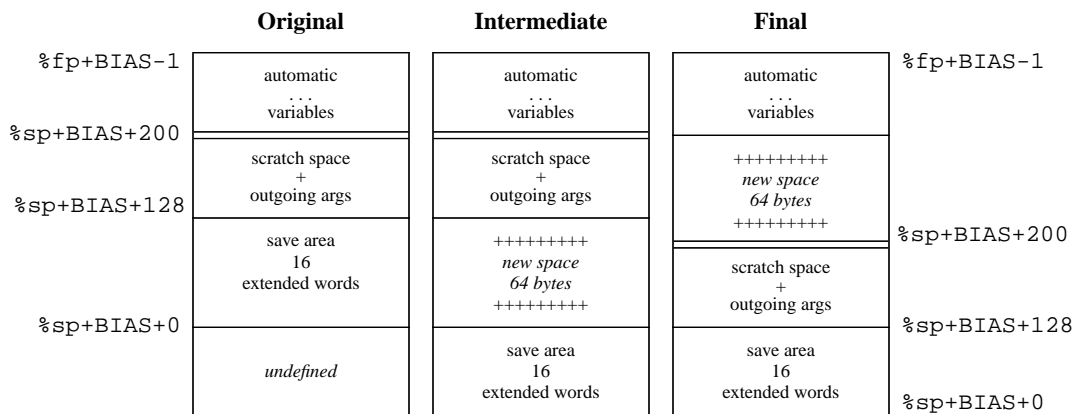
happen.

- The signal handler can return. The process then resumes the dynamic allocation from the point of interruption.
- The signal handler can execute a non-local *goto*, or *longjmp* [see *setjmp(BA_LIB)*]. This reset the process to a new context in a previous stack frame, automatically discarding the dynamic allocation.
- The process can terminate

Regardless of when the signal arrives during dynamic allocation, the result is a consistent (though possibly dead) process.

To illustrate, assume a program wants to allocate 50 bytes; its current stack frame has 24 bytes of compiled scratch space. The first step is rounding the 50 to 64, making it a multiple of 16. Figure 3-49 shows how the stack changes.

Figure 3-49: Dynamic Stack Allocation



New space starts at $\%sp+BIAS+200$. As described, every dynamic allocation in *this* function will return a new area starting at $\%sp+BIAS+200$, leaving previous stack objects untouched (other functions would have different stack addresses). Consequently, the compiler should compute the absolute address for each area, avoiding relative references. Otherwise future allocations in the same frame would destroy the stack's integrity.

Existing stack objects reside at fixed offsets from the frame and stack pointers; stack heap allocation preserves those offsets. Objects relative to the frame pointer don't move. Objects relative to the stack pointer move, but their $\%sp$ -relative positions do not change. Accordingly, compilers arrange not to publicize the absolute address of any object in the bottom half of the stack frame (in a way that violates the scope rules). $\%sp$ -relative references stay valid after dynamic allocation, but absolute addresses do not.

No special code is needed to free dynamically allocated stack memory. The function return resets the stack pointer and removes the entire stack frame, including the heap, from the stack. Naturally, a program should not reference heap objects after they have gone out of scope.

SCD

2.4

Object Files

Introduction

This chapter is split into a 32-bit ABI, a 64-bit gABI, and a 64-bit psABI section. Processor independent descriptions of the object file format for System V Release 4 may be found in Chapter 4 of the System V ABI. Information specific to the 32-bit psABI may be found in Chapter 4 of the System V ABI, SPARC Processor Supplement. On the other hand, unless explicitly stated, information in the 64-bit psABI section of this chapter is independent of System V ABI documentation.

Processor independent descriptions of the object file format for System V Release 4 may be found in Chapter 4 of the *System V ABI*. Information specific to SPARC platforms may be found in Chapter 4 of the *System V ABI, SPARC Processor Supplement*.

Object Files Changes (32-bit ABI)

The following are changes to the *System V Application Binary Interface* as reported to SPARC International.

#	Facility	Location	Description
1	SHT_DYNSYM	gABI	On page 4-14 add before the last sentence: "However this minimal set of symbols will always include all symbols of STB_GLOBAL binding."
2	Relocation Types	psABI	page 4-5, Figure 4-4 add the types in the table below. The descriptive portions of this table are for background information only. The primary significance of the table, given the presence of these interfaces as EXPERIMENTAL is to reserve the space of relocation values indicated in the table for use in EXPERIMENTAL system implementations. No conforming application will employ these values for any purpose, and no system is required to demonstrate conformance to any interpretation of these relocation types.

Name	Value	Field	Calculation
R_SPARC_PLT32	24	V-word32	L + A
R_SPARC_HIPLT22	25	T-imm22	(L + A) >> 10
R_SPARC_LOPLT10	26	T-simm13	(L + A) & 0x3ff
R_SPARC_PCPLT32	27	V-word32	L + A - P
R_SPARC_PCPLT22	28	V-disp22	(L + A - P) >> 10
R_SPARC_PCPLT10	29	V-simm12	(L + A - P) & 0x3ff
R_SPARC_10	30	V-simm10	S + A
R_SPARC_11	31	V-simm11	S + A
R_SPARC_64	32	V-xword64	S + A
R_SPARC_OLO10	33	V-simm13	((S + A) & 0x3ff) + 0
R_SPARC_HH22	34	V-imm22	(S + A) >> 42
R_SPARC_HM10	35	T-simm13	((S + A) >> 32) & 0x3ff
R_SPARC_LM22	36	T-imm22	(S + A) >> 10
R_SPARC_PC_HH22	37	V-imm22	(S + A - P) >> 42
R_SPARC_PC_HM10	38	T-simm13	((S + A - P) >> 32) & 0x3ff
R_SPARC_PC_LM22	39	T-imm22	(S + A - P) >> 10
R_SPARC_WDISP16	40	V-d2/disp14	(S + A - P) >> 2
R_SPARC_WDISP19	41	V-disp19	(S + A - P) >> 2
R_SPARC_GLOB_JMP	42	V-xword64	S + A
R_SPARC_7	43	V-imm7	(S + A) & 0x7f
R_SPARC_5	44	V-imm5	(S + A) & 0x1f
R_SPARC_6	45	V-imm6	(S + A) & 0x3f

R_SPARC_OLO10: This relocation type resembles **R_SPARC_LO10**, except an extra offset is added to make full use of the 13-bit signed immediate field.

R_SPARC_HH22: This relocation type is used by the assembler when it sees an instruction of the form “imm22-instruction ... %hh(absolute) ...”.

R_SPARC_HM10: This relocation type is generated by the assembler when it sees an instruction of the form “simm13-instruction ... %hm(absolute) ...”.

R_SPARC_LM22: This relocation type is used by the assembler when it sees an instruction of the form “imm22-instruction ... %lm(absolute) ...”. This resembles **R_SPARC_HI22**, except it truncates rather than validates.

R_SPARC_PC_HH22: This relocation type is used by the assembler when it sees an instruction of the form “imm22-instruction ... %hh(pc-relative) ...”.

R_SPARC_PC_HM10: This relocation type is generated by the assembler when it sees an instruction of the form “simm13-instruction ... %hm(pc-relative) ...”.

R_SPARC_PC_LM22: This relocation type is used by the assembler when it sees an instruction of the form “imm22-instruction ... %lm(pc-relative) ...”. This resembles **R_SPARC_PC22**, except it truncates rather than validates.

R_SPARC_GLOB_JMP: This relocation type resembles **R_SPARC_GLOB_DAT**, except that it is guaranteed to be associated with a procedure call and therefore the dynamic linker may evaluate the relocation lazily.

R_SPARC_7: This relocation type is used by the assembler for 7 bit software trap numbers.

Object Files Changes (64-bit gABI) - EXPERIMENTAL

#	Facility	Location	Description
1	Data Representation	gABI	Add to figure 4-2 (Page 4-3):

Figure 4-2: 64-Bit Data Types

Name	Size	Alignment	Purpose
Elf64_Addr	8	8	Unsigned program address
Elf64_Half	2	2	Unsigned small integer
Elf64_Off	8	8	Unsigned file offset
Elf64_Sword	4	4	Signed medium integer
Elf64_Sxword	8	8	Signed large integer
Elf64_Word	4	4	Unsigned medium integer
Elf64_Xword	8	8	Unsigned large integer

2 Data Representation gABI Add after the last line of the second paragraph (Page 4-3): Similarly, a structure containing an **Elf64_Addr** member will be aligned on a 8-byte boundary within the file.

3 ELF Headers gABI Add to Figure 4-3:

Figure 4-3: 64-bit ELF Header (ELFCLASS64)

```
typedef struct {
    unsigned char  e_ident[EI_NIDENT];
    Elf64_Half    e_type;
    Elf64_Half    e_machine;
    Elf64_Word    e_version;
    Elf64_Addr    e_entry;
    Elf64_Off     e_phoff;
    Elf64_Off     e_shoff;
    Elf64_Word    e_flags;
    Elf64_Half    e_ehsize;
    Elf64_Half    e_phentsize;
    Elf64_Half    e_phnum;
    Elf64_Half    e_shentsize;
    Elf64_Half    e_shnum;
    Elf64_Half    e_shstrndx;
} Elf64_Ehdr;
```

4 ELF Identification gABI add the following line to the end of the first paragraph on page 4-8: Class ELFCLASS64 supports machines with files and virtual address spaces up to 16 exabytes.

5 ELF Identification gABI Change last paragraph on page 4-8 to: A file's data encoding specifies how to interpret the basic objects in a file. As described above, ELF files use objects that occupy 1, 2, 4 and 8 bytes. Under defined encodings, objects are represented as shown below. Byte numbers appear in the upper left corners.

- 6 ELF Identification gABI Add the following data encoding to figure 4-5:

Figure 4-5: Data Encoding ELFDATA2LSB

0x0807060504030201	0 08	1 07	2 06	3 05	4 04	5 03	6 02	7 01
--------------------	---------	---------	---------	---------	---------	---------	---------	---------

- 7 ELF Identification gABI Add the following data encoding to figure 4-6:

Figure 4-6: Data Encoding ELFDATA2MSB

0x0102030405060708	0 01	1 02	2 03	3 04	4 05	5 06	6 07	7 08
--------------------	---------	---------	---------	---------	---------	---------	---------	---------

- 8 Sections gABI Change first line of page 4-10 to: An object file's section header table lets one locate all the file's sections. The section header table is an array of `Elf32_Shdr` or `Elf64_Shdr` structures as described below.
- 9 Sections gABI Add to Figure 4-8 page 4-12:

Figure 4-8: Section Header (ELFCLASS64)

```
typedef struct {
    Elf64_Word      sh_name;
    Elf64_Word      sh_type;
    Elf64_Xword     sh_flags;
    Elf64_Addr      sh_addr;
    Elf64_Off       sh_offset;
    Elf64_Xword     sh_size;
    Elf64_Word      sh_link;
    Elf64_Word      sh_info;
    Elf64_Xword     sh_addralign;
    Elf64_Xword     sh_entsize;
} Elf64_Shdr;
```

- 10 Symbol table gABI Add to Figure 4-15 page 4-22:

Figure 4-15: Symbol Table Entry (ELFCLASS64)

```
typedef struct {
    Elf64_Word      st_name;
    unsigned char   st_info;
    unsigned char   st_other;
    Elf64_Half      st_shndx;
    Elf64_Addr      st_value;
    Elf64_Xword     st_size;
} Elf64_Sym;
```

- 11 Relocation gABI Add to Figure 4-19 page 4-27:

Figure 4-19: Relocation Entries (ELFCLASS64)

```
typedef struct {
    Elf64_Addr  r_offset;
    Elf64_Xword r_info;
} Elf64_Rel;

typedef struct {
    Elf64_Addr  r_offset;
    Elf64_Xword r_info;
    Elf64_Sxword r_addend;
} Elf64_Rela;
```

- | | | | |
|----|------------|------|--|
| 12 | Relocation | gABI | Change last sentence under <code>r_info</code> page 4-27 to:
“When the text in the processor supplement refers to a relocation entry’s relocation type it means the result of applying <code>ELF32_R_TYPE</code> or <code>ELF64_R_TYPE</code> to an entry’s <code>r_info</code> member. When the text refers to a relocation entry’s symbol table index it means the result of applying <code>ELF32_R_SYM</code> or <code>ELF64_R_SYM</code> to the entry’s <code>r_info</code> member. |
| 13 | Relocation | gABI | Add the following (which apply to the 64-bit relocation types) to the figure at the top of page 4-28: |

```
#define ELF64_R_SYM(info)      (info)>>32)
#define ELF64_R_TYPE(info)    ((Elf64_Word)(info))
#define ELF64_R_INFO(sym,type) (((Elf64_Xword)(sym)<<32)+(Elf64_Xword)(type))
```

- | | | | |
|----|------------|------|--|
| 14 | Relocation | gABI | Change the first two sentences of the paragraph that begins “As shown above, only <code>ELF32_Rela</code> entries ...”, to:

As shown above, only <code>ELF32_Rela</code> and <code>ELF64_Rela</code> entries contain an explicit addend. Entries of type <code>Elf32_Rel</code> and <code>Elf64_Rel</code> store an implicit addend in the location to be modified. |
|----|------------|------|--|

Object Files (64-bit psABI) - EXPERIMENTAL

ELF Header

Machine Information

For file identification in *e_ident*, SPARC 64-bit ABI requires the following values.

Figure 4-1: SPARC 64bit Identification, *e_ident*

Position	Value
<i>e_ident</i> [EI_CLASS]	ELFCLASS64
<i>e_ident</i> [EI_DATA]	ELFDATA2MSB

Processor identification resides in the ELF header's *e_machine* member and must have the value 43, defined as the name **EM_SPARC64**. The ELF header's *e_flags* member holds bit flags associated with the file. The SPARC 64-bit ABI defines the following flags:

Figure 4-2: SPARC 64-bit Flags, *e_flags*

Name	Value	Meaning
EF_SPARC64_MM	0x3	Mask for Memory Model
EF_SPARC64_TSO	0x0	Total Store Ordering
EF_SPARC64_PSO	0x1	Partial Store Ordering
EF_SPARC64_RMO	0x2	Relaxed Memory Ordering
EF_SPARC_EXT_MASK	0xffff00	Reserved for vendor extensions
EF_SPARC_SUN_US1	0x000200	Sun UltraSPARC1 extensions
EF_SPARC_HAL_R1	0x000400	HAL R1 extensions
EF_SPARC_SUN_US3	0x000800	Sun UltraSPARC III extensions

All unspecified bits are reserved and should be set to zero. The compilation system sets the **EF_SPARC64_MM** field to the value required for the correct execution of the object. Typically, the programmer specifies what value to use for compiling a given source unit. **TSO** is the most restrictive memory model, followed by **PSO**, followed by **RMO**, in that order.

It is recommended that the default compilation model should be **RMO** to realize the performance advantages of this memory model. A binder that statically links input objects into a single output object will set **EF_SPARC64_MM** to the most restrictive model specified by any of the input objects.

At execution time, the dynamic linker will inform the operating system of the most restrictive model required by any of the objects that are part of the execution environment. The operating system will use this information to provide the memory order semantics of that model to the application, if available, or a more restrictive one.

The memory model flag expresses a requirement that the program has on the memory model semantics of the execution environment, but does *not* constrain the implementation in how it provides that model. For example, on a uniprocessor, the implementation can usually ignore the memory model flags, and set the processor into **RMO** mode because the program can only observe **TSO** memory ordering semantics.

Bits in **EF_SPARC_EXT_MASK** are assigned by SPARC International to vendors who implement user-accessible extensions to the *SPARC V9 instruction set*, and want to prevent accidental execution of binaries that use those extensions on machines that do not have them. The compilation system sets these bits according to the extensions used and the static linker propagates these bits into the objects it creates. The kernel and the dynamic linker cooperate to prohibit execution of objects that use specific extensions on systems that do not provide those extensions.

Note that objects that use vendor extensions are not portable and do not conform to the 64-bit ABI.

Sections

Special Section Indices

Two additional section index values are defined.

- **SHN_BEFORE** (0xff00) is used in conjunction with the **SHF_ORDERED** flag (see below).
- **SHN_AFTER** (0xff01) is used in conjunction with the **SHF_ORDERED** flag (see below).

Special Section Flags

Two new section flags are defined.

- The **SHF_EXCLUDE** flag specifies that the link editor is to exclude this section from executable and shared objects that it builds when those objects are not to be further relocated. **SHF_EXCLUDE** has the value 0x80000000.
- The **SHF_ORDERED** flag specifies that the *sh_link* and *sh_info* fields of the section header are to be interpreted specially (see below). **SHF_ORDERED** has the value 0x40000000.

For sections with the **SHF_ORDERED** flag set, a non-zero value of the *sh_link* field of the section header indicates that the data in this section should be combined into the section pointed at when the output file is constructed by the link editor (a section is allowed to point to itself). In the absence of alternate ordering information (see below), sections from a single object file collected into one section in the output shall be contiguous and have the same relative ordering as they did in the input file and the contributions from each input file shall appear in command-line order.

The **SHF_ORDERED** flag when applied to a set of sections all of whose *sh_link* fields point to the same section (within one object file) or to sections having the same name (across object files) specifies that the link editor is to sort the sections when it combines them (see above) based on the relative ordering in the output file of the sections pointed at by the *sh_info* field of each section in the set. In each original object file input to the link editor, the relative ordering shall be correct so that the link editor is not required to do any special processing unless it explicitly re-orders the sections which are the targets of the *sh_info* fields.

When used in conjunction with the **SHF_ORDERED** flag *sh_info* values of **SHN_BEFORE** and **SHN_AFTER** imply that those sections are to proceed or follow, respectively, all other sections in the set being ordered. File/command-line order is preserved when multiple sections in an ordered set each have the same *sh_info* value.

Special Sections

Various sections hold program and control information. Sections in the list below are used by the system and have the indicated types and attributes.

A special relocation section is used to initialize register symbols.

Figure 4-2: Special Sections

Name	Type	Attributes
<i>.rela</i> (*) <i>.got</i> <i>.plt</i>	SHT_RELA SHT_PROGBITS SHT_PROGBITS	None SHF_ALLOC + SHF_WRITE SHF_ALLOC + SHF_WRITE + SHF_EXECINSTR

- .rela** sh_link contains the section header index of the associated symbol table. sh_info is 0; this indicates that this section only contains register relocations. See **R_SPARC_REGISTER** in “Relocation Types” for more information.
(*) Note that the actual name doesn’t matter; it’s the type that counts.
- .got** This section holds the global offset table. See “Coding Examples” in Chapter 3 and “Global Offset Table” in Chapter 5 for more information
- .plt** This section holds the procedure linkage table. See “Procedure Linkage Table” in Chapter 5 for more information.

Symbol Table

Usage of a global register reserved to the application is indicated by a SPARC 64-bit specific Symbol Type.

Figure 4.2b: Symbol Type for Register Symbols

Name	Value
STT_SPARC_REGISTER	13

A symbol table entry for a register symbol contains the following:

Figure 4.2c: Symbol Table Entry

Field	Meaning
st_name	Index into the string table of the name of the symbol (*1).
st_value	Register number (*2).
st_size	unused (0).
st_info	ELF64_ST_INFO (bind, type) bind is typically STB_GLOBAL , but does reflect the actual declared scope of the name (that is, it could be STB_WEAK or STB_LOCAL). type must be STT_SPARC_REGISTER .
st_other	unused (0).
st_shndx	SHN_ABS if this object initializes this register symbol; SHN_UNDEF otherwise. (*3)

(*1) An index value of 0, which points to the null name in the string table, indicates that the register is used for scratch. A scratch register must have binding **STB_GLOBAL**.

(*2) Register numbers correspond to the assignments in *The SPARC Architecture Manual, Version 9* for integer registers. For example:

Figure 4.2d: Sample Register Numbers

Name	Value	Meaning
STO_SPARC_REGISTER_G2	0x2	%g2
STO_SPARC_REGISTER_G3	0x3	%g3

(*3) An initializer for a **SHN_ABS** register symbol is specified with a special register relocation type; See “Relocation Types” for details.

Absence of an entry for a particular global register means that the particular global register is not used at all by the object. An object that uses one or more of the application-reserved global registers but does not indicate this with an appropriate symbol-table entry does not conform to the 64-bit ABI. System objects may, but are not required to, use the same mechanism to indicate how they have used the global registers reserved for system software (%g6, %g7).

If an executable file contains a reference to a function defined in one of its associated shared objects, the symbol table section for that file will contain an entry for that symbol. The *st_shndx* member of that symbol table entry contains **SHN_UNDEF**. This informs the dynamic linker that the symbol definition for that function is not contained in the executable file itself. If that symbol has been allocated a procedure linkage table entry in the executable file, and the *st_value* member for that symbol table entry is non-zero, the value will contain the virtual address of the first instruction of that procedure linkage table entry. Otherwise, the *st_value* member contains zero. This procedure linkage table entry address is used by the dynamic linker in resolving references to the address of the function. See "Function Addresses" in Chapter 5 for details.

Relocation

The *r_info* field is composed of two 32-bit parts, the symbol table index and the relocation type. The relocation type on SPARC 64-bit ABI systems is further decomposed into an 8-bit type identifier and a 24-bit type dependent data field. For the existing ELF-32 relocation types, that data field is zero. New relocation types, however, may make use of these bits.

Figure 4-3a: Relocation Macros

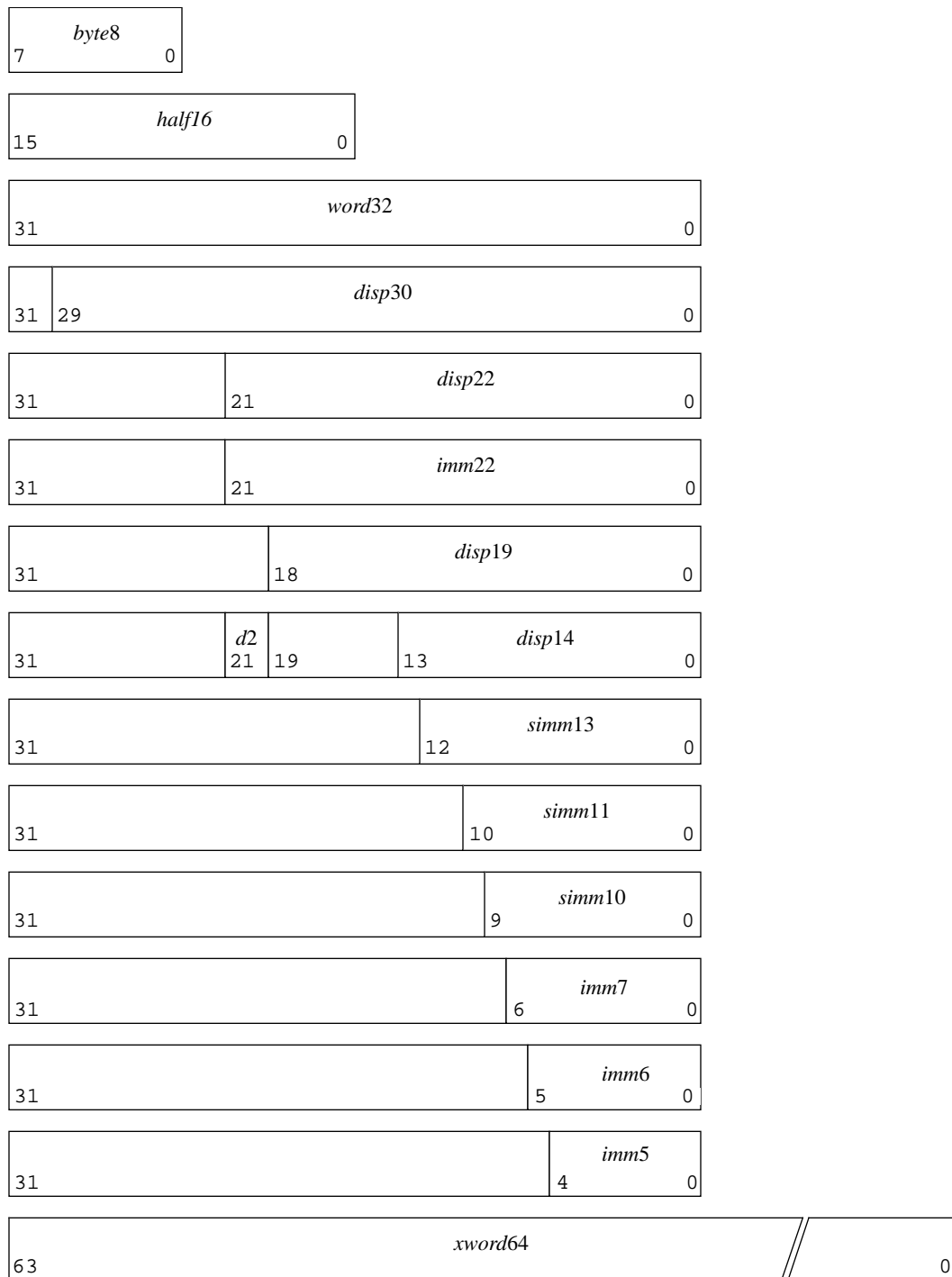
```
#define ELF64_R_TYPE_DATA(info)      (((Elf64_Xword)(info) << 32) >> 40)
#define ELF64_R_TYPE_ID(info)       (((Elf64_Xword)(info) << 56) >> 56)
#define ELF64_R_TYPE_INFO(data, type) (((Elf64_Xword)(data) << 8) + (Elf64_Xword)(type))
```

Relocation Types

An overview of the instruction and data formats from *The SPARC™ Architecture Manual, Version 9* makes relocation easier to understand. Relocation entries describe how to alter the following instruction and data fields (bit numbers

appear in the lower box corners).

Figure 4-3: Relocatable Fields



Calculations below assume the actions are transforming a relocatable file into either an executable or a shared object file. Conceptually, the link editor merges one or more relocatable files to form the output. It first decides how to combine and relocate the input files, then updates the symbol values, and finally performs the relocation. Relocations applied to executable or shared object files are similar and accomplish the same result. Descriptions below use the following

notation.

- A This means the addend used to compute the value of the relocatable field.
- B This means the base address at which a shared object has been loaded into memory during execution. Generally a shared object file is built with a 0 base virtual address, but the execution address will be different. See “Program Header” in the *System V ABI* for more information about base addresses.
- G This means the offset into the global offset table at which the address of the relocation entry’s symbol will reside during execution. See “Coding Examples” in Chapter 3 and “Global Offset Table” in Chapter 5 for more information.
- L This means the place (section offset or address) of the procedure linkage table entry for a symbol. A procedure linkage table entry redirects a function call to the proper destination. The link editor builds the initial procedure linkage table, and the dynamic linker modifies the entries during execution. See “Procedure Linkage Table” in Chapter 5 for more information.
- O This means the secondary addend used to compute the value of the relocation field. The secondary addend is extracted from the *r_info* field in the relocation entry by applying the **ELF64_R_TYPE_DATA** macro.
- P This means the place (section offset or address) of the storage unit being relocated (computed using *r_offset*).
- S This means the value of the symbol whose index resides in the relocation entry.

Relocation entries apply to bytes (*byte8*), halfwords (*half16*), extended-words, (*xword64*), or words (the others). In any case, the *r_offset* value designates the offset or virtual address of the first byte of the affected storage unit. The relocation type specifies which bits to change and how to calculate their values. SPARC 64-bit ABI uses only *ELF64_Rela* relocation entries with explicit addends. Thus the *r_addend* member serves as the relocation addend.

NOTE

Field names in the following tables tell whether the relocation type checks for “overflow”. A calculated relocation value may be larger than the intended field, and a relocation type may verify (V) the value fits or truncate (T) the result. As an example, *V-imm22* means the computed value may not have significant, non-zero bits outside the *imm22* field.

Figure 4-4: Relocation Types

Name	Value	Field	Calculation
R_SPARC_NONE	0	<i>none</i>	none
R_SPARC_8	1	<i>V-byte8</i>	S + A
R_SPARC_16	2	<i>V-half16</i>	S + A
R_SPARC_32	3	<i>V-word32</i>	S + A
R_SPARC_DISP8	4	<i>V-byte8</i>	S + A - P
R_SPARC_DISP16	5	<i>V-half16</i>	S + A - P
R_SPARC_DISP32	6	<i>V-disp32</i>	S + A - P
R_SPARC_WDISP30	7	<i>V-disp30</i>	(S + A - P) >> 2
R_SPARC_WDISP22	8	<i>V-disp22</i>	(S + A - P) >> 2
R_SPARC_HI22	9	<i>V-imm22</i>	(S + A) >> 10
R_SPARC_22	10	<i>V-imm22</i>	S + A
R_SPARC_13	11	<i>V-simm13</i>	S + A
R_SPARC_LO10	12	<i>T-simm13</i>	(S + A) & 0x3ff
R_SPARC_GOT10	13	<i>T-simm13</i>	G & 0x3ff
R_SPARC_GOT13	14	<i>V-simm13</i>	G
R_SPARC_GOT22	15	<i>T-imm22</i>	G >> 10
R_SPARC_PC10	16	<i>T-simm13</i>	(S + A - P) & 0x3ff
R_SPARC_PC22	17	<i>V-imm22</i>	(S + A - P) >> 10
R_SPARC_WPLT30	18	<i>V-disp30</i>	(L + A - P) >> 2
R_SPARC_COPY	19	<i>none</i>	none
R_SPARC_GLOB_DAT	20	<i>V-xword64</i>	S + A
R_SPARC_JMP_SLOT	21	<i>none</i>	see below
R_SPARC_RELATIVE	22	<i>V-xword64</i>	B + A
R_SPARC_UA32	23	<i>V-word32</i>	S + A
R_SPARC_PLT32	24	<i>V-word32</i>	L + A
R_SPARC_HIPLT22	25	<i>T-imm22</i>	(L + A) >> 10
R_SPARC_LOPLT10	26	<i>T-simm13</i>	(L + A) & 0x3ff
R_SPARC_PCPLT32	27	<i>V-disp32</i>	L + A - P
R_SPARC_PCPLT22	28	<i>V-disp22</i>	(L + A - P) >> 10
R_SPARC_PCPLT10	29	<i>V-simm13</i>	(L + A - P) & 0x3ff
R_SPARC_10	30	<i>V-simm10</i>	S + A
R_SPARC_11	31	<i>V-simm11</i>	S + A
R_SPARC_64	32	<i>V-xword64</i>	S + A
R_SPARC_OLO10	33	<i>V-simm13</i>	((S + A) & 0x3ff) + O
R_SPARC_HH22	34	<i>V-imm22</i>	(S + A) >> 42
R_SPARC_HM10	35	<i>T-simm13</i>	((S + A) >> 32) & 0x3ff
R_SPARC_LM22	36	<i>T-imm22</i>	(S + A) >> 10
R_SPARC_PC_HH22	37	<i>V-imm22</i>	(S + A - P) >> 42
R_SPARC_PC_HM10	38	<i>T-simm13</i>	((S + A - P) >> 32) & 0x3ff
R_SPARC_PC_LM22	39	<i>T-imm22</i>	(S + A - P) >> 10
R_SPARC_WDISP16	40	<i>V-d2/disp14</i>	(S + A - P) >> 2
R_SPARC_WDISP19	41	<i>V-disp19</i>	(S + A - P) >> 2
R_SPARC_7	43	<i>V-imm7</i>	(S + A) & 0x7f
R_SPARC_5	44	<i>V-imm5</i>	(S + A) & 0x1f
R_SPARC_6	45	<i>V-imm6</i>	(S + A) & 0x3f
R_SPARC_DISP64	46	<i>V-xword64</i>	S + A - P
R_SPARC_PLT64	47	<i>V-xword64</i>	L + A
R_SPARC_HIX22	48	<i>V-imm22</i>	((S + A) ^ 0xffffffff) >> 10
R_SPARC_LOX10	49	<i>T-simm13</i>	((S + A) & 0x3ff) 0x1c00
R_SPARC_H44	50	<i>V-imm22</i>	(S + A) >> 22
R_SPARC_M44	51	<i>T-imm10</i>	((S + A) >> 12) & 0x3ff
R_SPARC_L44	52	<i>T-imm13</i>	(S + A) & 0xff
R_SPARC_REGISTER	53	<i>V-xword64</i>	S + A
R_SPARC_UA64	54	<i>V-xword64</i>	S + A
R_SPARC_UA16	55	<i>V-half16</i>	S + A

Some relocation types have semantics beyond simple calculation.

R_SPARC_GOT10

This relocation type resembles **R_SPARC_LO10**, except it refers to the address of the symbol's global offset table entry and additionally instructs the link editor to build a global offset table.

R_SPARC_GOT13	This relocation type resembles R_SPARC_13 , except it refers to the address of the symbol's global offset table entry and additionally instructs the link editor to build a global offset table.
R_SPARC_GOT22	This relocation type resembles R_SPARC_22 , except it refers to the address of the symbol's global offset table entry and additionally instructs the link editor to build a global offset table.
R_SPARC_WPLT30	This relocation type resembles R_SPARC_WDISP30 , except it refers to the address of the symbol's procedure linkage table entry and additionally instructs the link editor to build a procedure linkage table.
R_SPARC_COPY	The link editor creates this relocation type for dynamic linking. Its offset member refers to a location in a writable segment. The symbol table index specifies a symbol that should exist both in the current object file and in a shared object. During execution, the dynamic linker copies data associated with the shared object's symbol to the location specified by the object.
R_SPARC_GLOB_DAT	This relocation type resembles R_SPARC_64 , except it is used to set a global offset table entry to the address of the specified symbol. The special relocation type allows one to determine the correspondence between symbols and global offset table entries.
R_SPARC_JMP_SLOT	The link editor creates this relocation type for dynamic linking. Its offset member gives a location of a procedure linkage table entry. The dynamic linker modifies the procedure linkage table entry to transfer control to the designated symbol's address [See "Procedure Linkage Table" in chapter 5].
R_SPARC_RELATIVE	The link editor creates this relocation type for dynamic linking. Its offset member gives a location within a shared object that contains a value representing a relative address. The dynamic linker computes the corresponding virtual address by adding the virtual address at which the shared object was loaded to the relative address. Relocation entries for this type must specify 0 for the symbol table index.
R_SPARC_UA32	This relocation type resembles R_SPARC_32 , except it refers to an unaligned word. That is the "word" to be relocated must be treated as four separate bytes with arbitrary alignment, not as a word aligned according to the architecture requirements.
R_SPARC_OLO10	This relocation type resembles R_SPARC_LO10 , except an extra offset is added to make full use of the 13-bit signed immediate field.
R_SPARC_HH22	This relocation type is used by the assembler when it sees an instruction of the form " <i>imm22-instruction ... %hh(absolute) ...</i> ".
R_SPARC_HM10	This relocation type is generated by the assembler when it sees an instruction of the form " <i>simm13-instruction ... %hm(absolute) ...</i> ".
R_SPARC_LM22	This relocation type is used by the assembler when it sees an instruction of the form " <i>imm22-instruction ... %lm(absolute) ...</i> ". This resembles R_SPARC_HI22 , except it truncates rather than validates.
R_SPARC_PC_HH22	This relocation type is used by the assembler when it sees an instruction of the form " <i>imm22-instruction ... %hh(pc-relative) ...</i> ".
R_SPARC_PC_HM10	This relocation type is generated by the assembler when it sees an instruction of the form " <i>simm13-instruction ... %hm(pc-relative) ...</i> ".
R_SPARC_PC_LM22	This relocation type is used by the assembler when it sees an instruction of the form " <i>imm22-instruction ... %lm(pc-relative) ...</i> ". This resembles R_SPARC_PC22 , except it truncates rather than validates.
R_SPARC_7	This relocation type is used by the assembler for 7 bit software trap numbers.

R_SPARC_HIX22	This relocation type is used with R_SPARC_LOX10 for executables that will be confined to the uppermost 4GB of the 64-bit address space. Similar to R_SPARC_HI22 , but supplies ones complement of linked value.
R_SPARC_LOX10	Used with R_SPARC_HIX22 . Similar to R_SPARC_LO10 , but always sets bits 10..12 of the linked value.
R_SPARC_H44	This relocation type is used by the assembler when it sees an instruction of the form “ <i>imm44-instruction ... %h44(absolute) ...</i> ”. For example, see the code in Section 3.5.1.6, Figure 3.35b.
R_SPARC_M44	This relocation type is generated by the assembler when it sees an instruction of the form “ <i>imm44-instruction ... %m44(absolute) ...</i> ”. For example, see the code in Section 3.5.1.6, Figure 3.35b.
R_SPARC_L44	This relocation type is used with the R_SPARC_H44 and R_SPARC_M44 relocation types to generate a 44-bit absolute addressing model. The assembler will generate this type when it sees an instruction of the form “ <i>imm44-instruction ... %l44(absolute) ...</i> ”. For example, see the code in Section 3.5.1.6, Figure 3.35b.
R_SPARC_REGISTER	This relocation type is used to initialize a register symbol. Its offset member contains the register number to be initialized. There must be a corresponding register symbol for this register of type SHN_ABS .
R_SPARC_UA64	This relocation type resembles R_SPARC_64 , except it refers to an unaligned extended word. That is the “ <i>xword</i> ” to be relocated must be treated as eight separate bytes with arbitrary alignment, not as an extended word aligned according to architecture requirements.
R_SPARC_UA16	This relocation type resembles R_SPARC_16 , except it refers to an unaligned halfword. That is the “ <i>halfword</i> ” to be relocated must be treated as two separate bytes with arbitrary alignment, not as a halfword aligned according to architecture requirements.

CHAPTER 5: Program Loading and Dynamic Linking

SCD
2.4

Program Loading & Dynamic Linking

Introduction

This chapter is split into a 32-bit ABI, a 64-bit gABI, and a 64-bit psABI section. Processor independent descriptions of program loading and linking for System V Release 4 may be found in Chapter 5 of the System V Application Binary Interface. Information specific to the 32-bit psABI may be found in Chapter 5 of the *System V Application Binary Interface, SPARC Processor Supplement*. On the other hand, unless explicitly stated, information in the 64-bit psABI section of this chapter is independent of *System V ABI* documentation.

Program Loading and Dynamic Linking Changes (32-bit ABI)

The following are changes to the *System V Application Binary Interface*, and the *System V Application Binary Interface, SPARC Processor Supplement* as reported to SPARC International.

#	Facility	Location	Description
1	LD_LIBRARY_PATH	gABI	Change the order of the first and the second bullets in page 5-20 such that the influence of LD_LIBRARY_PATH takes precedence over DT_RPATH specifications.
2	Dynamic Linking	gABI	Add a new third bullet to the entries on page 5-20: "DT_RPATH specifications influence search operations for their own DT_NEEDED objects. Each evaluation of a given object's set of DT_NEEDED specification uses <i>that object's</i> DT_RPATH. Thus, if an executable specifies a set of DT_NEEDED objects (e.g., a, b, and c) and a DT_RPATH specification of x:y, then the search for a, b, and c will involve the paths x and y. If, when later evaluating the DT_NEEDED object for a (e.g., d), then x and y will not be used for that search unless a also specifies a DT_RPATH containing them."
3	Initialization and Termination Functions	gABI	Add the following new third paragraph on page 5-22: "Initialization and Termination functions can expect to use all libsyes and libc ABI-defined services in their execution."
4	Shared Object Dependencies	gABI	In page 5-20, the gABI specifies in a "NOTE" that for set-user and set-group ID programs, LD_LIBRARY_PATH is ignored and DT_RPATH entries are used. This statement is incomplete. DT_RPATH entries should be used only to the extent that components beginning with "/" are acceptable (not relative path), in particular, relative path names are not used as these constitute a security hazard. Further, the prohibition against LD_LIBRARY_PATH is unnecessarily restrictive and conflicts with at least some widespread existing practice, in which those items contained in LD_LIBRARY_PATH which are also acceptable DT_RPATH entries or are "/usr/lib" are also used.
5	Dynamic linking	psABI	Add a section entitled "Dynamic Linker" as the first subsection of the "Dynamic Linking" section, which is: "The value of the program header element PT_INTERP in an ABI-conforming program is the reference name for the runtime linker. As a special case, the reference name for version 1 of the C library reference name is also accepted as a legitimate PT_INTERP specification."

Program Load. & Dyn. Linking (64-bit gABI) - EXPERIMENTAL

Program Loading & Dynamic Linking Changes

#	Facility	Location	Description
1	Program Header	gABI	<p>page 5-2, add to figure 5-1 the following header:</p> <pre> typedef struct { Elf64_Word p_type; Elf64_Word p_flags; Elf64_Off p_offset; Elf64_Addr p_vaddr; Elf64_Addr p_paddr; Elf64_Xword p_filesz; Elf64_Xword p_memsz; Elf64_Xword p_align; } Elf64_Phdr; </pre>
2	Dynamic Structure	gABI	<p>page 5-15, add to figure 5-9 the following structure:</p> <pre> typedef struct { Elf64_Xword d_tag; union { Elf64_Xword d_val; Elf64_Addr d_ptr; } d_un; } Elf64_Dyn; extern Elf64_Dyn _DYNAMIC[]; </pre>
3	Shared Object	gABI	<p>page 5-20: Add: A processor supplement may specify additional Dependencies environment variables to be used instead of or in addition to LD_LIBRARY_PATH.</p> <p>A processor supplement may specify additional or different default directories to be searched for shared objects.</p>

Program Load. & Dyn. Linking (64-bit psABI) - EXPERIMENTAL

Program Loading

As the system creates or augments a process image, it logically copies a file’s segment to a virtual memory segment. When—and if—the system physically reads the file depends on the program’s execution behavior, system load, etc. A process does not require a physical page unless it references the logical page during execution, and processes typically leave many pages unreferenced. Therefore delaying physical reads frequently obviates them, improving system performance. To obtain this efficiency in practice, executable and shared object files must have segment images whose file offsets and virtual addresses are congruent, modulo the page size. Virtual addresses and file offsets for SPARC 64-bit ABI segments are congruent modulo 1 MB (0x100000) or larger powers of 2. Because 1 MB is the maximum page size, the files will be suitable for paging regardless of physical page size.

Figure 5-1: Executable File

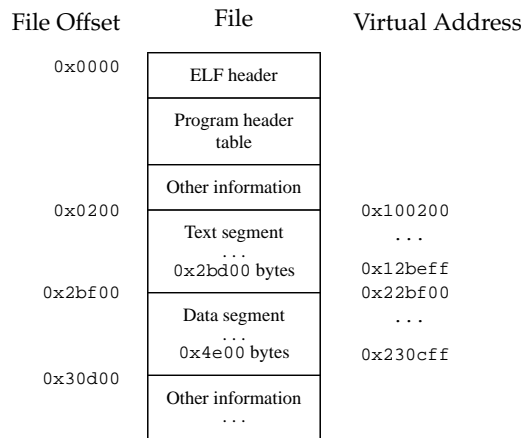


Figure 5-2: Program Header Segments

Member	Text	Data
p_type	PT_LOAD	PT_LOAD
p_offset	0x200	0x2bf00
p_vaddr	0x100200	0x22b000
p_paddr	unspecified	unspecified
p_filesz	0x2bd00	0x4e00
p_memsz	0x2bd00	0x6b24
p_flags	PF_R+PF_X	PF_R + PF_W + PF_X
p_align	0x100000	0x100000

Although the example’s file offsets and virtual addresses are congruent modulo 1MB for both text and data, up to four file pages hold impure text or data (depending on page size and file system block size).

- The first text page contains the ELF header, the program header table, and other information
- The last text page holds a copy of the beginning of data.
- The first data page has a copy of the end of text.
- The last data page may contain file information not relevant to the running process.

Logically, the system enforces the memory permissions as if each segment were complete and separate; segments’ addresses are adjusted to ensure each logical page in the address space has a single set of permissions. In the example

above, the region of the file holding the end of text and the beginning of data will be mapped twice: at one virtual address for text and at a different virtual address for data.

The end of the data segment requires special handling for uninitialized data, which the system defines to begin with zero values. Thus if a file's last data page includes information not in the logical memory page, the extraneous data must be set to zero, not the unknown contents of the executable file. "Impurities" in the other three pages are not logically part of the process image; whether the system expunges them is unspecified. The memory image for this program follows, assuming 4 KB (0x1000) pages.

Figure 5-3: Process Image Segments (4K page size)

Virtual Address	Contents	Segment
0x100000	Header padding 0x200 bytes	Text
0x100200	Text segment ... 0x2bd00 bytes	
0x12bf00	Data padding 0x100 bytes	
0x22b000	Text padding 0xf00 bytes	Data
0x22bf00	Data segment ... 0x4e00 bytes	
0x230d00	Page padding 0x300 zero bytes	
0x300000	Uninitialized data ... 0x1d24 bytes	Data
0x301d24	Page padding 0x2dc zero bytes	

Dynamic Section

Dynamic section entries give information to the dynamic linker. Some of this information is processor-specific, including the interpretation of some entries in the dynamic structure.

DT_PLTGOT On SPARC, this entry's *d_ptr* member gives the address of the first entry in the procedure linkage table. As described below, the first entry is special, and the dynamic linker must know its address.

DT_JMP_REL As explained in the *System V ABI*, this entry is associated with a table of relocation entries for the procedure linkage table. For the SPARC processor, this entry is mandatory both for executable and shared object files. Moreover, the relocation table's entries must have a one-to-one correspondence with the procedure linkage table. See "Procedure Linkage Table", below for more information.

Figure 5-5a: Dynamic Array Tags

Name	Value	d_un	Executable	Shared Object
DT_SPARC_REGISTER	0x7000001	d_val	optional	optional

DT_SPARC_REGISTER This element contains the index of an **STT_SPARC_REGISTER** symbol. There is one of these entries for every **STT_SPARC_REGISTER** symbol table entry in the symbol table.

Shared Object Dependencies

If the **LD_LIBRARY_PATH_64** variable is set in the process environment, its value is used (instead of the value of **LD_LIBRARY_PATH**) as an ordered list of directory locations to be searched for those shared objects upon which an application executable object depends. If the **LD_LIBRARY_PATH_64** variable is not set, the value of **LD_LIBRARY_PATH** (if set) is used. **LD_LIBRARY_PATH_64** follows the same syntactic convention as for **LD_LIBRARY_PATH** (i.e. it is a colon-separated ordered list of directory names).

In the SPARC 64-bit ABI (i.e. when the loading and dynamic linking of object files of **ELFCLASS64** takes place), the default (and last) directory location searched in order to locate shared objects needed by an executable object is */usr/lib/sparcv9*. The directory */usr/lib*, which is used as the default and last search location in the SPARC 32-bit ABI (i.e. when the runtime loading and linking of **ELFCLASS32** object files takes place) is not searched to locate shared objects in the SPARC 64-bit ABI.

Global Offset Table

Position-independent code cannot, in general, contain absolute virtual addresses. Global offset tables hold absolute addresses in private data, thus making the addresses available without compromising the position-independence and sharability of a program's text. A program references its global offset table using position-independent addressing and extracts absolute values, thus redirecting position-independent references to absolute locations.

Initially, the global offset table holds information as required by its relocation entries [see "Relocation" in chapter 4]. After the system creates memory segments for a loadable object file, the dynamic linker processes the relocation entries, some of which will be type **R_SPARC_GLOB_DAT** referring to the global offset table. The dynamic linker determines the associated symbol values, calculates their absolute addresses, and sets the appropriate memory table entries to the proper values. Although the absolute addresses are unknown when the link editor builds an object file, the dynamic linker knows the addresses of all memory segments and can thus calculate the absolute addresses of the symbols contained therein.

If a program requires direct access to the absolute address of a symbol, that symbol will have a global offset table entry. Because the executable file and shared objects have separate global offset tables, a symbol's address may appear in several tables. The dynamic linker processes all the global offset table relocations before giving control to any code in the process image, thus ensuring the absolute addresses are available during execution.

The table's entry zero is reserved to hold the address of the dynamic structure, referenced with the symbol **_DYNAMIC**. This allows a program, such as the dynamic linker, to find its own dynamic structure without having yet processed its relocation entries. This is especially important for the dynamic linker, because it must initialize itself without relying on other programs to relocate its memory image.

The system may choose different memory segment addresses for the same shared object in different programs; it may even choose different library addresses for different executions of the same program. Nonetheless, memory segments do not change addresses once the process image is established. As long as a process exists, its memory segments reside at fixed virtual addresses.

A global offset table's format and interpretation are processor-specific. For 64-bit ABI, the symbol `_GLOBAL_OFFSET_TABLE_` may be used to access the table.

Figure 5-5: Global Offset Table

```
extern Elf64_Addr    _GLOBAL_OFFSET_TABLE[];
```

The symbol `_GLOBAL_OFFSET_TABLE_` may reside in the middle of the `.got` section, allowing both negative and non-negative "subscripts" into the array of addresses.

Function Addresses

References to the address of a function from an executable file and the shared objects associated with it might not resolve to the same value. References from within shared objects will normally be resolved by the dynamic linker to the virtual address of the function itself. References from within the executable file to a function defined in shared object will normally be resolved by the link editor to the address of the procedure linkage table entry for that function within the executable file.

To allow comparisons of function addresses to work as expected, if an executable file references a function defined in a shared object, the link editor will place the address of the procedure linkage table entry for that function in its associated symbol table entry. [See "Symbol Values" in Chapter 4]. The dynamic linker treats such symbol table entries specially. If the dynamic linker is searching for a symbol, and encounters a symbol table entry for that symbol in the executable file, it normally follows the rules below.

1. If the `st_shndx` member of the symbol table entry is not `SHN_UNDEF`, the dynamic linker has found a definition for the symbol and uses its `st_value` member as the symbol's address
2. If the `st_shndx` member is `SHN_UNDEF` and the symbol is of type `STT_FUNC` and the `st_value` member is not zero, the dynamic linker recognizes this entry as special and uses the `st_value` member as the symbol's address.
3. Otherwise, the dynamic linker considers the symbol to be undefined within the executable file and continues processing.

Some relocation are associated with procedure linkage table entries. These entries are used for direct function calls rather than for references to function addresses. These relocations are not treated in the special way described above because the dynamic linker must not redirect procedure linkage table entries to point to themselves.

Procedure Linkage Table

Much as the global offset table redirects position-independent address calculations to absolute locations, the procedure linkage table redirects position-independent function calls to absolute locations. The link editor cannot resolve

execution transfers (such as function calls) from one executable or shared object to another. Consequently, the link editor arranges to have the program transfer control to entries in the procedure linkage table. For the 64-bit ABI, procedure linkage tables reside in private data. The dynamic linker determines the destinations' absolute addresses and modifies the procedure linkage table's memory image accordingly. The dynamic linker thus can redirect the entries without compromising the position-independence and sharability of the program's text. Executable files and shared object files have separate procedure linkage tables.

The first four procedure linkage table entries are reserved. (The original contents of these entries are unspecified, despite the example below.) Each of the first 32,768 entries in the table occupies 8 instructions (32 bytes) and must be aligned on a 32-byte boundary (the table as a whole must be aligned on a 256-byte boundary). In the unlikely event that more than 32,764 entries are needed, the remaining entries consist of 6 instructions (24 bytes) and 1 pointer (8 bytes). The instructions are collected together in blocks of 160 entries followed by 160 pointers. (The last group of entries and pointers may contain less than 160 items. No padding is required.)

NOTE The numbers 32,768 and 160 are based on the limits of branch and load displacements respectively with the second rounded down to make the divisions between code and data fall on 256-byte boundaries so as to improve cache performance.

Figure 5-6a below shows three of the initial 32,768 entries together with possible resolved forms that might apply if the target address was within +/- 2 Gb of the entry, within the lower 4 Gb of the address space, or anywhere respectively. The 64-bit ABI specifies the code sequence in the file. However, the only requirement placed on the resolved form is that the first instruction has no effect on the execution of the subsequent instructions of the entry. Figure 5-6b below shows 2 of

Figure 5-6a: Procedure Linkage Table - Example Early Entries

File	Memory Segment
.PLT101: sethi (. - .PLT0), %g1 ba,a %xcc,.PLT1 nop nop nop nop nop nop nop	.PLT101 nop mov %o7, %g1 call name101 mov %g1, %o7 nop nop nop
.PLT102: sethi (. - .PLT0), %g1 ba,a %xcc,.PLT1 nop nop nop nop nop nop	.PLT102: nop sethi %hi(name102), %g1 jmpl %g1+%lo(name102), %g0 nop nop nop
.PLT103: sethi (. - .PLT0), %g1 ba,a %xcc,.PLT1 nop nop nop nop nop nop	.PLT103: nop sethi %hh(name103), %g1 sethi %lm(name103), %g5 or %hm(name103), %g1 sllx %g1, 32, %g1 or %g1, %g5, %g5 jmpl %g5+%lo(name103), %g0 nop

the later entries. In this case, both the file and resolved forms (which differ only in the value of the pointer) are defined by the 64-bit ABI.

Figure 5-6b: Procedure Linkage Table - Example Later Entries

File	Memory Segment
.PLT32768: mov %o7, %g5 call . + 8 nop	.PLT32768: <unchanged> <unchanged> <unchanged>

Figure 5-6b: Procedure Linkage Table - Example Later Entries

File	Memory Segment
ldx [%07+.PLTP32768-(.PLT32768+4)], %g1	<unchanged>
jmp1 %07+%g1, %g1	<unchanged>
mov %g5, %07	<unchanged>
.....
.PLT32927:	.PLT32927:
mov %07, %g5	<unchanged>
call . + 8	<unchanged>
nop	<unchanged>
ldx [%07+.PLTP32927-(.PLT32927+4)], %g1	<unchanged>
jmp1 %07+%g1, %g1	<unchanged>
mov %g5, %07	<unchanged>
.PLTP32768:	.PLTP32768:
.xword .PLT0-(.PLT32768+4)	.xword name32768-(.PLT32768+4)
.....
.PLTP32927:	.PLTP32927:
.xword .PLT0-(.PLT32927+4)	.xword name32927-(.PLT32927+4)

Figure 5-6c below shows an example (not required by the 64-bit ABI) of the layout of the reserved entries at the beginning of the table.

Figure 5-6c: Procedure Linkage Table - Reserved Entries

File	Memory Segment
.PLT0:	.PLT0:
illtrap 0	save %sp, -176, %sp
illtrap 0	sethi %hh(dynamic_linker_0), %10
illtrap 0	sethi %lm(dynamic_linker_0), %11
illtrap 0	or %10, %hm(dynamic_linker_0), %10
illtrap 0	sllx %10, 32, %10
illtrap 0	or %10, %11, %10
illtrap 0	jmp1 %10+%10(dynamic_linker_0),%01
illtrap 0	mov %g1, %00
.PLT1	.PLT1:
illtrap 0	save %sp, -176, %sp
illtrap 0	sethi %hh(dynamic_linker_1), %10
illtrap 0	sethi %lm(dynamic_linker_1), %11
illtrap 0	or %10, %hm(dynamic_linker_1), %10
illtrap 0	sllx %10, 32, %10
illtrap 0	or %10, %11, %10
illtrap 0	jmp1 %10+%10(dynamic_linker_1),%01
illtrap 0	mov %g1, %00
.PLT2	.PLT2:
illtrap 0	.xword identification
.....

In this example the two entry points to the dynamic linker receive arguments as if they were normal C procedures. There are two entry points because the Procedure Linkage Table may contain two different kinds of unresolved entries. The unresolved form of any of the first 32,768 entries must branch to **.PLT1** and thus is in this example resolved by:

```
dynamic_linker_1(long x, long y)
```

which computes the table index as:

```
n = x >> 15;
```

The unresolved form of any later entries must jump to **.PLT0** and thus is in this example resolved by:

```
dynamic_linker_0(long x, long y)
```

which computes the table index as

```
i = x - y - 1048596;
n = 32768 + (i/5120)*160 + (i%5120)/24;
```

As mentioned before, a relocation table entry is associated with the procedure linkage table. The **DT_JMP_REL** entry in the **_DYNAMIC** array gives the location of the first relocation entry. The relocation table's entries parallel the procedure

linkage table in a one-to-one correspondence. That is, relocation table entry 0 applies to procedure linkage table entry 0, and so on.

For slots 4 through 32,767, the relocation type will be `R_SPARC_JMP_SLOT`, the relocation offset will specify the address of the first byte of the associated table entry, the addend field will be zero, and the symbol table index will reference the appropriate symbol. For slots 32,768 and beyond, the relocation type will be `R_SPARC_JMP_SLOT`, the relocation offset will specify the address of the first byte of the associated pointer, the addend field will contain the (un-relocated) value $-(\text{pltN}+4)$, and the symbol table index will reference the appropriate symbol.

NOTE

The pointer value generated by a `R_SPARC_JMP_SLOT` relocation with a non-zero addend is given by the formula

$$S + A - B$$

Following the steps below, the dynamic linker and the program “cooperate” to resolve symbolic references through the global offset table and the procedure linkage table. Again, the steps described below are for explanation only. The precise execution-time behavior of the dynamic linker is not specified.

1. When first creating the memory image of the program, the dynamic linker changes the initial procedure linkage table entries, making them transfer control to one of the dynamic linker’s own routines as described above. It also stores an extended word of *identification* information in the third entry. When it receives control, it can examine this extended word to determine what object called it.
2. All other procedure linkage table entries initially transfer to the first or second entry. Those entries establish a stack frame and call the dynamic linker.
3. Using the *identification* value, the dynamic linker finds its data structures associated with the object in question, including the relocation table.
4. The dynamic linker computes the index of the relocation entry for the table slot.
5. Knowing this, the dynamic linker finds the symbols “real” value, unwinds the stack, modifies the procedure linkage table entry, and transfers control to the desired destination.

Whenever the dynamic linker is modifying the instructions of a procedure table entry it must do so “carefully”.

- To make the code re-entrant, the procedure linkage table’s instructions must be changed in a particular sequence. That is, if the dynamic linker is “resolving” a function’s procedure linkage table entry and a signal arrives, the signal handling code must be able to call the original function with predictable (and correct) results.
- The dynamic linker may change up to eight words to convert an entry. Since it can update only a single word atomically with regard to instruction execution, re-entrancy must be achieved by first overwriting the `nop` instructions with their replacement instructions and then patching the `ba, a` (and the `sethi` instruction if using a 64-bit ABI store). If a re-entrant function call occurs just prior to the last patch, the dynamic linker gains control a second time. Although both invocations of the dynamic linker modify the same procedure linkage table entry, their changes do not interfere with each other.
- If the initial `sethi` instruction is changed, it can only be replaced by a `nop`.

Changing the pointer for the second form of entry is done using a single atomic 64-bit store.

The `LD_BIND_NOW` environment variable can change dynamic linking behavior. If its value is non-null, the dynamic linker evaluates all global offset table and procedure linkage table entries before transferring control to the program. That is, the dynamic linker processes relocation entries of type `R_SPARC_JMP_SLOT` during process initialization. Otherwise, the dynamic linker has the option of evaluating these entries lazily, delaying symbol resolution and relocation until the first execution of the related function.

NOTE

Lazy binding generally improves overall application performance, because unused symbols do not incur the dynamic linking overhead. Nevertheless, two situations make lazy binding undesirable for some applications. First, the initial reference to a shared object function takes longer than subsequent calls, because the dynamic linker intercepts the call to resolve the symbol. Some applications cannot tolerate this unpredictability. Second, if an error occurs and the dynamic linker cannot resolve the symbol, the dynamic linker will terminate the program. Under lazy binding, this might occur at arbitrary times. Once again, some applications cannot tolerate this unpredictability. By turning off lazy binding, the dynamic linker forces the failure to occur during process initialization, before the application receives control.

Program Interpreter

There is one (and only one) valid program interpreter for programs conforming to the SPARC 64-bit ABI. It is:

```
/usr/lib/sparcv9/ld.so.1
```

SCD

2.4

Libraries

Introduction

This chapter describes the libraries making up the SPARC 32-bit ABI, and those constituting the SPARC 64-bit ABI, and then describes the interface sets represented by each of those libraries. In addition, this chapter provides overviews of certain aspects of the ABI which either span several libraries, or relate to the management of the library interfaces in general.

This chapter is split into three sections: a common section, followed by a 64-bit gABI section, and a 64-bit psABI section. The common section applies to both the 32-bit ABI and 64-bit ABI, except where explicitly noted otherwise. Processor independent information relevant to libraries for System V Release 4 may be found in Chapter 6 of the *System V Application Binary Interface, SPARC Processor Supplement*. Information relevant to libraries for the 32-bit psABI and 64-bit psABI may be found in Chapter 6 of the *System V Application Binary Interface, SPARC Processor Supplement*. Windowing and Terminal Interface libraries are defined in detail in Chapter 10. All other libraries are defined in detail in this chapter.

Individual sections in this chapter list changes to the base System V ABI documents, give a large files support interfaces overview (32-bit ABI only), and define non-windowing and terminal interface libraries.

This chapter provides the following overviews:

- Support for Large Files (32-bit ABI)
- SCD “native” threads vs. POSIX threads
- Conventions and Techniques for Library Versioning

The libraries corresponding to the 32-bit and 64-bit SPARC ABIs are as follows:

Libraries constituting the SPARC 32-bit ABI:

- | | |
|---|--------------------------------|
| • Asynchronous I/O Library | <i>libaio (32-bit ABI)</i> |
| • C Library | <i>libc (32-bit ABI)</i> |
| Includes what was previously known as the System Library ¹ — | |
| <i>libsys (32-bit ABI)</i> | |
| • Dynamic Shared Object Handling Interfaces | <i>libdl (32-bit ABI)</i> |
| • ELF Object Utilities Interfaces | <i>libelf (32-bit ABI)</i> |
| • Internationalization Interfaces | <i>libintl (32-bit ABI)</i> |
| • Math Library | <i>libm (32-bit ABI)</i> |
| • NIS DataBase library | <i>libnisdb (32-bit ABI)</i> |
| • Network Services Library | <i>libnsl (32-bit ABI)</i> |
| • POSIX real-time and aio interfaces (POSIX 1003.4) | <i>libposix4 (32-bit ABI)</i> |
| • POSIX threads interfaces (POSIX 1003.1a) | <i>libpthread (32-bit ABI)</i> |
| • The Domain Name Service Interfaces | <i>libresolv(32-bit ABI)</i> |

1. In version 2.3 of the SCD (and earlier specifications), a proper subset of interfaces present in libc was also offered via the libsys library. It has been observed that no real applications have exploited this more minimal low-level C runtime support interface (preferring to use the full libc), and therefore libsys as a library interface distinct from libc has been removed.

- The RPC Services library *librpcsvc (32-bit ABI)*
- Socket Library *libsocket (32-bit ABI)*
- Multithreading Library *libthread (32-bit ABI)*
- UCB BSD dynamic-compatibility interfaces *libucb (32-bit ABI)*
 - Provides runtime compatibility for earlier applications which use dynamic binding to the University of California Berkeley's BSD-system programming interface
- Multi-byte Characters (wide char) Interfaces *libw (32-bit ABI)*

Libraries constituting the SPARC 64-bit ABI:

- Asynchronous I/O Library *sparcv9/libaio (64-bit ABI)*
- C Library *sparcv9/libc (64-bit ABI)*
- Dynamic (64-bit) Shared Object Handling Interfaces *sparcv9/libdl (64-bit ABI)*
- ELF 64-bit Object Utilities Interfaces *sparcv9/libelf(64-bit ABI)*
- Math Library *sparcv9/libm (64-bit ABI)*
- Network Services Library *sparcv9/libnsl (64-bit ABI)*
- POSIX real-time and aio interfaces (POSIX 1003.4) *sparcv9/libposix4 (64-bit ABI)*
- POSIX threads interfaces (POSIX 1003.1a) *sparcv9/libpthread (64-bit ABI)*
- The Domain Name Service Interfaces *sparcv9/libresolv(64-bit ABI)*
- Socket Library *sparcv9/libsocket (64-bit ABI)*
- Multithreading Library *sparcv9/libthread (64-bit ABI)*
- Multi-byte Characters (wide char) Interfaces *sparcv9/libw (64-bit ABI)*

Some of the entries in the tables which define the function interfaces provided by various libraries have a superscript. All entries with a superscript have an entry in the changes table describing differences between the SCD definition and the System V gABI, psABI, or *System V Interface Definition, Third Edition* definition of the function.

The first part of this chapter is the changes to the *System V Application Binary Interface, SPARC Processor Supplement*, and the *System V Interface Definition* as reported to SPARC International.

C library Changes

#	Facility	Location	Description
1	<code>__dtou</code>	psABI	Change - On page 6-6, replace the description of exceptions for <code>__dtou</code> with “If $-2^{31} \leq a < 2^{32}$ then the operation is successful. If a is not a whole number, the inexact exception is raised. Otherwise, the value returned by <code>__dtou</code> is unspecified, and the invalid exception is raised. Note that negative values of a , in a successful operation, are first converted to integer and then cast to an unsigned integer.”
2	<code>__ftou</code>	psABI	Change - On page 6-7, replace the description of exceptions for <code>__ftou</code> with “If $-2^{31} \leq a < 2^{32}$ then the operation is successful. If a is not a whole number, the inexact exception is raised. Otherwise, the value returned by <code>__ftou</code> is unspecified, and the invalid exception is raised. Note that negative values of a , in a successful operation, are first converted to integer and then cast to an unsigned integer.”
3	<code>_Q_qtou</code>	psABI	Change - On page 6-5, replace the description of exceptions for <code>_Q_qtou</code> with “If $-2^{31} \leq a < 2^{32}$ then the operation is successful. If a is not a whole number, the inexact exception is raised. Otherwise, the value returned by <code>_Q_qtou</code> is unspecified, and the invalid exception is raised. Note that negative values of a , in a successful operation, are first converted to integer and then cast to an unsigned integer.”
4	<code>_environ</code>	gABI	Addition - On page 6.6, add the symbol <code>_environ</code> to Figure 6-5.
5	Additional Entry Points	psABI	Page 6-5 of the <i>System V Application Binary Interface</i> states “ABI-conforming systems must provide a <code>libs</code> entry point for each of [<code>fstat</code> , <code>lstat</code> , <code>mknod</code> , <code>stat</code> , and <code>uname</code>]. The name and syntax of [these entry points] may be the same as those characteristics of the source-level service or they may vary across processor architectures. The actual names of the entry points are specified in each processor’s supplement to the ABI, together with the entry points’ syntax information if names differ from those of the source-level services.” The <i>System V Application Binary Interface, SPARC Processor Supplement</i> (psABI) is missing the required specification. A section titled <i>Additional Entry Points (Processor -Specific)</i> should be added to the beginning of chapter 6 of the psABI which states “The binary entry points for <code>fstat</code> , <code>lstat</code> , <code>mknod</code> , <code>stat</code> , <code>uname</code> exist with these names and with the same calling sequence as described in their source-level interface. Synonyms exist for each of these entry points.”
6	<code>errno</code>	gABI	Addition - On page 6-6, add the symbol <code>errno</code> to Figure 6-5.
7	<code>fcntl(BA_OS)</code>	SVID, Vol. 1	Add a description of the command <code>F_FREESP</code> which reads: “Free storage space associated with a section of the ordinary file <i>fil</i> . The section is specified by a variable of data type <code>struct flock</code> pointed to by the third argument <i>arg</i> . <i>l_whence</i> is <code>SEEK_SET</code> , <code>SEEK_CUR</code> , or <code>SEEK_END</code> to indicate that the relative offset <i>l_start</i> will be measured from the start of the file, the current position, or the end of the file, respectively. <i>l_start</i> is the offset from the position specified in <i>l_whence</i> . <i>l_len</i> is the size of the section. An <i>l_len</i> of 0 frees up to the end of the file; in this case, the end of file (i.e., file size) is set to the beginning of the section freed. Any data previously written into this section is no longer accessible.”
8	<code>fcntl(BA_OS)</code>	SVID, Vol. 1	Change - The <code>EAGAIN</code> error return value only applies to files for which mandatory locking is enabled.

9	<i>getcwd</i> (BA_OS)	SVID, Vol.1	Change - For POSIX conformance, the type of the second argument size should be <code>size_t</code> rather than <code>int</code> .
10	<i>getgrent</i> (BA_LIB)	gABI	Add the functions <i>getgrent</i> , <i>setgrent</i> , <i>endgrent</i> and <i>fgetgrent</i> to Figure 6-2 on page 6-4.
11	<i>getgrent</i> (BA_LIB)	SVID, Vol. 1	The description that the information in the group structure comes from the <i>/etc/group</i> file is too restrictive; the information may come from other sources. These sources are collectively called "group database". Applications should not depend on the implementation of the group database.
12	<i>getpwent</i> (BA_LIB)	gABI	Add the functions <i>getpwent</i> , <i>setpwent</i> , <i>endpwent</i> and <i>fgetpwent</i> to Figure 6-2 on page 6-4.
13	<i>getpwent</i> (BA_LIB)	SVID, Vol. 1	The description that the information in the passwd structure comes from <i>/etc/passwd</i> file is too restrictive; the information may come from other sources. These sources are collectively called "user database". Applications should not depend on the implementation of the user database.
14	Global Data Symbols	gABI	Change the description of <code>_altzone</code> . Replace " <i>tzset</i> (BA_LIB)" with " <i>tzset</i> ()". See <i>ctime</i> (BA_LIB)."
15	<i>mmap</i> (KE_OS)	SVID, Vol. 1	Add to the paragraph which begins "Not all implementations..." insert "No implementation will permit an access to succeed where <code>PROT_NONE</code> has been set." after "... where <code>PROT_WRITE</code> has not been set."
16	<i>read</i> , <i>readv</i> (BA_OS)	SVID, Vol. 1	Addition - The SVID specifies that the write, <i>writew</i> (BA_OS) length of the struct <i>iov</i> [] in calls to <i>readv</i> ()/ <i>writew</i> () must be in the range <code>0 =< iovcnt =< IOV_MAX</code> . However, <code>IOV_MAX</code> is never defined. SCD compliant systems will support a minimum of 16 elements in a struct <i>iov</i> [].
17	<i>rename</i>	gABI	Change - On page 6-4, move <i>rename</i> from Figure 6-2 to Figure 6-3.
18	<i>sbrk</i>	SVID, Vol. 1	Add description of the function <i>sbrk</i> . See the man page for this function in the SCD 2.4 Interface Semantics.
19	<i>sbrk</i>	gABI	Add the function <i>sbrk</i> to Figure 6-2 on page 6-4.
20	<i>symlink</i> (BA_OS)	SVID, Vol. 1	Change description of <code>ENAMETOOLONG</code> to "if the length of <i>path2</i> exceeds <code>{PATH_MAX}</code> , or pathname component of <i>path2</i> is longer than <code>{NAME_MAX}</code> while <code>{_POSIX_NO_TRUNC}</code> is in effect."
21	<i>system</i> (BA_OS)	SVID, Vol.1	Change - For POSIX conformance, <i>system</i> () will ignore <code>SIGINT</code> and <code>SIGQUIT</code> , and block <code>SIGCHLD</code> while waiting for the command it invokes to terminate. Receipt of these signals will not result in <i>system</i> () returning with a <code>-1</code> result and with <code>errno</code> set to <code>EINTR</code> .
22	<i>waitid</i> (BA_OS)	SVID, Vol. 1	Change - The flag <code>WTRACED</code> should be replaced with <code>WTRAPPED</code> .
23	<i>crypt</i> (BA_LIB)	gABI	Add the function <i>crypt</i> to Figure 6-7 on page 6-10.
24	<i>crypt</i> (BA_LIB)	gABI	Add the function <i>encrypt</i> to Figure 6-7 on page 6-10.
25	<i>crypt</i> (BA_LIB)	gABI	Add the function <i>setkey</i> to Figure 6-7 on page 6-10.
26	<i>fdopen</i> (BA_OS)	SVID, Vol. 1	Change - The requirement that the <i>fildev</i> argument be open is incorrect.
27	<i>getitimer</i> (RT_OS)	SVID, Vol.3	Change - The description of canonical form is incorrect. The

- microsecond value can be zero. Hardware platforms must provide at least 60 Hz resolution. Platforms may provide greater than 60 Hz resolution, but applications that rely on a faster clock will not be portable.
- 28 *getitimer(RT_OS)* gABI Add the function setitimer to Figure 6-7 on page 6-10.
- 29 *gettimeofday(RT_OS)* gABI Add the function gettimeofday to Figure 6-7 on page 6-10.
- 30 *lockf(BA_OS)* SVID, Vol. 1 Addition - The **EAGAIN** error return value only applies to files for which mandatory locking is enabled.
- 31 *sysinfo* gABI Add the function sysinfo to Figure 6-7 on page 6-1
- 32 *termios(BA_OS)* SVID, Vol.1 Change - On page 6-152, in the description of tcsendbreak(), “If duration is not zero, zero-valued bits are not transmitted” is incorrect. Zero valued bits will be sent for implementation dependent period of time.
- 33 *elf_hash* gABI p5-22 Figure 5-12: Hashing Function The *elf_hash(3E)* routine has ‘unsigned long’ as a return type, and two ‘unsigned long’ variables are used as automatic variables. Change: These should all be ‘unsigned int’. [64 bit sparcs]
- 34 *scalb(BA_LIB)* gABI Move the function logb from Figure 6-6 on page 6-9 to Figure 6-7 on page 6-10. logb is non-ANSI function.
- 35 *signal.h* psABI change structure sigaltstack figure 6-33 page 6-41 from
- ```

struct sigaltstack {
 char *ss_sp;
 int ss_size;
 int ss_flags;
};

```
- to
- ```

struct sigaltstack {
    void                    *ss_sp;
    size_t                  ss_size;
    int                     ss_flags;
};

```
- 36 *signal.h* SVID, Vol.1 Delete *abort()*, *exit()*, and *longjmp()* from the list on page 5-68.

Rationale:

Async-signal-safe function list in SVID3 is different from XPG4.2/XPG5/POSIX. The three functions above are only on SVID3 and also not on both UXP/DS and Solaris2.5.1 manuals. So these differences could be a problem for applications. the Async-signal-safe function list for SVID3 is different than that for XPG4.2/XPG5/POSIX.

From POSIX Part 1, section 3.3.1.3, page 78, : “All POSIX.1 functions not in the preceding table and all functions defined in the C Standard not stated to be callable from a signal-catching function are considered to be unsafe with respect to signals. In the presence of signals, all functions defined by this part of ISO/IEC 9945 or by the C Standard shall behave as defined (by the defining standard) when called from or interrupted by a signal-catching function, with a single exception: when a signal interrupts an unsafe function and the

signal-catching function calls an unsafe function, the behavior is undefined." The language from XPG5 (under sigaction of volume 2) is basically a subset of this and includes a similar table, with *abort()*, *exit()* and *longjmp()* removed from the table.

37 *stdlib.h*

psABI

add to figure 6-40 (page 6-48), definition of *wchar_t*:

```
typedef long wchar_t;
```

Network Services Library Changes

#	Facility	Location	Description
1	netconfig(RS_ENV)	SVID, Vol. 3	Change - On page 17-20 the type declaration of <i>nc_flag</i> should be changed from <i>char *</i> to <i>unsigned long</i> .
2	rpc_broadcast_exp	gABI	Add the function <i>rpc_broadcast_exp</i> to Figure 6-11 on page 6-13.
3	rpc_clnt_calls(RS_LIB)	SVID, Vol. 3	Change - On page 18-11 the function prototype of <i>rpc_call()</i> should be: <pre>rpc_call (char *host, u_long prognum, u_long versnum, u_long procnum, xdrproc_t inproc, char *in, xdrproc_t outproc, char *out, char *nettype)</pre>
4	rpc_svc_err(RS_LIB)	SVID, Vol. 3	Change - Description of the function <i>svcerr_progvers()</i> is missing its last two arguments in the function prototype. Prototype should be: <pre>void svcerr_progvers(const SVCXPRT *xpert, ulong_t low, ulong_t high)</pre> <p>where <i>low</i> and <i>high</i> represent the lowest and highest, respectively, of the versions of the service provided.</p>
5	svc_fdset	gABI	<i>svc_fds</i> in Figure 6-12 should be changed to <i>svc_fdset</i> .
6	t_alloc(BA_LIB)	SVID, Vol. 1	Change the sentence starting with "If the size value associated with any specified field is -1 or -2..." to "If the size value associated with any specified field is -1, <i>t_alloc()</i> will allocate the buffer with the size of 1024 bytes. If the size value is -2, <i>t_alloc()</i> will set the buffer pointer to NULL and the buffer maximum size to 0 and will return with success."
7	t_getstate(BA_LIB)	SVID, Vol. 1	Delete the phrase beginning with "or <i>t_getstate()</i> was called..."

System Data Interface Changes

#	Facility	Location	Description
1	<dirent.h>	psABI	Change the declaration of DIR, in Figure 6-5, to be an opaque type. Application programs can know neither the size nor the layout of this type.: <pre>typedef struct{ /* unspecified */ } DIR;</pre>
2	<fcntl.h>	psABI	The following manifest constant is needed for implementing ftruncate() and truncate() operations but is missing from Figure 6-7: <pre>#define F_FREESP 11</pre>
3	<rpc.h>	psABI	page 6-31, The identity of this header file is incorrect, it is specified in the SVID (and in existing practice) to be <rpc/rpc.h>.
4	<rpc.h>	psABI	Change - Delete the definition of RPC_ANYSOCK , and change the definition of RPC_ANYFD to be -1
5	<signal.h>	psABI	Change - On page 6-41, in Figure 6-33 for the struct sigaction type declaration change <i>sigdisp_t sa_disp</i> to void (*sa_handler)(). <pre>struct sigaction { int sa_flags; void (*sa_handler)(); sigset_t sa_mask; int sa_resv[2]; };</pre>
6	<signal.h>	psABI	page 6-41, The values "28" (SIGVTALRM) and "29" (SIGPROF) are missing from <signal.h> in the psABI. These are needed now that <i>getitimer()</i> and <i>setitimer()</i> are part of the SCD 2.4 (figure 6-33).
7	<signal.h>	psABI	page 6-41, The signal of values 32 and above are reserved to the system implementation and must not be used by an SCD-compliant application.
8	<sys/param.h>	psABI	Remove definition of HZ from Figure 6-23.
9	<sys/tiuser.h>	psABI	Change - In Figure 6-52 through 6-58 on page 6-59 through 6-63, header name <sys/tiuser.h> should be changed to <tiuser.h>.
10	<sys/types.h>	psABI	Addition - On page 6-63, in Figure 6-59, add the following type definitions: <pre>typedef unsigned int u_int; typedef unsigned long u_long; typedef unsigned short u_short; typedef char caddr_t;</pre>
11	<wait.h>	psABI	Change - In Figure 6-66 on page 6-68, header name <wait.h> should be changed to <sys/wait.h>.

Miscellaneous ABI Changes

#	Facility	Location	Description
1	Shared Library Names	psABI	Addition - A section should be inserted that identifies the actual version numbers and reference names for shared objects on a SPARC system.

Table 6-2 Library Logical and Reference Names

Library	Reference Name
(runtime linker)	/usr/lib/ld.so.1
libaio	/usr/lib/libaio.so.1
libc	/usr/lib/libc.so.1
libdl	/usr/lib/libdl.so.1
libelf	/usr/lib/libelf.so.1
libintl	/usr/lib/libintl.so.1
libm	/usr/lib/libm.so.1
libnisdb	/usr/lib/libnisdb.so.1
libnsl	/usr/lib/libnsl.so.1
libposix4	/usr/lib/libposix4.so.1
libpthread	/usr/lib/libpthread.so.1
libresolv	/usr/lib/libresolv.so.1
librpcsvc	/usr/lib/librpcsvc.so.1
libsocket	/usr/lib/libsocket.so.1
libucb	/usr/lib/libucb.so.1
libw	/usr/lib/libw.so.1
libthread	/usr/lib/libthread.so.1

2	Dependencies Among	gABI	<p>Change - On page 6-2, at the statement which begins "Application Libraries executable and shared object files..." replace to the end of the paragraph with "Application executables must provide a complete list of those shared objects which the application uses directly. Each system library must supply a complete dependency graph for its own execution as DT_NEEDED entries.</p> <p>Rationale: No application should be required to know what secondary dependencies any platform system library may have. Such dependencies may vary from system to system.</p>
3	Shared Library Names	gABI	<p>Deletion - Delete Table 6-1 on page 6-2.</p> <p>Addition - Actual full path names (reference names) of these shared libraries are specified in the appropriate processor supplement.</p>

4	Shared Library Names	gABI	<p>Addition page 6-2 - A second paragraph should be inserted to this section that states: "The version numbers of shared objects are set on a per-processor basis with the constraint that they are derived from a Generic ABI 'reference version number' for each interface and must change their current value whenever that reference version number changes. In this manner, the reference names can reflect the often combined generic and processor specific portions of the interface in a consistent manner." A shared object version number must change whenever one or more of the following occurs:</p> <ul style="list-style-type: none"> • an entry point is deleted, • an entry point is added, • an entry point is changed, • program visible semantic properties change, or • changes to exported data objects change in size, type, or name."
5	<code><resource.h></code>	psABI	<p>Change the declaration of <code>rlim_t</code>, in Figure 6-27, as follows:</p> <pre>#if _FILE_OFFSET_BITS == 64 typedef unsigned long long rlim_t; #else typedef unsigned long rlim_t; #endif</pre>
6	<code><resource.h></code>	psABI	<p>Change the constant value <code>RLIM_INFINITY</code>, in Figure 6-27, as follows:</p> <pre>#if _FILE_OFFSET_BITS == 64 #define RLIM_INFINITY ((rlim_t) -3) #else #define RLIM_INFINITY 0x7fffffff #endif</pre>
7	<code><stdio.h></code>	psABI	<p>Change the declaration of <code>fpos_t</code>, in Figure 6-39, as follows:</p> <pre>#if _FILE_OFFSET_BITS == 64 typedef long long fpos_t; #else typedef long fpos_t; #endif</pre>
8	<code><stat.h></code>	psABI	<p>Change -On page 6-43, in Figure 6-35 for the struct <code>stat</code> type declaration change as follows:</p> <pre>struct stat { dev_t st_dev; long st_pad1[3]; ino_t st_ino; mode_t st_mode; nlink_t st_nlink; uid_t st_uid; gid_t st_gid; dev_t st_rdev; long st_pad2[2]; off_t st_size; #if _FILE_OFFSET_BITS != 64 long st_pad3; #endif timestruc_t st_atim; timestruc_t st_mtim; timestruc_t st_ctim; long st_blksize;</pre>

-
- | | | | |
|----|---------------|-------|--|
| | | | <pre> blkcnt_t st_blocks; char st_fstype[_ST_FSTYPSZ]; long st_pad4[8]; }; </pre> |
| 9 | <statvfs.h> | psABI | <p>Change -On page 6-45, in Figure 6-36 for the struct <i>statvfs</i> type declaration change as follows:</p> <pre> typedef struct statvfs { unsigned long f_bsize; unsigned long f_frsize; blkcnt_t f_blocks; fsblkcnt_t f_bfree; fsblkcnt_t f_bavail; fsfilcnt_t f_files; fsfilcnt_t f_ffree; fsfilcnt_t f_favail; unsigned long f_fsid; char f_basetype[FSTYPSZ]; unsigned long f_flag; unsigned long f_namemax; char f_fstr[32]; unsigned long f_filler[16]; } statvfs_t; </pre> |
| 10 | <sys/types.h> | psABI | <p>Change the declaration of <i>off_t</i>, in Figure 6-59, as follows:</p> <pre> #if _FILE_OFFSET_BITS == 64 typedef long long off_t; #else typedef long off_t; #endif </pre> |

Overview of Large Files Support (32bit-ABI)

Overview

This section of the SCD defines a set of interfaces introduced within the 32-bit ABI in order to support the use of large files (i.e. files larger than 2^{31} bytes in length). This is essentially a transitional interface which allows developers to use large files in a SPARC 32-bit ABI application. This facility is provided through the addition of a set of large file-specific interfaces with specially-qualified names (e.g. `open64`, `readdir64`, etc.), to complement the historical unqualified names (e.g. `open`, `readdir`, etc.) already in the 32-bit ABI, which operate on small files (those smaller than 2^{31} bytes). Interfaces with these qualified names do not apply within the 64-bit ABI, since the interfaces with the historical unqualified names (e.g. `open`, `readdir`, etc.) all provide access to large files.

The table below summarizes the large file support interfaces and specifies those interface members which are REQUIRED and those which are EXPERIMENTAL.

SCD Extensions to the System V ABI

The SCD includes additional functions which are not specified by the gABI. These functions are not defined in the SVID. The semantics for these additional function definitions are available in the SCD Interface Semantics document. Structures and manifest constants pertaining to these functions are as specified further below.

Library Contents

The following table (Table 6-3) lists the complete set of interfaces added to the 32-bit ABI to support access to large files. The table also indicates the library within which each respective interface is found.

Table 6-3 Large files Interface content (in `libc`, `libaio`, `libthread`, `libposix4`, and `libuch`)

<code>alphasort64</code> (5E)	<code>freopen64</code> (1R)	<code>lstat64</code> (1R)	<code>tmpfile64</code> (1R)
<code>aioread64</code> (3E)	<code>fseeko64</code> (1R)	<code>mkstemp64</code> (1E)	<code>truncate64</code> (1R)
<code>aio_cancel64</code> (4E)	<code>fsetpos64</code> (1R)	<code>mmap64</code> (1R)	
<code>aio_error64</code> (4E)	<code>fstat64</code> (1R)	<code>nftw64</code> (1R)	
<code>aio_fsync64</code> (4E)	<code>fstatvfs64</code> (1R)	<code>open64</code> (1R) (2E)	(1) <code>libc</code>
<code>aio_read64</code> (4E)	<code>ftello64</code> (1R)	<code>pread64</code> (1E)	(2) <code>libthread</code>
<code>aio_return64</code> (4E)	<code>ftruncate64</code> (1R)	<code>pwrite64</code> (1E)	(3) <code>libaio</code>
<code>aio_suspend64</code> (4E)	<code>ftw64</code> (1R)	<code>readdir64</code> (1R) (5E)	(4) <code>libposix4</code>
<code>aio_write64</code> (4E)	<code>getdents64</code> (1E)	<code>readdir64_r</code> (1E)	(5) <code>libuch</code>
<code>aiowrite64</code> (3E)	<code>getrlimit64</code> (1R)	<code>scandir64</code> (5E)	(E) EXPERIMENTAL
<code>creat64</code> (1R) (2E)	<code>lio_listio64</code> (4E)	<code>setrlimit64</code> (1R)	(R) REQUIRED
<code>fgetpos64</code> (1R)	<code>lockf64</code> (1R)	<code>stat64</code> (1R)	
<code>fopen64</code> (1R)	<code>lseek64</code> (1R)	<code>statvfs64</code> (1R)	

Structures and Manifest Constants

The following table summarizes binary-level data types and data structures for the small file (standard, 32-bit) ABI's and the types for the corresponding large file (64-bit) interfaces. The absence of an entry in the Small File Definition column indicates that there is no existing small file (32-bit) type corresponding to the type listed in the Large File Definition column.

Table 6-4 Data structures and types—Standard vs. Large Files (32-bit ABI only)

Standard Definition	Large Files Definition	Header File
<i>struct aiocb</i> <i>off_t aio_offset;</i>	<i>struct aiocb64</i> <i>off64_t aio_offset;</i>	< <i>aio.h</i> >
Note: <i>off_t</i> is a 32-bit scalar indicating an offset within a small file	Note: <i>off64_t</i> is a 64-bit scalar indicating an offset within a large file	
<i>struct dirent</i> <i>ino_t d_ino;</i> <i>off_t d_off;</i>	<i>struct dirent64</i> <i>ino64_t d_ino;</i> <i>off64_t d_off;</i>	< <i>sys/dirent.h</i> >
<i>struct flock</i> <i>off_t l_start;</i> <i>off_t l_len;</i>	<i>struct flock64</i> <i>off64_t l_start;</i> <i>off64_t l_len;</i>	< <i>sys/fcntl.h</i> >
F_SETLK, F_SETLKW, F_GSTLK, F_FREESP	F_SETLK64, F_SETLKW64, F_GETLK64, F_FREESP64, O_LARGEFILE	
<i>fpos_t</i>	<i>fpos64_t</i>	< <i>sys/stdio.h</i> >
<i>rlim_t</i> <i>rlim_t rlim_max;</i>	<i>rlim64_t</i> <i>rlim64_t rlim_max;</i>	< <i>sys/resource.h</i> >
<i>struct rlimit</i> <i>rlim_t rlim_cur;</i>	<i>struct rlimit64</i> <i>rlim64_t rlim_cur;</i>	
RLIM_INFINITY, RLIM_SAVED_MAX, RLIM_SAVED_CUR	RLIM64_INFINITY, RLIM64_SAVED_MAX, RLIM64_SAVED_CUR	
<i>struct stat</i>	<i>struct stat64</i>	< <i>sys/stat.h</i> >
<i>ino_t st_ino;</i> <i>off_t st_size;</i> <i>blkcnt_t st_blocks</i>	<i>ino64_t st_ino;</i> <i>off64_t st_size;</i> <i>blkcnt64_t st_blocks;</i>	
<i>struct statvfs</i> <i>fsblkcnt_t f_blocks;</i> <i>fsblkcnt_t f_bfree;</i> <i>fsblkcnt_t f_bavail;</i> <i>fsfilcnt_t f_files;</i> <i>fsfilcnt_t f_ffree;</i> <i>fsfilcnt_t f_favail;</i>	<i>struct statvfs64</i> <i>fsblkcnt64_t f_flocks;</i> <i>fsblkcnt64_t f_bfree;</i> <i>fsblkcnt64_t f_bavail;</i> <i>fsfilcnt64_t f_files;</i> <i>fsfilcnt64_t f_ffree;</i> <i>fsfilcnt64_t f_favail;</i>	< <i>sys/statvfs.h</i> >

Table 6-4 Data structures and types—Standard vs. Large Files (32-bit ABI only)

Standard Definition	Large Files Definition	Header File
<i>off_t</i> ; <i>ino_t</i> ; <i>blkcnt_t</i> ; <i>fsblkcnt_t</i> ; <i>fsfilcnt_t</i> ;	<i>off64_t</i> ; <i>ino64_t</i> ; <i>blkcnt64_t</i> ; <i>fsblkcnt64_t</i> ; <i>fsfilcnt64_t</i> ;	<sys/types.h>
	LFS64_LARGEFILE LFS64_STDIO	<unistd.h>
	_CS_LFS64_CFLAGS _CS_LFS64_LDFLAGS _CS_LFS64_LIBS _CS_LFS64_LINTFLAGS	<sys/unistd.h>

Figure 6-1 Manifest Constants and Data Types from <dirent.h>

```

struct dirent64 {
    ino64_t      d_ino;
    off64_t     d_off;
    unsigned short d_reclen;
    char        d_name[1];
};

```

Figure 6-2 Manifest Constants and Data Types from <fcntl.h>

```

#define F_SETLK64          34
#define F_SETLKW64       35
#define F_GETLK6         33
#define F_FREESP64       27
#define O_LARGEFILE      0x2000
#if _FILE_OFFSET_BITS == 64
#define F_SETLK          F_SETLK64
#define F_SETLKW         F_SETLKW64
#define F_GETLK          F_GETLK64
#define F_FREESP         F_FREESP64
#else
#define F_SETLK          6
#define F_SETLKW         7
#define F_GETLK          14
#define F_FREESP         11
#endif
struct flock64 { short l_type; short l_whence; off64_t l_start; off64_t l_len; long l_sysid; pid_t l_pid; long l_pad[4]; };

```

Figure 6-3 Manifest Constants and Data Types from <stdio.h>

```

typedef long long    fpos64_t;

```

Figure 6-4 Manifest Constants and Data Types in <sys/resource.h>

```

#define RLIM64_INFINITY                ((rlim64_t) -3)
#define RLIM64_SAVED_MAX               ((rlim64_t) -2)
#define RLIM64_SAVED_CUR              ((rlim64_t) -1)
#if _FILE_OFFSET_BITS == 64
#define RLIM_SAVED_MAX                 ((rlim_t) -2)
#define RLIM_SAVED_CUR                ((rlim_t) -1)
#else
#define RLIM_SAVED_MAX                 0x7fffffff
#define RLIM_SAVED_CUR                0x7ffffffd
#endif
typedef unsigned long longrlim64_t;
struct rlimit64 {
    rlim64_t      rlim64_t      rlim_cur;
                                rlim64_t      rlim_max;
};

```

Figure 6-5 Manifest Constants and Data Types from <sys/stat.h>

```

struct stat64{
    dev_t          st_dev;          long          st_pad1[3];
    ino64_t        st_ino;          mode_t        st_mode;
    nlink_t        st_nlink;       uid_t         st_uid;
    gid_t          st_gid;         dev_t         st_rdev;
    long           st_pad2[2];     off64_t       st_size;
    timestruc_t   st_atim;         timestruc_t   st_mtim;
    timestruc_t   st_ctim;         long          st_blksize;
    blkcnt64_t    st_blocks;       char          st_fstype[16];
    long           st_pad4[8];
};

```

Figure 6-6 Manifest Constants and Data Types from <sys/statvfs.h>

```

typedef struct statvfs64 {
    unsigned int   f_bsize;         unsigned int   f_frsize;
    fsblkcnt64_t  f_blocks;        fsblkcnt64_t  f_bfree;
    fsblkcnt64_t  f_bavail;        fsfilcnt64_t  f_files;
    fsfilcnt64_t  f_ffree;         fsfilcnt64_t  f_favail;
    unsigned int  f_fsid;          char           f_basetype[16];
    unsigned int  f_flag;          unsigned int   f_namemax;
    char          f_fstr[32];      unsigned long  f_filler[16];
} statvfs64_t;

```

Figure 6-7 Manifest Constants and Data Types from <sys/types.h>

```

typedef long long          off64_t;
typedef unsigned long long ino64_t;
typedef long long         blkcnt64_t;
typedef unsigned long long fsblkcnt64_t;
typedef unsigned long long fsfilcnt64_t;
#if _FILE_OFFSET_BITS == 64
typedef unsigned long long ino_t;
typedef long long         blkcnt_t;
typedef unsigned long long fsblkcnt_t;
typedef unsigned long long fsfilcnt_t;
#else
typedef unsigned long     ino_t;
typedef long              blkcnt_t;
typedef unsigned long     fsblkcnt_t;
typedef unsigned long     fsfilcnt_t;
#endif

```

Figure 6-8 Manifest Constants and Data types defined in <unistd.h>

```
#define _PC_FILESIZEBITS          67
#define _CS_LFS_CFLAGS            68
#define _CS_LFS_LDFLAGS          69
#define _CS_LFS_LIBS              70
#define _CS_LFS_LINTFLAGS        71
#define _CS_LFS64_CFLAGS         72
#define _CS_LFS64_LDFLAGS       73
#define _CS_LFS64_LIBS           74
#define _CS_LFS64_LINTFLAGS      75
```


Conventions and Techniques for Library Versioning

Overview

This section of the SCD describes the policies ascribed to by the SCD for versioning of the system libraries and interfaces which define the SPARC ABI. Specifically, the policies apply to conventions for naming and numbering of the shared libraries, so as to indicate a successive, upward-compatible evolution of the runtime interface for applications offered by SCD-conformant system implementations.

Standard Syntax for Library Names and Version Numbers

Each library identified by the SCD has a *full-name*, consisting of a *name part* and a *version part*. The name part always commences with the string “lib”. This initial string is followed by one or more other alphanumeric characters that typically give some idea of the logical interface content contained within the library. To conclude the name part, and to separate it from the version part, is the string “.so.”. The “.so” in the name indicates that the library is a dynamically-linked shared object of the sort that SCD-compliant applications may safely depend upon. After the separating “.so.”, is the version part, which indicates the *major version* number (major revision level) of the library.

In the SCD, the convention is for library names to have only a single-level revision number following the name (e.g. libc.so.1, libnsl.so.1, as compared to libc.so.1.3 or libnsl.so.1.4), and this number indicates the major revision level of the library. Minor revision levels on SCD-conformant systems are indicated by a library-internal labelling technique described below. In certain exceptional cases SCD libraries have both a major and minor number (each number separated by a period following the name part). This situation, which is a departure from the standard naming convention, is the case for example, when the library names have been prescribed earlier and/or by some other standard, and it was either necessary or most practical to preserve that specific name for compatibility with existing applications or other existing usage in the field (e.g. libXm.so.1.2). In such cases, the SCD will typically also offer another name for the same library which conforms to the SCD’s standard naming convention (i.e. the two names will be aliases—e.g. *libXm.so.1.2* and *libXm.so.3* in this specification are equivalent names for the same library).

The *full-name* of each given library is also recorded as the library’s *so-name* recorded as the **DT_SONAME** within the shared object (i.e. the ELF binary) which implements the library, so that this name can be recorded in application binaries which depend upon it (see below under “Application Binary Objects Record Library Names They Depend On”).

Purpose of Versions

The purpose of versioning of the SCD’s shared libraries is to allow an SCD-conformant system implementation to locate the correct library and interface content needed by an application executable (or other shared object) when the application is run. Since there may be more than one revision level of a library present on the number of system implementations and/or releases available in the field at any point in time, or even on the number of revision levels of a library present on a single system implementation, it is important that the application be associated with the correct one at runtime if its interface content and functionality needs are to be met.

Versioning of the libraries has the purpose of indicating one or more of the following properties:

- Additions to the interface content contained within the library.
- Introduction of an entirely new instance of a library which is incompatible with a preceding version of a library by the same name
- Indication of the content level present in the libraries found on a given system implementation
- Indication of the content level depended upon by an SCD-compliant application or other executable

Revision levels (versions)—Major and Minor

Different revision levels (versions) of a library, be they a difference in *major* or *minor* revision, indicate different interface

content within the library. Successive minor revisions of a library contain strictly upward compatible changes to the interface content and/or function within the library. For example minor revision 4 of libc.so.1 has strictly upward compatible content to that which is present in minor revision 3 of libc.so.1. This might mean, for example that minor revision 4 of libc has some additional interface content, beyond what is present in minor revision 3, or that some of the interfaces that were present in revision 3 have had their functionality extended in minor revision 4, or both. Successive major revisions have no specific relation to one another. They might, other than being related to one another in the general sort of functional content they contain, have been given entirely different names. For example libc.so.1 and libc.so.2 would indicate two different major revisions of the C library (libc.so). The presumption is that the two are incompatible in the interface content and/or behavior of the the interfaces contained within them, and that an application may depend on one or the other, but *not* both.

Managing major revisions

In the case that a major revision of a library takes place, system implementations will continue to carry (offer) the earlier major revision of the library (as a complete and separate ELF object with the appropriate major version number) so that applications which were built earlier and depend on that major revision level can continue to be run. For example if a new major revision of libc were to be introduced, it would be libc.so.2, and SCD-conformant systems would also continue to offer the earlier major revision of libc as libc.so.1.

Such a situation would be viewed as the practical technique and basis for dealing with a DEPRECATED set of interfaces. Earlier major revision levels of a library could be removed from a system implementation at the point at which there were no longer sufficiently many application binaries in the field.

In some cases the name part of the library might also be changed to indicate an incompatible departure in the semantics of the interface elements contained within its successor (or alternate). In SCD2.4 we observe this in respect of libpthread.so.1 (the multi-threading interfaces conforming to the POSIX standard) vs. libthread.so.1 (the SCD-native multi-threading interfaces introduced earlier, prior to the POSIX standard). Note that both (incompatible) library interfaces are present in SCD2.4: the former in order to support earlier application binaries based upon SCD's native threads interfaces, the latter in order to support later and current application binaries based upon the POSIX threads standard. A similar situation exists with libaio.so.1 and libposix4.so.1—successive but incompatible versions of the programming interface for asynchronous i/o and real-time programming.

Managing minor revisions

In the case that a minor revision of a library takes place (and this is the most common evolutionary change to the SCD), system implementations continue to carry (offer) the library with the same name and major revision number as before, but with additional interface content and/or functionality in that library. The additional interface content from one minor release of a library to the next is indicated by a library-internal versioning technique described below.

Application Binary Objects Record Library Names They Depend On

An important aspect of library names is that they are recorded in the binary (ELF object) of any executable object or other shared object that depends directly upon the library. This is described in the Extensible Linking Format specification of the System V ABI specification. A set of DT_NEEDED entries recorded within the application's object file (ELF object) indicate the names of any depended-upon shared objects and a DT_RPATH entry indicates any runtime directories (beyond the default /usr/lib) in which the system implementation is to search for them.

It is important to note that the *full-name* described above (in the case of SCD-conformant systems a string containing both a name part and version part: e.g. "libc.so.1") is recorded in a DT_NEEDED entry within the application's executable object, for each library (shared object) it was linked with when it was built. This is a way of recording the library dependency information within the application binary. This both allows and requires that exact matches in the names of the shared library be able to be found at runtime on the system implementation in question if the application is to be dynamically linked with the library and thus given the chance to run.

Library-Internal Minor Versioning—EXPERIMENTAL

In SCD-conformant systems, shared libraries may contain internal definitions indicating minor revision levels of the library. This minor versioning technique relies upon and exploits a *versioning section* within the shared library's ELF object which allows a collection of symbols within the library to be associated with a named set. Several such named sets may be defined within a versioned shared library, and one named set may absorb the content of ("inherit") that of another named set to keep track of an upward-compatible evolution of the interface-member content of a library as interface members were added from one minor release of the library to the next. Library-internal named sets can thus be used for minor versioning and such named sets are referred to as version definitions.

A shared library can have associated with it one or more internal *version definitions*. Each version definition is commonly associated with one or more symbol names. Each symbol name can only be associated with *one* version definition, however a version definition can inherit the symbols from other version definitions. Thus, a structure exists to define one or more independent, or related, version definitions within the object being created. As new changes are made to the object, new version definitions can be added to express these changes.

There are two consequences of providing version definitions within a shared object:

- Dynamic objects that are built using this shared object can record their dependency on the version definitions (interface content) they rely upon. These version dependencies will be verified at runtime (on those system implementations containing versioned shared libraries) to ensure that the appropriate interfaces, or functionality, are available for the correct execution of an application.
- Dynamic objects can select (during their link-edit) only those version definitions of a shared object that they wish to bind to. This mechanism allows developers to control their dependencies on a shared library to the interfaces, or functionality, that provide them the most flexibility (i.e. their desired minor revision level).

Creating a Shared Library with Minor Version Definitions

Version definitions consist of an association of symbol names to a unique version *name*. These associations are established within the shared libraries offered by SCD-conformant system implementations by the vendors of these implementations when they construct the shared libraries at build-time. Such shared libraries are constructed with the use of a versioning `mapfile` supplied to the final link-edit of an object using the `link-editor's-M` option. A version definition is established whenever a version name is specified as part of the `mapfile` directive.

Overview of Support for Multi-threaded Applications

Overview

This section of the SCD describes the SPARC ABI's handling of the interfaces for multi-threaded programs. In particular, there are two semantically distinct interface sets provided—`libthread.so.1`, providing the original SCD “native” interface for multi-threaded applications (those written prior to the POSIX 1003.1c threads standard), and a subsequent one—`libpthread.so.1`, providing a runtime interface for multi-threaded applications which conform to the POSIX threads standard.

Specifically, this section describes certain constraints on applications' use of the threads interfaces, and also the techniques used in SCD-conformant system implementations to decide on and select the correct interface semantics to use (for certain shared interfaces) in a multi-threaded application.

Identification of a Multi-threaded Application and its MT type

A multi-threaded (MT) application is identified by the fact that it depends upon either `libthread.so.1` or `libpthread.so.1`. Specifically, the loading of either `libthread.so.1` or `libpthread.so.1` as a result of the application's dependence upon it, or its loading as needed by any other shared object used by the application causes the multi-threaded interfaces to be present, and the multi-threaded semantics of certain interfaces within `libc` to be in effect for that process. In the case of an MT application, certain interfaces in `libc` have additional thread-specific behaviors which are not required in a non-threaded application.

SCD-native threads vs. POSIX threads—Differing Interface Semantics

The following table (Table 6-5) lists the set of interfaces whose signature or semantics is different in the native threads case (use of `libthread.so.1`) vs. the POSIX threads case (use of `libpthread.so.1`). All the interfaces indicated are found in `libc`.

Table 6-5 Interfaces with Signature or Semantic Behavior Dependent on MT Choice (in libc)

native: char *	asctime_r (const struct tm *tm, char *buf, int buflen);
POSIX: char *	__posix_asctime_r (const struct tm *tm, char *buf);
native: char *	ctime_r (const time_t *clock, char *buf, int buflen);
POSIX: char *	__posix_ctime_r (const time_t *clock, char *buf);
native: int	fork () /* Fork all threads in the process */
POSIX: int	fork () /* Fork just the current thread */
native: struct group *	getgrgid_r (gid_t gid, struct group *result,
	char *buffer, int buflen);
POSIX: int	__posix_getgrgid_r (gid_t gid, struct group *grp,
	char *buffer, size_t bufsize,
	struct group **result);
native: struct group *	getgrnam_r (const char *name, struct group *result,
	char *buffer, int buflen);
POSIX: int	__posix_getgrnam_r (const char *name, struct group *grp,
	char *buffer, size_t bufsize,
	struct group **result);
native: char *	getlogin_r (char *name, int namelen);
POSIX: int	__posix_getlogin_r (char *name, size_t namesize);
native: struct passwd *	getpwnam_r (const char *name, struct passwd *result,
	char *buffer, int buflen);
POSIX: int	__posix_getpwnam_r (const char *name, struct passwd *pwd,
	char *buffer, size_t bufsize,
	struct passwd **result);
native: struct passwd *	getpwuid_r (uid_t uid, struct passwd *pwd,
	char *buffer, int buflen);
POSIX: int	__posix_getpwuid_r (uid_t uid, struct passwd *pwd,
	char *buffer, size_t bufsize,
	struct passwd **result);
native: struct dirent *	readdir_r (DIR *dirp, struct dirent *entry);
POSIX: int	__posix_readdir_r (DIR *dirp, struct dirent *entry,
	struct dirent **result);
native: int	sigwait (sigset_t *setp);
POSIX: int	__posix_sigwait (sigset_t *setp, int *signo)
native: char *	ttyname_r (int fildes, char *name, int namelen);
POSIX: int	__posix_ttyname_r (int fildes, char *name,
	size_t namesize);

To deal with the fact that the interface signature (number and/or types of arguments), and/or the interface semantics (behavior) are different in the native threads mode vs. the POSIX threads mode, two different techniques are used: For the interface `fork(1)`, a runtime decision is made as to the appropriate semantics. For `sigwait(1)` and the reentrant interfaces whose signature is different in the two cases, one of the two possible binary level interface names is selected (typically at compile-time). Each of these is described further below.

Runtime Semantic Dispatch

For `fork(1)`, in which a single binary interface name is provided, SCD conformant system implementations select one or the other semantic variant at runtime. This choice is made on a per-application (i.e. per-process) basis depending on whether the application has linked with `libthread.so.1` or `libpthread.so.1`.

Compile-time Dispatch

For the reentrant interfaces (all but `fork(1)` above), two distinct sets of binary level interfaces (ABI's) are provided. Those ABI's providing the native semantics have the same name as the source level interface, those ABI's providing the POSIX

semantics have a name which is the same as the source name prefixed with “__posix_”. It is the expectation that the software generation system (header files and/or compilation system) maps the source level interface to the correct binary interface.

The following table (Table 6-6) provides an indication of the mapping from the source-level interfaces to the respective binary-level interfaces in the case of native threads and POSIX threads usage¹.

Table 6-6 Mapping from API to ABI

Header File	API	Native Threads ABI	POSIX Threads ABI
<i>unistd.h</i>	fork	fork ^a	fork ^b
	getlogin_r	getlogin_r	__posix_getlogin_r
	ttyname_r	ttyname_r	__posix_ttyname_r
<i>signal.h</i>	sigwait	sigwait	__posix_sigwait
<i>dirent.h</i>	readdir_r	readdir_r	__posix_readdir_r
<i>grp.h</i>	getgrgid_r	getgrgid_r	__posix_getgrgid_r
	getgrnam_r	getgrnam_r	__posix_getgrnam_r
<i>pwd.h</i>	getpwuid_r	getpwuid_r	__posix_getpwuid_r
	getpwnam_r	getpwnam_r	__posix_getpwnam_r
<i>time.h</i>	ctime_r	ctime_r	__posix_ctime_r
	asctime_r	asctime_r	__posix_asctime_r

a. Native semantics apply at runtime (all threads of the current process are duplicated in the fork)

b. POSIX semantics apply at runtime (just the current thread is duplicated in the fork)

MT type is exclusive

A single SCD compliant application may not combine the use of native threads and POSIX threads, as the result is undefined behavior. That is, no single application may link both `libthread.so.1` and `libpthread.so.1`. However SCD conformant system implementations do support the simultaneous execution of applications which use native threads (`libthread` only) and those which use POSIX-threads (`libpthread` only).

Summary of the Threads-related Interfaces

The following tables (Table 6-7, Table 6-8), and Table 6-21, on page 6-61 (in the POSIX threads interface section later in this chapter), list the complete set of interfaces within the 32-bit ABI to support multi-threaded applications. The tables also indicate the library within which each respective interface is found.

1. Note that native interfaces were available as the development (build-time) interface on earlier systems, and POSIX-style interfaces are available on more recent systems. On system implementations which offer the POSIX build-time interface, the earlier interface is typically unavailable, in favor of the newer POSIX threads interface.

Table 6-7 Interfaces to support multi-threaded applications (in libc)

__posix_asctime_r	2.4 E	getpwuid_r	2.3	putc_unlocked	2.3
__posix_ctime_r	2.4 E	getpwnam_r	2.3	putchar_unlocked	2.3
__posix_getgrgid_r	2.4 E	readdir_r	2.3	rand_r	2.3
__posix_getgrnam_r	2.4 E	sigwait	2.3	strtok_r	2.3
__posix_getlogin_r	2.4 E	ttyname_r	2.3		
__posix_getpwuid_r	2.4 E	getc_unlocked	2.3		
__posix_getpwnam_r	2.4 E	funlockfile	2.3		
__posix_readdir_r	2.4 E	flockfile	2.3		
__posix_sigwait	2.4 E	___errno	2.3	2.3 - Interfaces added in SCD2.3	
__posix_ttyname_r	2.4 E	fgetgrent_r	2.3	2.4 - Interfaces added in SCD2.4	
asctime_r	2.3	fgetpwent_r	2.3		
ctime_r	2.3	getchar_unlocked	2.3	* Two versions of fork	
fork E,R*	2.3	getgrent_r	2.3	E - EXPERIMENTAL Interface	
getgrgid_r	2.3	getpwent_r	2.3		
getgrnam_r	2.3	gmtime_r	2.3		
getlogin_r	2.3	localtime_r	2.3		

Table 6-8 Interfaces to support multi-threaded applications (in libc, libthread, libpthread)

cond_broadcast	(1,2)	setcontext	(1)	
cond_destroy	(1,2)	sigaction	(1,2)	(1) libthread
cond_init	(1,2)	siglongjmp	(1,2)	(2) libpthread
cond_signal	(1,2)	sigprocmask	(1,2)	L - 32-bit ABI Large file support routine.
cond_timedwait	(1,2)	sigsetjmp	(1,2)	E - EXPERIMENTAL interfaces.
cond_wait	(1,2)	sigsuspend	(1,2)	
creat64	(1, LE) 2.4	sigwait	(1,2)	2.4 - Interfaces added in SCD2.4.
fork1	(1)	sleep	(1,2)	
mutex_destroy	(1,2)	thr_continue	(1,2)	Note: Each interface in this table also has a corresponding interface by the same name in libc.so.1, which provides the non-threaded (stub) implementation.
mutex_init	(1,2)	thr_create	(1,2)	
mutex_lock	(1,2)	thr_exit	(1,2)	
mutex_trylock	(1,2)	thr_getconcurrency	(1,2)	
mutex_unlock	(1,2)	thr_getprio	(1,2)	
open64	(1, LE) 2.4	thr_getspecific	(1,2)	
rw_rdlock	(1,2)	thr_join	(1,2)	
rw_tryrdlock	(1,2)	thr_keycreate	(1,2)	
rw_trywrlock	(1,2)	thr_kill	(1,2)	
rw_unlock	(1,2)	thr_main	(1,2)	
rw_wrlock	(1,2)	thr_min_stack	(1,2)	
rwlock_destroy	(1)	thr_self	(1,2)	
rwlock_init	(1,2)	thr_setconcurrency	(1,2)	
sema_destroy	(1,2)	thr_setprio	(1,2)	
sema_init	(1,2)	thr_setspecific	(1,2)	
sema_post	(1,2)	thr_sigsetmask	(1,2)	
sema_trywait	(1,2)	thr_suspend	(1,2)	
sema_wait	(1,2)	thr_yield	(1,2)	

libaio - Asynchronous I/O Library - DEPRECATED

Overview

The services specified in this section provide applications with the ability to invoke a number of file operations asynchronously with the execution of the application program.

The interface set described here resides entirely in the OPTIONAL and DEPRECATED dynamic library */usr/lib/libaio.so.1*. Interface members of this library, listed in the table below, are OPTIONAL and DEPRECATED unless explicitly noted otherwise.

This interface set is DEPRECATED effective August 1995. This interface set will not be removed from the SCD before August 1998.

Rationale:

These interfaces are DEPRECATED as they may eventually be replaced by POSIX 1003.4 Asynchronous I/O interfaces.

SCD Extensions to the System V ABI

The SCD specifies a new library: */usr/lib/libaio.so.1*, which has functions not specified by the gABI. These functions are not defined in the SVID. The semantics for these additional function definitions are available in the SCD 2.4 Interface Semantics document. Structures and manifest constants pertaining to these functions are as specified further below.

Library Contents

Table 6-9 libaio Contents

aiocancel	2.3	L - 32-bit ABI Large File Support interface. See	
aioread	2.3	Large File Support	2.3 - Interfaces added in SCD2.3.
aioread64 L(E)	2.4	Interfaces Overview	
aiowait	2.3	section.	2.4 - Interfaces added in SCD2.4.
aiowrite	2.3	E - EXPERIMENTAL	
aiowrite64 L(E)	2.4	interfaces.	Note: All interfaces defined in the SCD Interface Semantics document.

Structures and Manifest Constants

Figure 6-9 Manifest Constants and Data Types from <sys/async.h>

```
#define AIO_INPROGRESS -2 /* values not set by the system */

typedef struct aio_result_t {
    int      aio_return; /* return value of read or write */
    int      aio_errno; /* errno generated by the IO */
} aio_result_t;
```

libc - The C Library

Overview

This section contains the REQUIRED libc and former libsys interfaces to basic system services listed in the *System V Application Binary Interface* and described in sections **BA_OS**, **BA_LIB**, **BA_ENV**, **KE_OS**, and **RT_OS** of the *System V Interface Definition, Third Edition*, along with additional ABI extensions.

The interface set described here resides entirely in the REQUIRED dynamic library: */usr/lib/libc.so.1*. Interface members of this library, listed in the tables below, are REQUIRED unless explicitly noted otherwise. Effective November 1st, 1993 the *sbrk* function interface is DEPRECATED. This interface may be removed as early as November 1st, 1996.

The libc ABI Interfaces

The ABI interfaces listed in the tables further below have been included in the SCD because they are REQUIRED to be present in the dynamic library: */usr/lib/libc.so.1*. Issues regarding synonyms and global data symbols associated with this library can be found in the *System V Application Binary Interface*.

SCD Extensions to the System V ABI

The SCD requires */usr/lib/libc.so.1*, which has functions which are either not specified by, or are different from, the gABI. These functions are either not defined in the SVID, or are defined differently in the SCD than the SVID. The semantics pages for these additional/modified function definitions are available in the SCD Interface Semantics document. Structures and manifest constants pertaining to these functions are as specified further below.

Long Long Intrinsic Support

long long intrinsics are routines for the support of operations on a 64-bit integer type (“long long”) for both signed and unsigned quantities. Their descriptions are available in the SCD 2.4 Interface Semantics document. Calling sequence for 64-bit integer arguments and return value is described in Low-level System Information Changes in Chapter 3. The long long type is supported in *printf(BA_LIB)* and *scanf(BA_LIB)* as follows: In format string, conversion specifiers d, i, o, u, x, and X may be preceded by ll (ell ell) to indicate that corresponding argument is of type long long integer (printf) or pointer to long long integer (scanf).

Exported Data

The exported data in the following table is REQUIRED to be present in */usr/lib/libc.so.1*. The format of these entries is: data[size], where size is a hexadecimal byte count.

Table 6-10 Exported Data for libc

__iob[0x140]	daylight[0x4] +1, 2	+ - Previously found in the libsys system library.	3 - New interfaces added to SCD2.3.
__huge_val[0x8] +	environ[0x4] +1, 2		
_altzone[0x4] +	errno[0x4] +1, 2, 3		
_ctype[0x209] +	getdate_err[0x4]	1 - See C Library Changes section at the beginning of this chapter.	
_daylight[0x4] +	optarg[0x4]		
_environ[0x4]+1,2,3	opterr[0x4]	2 - Conforming systems are required to reserve space to _environ in libc.	
_getdate_err[0x4]	optind[0x4]		
_numeric[0x2] +	optopt[0x4]		
_timezone[0x4] +	timezone[0x4] +	It is the responsibility of either the compilation system or the application to ensure that _environ is properly initialized.	
_tzname[0x8] +	tzname[0x8] +		
altzone[0x4] +			

Table 6-11 libc SPARC Support Routines (32-Bit ABI)

.div +	__rem64 +,*2,3	_Q_mul +	+ - Previously found in the libsys system library.	described in Low-Level System Information Changes in Chapter 3.
.mul +	__udiv64 +*2,3	_Q_neg +		
.rem +	__umul64 +*2,3	_Q_qtod +		
.stret1 +	__urem64 +*2,3	_Q_qtoi +	* - Found in the SCD IS.	3 - New interfaces added to SCD2.3.
.stret2 +	_Q_add +	_Q_qtos +		
.stret4 +	_Q_cmp +	_Q_qtoll +*2,3	1 - See C Library Changes section at the beginning of this chapter.	
.stret8 +	_Q_cmpe +	_Q_qtou+ 1		
.udiv +	_Q_div +	_Q_qtoull +*2,3		
.umul +	_Q_dtoq +	_Q_sqrt +	2 - Long long intrinsics support routines - routines supporting operations on a 64-bit integer ("long long") for both signed and unsigned quantities.	
.urem +	_Q_feq +	_Q_stoq +		
__div64 +*2,3	_Q_fge +	_Q_sub +		
__dtoll +*2,3	_Q_fle +	_Q_ulltoq *2,3		
__dtoull +*2,3	_Qflt +	_Q_utoq +		
__ftoll +*2,3	_Q_fne +	__dtou +1		
__ftoull +*2,3	_Q_itoq +	__ftou +1	Calling sequence for 64-bit integer arguments and return value is	
__mul64 +*2,3	_Q_lltoq +*2,3			

Library Contents

Table 6-12 libc content

__errno	*,2.3	chdir	+	fattach	+
__assert		chmod	+	fchdir	+
__errno		chown	+	fchmod	+
__filbuf		chroot	+	fchown	+
__flsbuf		clearerr		fclose	
__posix_asctime_r	2.4,E	clock		fcntl	+
__posix_ctime_r	2.4,E	close	+	fconvert	*,2.4,E
__posix_getgrgid_r	2.4,E	closedir	+	fcvt	*,2.4
__posix_getgrnam_r	2.4,E	closelog	*,2.4	fdetach	+
__posix_getlogin_r	2.4,E	creat	+	fdopen	++
__posix_getpwnam_r	2.4,E	creat64	*,2.4,L	feof	
__posix_getpwuid_r	2.4,E	crypt	++	ferror	
__posix_readdir_r	2.4,E	ctermid		fflush	
__posix_sigwait	2.4,E	ctime		ffs	*,2.4
__posix_ttyname_r	2.4,E	ctime_r	*,2.3,E	fgetc	
_cleanup	*	cuserid		fgetgrent	+,++
_exit	+	dbm_close	*,2.4	fgetgrent_r	*,2.3
_prioset	*,2.4	dbm_delete	*,2.4	fgetpos	
_tolower		dbm_fetch	*,2.4	fgetpos64	*,2.4,L
_toupper		dbm_firstkey	*,2.4	fgetpwent	+,++
_xftw		dbm_nextkey	*,2.4	fgetpwent_r	2.3
abort		dbm_open	*,2.4,E	fgets	
abs		dbm_store	*,2.4	fgetspent	*,2.4,E
access	+	decimal_to_double	*,2.4,E	file_to_decimal	*,2.4,E
acct	+	decimal_to_extended*	*,2.4,E	fileno	
addsev	E	decimal_to_quadruple	*,2.4,E	finite	*,2.4,E
addseverity	*	decimal_to_single	*,2.4,E	flockfile	*
adjtime		difftime		fmtmsg	
alarm	+	div		fopen	
ascftime	*,2.4,E	double_to_decimal	*,2.4,E	fopen64	*,2.4,L
asctime		dup	+	fork	+,E,R
asctime_r	*,2.3,E	econvert	*,2.4,E	fpathconf	+
atexit		ecvt	*,2.4	fpclass	*,2.4,E
atof		encrypt	*,++	fpgetmask	*,2.4,E
atoi		endgrent	+,++	fpgetround	*,2.4,E
atol		endpwent	+,++	fpgetsticky	*,2.4,E
bsearch		endspent	*,2.4,E	fprintf	
calloc	+,*	endutxent	*,2.4	fpsetmask	*,2.4,E
catclose	+	execl	+	fpsetround	*,2.4,E
catgets	+	execle	+	fpsetsticky	*,2.4,E
catopen		execlp	+	fputc	
cfgetispeed	+	execv	+	fputs	
cfgetospeed		execve	+	fread	
cfsetispeed		execvp	+	free	+,*
cfsetospeed		exit	+	freopen	
cftime	*,E	extended_to_decimal*	*,2.4,E	freopen64	*,2.4,L
				frexp	

Table 6-12 (cont'd) libc content

fscanf		getmsg	+	isalnum	
fseek		getopt		isalpha	
fseeko		getpass		isascii	
fseeko64	*,2.4,L	getpgid	+	isastream	+
fsetpos		getpgrp	+	isatty	
fsetpos64	*,2.4,L	getpid		iscntrl	
fstat	+	getpmsg		isdigit	
fstat64	*,2.4,L	getppid		isgraph	
fstatvfs	+	getpw	*,2.4,E	islower	
fstatvfs64	*,2.4,L	getpwent	+,++	isnan	*
fsync	+	getpwent_r	*,2.3,E	isnand	*
ftell		getpwnam	+	isnanf	*,2.4,E
ftello		getpwnam_r	*,2.3	isprint	
ftello64	*,2.4,L	getpwuid	+	ispunct	
ftok	+	getpwuid_r	*,2.3,E	isspace	
ftruncate	*,2.4	getrlimit	+	isupper	
ftruncate64	*,2.4,L	getrlimit64	*,2.4,L	isxdigit	
ftw64	*,2.4,L	gets		kill	+
func_to_decimal	*,2.4,E	getsid	+	labs	
funlockfile		getspent	*,2.4,E	lchown	+
fwrite		getspnam	*,2.4	lckpddf	E
gconvert	*,2.4,E	getsubopt		ldexp	
gcvt	*,2.4	gettimeofday	++	ldiv	
getc		gettxt	+	lfind	
getc_unlocked	*,2.3	getuid	+	lfmt	E
getchar		getutmp	*,2.4,E	link	+
getchar_unlocked	*,2.3	getutmpx	*,2.4,E	llseek	+
getcontext	+	getutxent	*,2.4	localeconv	+
getcwd	+	getutxid	*,2.4	localtime	
getdate		getutxline	*,2.4	localtime_r	*,2.3
getdents	*,2.4,E	getvfsany	*,2.4	lockf	
getdents64	*,2.4,L,E	getvfssent	*,2.4,E	lockf64	*,2.4,L
getegid	+	getvfssfile	*,2.4,E	logb	
getenv		getvfsspec	*,2.4	longjmp	
geteuid	+	getw		lsearch	
getgid	+	gmtime		lseek	+
getgrent	+,++	gmtime_r	*,2.3	lseek64	*,2.4,L
getgrent_r	*,2.3	grantpt	+	lstat	+
getgrgid	+	hasmntopt	*,2.4,E	lstat64	*,2.4,L
getgrgid_r	*,2.3,E	hcreate		madvise	*,2.4,E
getgrnam	+	hdestroy		makecontext	+,*
getgrnam_r	*,2.3,E	hsearch		malloc	+,*
getgroups	+	iconv	*	mblen	
getitimer	++	iconv_close	*	mbstowcs	
getlogin	+	iconv_open	*	mbtowc	
getlogin_r	*,2.3,E	initgroups	+	memalign	*,2.4,E
getmntany	*,2.4	insque	*,2.4		
getmntent	*,2.4,E	ioctl	+		

Table 6-12 (cont'd) libc content

memccpy		pread	*,2.4,E	scalb	
memchr		pread64	*,2.4,L,E	scanf	
memcmp		printf		seconvert	*,2.4,E
memcntl	+	prcntl		seekdir	+
memcpy		processor_bind	*,2.4,E	select	*,2.4
memmove		processor_info	*,2.4,E	semctl	+
memset		profil		semget	+
mincore	*,2.4,E	psiginfo	*,2.4,E	semop	+
mkdir	+	psignal	*,2.4,E	setbuf	
mkfifo	+	ptrace	+	setcat	E
mkstemp64	*,2.4,L,E	ptsname	+	setcontext	+
mktemp		putc		setegid	*,2.4
mktime		putc_unlocked	*,2.3	seteuid	*,2.4
mlock	+	putchar		setgid	+
mlockall		putchar_unlocked	*,2.3	setgrent	+,++
mmap		putenv		setgroups	+
mmap64	*,2.4,L	putmsg	+	setitimer	++
modf	*	putpmsg	+	setjmp	
modff	*,2.4,E	puts		setkey	++
monitor		putspent	E	setlabel	
mount		pututxline		setlocale	+
mprotect	+	putw		setlogmask	*,2.4
msgctl	+	pwrite	*,2.4,E	setpgid	+
msgget	+	pwrite64	*,2.4,L,E	setpgrp	+
msgrcv	+	qeconvert	*,2.4,E	setpwent	+
msgsnd	+	qfconvert *	2.4 (E)	setrlimit	+
msync	+	qgconvert *	2.4 (E)	setrlimit64	*,2.4,L
munlock	+	qsort		setsid	+
munlockall		quadruple_to_decimal	*,2.4,E	setspent	*,2.4,E
munmap	+	raise		settimeofday	E
nextafter		rand		setuid	+
nftw		rand_r	*,2.3	setutxent	*,2.4
nftw64	*,2.4,L	read	+,++	setvbuf	
nice	+	readdir	+	sfconvert	*,2.4,E
nl_langinfo		readdir64	*,2.4,L	sgconvert	*,2.4,E
open	+	readdir64_r	*,2.4,L,E	shmat	+
open64	*,2.4,L	readdir_r	*,2.3,E	shmctl	+
opendir	+	readlink	+	shmdt	+
openlog	*,2.4	readv	+,++	shmget	+
p_online	*,2.4,E	realloc	+	sigaction	+
pathconf	+	realpath	*,2.4	sigaddset	+
pause	+	remove	+	sigaltstack	+
pclose		remque	*,2.4	sigdelset	+
perror		rename	+,++	sigemptyset	+
pfmt	E	rewind		sigfillset	+
pipe	+	rewinddir	+	sighold	+
poll	+	rmdir	+	sigignore	+
popen		sbrk	+,++		

Table 6-12 (cont'd) libc content

sigismember	+	strtol		uname	+
siglongjmp	+	strtoul		ungetc	
signal	+	strxfrm	+	unlink	
sigpause	+	swab		unlockpt	+
sigpending	+	swapcontext	+,*	unordered	*,E
sigprocmask	+	swapctl		updwtmp	*,E
sigrelse	+	symlink	+,++	updwtmpx	*,2.4,E
sigsend	+	sync	+	utime	+
sigsendset	+	sysconf	+	utmpxname	*,2.4,E
sigset	+	sysfs	*,2.4,E	valloc	*,2.4
sigsetjmp	+	sysinfo	++	vfork	*,2.4
sigsuspend	+	syslog	*,2.4	vfprintf	
sigwait	+,2.3,E,Rsystem	system	+,++	vhangup	*,E
single_to_decimal	*,2.4,E	tcdrain		vlfmt	E
sleep		tcflow		vpfmt	E
sprintf		tcflush		vprintf	
srand		tcgetattr		vsprintf	
sscanf		tcgetpgrp		vsyslog	*,2.4,E
stat	+	tcgetsid		wait	+
stat64	*,2.4,L	tcsendbreak		waitid	+,++
statvfs	+	tcsetattr		waitpid	+
statvfs64	*,2.4,L	tcsetpgrp		wcstombs	
stime	+	tdelete		wctomb	
strcasecmp	*,2.4	tell		write	+,++
strcat		telldir	+	writev	+,++
strchr		tempnam			
strcmp		tfind			
strcoll	+	time	+		
strcpy		times			
strcspn		tmpfile			
strdup		tmpfile64	*,2.4,L		
strerror	+	tmpnam			
strftime	+	toascii			
string_to_decimal	*,2.4,E	tolower			
strlen		toupper			
strncasecmp	*,2.4	truncate	*,2.4		
strncat		truncate64	*,2.4,L	+ - previously in libsys	
strncmp		tsearch			
strncpy		ttyname	+,*	++ - see C Library Changes Section at the beginning of this chapter	
strpbrk		ttyname_r	*,2.3,E		
strptime		ttyslot	*,2.4	* - Defined in the SCD IS	
strrchr		twalk		L- 32-bit ABI large files interface	
strsignal	*,2.4,E	tzset			
strspn		uadmin	*,2.4,E	E - EXPERIMENTAL interface	
strstr		ulckpwwdf	E	R - REQUIRED interface	
strtod		ulimit	+		
strtok		umask	+	2.3 - Interface added in SCD2.3	
strtok_r	*,2.3	umount	+	2.4 - Interface added in SCD2.4	

Structures and Manifest Constants

Figure 6-10 Manifest Constants and Data Types from <alloca.h>

```
#define alloca(x)      __builtin_alloca(x)
```

Figure 6-11 Manifest Constants and Data Types from <floatingpoint.h>

```
typedef float          single;
typedef unsigned long extended[3];
typedef long double   quadruple;
typedef unsigned      fp_exception_field_type;
#define DECIMAL_STRING_LENGTH 512
typedef char decimal_string[DECIMAL_STRING_LENGTH];
typedef struct {
    enum fp_class_type    fpclass;
    int                  sign;
    int                  exponent;
    decimal_string       ds;
    int                  more;
    int                  ndigits;
} decimal_record;

enum decimal_form {fixed_form, floating_form};

typedef struct {
    enum fp_direction_type rd;
    enum decimal_form      df;
    int                    ndigits;
} decimal_mode;

enum decimal_string_form {
    invalid_form,          whitespace_form,          fixed_int_form,
    fixed_intdot_form,    fixed_dotfrac_form,    fixed_intdotfrac_form,
    floating_int_form,    floating_intdot_form,  floating_dotfrac_form,
    floating_intdotfrac_form, inf_form,          infinity_form,
    nan_form,            nanstring_form
};

enum fp_direction_type {
    fp_nearest = 0,
    fp_tozero  = 1,
    fp_positive = 2,
    fp_negative = 3
};

enum fp_class_type {
    fp_zero          = 0,
    fp_subnormal    = 1,
    fp_normal        = 2,
    fp_infinity     = 3,
    fp_quiet        = 4,
    fp_signaling    = 5
};
```

Figure 6-12 Manifest Constants and Data Types from <ftw.h>

```
#define FTW_SLN          7
#define _XFTWVER        2
#define ftw(p, f, d)    _xftw(_XFTWVER, p, f, d)
```

Figure 6-13 Manifest Constants and Data Types from <iconv.h>

```
typedef void    *iconv_t;
```

Figure 6-14 Manifest Constants and Data Types from <ieeefp.h>

```
typedef enum fpclass_t {
    FP_SNAN = 0, FP_QNAN = 1, FP_NINF = 2,
    FP_PINF = 3, FP_NDENORM = 4, FP_PDENORM = 5,
    FP_NZERO = 6, FP_PZERO = 7, FP_NNORM = 8, FP_PNORM = 9
} fpclass_t;
typedef enum    fp_rnd {FP_RN = 0, FP_RZ = 1, FP_RP = 2, FP_RM = 3} fp_rnd;
#define fp_except    int
#define FP_X_INV     0x10
#define FP_X_OFL     0x08
#define FP_X_UFL     0x04
#define FP_X_DZ      0x02
#define FP_X_IMP     0x01
```

Figure 6-15 Manifest Constants and Data Types from <limits.h>

```
#define LOGNAME_MAX    8
#define LLONG_MIN      (-9223372036854775807LL-1LL)
#define LLONG_MAX      9223372036854775807LL
#define ULLONG_MAX     18446744073709551615ULL
```

Figure 6-16 Manifest Constants and Data Types from <ndbm.h>

```
#define PBLKSIZ          1024
#define DBLKSIZ          4096
#define _DBM_IOERR       0x2
#define DBM_INSERT      0
#define DBM_REPLACE     1
typedef struct{
    int      dbm_dirf;   int      dbm_pagf;   int      dbm_flags;
    long     dbm_maxbno; long     dbm_bitno; long     dbm_hmask;
    long     dbm_blkptr; int      dbm_keyptr; long     dbm_blkno;
    long     dbm_pagbno; char     dbm_pagbuf[PBLKSIZ]; long     dbm_dirbno;
    char     dbm_dirbuf[DBLKSIZ];
} DBM;
#define dbm_error(db)    ((db)->dbm_flags & _DBM_IOERR)
#define dbm_clearerr(db) ((db)->dbm_flags &= ~_DBM_IOERR)
typedef struct {
    char     *dptr;
    int      dsize;
} datum;
```

Figure 6-17 Manifest Constants and Data Types from <pfmt.h>

```
#define MM_STD           0
#define MM_NOSTD        0x100
#define MM_GET          0
#define MM_NOGET        0x200
#define MM_ACTION       0x400
#define MM_NOCONSOLE    0
#define MM_CONSOLE      0x800
#define MM_HARD         0x1000
#define MM_SOFT         0x2000
#define MM_FIRM         0x4000
#define MM_APPL         0x8000
#define MM_UTIL         0x10000
#define MM_OPSYS        0x20000
#define MM_ERROR        0
#define MM_HALT         1
#define MM_WARNING      2
#define MM_INFO         3
```

Figure 6-18 Manifest Constants and Data Types from <search.h>

```

struct qelem {
    struct qelem *q_forw;
    struct qelem *q_back;
};

```

Figure 6-19 Manifest Constants and Data Types from <shadow.h>

```

struct spwd {
    char *sp_namp;
    char *sp_pwdp;
    long sp_lstchg;
    long sp_min;
    long sp_max;
    long sp_warn;
    long sp_inact;
    long sp_expire;
    unsigned long sp_flag;
};

```

Figure 6-20 Manifest Constants and Data Types from <siginfo.h>

```

union sigval {
    int sival_int;
    void *sival_ptr;
};
struct sigevent {
    int sigev_notify;
    int sigev_signo;
    union sigval sigev_value;
};
#define SIGEV_NONE 1
#define SIGEV_SIGNAL 2
#define SI_NOINFO 32767
#define SI_USER 0
#define SI_LWP (-1)
#define SI_QUEUE (-2)
#define SI_TIMER (-3)
#define SI_ASYNCIO (-4)
#define SI_MESGQ (-5)

```

Figure 6-21 Manifest Constants and Data Types from <stddef.h>

```

#define offsetof(s, m) ((size_t)(&(((s *)0)->m))

```

Figure 6-22 Manifest Constants and Data Types from <sys/dirent.h>

```
struct dirent {
    ino_t          d_ino;
    off_t          d_off;
    ushort_t      d_reclen;
    char           d_name[1];
};
```

Figure 6-23 Manifest Constants and Data Types from <sys/fstyp.h>

```
#define FSTYPSZ      16
#define GETFSIND     1
#define GETFSTYP     2
#define GETNFSSTYP  3
```

Figure 6-24 Manifest Constants and Data Types from <sys/mnttab.h>

```
#define MNT_LINE_MAX 1024
#define MNT_TOOLONG  1
#define MNT_TOOMANY  2
#define MNT_TOOFEW   3
#define putmntent(fd, mp)\
    fprintf((fd), "%s\t%s\t%s\t%s\t%s\n",\
            (mp)->mnt_special ? (mp)->mnt_special : "-",\
            (mp)->mnt_mountp ? (mp)->mnt_mountp : "-",\
            (mp)->mnt_fstype ? (mp)->mnt_fstype : "-",\
            (mp)->mnt_mntopts ? (mp)->mnt_mntopts : "-",\
            (mp)->mnt_time ? (mp)->mnt_time : "-")
struct mnttab {
    char    *mnt_special;
    char    *mnt_mountp;
    char    *mnt_fstype;
    char    *mnt_mntopts;
    char    *mnt_time;
};
```

Figure 6-25 Manifest Constants and Data Types from <sys/systeminfo.h>

```

/* Commands to sysinfo() */
#define SI_SYSNAME      1          /* return name of operating system */
#define SI_HOSTNAME     2          /* return name of node */
#define SI_RELEASE      3          /* return release of operating system */
#define SI_VERSION      4          /* return version field of utsname */
#define SI_MACHINE      5          /* return kind of machine */
#define SI_ARCHITECTURE 6          /* return instruction set arch */
#define SI_HW_SERIAL    7          /* return hardware serial number */
#define SI_HW_PROVIDER  8          /* return hardware manufacturer */
#define SI_SRPC_DOMAIN  9          /* return secure RPC domain */

```

Figure 6-26 Manifest Constants and Data Types from <sys/uadmin.h>

```

#define A_REBOOT        1
#define A_SHUTDOWN     2
#define A_REMOUNT      4
#define AD_HALT        0
#define AD_BOOT        1
#define AD_IBOOT       2

```

Figure 6-27 Manifest Constants and Data Types from <sys/vfstab.h>

```

#define VFS_LINE_MAX    1024
#define VFS_TOOLONG     1
#define VFS_TOOMANY     2
#define VFS_TOOFEW     3
struct vfstab {
    char    *vfs_special;
    char    *vfs_fsckdev;
    char    *vfs_mountp;
    char    *vfs_fstype;
    char    *vfs_fsckpass;
    char    *vfs_automnt;
    char    *vfs_mntopts;
};

```

Figure 6-28 Manifest Constants and Data Types from <sys/lock.h>

```
#define UNLOCK          0
#define PROCLOCK       1
#define TXTLOCK        2
#define DATLOCK        4
```

Figure 6-29 Manifest Constants and Data Types from <sys/mman.h>

```
#define PROT_NONE      0x0
#define MAP_NORESERVE  0x40
#define PROC_TEXT      (PROT_EXEC | PROT_READ)
#define PROC_DATA      (PROT_READ | PROT_WRITE | PROT_EXEC)
#define SHARED         0x10
#define PRIVATE        0x20
#define MAP_FAILED     ((void *) -1)
#define MADV_NORMAL    0
#define MADV_RANDOM    1
#define MADV_SEQUENTIAL 2
#define MADV_WILLNEED 3
#define MADV_DONTNEED 4
#define MC_SYNC        1
#define MC_LOCK        2
#define MC_UNLOCK      3
#define MC_ADVISE      4
#define MC_LOCKAS      5
#define MC_UNLOCKAS    6
#define MCL_CURRENT    0x1
#define MCL_FUTURE     0x2
```

Figure 6-30 Manifest Constants and Data Types from <sys/priocntl.h>

```

#define PC_VERSION          1
#define priocntl(idtype, id, cmd, arg) __priocntl(PC_VERSION, idtype, id, cmd, arg)
#define priocntlset( psp, cmd, arg) __priocntlset(PC_VERSION, psp, cmd, arg)
#define PC_GETCID          0
#define PC_GETCLINFO      1
#define PC_SETPARMS       2
#define PC_GETPARMS       3
#define PC_ADMIN           4
#define PC_CLNULL         -1
#define PC_CLNMSZ         16
#define PC_CLINFOSZ       (32 / sizeof(long))
#define PC_CLPARMSZ       (32 / sizeof(long))
typedef struct pcinfo {
    id_t    pc_cid;
    char    pc_clname[PC_CLNMSZ];
    long    pc_clinfo[PC_CLINFOSZ];
} pcinfo_t;
typedef struct pcparms {
    id_t    pc_cid;
    long    pc_clparms[PC_CLPARMSZ];
} pcparms_t;

```

Figure 6-31 Manifest Constants and Data Types from <sys/processor.h>

```

#define P_OFFLINE          1
#define P_ONLINE           2
#define P_STATUS           3
#define PI_TYPELEN        16
#define PI_FPUTYPE        32
typedef struct {
    int    pi_state;
    char    pi_processor_type[PI_TYPELEN];
    char    pi_fputypes[PI_FPUTYPE];
    int    pi_clock;
} processor_info_t;
#define PBIND_NONE        -1
#define PBIND_QUERY       -2

```

Figure 6-32 Manifest Constants and Data Types from <sys/procset.h>

```
typedef enum idtype {P_PID,P_PPID,P_PGID,P_SID,P_CID,P_UID,P_GID,P_ALL,P_LWPID }
idtype_t;
#define setprocset(psp, op, ltype, lid, rtype, rid) \
        ((psp)->p_op           = (op), \
         (psp)->p_lidtype      = (ltype), \
         (psp)->p_lid         = (lid), \
         (psp)->p_ridtype     = (rtype), \
         (psp)->p_rid        = (rid))
```

Figure 6-33 Manifest Constants and Data Types from <sys/rtprioctl.h>

```
typedef struct rtparms {
    short  rt_pri;
    ulong  rt_tqsecs;
    long   rt_tqnsecs;
} rtparms_t;
typedef struct rtinfo {
    short  rt_maxpri;
} rtinfo_t;
#define RT_NOCHANGE      -1
#define RT_TQINF        -2
#define RT_TQDEF        -3
```


Figure 6-34 Manifest Constants and Data Types from <sys/swap.h>

```

#define ST_INDEL          0x01
#define SC_ADD            1
#define SC_LIST           2
#define SC_REMOVE        3
#define SC_GETNSWP       4
typedef struct swapres {
    char    *sr_name;
    off_t   sr_start;
    off_t   sr_length;
} swapres_t;
typedef struct swapent {
    char    *ste_path;
    off_t   ste_start;
    off_t   ste_length;
    long    ste_pages;
    long    ste_free;
    long    ste_flags;
} swapent_t;
typedef struct swaptable {
    int      swt_n;
    struct   swapent swt_ent[1];
} swaptbl_t;

```

Figure 6-35 Manifest Constants and Data Types from <sys/time.h>

```

typedef longlong_t       hrttime_t;
#define FD_SETSIZE       1024
#define NBBY              8
typedef long              fd_mask;
#define NFDBITS           (sizeof(fd_mask) * NBBY) /* bits per mask */
#define howmany(x, y)     (((x)+(y)-1)/(y))
typedef struct fd_set {
    fd_mask fds_bits[howmany(FD_SETSIZE, NFDBITS)];
} fd_set;
#define FD_SET(n, p)     ((p)->fds_bits[(n)/NFDBITS] |= (1 << ((n) % NFDBITS)))
#define FD_CLR(n, p)     ((p)->fds_bits[(n)/NFDBITS] &= ~(1 << ((n) % NFDBITS)))
#define FD_ISSET(n,p)    ((p)->fds_bits[(n)/NFDBITS] & (1 << ((n) % NFDBITS)))
#define FD_ZERO(p)      memset((char *) (p), 0, sizeof(*(p)))

```

Figure 6-36 Manifest Constants and Data Types from <sys/tspriocntl.h>

```
typedef struct tsparms {
    short      ts_uprilm;
    short      ts_upri;
} tsparms_t;

typedef struct tsinfo {
    short      ts_maxupri;
} tsinfo_t;

#define TS_NOCHANGE    -32768
```

Figure 6-37 Manifest Constants and Data Types from <sys/types.h>

```
typedef int          processorid_t;
typedef long long    longlong_t;
typedef unsigned long long    u_longlong_t;
typedef unsigned long    major_t;
typedef unsigned long    minor_t;
#define NODEV          (dev_t)(-1)
```

Figure 6-38 Manifest Constants and Data Types from <syslog.h>

```

#define LOG_KERN          (0<<3)
#define LOG_USER         (1<<3)
#define LOG_MAIL         (2<<3)
#define LOG_DAEMON       (3<<3)
#define LOG_AUTH         (4<<3)
#define LOG_SYSLOG       (5<<3)
#define LOG_LPR          (6<<3)
#define LOG_NEWS         (7<<3)
#define LOG_UUCP         (8<<3)
#define LOG_LFMT         (14<<3)
#define LOG_CRON         (15<<3)
#define LOG_LOCAL0       (16<<3)
#define LOG_LOCAL1       (17<<3)
#define LOG_LOCAL2       (18<<3)
#define LOG_LOCAL3       (19<<3)
#define LOG_LOCAL4       (20<<3)
#define LOG_LOCAL5       (21<<3)
#define LOG_LOCAL6       (22<<3)
#define LOG_LOCAL7       (23<<3)
#define LOG_EMERG        0
#define LOG_ALERT        1
#define LOG_CRIT         2
#define LOG_ERR          3
#define LOG_WARNING      4
#define LOG_NOTICE       5
#define LOG_INFO         6
#define LOG_DEBUG        7
#define LOG_MASK(pri)    (1 << (pri))
#define LOG_UPTO(pri)    ((1 << ((pri)+1)) - 1)
#define LOG_PID          0x01
#define LOG_CONS         0x02
#define LOG_ODELAY       0x04
#define LOG_NDELAY       0x08
#define LOG_NOWAIT       0x10

```

Figure 6-39 Manifest Constants and Data Types from <unistd.h>

```

#define _CS_PATH        65

```

Figure 6-40 Manifest Constants and Data Types from <utmpx.h>

```
struct exit_status {short e_termination;short e_exit;};
struct utmpx {
    char          ut_user[32];
    char          ut_id[4];
    char          ut_line[32];
    pid_t         ut_pid;
    short         ut_type;
    struct exit_status ut_exit;
    struct timeval ut_tv;
    long          ut_session;
    long          pad[5];
    short         ut_syslen;
    char          ut_host[257];
};
struct utmp {
    char          ut_user[8];
    char          ut_id[4];
    char          ut_line[12];
    short         ut_pid;
    short         ut_type;
    struct        exit_status ut_exit;
    time_t        ut_time;
};

#define EMPTY          0
#define RUN_LVL        1
#define BOOT_TIME      2
#define OLD_TIME       3
#define NEW_TIME       4
#define INIT_PROCESS   5
#define LOGIN_PROCESS  6
#define USER_PROCESS   7
#define DEAD_PROCESS   8
```

libdl - Dynamic Object File Loading Library

Overview

The run-time dynamic linking facilities of the system are specified in this section. The particulars on dynamic linking and loading, path name resolution, data initialization functions, symbol relocation and binding, and automatic loading of secondary objects are given in Chapter 5 of this document and in the normative documents it references, the *System V Application Binary Interface* and the *System V Application Binary Interface, SPARC Processor Supplement*.

The interface set described here resides entirely in the REQUIRED dynamic library: */usr/lib/libdl.so.1*. Interface members of this library, listed in the table below, are REQUIRED unless explicitly noted otherwise.

SCD Extensions to the System V ABI

The SCD specifies a new library: */usr/lib/libdl.so.1*, which has functions not specified by the gABI. These functions are not defined in the SVID. The semantics for these additional function definitions are available in the SCD Interface Semantics document. Structures and manifest constants pertaining to these functions are as specified further below.

Library Contents

Table 6-13 libdl contents

dladdr *	2.4	* - Defined in the SCD Interface Semantics.
dlclose *		2.4 - Interfaces added in SCD2.4
dLError *		
dlopen *		
dlsym *		

Structures and Manifest Constants

The manifest constants in Figure 6-41 are required for dlopen.

Figure 6-41 Manifest Constants and Data Types from <dlfcn.h>

```

/* Valid values for mode argument to dlopen. */
#define RTLD_LAZY 1 /* lazy function call binding */
#define RTLD_NOW 2 /* immediate function call binding */
#define RTLD_NEXT (void *)-1
#define RTLD_GLOBAL 0x100

typedef struct dl_info {
    const char *dli_fname;
    void *dli_fbase;
    const char *dli_sname;
    void *dli_saddr;
} Dl_info;

```

libelf - Executable Linking Format Library

Overview

The interfaces specified in this section provide applications with the ability to manipulate executable and linking format (ELF) object files, archive files, and archive members. The interface set described here resides entirely in the REQUIRED dynamic library: */usr/lib/libelf.so.1*. Interface members of this library, listed in the table below, are REQUIRED unless explicitly noted otherwise.

SCD Extension to the System V ABI

The SCD requires a new library: */usr/lib/libelf.so.1*, which has functions not specified by the gABI. These functions are not defined in the SVID. Except for *nlist*, the semantics for these additional function definitions are available in the SCD Interface Semantics document. Structures and manifest constants pertaining to these functions are as specified further below.

Library Contents

Table 6-14 libelf contents

elf32_fsize	2.4	elf_getarhdr	2.4	2.4 - Interfaces added in SCD2.4
elf32_getehdr	2.4	elf_getarsym	2.4	Note: Except for <i>nlist</i> , all interfaces defined in the SCD Interface Semantics.
elf32_getphdr	2.4	elf_getbase	2.4	
elf32_getshdr	2.4	elf_getdata	2.4	
elf32_newehdr	2.4	elf_getident	2.4	
elf32_newphdr	2.4	elf_getscn	2.4	
elf32_xlatetof	2.4	elf_hash	2.4	
elf32_xlatetom	2.4	elf_kind	2.4	
elf_begin	2.4	elf_ndxscn	2.4	
elf_cntl	2.4	elf_newdata	2.4	
elf_end	2.4	elf_newscn	2.4	
elf_errmsg	2.4	elf_next	2.4	
elf_errno	2.4	elf_nextscn	2.4	
elf_fill	2.4	elf_rand	2.4	
elf_flagdata	2.4	elf_rawdata	2.4	
elf_flagehdr	2.4	elf_rawfile	2.4	
elf_flagelf	2.4	elf_strptr	2.4	
elf_flagphdr	2.4	elf_update	2.4	
elf_flagscn	2.4	elf_version	2.4	
elf_flagshdr	2.4	nlist	2.4	

Structures and Manifest Constants

Figure 6-42 Manifest Constants and Data Types from <elf.h>

```

typedef unsigned long      Elf32_Addr;
typedef unsigned short    Elf32_Half;
typedef unsigned long     Elf32_Off;
typedef long              Elf32_Sword;
typedef unsigned long     Elf32_Word;
typedef long              off_t;
typedef long              time_t;
typedef unsigned int      size_t;
typedef void              Elf_Void;
typedef struct Elf        Elf;
typedef struct Elf_Scn    Elf_Scn;

#define ELF32_FSZ_ADDR      4
#define ELF32_FSZ_HALF    2
#define ELF32_FSZ_OFF     4
#define ELF32_FSZ_SWORD   4
#define ELF32_FSZ_WORD    4
#define EI_NIDENT          16
#define EI_MAG0            0
#define EI_MAG1            1
#define EI_MAG2            2
#define EI_MAG3            3
#define EI_CLASS           4
#define EI_DATA            5
#define EI_VERSION         6
#define ELFMAG0            0x7f
#define ELFMAG1            'E'
#define ELFMAG2            'L'
#define ELFMAG3            'F'
#define ELFCLASSNONE      0
#define ELFCLASS32        1
#define ELFCLASS64        2
#define ELFDATANONE       0
#define ELFDATA2LSB       1
#define ELFDATA2MSB       2
#define EV_NONE           0
#define EV_CURRENT        1
#define SHT_NULL           0
#define SHT_PROGBITS       1
#define SHT_SYMTAB         2
#define SHT_STRTAB         3
#define SHT_RELA           4
#define SHT_HASH            5
#define SHT_DYNAMIC        6
#define SHT_NOTE           7
#define SHT_NOBITS         8
#define SHT_REL            9
#define SHT_DYNSYM         11
#define ELF_F_DIRTY        0x1
#define ELF_F_LAYOUT       0x4

```

```
typedef struct {
    unsigned char e_ident[EI_NIDENT];
    Elf32_Half    e_type;
    Elf32_Half    e_machine;
    Elf32_Word    e_version;
    Elf32_Addr    e_entry;
    Elf32_Off    e_phoff;
    Elf32_Off    e_shoff;
    Elf32_Word    e_flags;
    Elf32_Half    e_ehsize;
    Elf32_Half    e_phentsize;
    Elf32_Half    e_phnum;
    Elf32_Half    e_shentsize;
    Elf32_Half    e_shnum;
    Elf32_Half    e_shstrndx;
} Elf32_Ehdr;

typedef struct {
    Elf32_Word    p_type;
    Elf32_Off    p_offset;
    Elf32_Addr    p_vaddr;
    Elf32_Addr    p_paddr;
    Elf32_Word    p_filesz;
    Elf32_Word    p_memsz;
    Elf32_Word    p_flags;
    Elf32_Word    p_align;
} Elf32_Phdr;

typedef struct {
    Elf32_Word    sh_name;
    Elf32_Word    sh_type;
    Elf32_Word    sh_flags;
    Elf32_Addr    sh_addr;
    Elf32_Off    sh_offset;
    Elf32_Word    sh_size;
    Elf32_Word    sh_link;
    Elf32_Word    sh_info;
    Elf32_Word    sh_addralign;
    Elf32_Word    sh_entsize;
} Elf32_Shdr;

typedef enum {
    ELF_C_NULL = 0, ELF_C_READ, ELF_C_WRITE, ELF_C_CLR, ELF_C_SET, ELF_C_FDDONE,
    ELF_C_FDREAD, ELF_C_RDWR, ELF_C_NUM
} Elf_Cmd;

typedef enum {
    ELF_T_BYTE = 0, ELF_T_ADDR, ELF_T_DYN, ELF_T_EHDR,
    ELF_T_HALF, ELF_T_OFF, ELF_T_PHDR, ELF_T_RELA, ELF_T_REL,
    ELF_T_SHDR, ELF_T_SWORD, ELF_T_SYM, ELF_T_WORD, ELF_T_NUM
} Elf_Type;

typedef struct {
    Elf_Void    *d_buf;
    Elf_Type    d_type;
    size_t      d_size;
    off_t       d_off;
    size_t      d_align;
    unsigned    d_version;
} Elf_Data;
```



```

typedef struct {
    char          *ar_name;
    time_t        ar_date;
    long          ar_uid;
    long          ar_gid;
    unsigned long ar_mode;
    off_t         ar_size;
    char          *ar_rawname;
} Elf_Arhdr;

typedef struct {
    char          *as_name;
    size_t        as_off;
    unsigned long as_hash;
} Elf_Arsym;

typedef enum {
    ELF_K_NONE = 0, ELF_K_AR, ELF_K_COFF, ELF_K_ELF, ELF_K_NUM
} Elf_Kind;

typedef struct {
    Elf32_Word    st_name;
    Elf32_Addr    st_value;
    Elf32_Word    st_size;
    unsigned char st_info;
    unsigned char st_other;
    Elf32_Half    st_shndx;
} Elf32_Sym;

typedef struct {
    Elf32_Addr    r_offset;
    Elf32_Word    r_info;
} Elf32_Rel;

typedef struct {
    Elf32_Addr    r_offset;
    Elf32_Word    r_info;
    Elf32_Sword   r_addend;
} Elf32_Rela;

typedef struct {
    Elf32_Sword   d_tag;
    union {
        Elf32_Word d_val;
        Elf32_Addr d_ptr;
        Elf32_Off  d_off;
    } d_un;
} Elf32_Dyn;

```

Figure 6-43 Manifest Constants and Data Types from <nlist.h>

```

struct nlist {
    char          *n_name;
    long          n_value;
    short         n_scnm;
    unsigned short n_type;
    char          n_sclass;
    char          n_numaux;
};

```

libintl - Internationalization Library - EXPERIMENTAL

Overview

The interfaces specified in this section provide support to applications in retrieving translated text strings from internationalized message objects. The interface set described here resides entirely in the EXPERIMENTAL dynamic library: */usr/lib/libintl.so.1*. Interface members of this library, listed in the table below, are EXPERIMENTAL unless explicitly noted otherwise.

SCD Extensions to the System V ABI

The SCD requires a new library: */usr/lib/libintl.so.1*, which has functions not specified by the gABI. These functions are not defined in the SVID. The semantics for these additional function definitions are available in the SCD Interface Semantics document.

Library Contents

Table 6-15 libintl contents

<code>bindtextdomain</code>	2.4 (E)	<code>gettext</code>	2.4 (E)	2.4 - Interfaces added to SCD2.4
<code>dcgettext</code>	2.4 (E)	<code>textdomain</code>	2.4 (E)	E - EXPERIMENTAL interface
<code>dgettext</code>	2.4 (E)			

libm - Math Library

Overview

The interfaces specified in this section provide several mathematical functions. The interface set described here resides entirely in the REQUIRED dynamic library: */usr/lib/libm.so.1*. Interface members of this library, listed in the table below, are REQUIRED unless explicitly noted otherwise.

SCD Extensions to the System V ABI

The SCD requires a new library: */usr/lib/libm.so.1*, which has functions not specified by the gABI. These functions are not defined in the SVID. The semantics for these additional function definitions are available in the SCD Interface Semantics document. Structures and manifest constants pertaining to these functions are as specified further below.

Library Contents

Table 6-16 libm contents

acos	2.4	erf	2.4	jn	2.4	sqrt	2.4
acosh	2.4	erfc	2.4	lgamma	2.4	tan	2.4
asin	2.4	exp	2.4	log	2.4	tanh	2.4
asinh	2.4	expml *	2.4	log10	2.4	y0	2.4
atan	2.4	fabs	2.4	loglp *	2.4	y1	2.4
atan2	2.4	floor	2.4	pow	2.4	yn	2.4
atanh	2.4	fmod	2.4	remainder	2.4		
cbrt	2.4	gamma	2.4	rint *	2.4		
ceil	2.4	hypot	2.4	scalbn *	2.4 E		
copysign *	2.4 E	ilogb *	2.4	significantd *	2.4 E		* - Defined in the SCD Interface Semantics.
cos	2.4	j0	2.4	sin	2.4		2.4 - Interfaces added in SCD2.4
cosh	2.4	j1	2.4	sinh	2.4		

Structures and Manifest Constants

Figure 6-44 Manifest Constants and Data Types from <math.h>

```

struct exception {
    int         type;
    char        *name;
    double      arg1;
    double      arg2;
    double      retval;
};

#define INT_MIN      (-2147483647-1)
#define INT_MAX      2147483647
#define DBL_MANT_DIG 53
#define DBL_MAX_EXP  1024

```

libnisdb - Network Infor. Services Database library - EXPERIMENTAL

Overview

The interfaces specified in this section provide network information services database access functions.

The interface set described here resides entirely in the EXPERIMENTAL dynamic library: */usr/lib/libnisdb.so.1*. Interface members of this library, listed in the table below, are EXPERIMENTAL unless explicitly noted otherwise.

SCD Extensions to the System V ABI

The SCD requires a new library: */usr/lib/libnisdb.so.1*, which has functions not specified by the gABI. These functions are not defined in the SVID. The semantics for these additional function definitions are available in the SCD Interface Semantics document. Structures and manifest constants pertaining to these functions are as specified further below.

Library Contents

Table 6-17 libnisdb contents

db_add_entry	2.4E	db_list_entries	2.4E	2.4E - EXPERIMENTAL interfaces added in SCD 2.4
db_checkpoint	2.4E	db_next_entry	2.4E	
db_create_table	2.4E	db_remove_entry	2.4E	
db_destroy_table	2.4E	db_reset_next_entry	2.4E	
db_first_entry	2.4E	db_standby	2.4E	
db_free_result	2.4E	db_table_exists	2.4E	
db_initialize	2.4E	db_unload_table	2.4E	

Structures and Manifest Constants

Figure 6-45 Manifest Constants and Data Types from <rpcsvc/nis_db.h>

```
enum db_status {
    DB_SUCCESS = 0,
    DB_NOTFOUND = 1,
    DB_NOTUNIQUE = 2,
    DB_BADTABLE = 3,
    DB_BADQUERY = 4,
    DB_BADOBJECT = 5,
    DB_MEMORY_LIMIT = 6,
    DB_STORAGE_LIMIT = 7,
    DB_INTERNAL_ERROR = 8
};
typedef enum db_status db_status;
enum db_action {
    DB_LOOKUP = 0,
    DB_REMOVE = 1,
    DB_ADD = 2,
    DB_FIRST = 3,
    DB_NEXT = 4,
    DB_ALL = 5,
    DB_RESET_NEXT = 6
};
typedef enum db_action db_action;
typedef entry_obj *entry_object_p;
typedef struct {
    u_int      db_next_desc_len;
    char      *db_next_desc_val;
} db_next_desc;

struct db_result {
    db_status  status;
    db_next_desc nextinfo;
    struct {
        u_int      objects_len;
        entry_object_p *objects_val;
    } objects;
    long      ticks;
};
typedef struct db_result db_result;
```

libnsl - The Network Services Library

Overview

The interfaces specified in this section provide REQUIRED libnsl network services and Internet address manipulation interfaces listed in the *System V Application Binary Interface*, and described in the *System V Interface Definition, Third Edition*, along with additional ABI extensions.

The interface set described here resides entirely in the REQUIRED dynamic library: */usr/lib/libnsl.so.1*. Interface members of this library, listed in the table below, are REQUIRED unless explicitly noted otherwise.

The libnsl ABI Interfaces

The interfaces listed below have been included in SCD the because they are REQUIRED to be present on all systems conforming to the *System V Application Binary Interface*, in the dynamic library: */usr/lib/libnsl.so.1*.

The ABI interfaces found in the tables below are REQUIRED to be present on an ABI-conforming system. Systems without networking capabilities are not required to implement these interfaces, but must provide an entry point in libnsl for each. Entry points which are provided as stubs and not implemented must fail normally and set the external symbol `errno` to `ENOSYS`.

SCD Extensions to the System V ABI

The SCD specifies the library: */usr/lib/libnsl.so.1*, which has functions not specified by, or different from, the gABI. These functions are either not defined in the SVID, or are defined differently in the SCD than in the SVID. The semantics for these additional/modified function definitions are available in the SCD Interface Semantics document. Exported data, structures and manifest constants pertaining to these functions are as specified further below

Exported Data

Table 6-18 Exported data for libnsl

<code>_nderror[0x4]</code>	<code>svc_fdset</code> ¹	1- see libnsl library changes at the beginning of this chapter.
<code>_null_auth</code> ²	<code>t_errno[0x4]</code>	2- DEPRECATED
<code>rpc_createerr[0x16]</code>		

Library Contents

Table 6-19 libnsl contents

auth_destroy	2.4	getnetname		nis_lerror	2.4,E
authdes_create	*2.4,E	getnetpath		nis_list	2.4,E
authdes_getucred	*2.4	getpublickey		nis_local_directory	2.4,E
authdes_seccreate		getrpcbyname	*2.4	nis_local_group	2.4,E
authnone_create		getrpcbynumber	*2.4	nis_local_host	2.4,E
authsys_create		getrpccent	*2.4	nis_local_principal	2.4,E
authsys_create_default		getsecretkey		nis_lookup	2.4,E
callrpc	*2.4,E	host2netname		nis_mkdir	2.4,E
clnt_broadcast	*2.4,E	inet_addr	2	nis_modify	2.4,E
clnt_call	*2.4	inet_netof	2	nis_modify_entry	2.4,E
clnt_control	*2.4	inet_ntoa	2	nis_name_of	2.4,E
clnt_create	*	key_decryptsession		nis_next_entry	2.4,E
clnt_create_vers	*2.4,E	key_encryptsession		nis_perror	2.4,E
clnt_destroy	*2.4	key_gendes		nis_ping	2.4,E
clnt_dg_create	*	key_setsecret		nis_print_group_entry	2.4,E
clnt_freeres	2.4	nc_perror		nis_print_object	2.4,E
clnt_geterr	2.4	nc_sperror	*2.4,E	nis_remove	2.4,E
clnt_pcreateerror	*	netdir_free		nis_remove_entry	2.4,E
clnt_perrno		netdir_getbyaddr		nis_removemember	2.4,E
clnt_perror		netdir_getbyname		nis_rmdir	2.4,E
clnt_raw_create	*	netdir_options		nis_servstate	2.4,E
clnt_spcreateerror	*	netdir_perror	*2.4,E	nis_sperrno	2.4,E
clnt_sperrno		netdir_sperror	*2.4,E	nis_sperror	2.4,E
clnt_sperror		netname2host		nis_stats	2.4,E
clnt_tli_create	*	netname2user		nis_verifygroup	2.4,E
clnt_tp_create	*	nis_add	2.4,E	pmap_getmaps	*2.4
clnt_vc_create	*	nis_add_entry	2.4,E	pmap_getport	*2.4
clntraw_create	*2.4,E	nis_addmember	2.4,E	pmap_rmtcall	*2.4
clnttcp_create	*2.4,E	nis_checkpoint	2.4,E	pmap_set	*2.4
clntudp_bufcreate	*2.4,E	nis_clone_object	2.4,E	pmap_unset	*2.4
clntudp_create	*2.4,E	nis_creategroup	2.4,E	registerrpc	*2.4,E
dial	*2.4,E	nis_destroy_object	2.4,E	rpc_broadcast	*2.4
doconfig	*2.4,E	nis_destroygroup	2.4,E	rpc_broadcast_exp	*1
endhostent	2.4	nis_dir_cmp	2.4,E	rpc_call	1
endnetconfig		nis_domain_of	2.4,E	rpc_reg	*
endnetpath		nis_first_entry	2.4,E	rpcb_getaddr	
endrpcent	*2.4	nis_freenames	2.4,E	rpcb_getmaps	
freenetconfigent		nis_freeresult	2.4,E	rpcb_gettime	
get_myaddress	*2.4,E	nis_freeservlist	2.4,E	rpcb_rmtcall	
gethostbyaddr	2	nis_freetags	2.4,E	rpcb_set	
gethostbyname	2	nis_getnames	2.4,E	rpcb_unset	
gethostent	2.4	nis_getservlist	2.4,E		
getnetconfig		nis_ismember	2.4,E		(See numbered notes in following table)
getnetconfigent		nis_leaf_of	2.4,E		

Table 6-19 (cont'd) libnsl contents

sethostent	2.4	t_free	xdr_reference
setnetconfig		t_getinfo	xdr_rejected_reply
setnetpath		t_getstate	1 xdr_replymsg
setrpcnt	* 2.4	t_listen	xdr_setpos *2.4
svc_auth_reg	* 2.4,E	t_look	xdr_short
svc_create		t_open	xdr_sizeof *E
svc_destroy	2.4	t_optmgmt	xdr_string
svc_dg_create		t_rcv	xdr_u_char
svc_dg_enablecache*	2.4,E	t_rcvconnect	xdr_u_int * 2.4
svc_exit	* 2.4,E	t_rcvdis	xdr_u_long
svc_fd_create		t_rcvrel	xdr_u_short
svc_freeargs	*2.4	t_rcvudata	xdr_union
svc_getargs	*2.4	t_rcvuderr	xdr_vector
svc_getcaller	* 2.4, E	t_snd	xdr_void
svc_getreq_common	* 2.4,E	t_snddis	xdr_wrapstring
svc_getreq_poll	* 2.4,E	t_sndrel	xdrmem_create
svc_getreqset		t_sndudata	xdrrec_create
svc_getrpccaller	*2.4	t_strerror	xdrrec_endofrecord* 2.4,E
svc_pollset	*2.4,E	t_sync	xdrrec_eof
svc_raw_create		t_unbind	xdrrec_readbytes * 2.4,E
svc_reg		taddr2uaddr	xdrrec_skiprecord *
svc_register	* 2.4	uaddr2taddr	xdrstdio_create
svc_run		undial	xprt_register
svc_sendreply		user2netname	xprt_unregister
svc_tli_create		xdr_accepted_reply	yp_all 2.4,E,
svc_tp_create		xdr_array	yp_bind 2.4,E
svc_reg		xdr_authsys_parms	yp_first 2.4,E
svc_unregister	* 2.4	xdr_authunix_params	yp_get_default_domain2.4,E
svc_vc_create		xdr_bool	yp_master 2.4,E
svcerr_auth		xdr_bytes	yp_match 2.4,E
svcerr_decode		xdr_callhdr	yp_next 2.4,E
svcerr_noproc		xdr_callmsg	yp_order 2.4,E
svcerr_noprog		xdr_char	yp_unbind 2.4,E
svcerr_progvers	1	xdr_control	yperr_string 2.4,E
svcerr_systemerr		xdr_destroy	ypprot_err 2.4,E
svcerr_weakauth		xdr_double	
svcfld_create	* 2.4,E	xdr_enum	
svccraw_create	* 2.4,E	xdr_float	
svctcp_create	* 2.4	xdr_free	
svcudp_bufcreate	* 2.4,E	xdr_getpos	*2.4
svcudp_create		xdr_inline	*2.4
t_accept		xdr_int	
t_alloc		xdr_long	
t_bind		xdr_opaque	
t_close		xdr_opaque_auth	
t_connect		xdr_pointer	
t_error		xdr_quadruple	*2.4,E

* - Defined in the SCD Interface Semantics document.

1 - See Network Services Library Changes section at the beginning of this chapter.

2 - Socket support routines.

E- EXPERIMENTAL Interfaces

2.3 - Interfaces added in SCD2.3

2.4 - Interfaces added or changed in SCD2.4

Structures and Manifest Constants

Figure 6-46 Manifest Constants and Data Types from <rpc/xdr.h>

```

struct netobj {
    u_int      n_len;
    char      *n_bytes;
};
typedef struct netobj  netobj;

```

Figure 6-47 Manifest Constants and Data Types from <dial.h>

```

#define INTRPT (-1)
#define D_HUNG (-2)
#define NO_ANS (-3)
#define ILL_BD (-4)
#define A_PROB (-5)
#define L_PROB (-6)
#define NO_Ldv (-7)
#define DV_NT_A (-8)
#define DV_NT_K (-9)
#define NO_BD_A (-10)
#define NO_BD_K (-11)
#define DV_NT_E (-12)
#define BAD_SYS (-13)
typedef struct {
    struct termio *attr;
    int          baud;
    int          speed;
    char        *line;
    char        *telno;
    int         modem;
    char        *device;
    int         dev_len;
} CALL;

```

Figure 6-48 Manifest Constants and Data Types from <sac.h>

```

#define NOASSIGN      0x1
#define NORUN        0x2

```

Figure 6-49 Manifest Constants and Data Types from <rpc/rpcent.h>

```
struct rpcent {
    char      *r_name;
    char      **r_aliases;
    int       r_number;
};
```

Figure 6-50 Manifest Constants and Data Types from <rpc/rpc.h>

```
typedef bool_t      (*resultproc_t)(caddr_t,...);
struct pmap {
    u_long          pm_prog;
    u_long          pm_vers;
    u_long          pm_prot;
    u_long          pm_port;
};
typedef struct pmap pmap;
typedef pmap        PMAP;
struct pmaplist{
    PMAP            pml_map;
    struct pmaplist *pml_next;
};
#define IPPROTO_TCP      6
#define IPPROTO_UDP      17
```

libposix4 - POSIX4 Library - EXPERIMENTAL

Overview

The interfaces specified in this section are based on interfaces provided by the POSIX 1003.4 standard. Many of these interfaces support POSIX asynchronous I/O. The interface set described here resides entirely in the EXPERIMENTAL dynamic library: */usr/lib/libposix4.so.1*. Interface members of this library, listed in the table below, are EXPERIMENTAL.

Rationale:

Interfaces specified here have not yet been implemented by multiple system vendors and are therefore EXPERIMENTAL.

SCD Extensions to the System ABI

The SCD requires a new library: */usr/lib/libposix4.so.1*, which has functions not specified by the gABI. These functions are not defined in the SVID.

Library Contents

Table 6-20 libposix4 contents

aio_cancel		mq_open	shm_open
aio_cancel64	L 2.4	mq_receive	shm_unlink
aio_error		mq_send.	sigqueue
aio_error64	L 2.4	mq_setattr	sigtimedwait
aio_fsync		mq_unlink	sigwaitinfo
aio_fsync64	L 2.4	nanosleep	timer_create
aio_read		sched_get_priority_max	timer_delete
aio_read64	L 2.4	sched_get_priority_min	timer_getoverrun
aio_return		sched_getparam	timer_gettime
aio_return64	L 2.4	sched_getscheduler	timer_settime
aio_suspend		sched_rr_get_interval	
aio_suspend64	L 2.4	sched_setparam	
aio_write		sched_setscheduler	L - 32-bit ABI Large file support routine.
aio_write64	L 2.4	sched_yield	2.4 - Interfaces added in SCD2.4.
clock_getres		sem_close	
clock_gettime		sem_destroy	
clock_settime		sem_getvalue	
fdatasync		sem_init	
lio_listio		sem_open	
lio_listio64	L 2.4	sem_post	
mq_close		sem_trywait	
mq_getattr		sem_unlink	
mq_notify		sem_wait	

libpthread - POSIX Multithreading Library - EXPERIMENTAL

Overview

The POSIX thread library, libpthread contains several functions listed in the table “libpthread Contents” below. The libposix4 library is a EXPERIMENTAL interface set with reference name */usr/lib/libpthread.so.1*. The interfaces specified in this section provide applications with POSIX multithreaded services (multiple “threads of control”).

The interface set described here resides entirely in the EXPERIMENTAL dynamic library: */usr/lib/libpthread.so.1*. Interface members of this library, listed in the table below, are EXPERIMENTAL.

Rationale:

Interfaces specified here have not yet been implemented by multiple system vendors and are therefore EXPERIMENTAL.

SCD Extensions to the System V ABI

The SCD requires a new library: */usr/lib/libpthread.so.1*, which has functions not specified by the gABI. These functions are not defined in the SVID. The semantics for these additional function definitions are available in the *Single UNIX Specification, Version 2*. Structures and manifest constants pertaining to these functions are as specified further below.

Library Contents

Table 6-21 libpthread contents

pthread_atfork	(1) 2.4	pthread_mutex_destroy	(1) 2.4
pthread_attr_destroy	(1) 2.4	pthread_mutex_getprioceiling	(1) 2.4
pthread_attr_getdetachstate	(1) 2.4	pthread_mutex_init	(1) 2.4
pthread_attr_getguardsize	(1) 2.4	pthread_mutex_lock	(1) 2.4
pthread_attr_getinheritsched	(1) 2.4	pthread_mutex_setprioceiling	(1) 2.4
pthread_attr_getschedparam	(1) 2.4	pthread_mutex_trylock	(1) 2.4
pthread_attr_getschedpolicy	(1) 2.4	pthread_mutex_unlock	(1) 2.4
pthread_attr_getscope	(1) 2.4	pthread_mutexattr_destroy	(1) 2.4
pthread_attr_getstackaddr	(1) 2.4	pthread_mutexattr_getprioceiling	(1) 2.4
pthread_attr_getstacksize	(1) 2.4	pthread_mutexattr_getprotocol	(1) 2.4
pthread_attr_init	(1) 2.4	pthread_mutexattr_getpshared	(1) 2.4
pthread_attr_setdetachstate	(1) 2.4	pthread_mutexattr_gettype	(1) 2.4
pthread_attr_setguardsize	(1) 2.4	pthread_mutexattr_init	(1) 2.4
pthread_attr_setinheritsched	(1) 2.4	pthread_mutexattr_setprioceiling	(1) 2.4
pthread_attr_setschedparam	(1) 2.4	pthread_mutexattr_setprotocol	(1) 2.4
pthread_attr_setschedpolicy	(1) 2.4	pthread_mutexattr_setpshared	(1) 2.4
pthread_attr_setscope	(1) 2.4	pthread_mutexattr_settype	(1) 2.4
pthread_attr_setstackaddr	(1) 2.4	pthread_once	(1) 2.4
pthread_attr_setstacksize	(1) 2.4	pthread_rwlock_destroy	(1) 2.4
pthread_cancel	(1) 2.4	pthread_rwlock_init	(1) 2.4
pthread_cleanup_pop	(1) 2.4	pthread_rwlock_rdlock	(1) 2.4
pthread_cleanup_push	(1) 2.4	pthread_rwlock_tryrdlock	(1) 2.4
pthread_cond_broadcast	(1) 2.4	pthread_rwlock_trywrlock	(1) 2.4
pthread_cond_destroy	(1) 2.4	pthread_rwlock_unlock	(1) 2.4
pthread_cond_init	(1) 2.4	pthread_rwlock_wrlock	(1) 2.4
pthread_cond_signal	(1) 2.4	pthread_rwlockattr_destroy	(1) 2.4
pthread_cond_timedwait	(1) 2.4	pthread_rwlockattr_getshared	(1) 2.4
pthread_cond_wait	(1) 2.4	pthread_rwlockattr_init	(1) 2.4
pthread_condattr_destroy	(1) 2.4	pthread_rwlockattr_setshared	(1) 2.4
pthread_condattr_getpshared	(1) 2.4	pthread_self	(1) 2.4
pthread_condattr_init	(1) 2.4	pthread_setcancelstate	(1) 2.4
pthread_condattr_setpshared	(1) 2.4	pthread_setcanceltype	(1) 2.4
pthread_create	(1) 2.4	pthread_setconcurrency	(1) 2.4
pthread_detach	(1) 2.4	pthread_setschedparam	(1) 2.4
pthread_equal	(1) 2.4	pthread_setspecific	(1) 2.4
pthread_exit	(1) 2.4	pthread_sigmask	(1) 2.4
pthread_getconcurrency	(1) 2.4	pthread_testcancel	(1) 2.4
pthread_getschedparam	(1) 2.4		
pthread_getspecific	(1) 2.4		
pthread_join	(1) 2.4		
pthread_key_create	(1) 2.4		
pthread_key_delete	(1) 2.4		
pthread_kill	(1) 2.4		

1 - Found in *The Single UNIX Specification, Version 2*.

2.4 - Interfaces added in SCD2.4.

Structures and Manifest Constants

Figure 6-51 Manifest Constants and Data types from <sched.h>

```
#define SCHED_OTHER    0
#define SCHED_FIFO    1
#define SCHED_RR      2
```

Figure 6-52 Manifest Constants and Data types from <pthread.h>

```
typedef unsigned int    pthread_t;
typedef unsigned int    pthread_key_t;

typedef struct _pthread_attr {
    void                *pthread_attrp;
} pthread_attr_t;

typedef struct _pthread_mutexattr {
    void                *pthread_mutexattrp;
} pthread_mutexattr_t;

typedef struct _pthread_mutex {
    struct {
        unsigned char    pthread_mutex_flag[4];
        unsigned long    pthread_mutex_type;
    } pthread_mutex_flags;
    union {
        struct {
            unsigned char    pthread_mutex_pad[8];
        } pthread_mutex_lock64;
        unsigned long long    pthread_mutex_owner64;
    } pthread_mutex_lock;
    unsigned long long    pthread_mutex_data;
} pthread_mutex_t;

typedef struct _pthread_cond {
    struct {
        unsigned char    pthread_cond_flag[4];
        unsigned long    pthread_cond_type;
    } pthread_cond_flags;
    unsigned long long    pthread_cond_data;
} pthread_cond_t;

typedef struct _pthread_condattr {
    void                *pthread_condattrp;
} pthread_condattr_t;

typedef struct _once {
    unsigned long long    pthread_once_pad[4];
} pthread_once_t;

typedef struct _pthread_rwlock {
    int                    __pthread_rwlock_readers;
    unsigned short        __pthread_rwlock_type;
    unsigned short        __pthread_rwlock_magic;
    unsigned long long    __pthread_rwlock_pad1[3];
    unsigned long long    __pthread_rwlock_pad2[2];
    unsigned long long    __pthread_rwlock_pad3[2];
} pthread_rwlock_t;
```

```
#define PTHREAD_CANCEL_ENABLE      0x00
#define PTHREAD_CANCEL_DISABLE     0x01
#define PTHREAD_CANCEL_DEFERRED   0x00
#define PTHREAD_CANCEL_ASYNCHRONOUS 0x02
#define PTHREAD_CANCELED           -19
#define PTHREAD_COND_INITIALIZER   {{{0}, 0}, 0}
#define PTHREAD_CREATE_DETACHED    0x40
#define PTHREAD_CREATE_JOINABLE    0
#define PTHREAD_INHERIT_SCHED      1
#define PTHREAD_EXPLICIT_SCHED     0
#define PTHREAD_MUTEX_DEFAULT      PTHREAD_MUTEX_NORMAL
#define PTHREAD_MUTEX_ERRORCHECK   2
#define PTHREAD_MUTEX_NORMAL       0
#define PTHREAD_MUTEX_INITIALIZER  {{{0}, 0}, {{{0}}}, 0}
#define PTHREAD_MUTEX_RECURSIVE   4
#define PTHREAD_ONCE_DONE          1
#define PTHREAD_ONCE_NOTDONE       0
#define PTHREAD_ONCE_INIT          {0, 0, 0, PTHREAD_ONCE_NOTDONE}
#define PTHREAD_PROCESS_SHARED     1
#define PTHREAD_PROCESS_PRIVATE    0
#define PTHREAD_SCOPE_SYSTEM       0x01
#define PTHREAD_SCOPE_PROCESS      0
#define PTHREAD_RWLOCK_INITIALIZER {{{0}, 0}, {{{0}}}, 0}
```

libresolv - Domain Name Services Library - EXPERIMENTAL

Overview

The interfaces specified in this section provide applications with domain name access services. The interface set described here resides entirely in the EXPERIMENTAL dynamic library: */usr/lib/libresolv.so.1*. Interface members of this library, listed in the table below, are EXPERIMENTAL unless explicitly noted otherwise.

SCD Extensions to the System V ABI

The SCD requires a new library: */usr/lib/libresolv.so.1*, which has functions not specified by the gABI. These functions are not defined in the SVID. The semantics for these additional function definitions are available in the SCD Interface Semantics document. Structures and manifest constants pertaining to these functions are as specified further below.

Library Contents

Table 6-22 libresolv contents

dn_comp	2.4 E	res_mkquery	2.4 E	2.4 - Interfaces added in SCD2.4.
dn_expand	2.4 E	res_send	2.4 E	E - EXPERIMENTAL interfaces
res_init	2.4 E			

Structures and Manifest Constants

Figure 6-53 Manifest Constants and Data Types from <arpa/nameser.h>

```
struct rrec {
    short    r_zone;
    short    r_class;
    short    r_type;
    u_long   r_ttl;
    int      r_size;
    char     *r_data;
};

#define QUERY           0x0
#define IQUERY         0x1
#define STATUS         0x2
#define UPDATEEA       0x9
#define UPDATED        0xa
#define UPDATEM        0xc
#define UPDATEMA       0xd
#define ZONEINIT       0xe
#define ZONEREF        0xf
#define MAXDNAME       256
```


Figure 6-54 Manifest Constants and Data Types from <resolv.h>

```
#define RES_INIT          0x0001
#define RES_DEBUG        0x0002
#define RES_AAONLY       0x0004
#define RES_USEVC        0x0008
#define RES_IGNTC        0x0020
#define RES_RECURSE      0x0040
#define RES_DEFNAMES     0x0080
#define RES_STAYOPEN     0x0100
#define RES_DNSRCH       0x0200
#define MAXNS            3

struct state {
    int             retrans;
    int             retry;
    long            options;
    int             nscount;
    struct sockaddr_in nsaddr_list[MAXNS];
#define nsaddr      nsaddr_list[0]
    u_short         id;
    char             defdname[MAXDNAME];
    char             *dnsrch[MAXDNSRCH+1];
    int             ascount;
    struct          in_addr sort_list[MAXADDR];
};
extern struct state _res;
```

librpcsvc - Remote Procedure Call Services Library - EXPERIMENTAL

Overview

The interfaces specified in this section provide applications with remote procedure call services. The interface set described here resides entirely in the EXPERIMENTAL dynamic library: */usr/lib/librpcsvc.so.1*. Interface members of this library, listed in the table below, are EXPERIMENTAL unless explicitly noted otherwise.

SCD Extensions to the System V ABI

The SCD requires a new library: */usr/lib/librpcsvc.so.1*, which has functions not specified by the gABI. These functions are not defined in the SVID. The semantics for these additional functions are not available in the SCD 2.4 Interface Semantics document.

Library Contents

Table 6-23 librpcsvc contents

rnusers	2.4 E	xdr_sprayarr	2.4 E	2.4 - Interfaces added in SCD2.4.
rusers	2.4 E	xdr_spraycumul	2.4 E	E - EXPERIMENTAL interfaces
rwall	2.4 E			

libsocket - Socket Library¹

Overview

The interfaces specified in this section provide applications with socket inter-networking interfaces, primarily used with the TCP/IP protocol suite.

The interface set described here resides entirely in the REQUIRED dynamic library: */usr/lib/libsocket.so.1*. Interface members of this library, listed in the table below, are REQUIRED unless explicitly noted otherwise.

ABI Extensions to the System V ABI

The SCD specifies a new library: */usr/lib/libsocket.so.1*, which has functions not specified by the gABI. These functions are not defined in the SVID. The semantics for these additional function definitions are available in the SCD Interface Semantics document. Structures and manifest constants pertaining to these functions are as specified further below.

Library Contents

Table 6-24 Socket Functions in libsocket

accept		getprotobyname		recv		2.4 - Interfaces added or changed in SCD2.4.
bind		getprotoent		recvfrom		
connect		getservbyname		recvmsg		E- EXPERIMENTAL interfaces
endnetent	E 2.4	getservbyport		rexec	E 2.4	
endprotoent	E 2.4	getservent	2.4	rresvport	E 2.4	
endservent	E 2.4	getsockname		ruserok	E 2.4	
ether_aton	E 2.4	getsockopt		send		
ether_hostton	E 2.4	htonl	E 2.4	sendmsg		
ether_line	2.4	htons	2.4	sendto		
ether_ntoa	E 2.4	inet_lnaof		setnetent	E	
ether_ntohost	2.4	inet_makeaddr		setprotoent	E 2.4	
getnetbyaddr	E 2.4	inet_network		setservent	E 2.4	
getnetbyname	2.4	listen		setsockopt		
getnetent	2.4	ntohl	E 2.4	shutdown		
getpeername		ntohs	2.4	socket		
getprotobyname		rcmd	E 2.4	socketpair	2.4	

Structures and Manifest Constants

The Figures below contain the values of manifest constants and type declarations of the data types needed for the socket functions.

1. Earlier this interface set was DEPRECATED (November 1st, 1993 for possible removal in Nov 1996) in favor of the transport independent interfaces offered as part of the System V standard. However the sockets interface has subsequently been seen to have widespread use in applications and is now considered to be of equal importance and validity as an application interface set. This interface was therefore restored to the REQUIRED state in SCD2.4 in December 1997.

Figure 6-55 Manifest Constants and Data Types from <errno.h>

```

#define NOBUFS                132
#define ENOPROTOPT           99
#define EADDRINUSE           125        /* Address already in use */
#define EADDRNOTAVAIL        126        /* Can't assign requested address */
#define EAFNOSUPPORT         124        /* Address family not supported by
#define EALREADY             149        /* operation already in progress */
#define ECONNREFUSED         146        /* Connection refused */
#define EINPROGRESS          150        /* operation now in progress */
#define EISCONN              133        /* Socket is already connected */
#define EMSGSIZE             97         /* Message too long */
#define ENETUNREACH          128        /* Network is unreachable */
#define ENOTCONN             134        /* Socket is not connected */
#define ENOTSOCK             95         /* Socket operation on non-socket */
#define EOPNOTSUPP           122        /* Operation not supported on socket */
#define EPROTONOSUPPORT      120        /* Protocol not supported */
#define EPROTOTYPE           98         /* Protocol wrong type for socket */
#define ETIMEDOUT            145        /* Connection timed out */
#define EWOULDBLOCK          EAGAIN

```

Figure 6-56 Manifest Constants and Data Types from <netdb.h>

```

struct netent {
    char          *n_name;
    char          **n_aliases;
    int           n_addrtype;
    unsigned long n_net;};

struct hostent {
    char          *h_name;           /* official name of host */
    char          **h_aliases;      /* alias list */
    int           h_addrtype;       /* host address type */
    int           h_length;         /* length of address */
    char          **h_addr_list;    /* list of addresses from name server */
#define h_addr   h_addr_list[0]   /* address, for backward compatibility */
};

struct servent {
    char          *s_name;
    char          **s_aliases;
    int           s_port;
    char          *s_proto;
};

struct protoent {
    char          *p_name;
    char          **p_aliases;
    int           p_proto;
};

#define HOST_NOT_FOUND      1        /* Authoritative Answer Host not found */
#define TRY_AGAIN           2        /* Non-Auth. Host not found, or SERVERFAIL */
#define NO_RECOVERY         3        /* Non recover errors:FORMERR,REFUSED,NOTIMP */
#define NO_DATA             4        /* Valid name, no data rec. of requested type */
#define NO_ADDRESS          NO_DATA  /* no address, look for MX record */

```

Figure 6-57 Manifest Constants and Data Types from <netinet/if_ether.h>

```

struct ether_addr {
    u_char    ether_addr_octet[6];
};

```

Figure 6-58 Manifest Constants and Data Types from <netinet/in.h>

```

/* IP address */
struct in_addr {
    union {
        struct { u_char s_b1, s_b2, s_b3, s_b4;} S_un_b;
        struct { u_short s_w1, s_w2;} S_un_w;
        u_long S_addr;
    } S_un;
};

/* socket address using IP */
struct sockaddr_in {
    short        sin_family;
    u_short      sin_port;
    struct in_addr sin_addr;
    char         sin_zero[8];
};

/* Options for use with [gs]etsockopt at the IP level. */
#define IP_OPTIONS        1        /* set/get IP per-packet options */
#define IP_HDRINCL       2        /* int; header is included with data (raw) */
#define IP_TOS           3        /* int; IP type of service and precedence */
#define IP_TTL           4        /* int; IP time to live */
#define IP_RECVOPTS      5        /* bool; receive all IP options w/datagram */
#define IP_RECVRETOPTS   6        /* bool; receive IP options for response */
#define IP_RECVDSTADDR   7        /* bool; receive IP dst addr w/datagram */
#define IP_RETOPTS       8        /* ip_opts; set/get IP per-packet options */
#define IP_MULTICAST_IF  0x10     /* set/get IP multicast interface */
#define IP_MULTICAST_TTL 0x11     /* set/get IP multicast timetolive */
#define IP_MULTICAST_LOOP 0x12    /* set/get IP multicast loopback */
#define IP_ADD_MEMBERSHIP 0x13    /* add an IP group membership */
#define IP_DROP_MEMBERSHIP 0x14   /* drop an IP group membership */
#define IP_DEFAULT_MULTICAST_TTL 1 /* normally limit m'casts to 1 hop */
#define IP_DEFAULT_MULTICAST_LOOP 1 /* normally hear sends if a member */

/* Argument structure for IP_ADD_MEMBERSHIP and IP_DROP_MEMBERSHIP. */
struct ip_mreq {
    struct in_addr imr_multiaddr;
    struct in_addr imr_interface;
};

```

Figure 6-59 Manifest Constants and Data Types from <rpcsvc/nis.h>

```

#define NIS_MAXSTRINGLEN      255
#define NIS_MAXNAMELEN       1024
#define NIS_MAXATTRNAME      32
#define NIS_MAXATTRVAL       2048
#define NIS_MAXCOLUMNS      64
#define NIS_MAXATTR          16
#define NIS_MAXPATH           1024
#define NIS_MAXREPLICAS      128
#define NIS_MAXLINKS         16
#define NIS_PK_NONE           0
#define NIS_PK_DH              1
#define NIS_PK_RSA             2
#define NIS_PK_KERB           3
#define EN_BINARY              1
#define EN_CRYPT                2
#define EN_XDR                  4
#define EN_MODIFIED            8
#define EN_ASN1                 64
#define TA_BINARY              1
#define TA_CRYPT                2
#define TA_XDR                  4
#define TA_SEARCHABLE          8
#define TA_CASE                 16
#define TA_MODIFIED            32
#define TA_ASN1                 64

struct nis_attr {
    char          *zattr_ndx;
    struct {
        u_int zattr_val_len;
        char *zattr_val_val;
    } zattr_val;
};

typedef struct nis_attr      nis_attr;
typedef char                *nis_name;

enum zotypes {
    BOGUS_OBJ = 0,          NO_OBJ = 1,
    DIRECTORY_OBJ = 2,     GROUP_OBJ = 3,
    TABLE_OBJ = 4,        ENTRY_OBJ = 5,
    LINK_OBJ = 6,          PRIVATE_OBJ = 7
};

typedef enum zotypes zotypes;

enum nstype {
    UNKNOWN = 0,           NIS = 1,
    SUNYP = 2,             IVY = 3,
    DNS = 4,               X500 = 5,
    DNANS = 6,             XCHS = 7,
    CDS = 8
};

typedef enum nstype nstype;

struct oar_mask {
    u_long          oa_rights;
    zotypes         oa_otype;
};

typedef struct oar_mask oar_mask;

struct endpoint {
    char          *uaddr;
};

```

```

    char      *family;
    char      *proto;
};
typedef struct endpoint endpoint;
struct nis_server {
    nis_name   name;
    struct {
        u_int   ep_len;
        endpoint *ep_val;
    } ep;
    u_long     key_type;
    netobj     pkey;
};
typedef struct nis_server nis_server;
struct directory_obj {
    nis_name   do_name;
    nstype     do_type;
    struct {
        u_int   do_servers_len;
        nis_server *do_servers_val;
    } do_servers;
    u_long     do_ttl;
    struct {
        u_int   do_armask_len;
        oar_mask *do_armask_val;
    } do_armask;
};
typedef struct directory_obj directory_obj;
struct entry_col {
    u_long     ec_flags;
    struct {
        u_int   ec_value_len;
        char    *ec_value_val;
    } ec_value;
};
typedef struct entry_col entry_col;
struct entry_obj {
    char      *en_type;
    struct {
        u_int   en_cols_len;
        entry_col *en_cols_val;
    } en_cols;
};
    typedef struct entry_obj entry_obj;
struct group_obj {
    u_long     gr_flags;
    struct {
        u_int   gr_members_len;
        nis_name *gr_members_val;
    } gr_members;
};
typedef struct group_obj group_obj;
struct link_obj {
    zotypes    li_rtype;
    struct {
        u_int   li_attrs_len;
        nis_attr *li_attrs_val;
    } li_attrs;
    nis_name   li_name;
};
typedef struct link_obj link_obj;
struct table_col {
    char      *tc_name;
    u_long     tc_flags;
    u_long     tc_rights;
};
typedef struct table_col table_col;

```

```
struct table_obj {
    char          *ta_type;
    int           ta_maxcol;
    u_char        ta_sep;
    struct {
        u_int     ta_cols_len;
        table_col *ta_cols_val;
    } ta_cols;
    char          *ta_path;
};

typedef struct table_obj table_obj;

struct objdata {
    zotypes          zo_type;
    union {
        struct directory_obj di_data;
        struct group_obj    gr_data;
        struct table_obj    ta_data;
        struct entry_obj    en_data;
        struct link_obj     li_data;
        struct {
            u_int     po_data_len;
            char      *po_data_val;
        } po_data;
    } objdata_u;
};

typedef struct objdata objdata;

struct nis_oid {
    u_long    ctime;
    u_long    mtime;
};

typedef struct nis_oid nis_oid;

struct nis_object {
    nis_oid    zo_oid;
    nis_name  zo_name;
    nis_name  zo_owner;
    nis_name  zo_group;
    nis_name  zo_domain;
    u_long    zo_access;
    u_long    zo_ttl;
    objdata  zo_data;
};

typedef struct nis_object nis_object;

enum nis_error {
    NIS_SUCCESS = 0,          NIS_S_SUCCESS = 1,          NIS_NOTFOUND = 2,
    NIS_S_NOTFOUND = 3,      NIS_CACHEEXPIRED = 4,      NIS_NAMEUNREACHABLE = 5,
    NIS_UNKNOWNOBJ = 6,     NIS_TRYAGAIN = 7,         NIS_SYSTEMERROR = 8,
    NIS_CHAINBROKEN = 9,    NIS_PERMISSION = 10,      NIS_NOTOWNER = 11,
    NIS_NOT_ME = 12,        NIS_NOMEMORY = 13,        NIS_NAMEEXISTS = 14,
    NIS_NOTMASTER = 15,    NIS_INVALIDOBJ = 16,      NIS_BADNAME = 17,
    NIS_NOCALLBACK = 18,   NIS_CBRESULTS = 19,       NIS_NOSUCHNAME = 20,
    NIS_NOTUNIQUE = 21,    NIS_IBMODERROR = 22,      NIS_NOSUCHTABLE = 23,
    NIS_TYPEMISMATCH = 24, NIS_LINKNAMEERROR = 25,   NIS_PARTIAL = 26,
    NIS_TOOMANYATTRS = 27, NIS_RPCERROR = 28,        NIS_BADATTRIBUTE = 29,
    NIS_NOTSEARCHABLE = 30, NIS_CBERROR = 31,         NIS_FOREIGNNS = 32,
    NIS_BADOBJECT = 33,     NIS_NOTSAMEOBJ = 34,      NIS_MODFAIL = 35,
    NIS_BADREQUEST = 36,   NIS_NOTEMPTY = 37,        NIS_COLDSTART_ERR = 38,
    NIS_RESYNC = 39,        NIS_FAIL = 40,            NIS_UNAVAIL = 41,
    NIS_RES2BIG = 42,       NIS_SRVAUTH = 43,         NIS_CLNTAUTH = 44,
    NIS_NOFILESPPACE = 45, NIS_NOPROC = 46,          NIS_DUMPLATER = 47
};

typedef enum nis_error nis_error;

struct nis_result {
    nis_error    status;
};
```



```

    struct {
        u_int      objects_len;
        nis_object *objects_val;
    } objects;
    netobj      cookie;
    u_long      zticks;
    u_long      dticks;
    u_long      aticks;
    u_long      cticks;
};

typedef struct nis_result nis_result;
#define FOLLOW_LINKS      (1<<0)
#define FOLLOW_PATH      (1<<1)
#define HARD_LOOKUP      (1<<2)
#define ALL_RESULTS      (1<<3)
#define NO_CACHE          (1<<4)
#define MASTER_ONLY      (1<<5)
#define EXPAND_NAME       (1<<6)

```

Figure 6-60 Manifest Constants and Data Types from <sys/socket.h>

```

/* Types */
#define SOL_SOCKET      0xffff
#define SOCK_STREAM    2          /* stream socket */
#define SOCK_DGRAM     1          /* datagram socket */
#define SOCK_RAW       4          /* raw-protocol interface */
#define SOCK_RDM       5          /* reliably-delivered message */
#define SOCK_SEQPACKET 6          /* sequenced packet stream */

/* Option flags per-socket. */
#define SO_DEBUG        0x0001    /* turn on debugging info recording */
#define SO_ACCEPTCONN  0x0002    /* socket has had listen() */
#define SO_REUSEADDR   0x0004    /* allow local address reuse */
#define SO_KEEPAALIVE  0x0008    /* keep connections alive */
#define SO_DONTROUTE   0x0010    /* just use interface addresses */
#define SO_BROADCAST   0x0020    /* permit sending of broadcast msgs */
#define SO_USELOOPBACK 0x0040    /* bypass hardware when possible */
#define SO_LINGER       0x0080    /* linger on close if data present */
#define SO_OOBINLINE   0x0100    /* leave received OOB data in line */

/* Additional options, not kept in so_options. */
#define SO_SNDBUF      0x1001    /* send buffer size */
#define SO_RCVBUF      0x1002    /* receive buffer size */
#define SO_SNDLOWAT    0x1003    /* send low-water mark */
#define SO_RCVLOWAT    0x1004    /* receive low-water mark */
#define SO_SNDTIMEO    0x1005    /* send timeout */
#define SO_RCVTIMEO    0x1006    /* receive timeout */
#define SO_ERROR       0x1007    /* get error status and clear */
#define SO_TYPE        0x1008    /* get socket type */
#define SO_PROTOCOL    0x1009    /* get/set protocol type */

/* Structure used for manipulating linger option. */
struct linger {
    int    l_onoff;
    int    l_linger;
}

```

```
};

/* Level number for (get/set) sockopt() to apply to socket itself. */
#define SOL_SOCKET      0xffff      /* options for socket level */

/* Address families. */
#define AF_UNSPEC      0              /* unspecified */
#define AF_UNIX      1              /* local to host (pipes, portals) */
#define AF_INET      2              /* internet: UDP, TCP, etc. */
#define AF_IMPLINK    3              /* arpanet imp addresses */
#define AF_PUP      4              /* pup protocols: e.g. BSP */
#define AF_CHAOS      5              /* mit CHAOS protocols */
#define AF_NS      6              /* XEROX NS protocols */
#define AF_NBS      7              /* nbs protocols */
#define AF_ECMA      8              /* european computer manufacturers */
#define AF_DATAKIT    9              /* datakit protocols */
#define AF_CCITT      10             /* CCITT protocols, X.25 etc */
#define AF_SNA      11             /* IBM SNA */
#define AF_DECnet     12             /* DECnet */
#define AF_DLI      13             /* Direct data link interface */
#define AF_LAT      14             /* LAT */
#define AF_HYLINK     15             /* NSC Hyperchannel */
#define AF_APPLETALK  16             /* Apple Talk */
#define AF_NIT      17             /* Network Interface Tap */
#define AF_802      18             /* IEEE 802.2, also ISO 8802 */
#define AF_OSI      19             /* umbrella for all families used */
#define AF_X25      20             /* CCITT X.25 in particular */
#define AF_OSINET    21             /* AFI = 47, IDI = 4 */
#define AF_GOSIP     22             /* U.S. Government OSI */
#define AF_MAX      22

/* Structure used by kernel to store most addresses. */
struct sockaddr {
    u_short      sa_family;
    char        sa_data[14];
};

/* Structure used by kernel to pass protocol * information in raw sockets. */
struct sockproto {
    u_short      sp_family;
    u_short      sp_protocol;
};

/* Protocol families, same as address families for now. */
#define PF_UNSPEC      AF_UNSPEC
#define PF_UNIX      AF_UNIX
#define PF_INET      AF_INET
#define PF_IMPLINK    AF_IMPLINK
#define PF_PUP      AF_PUP
#define PF_CHAOS      AF_CHAOS
#define PF_NS      AF_NS
#define PF_NBS      AF_NBS
#define PF_ECMA      AF_ECMA
#define PF_DATAKIT    AF_DATAKIT
#define PF_CCITT      AF_CCITT
#define PF_SNA      AF_SNA
#define PF_DECnet     AF_DECnet
```

```

#define PF_DLI          AF_DLI
#define PF_LAT          AF_LAT
#define PF_HYLINK       AF_HYLINK
#define PF_APPLETALK    AF_APPLETALK
#define PF_NIT          AF_NIT
#define PF_802          AF_802
#define PF_OSI          AF_OSI
#define PF_X25          AF_X25
#define PF_OSINET       AF_OSINET
#define PF_GOSIP        AF_GOSIP
#define PF_MAX          AF_MAX

/* Maximum queue length specifiable by listen. */
#define SOMAXCONN      5

/* Message header for recvmsg and sendmsg calls. */
struct msghdr {
    caddr_t      msg_name;          /* optional address */
    int          msg_namelen;       /* size of address */
    struct iovec *msg_iov;          /* scatter/gather array */
    int          msg_iovlen;        /* # elements in msg_iov */
    caddr_t      msg_accrights;     /* access rights sent/received */
    int          msg_accrightslen;
};

#define MSG_OOB          0x1          /* process out-of-band data */
#define MSG_PEEK         0x2          /* peek at incoming message */
#define MSG_DONTROUTE    0x4          /* send without using routing tables */
#define MSG_MAXIOVLEN    16

/* option header */
struct opthdr {
    long         level;
    long         name;
    long         len;
};

#define OPTLEN(x)        (((x) + sizeof (long) - 1) / sizeof (long)) * sizeof (long)
#define OPTVAL(opt)     ((char *) (opt + 1))

/* the optdefault structure is used for internal tables of option default values. */
struct optdefault {
    int          optname;
    char         *val;
    int          len;
};

/* the opproc structure is used to build tables of options processing functions for
dooptions(). */
struct opproc {
    int          level;
    int          (*func)();
};

/* This structure is used to encode pseudo system calls */
struct socksysreq {int     args[7];};

/* This structure is used for adding new protocols to the list supported by sockets. */
struct socknewproto {
    int          family;          /* address family (AF_INET, etc.) */

```

```
int      type;          /* protocol type (SOCK_STREAM, etc.) */
int      proto;        /* per family proto number */
dev_t    dev;          /* major/minor to use (must be a clone) */
int      flags;        /* protosw flags */
};
```

libthread - Multithreading Library

Overview

The interfaces specified in this section provide applications with the ability to create multiple “threads of control” within their address spaces.

The interface set described here resides entirely in the REQUIRED dynamic library: */usr/lib/libthread.so.1*. Interface members of this library, listed in the table below, are REQUIRED unless explicitly noted otherwise.

SCD Extensions to the System V ABI

The SCD requires a new library: */usr/lib/libthread.so.1*, which has functions not specified by the gABI. These functions are not defined in the SVID. The semantics for these additional function definitions are available in the SCD Interface Semantics document. Structures and manifest constants pertaining to these functions are as specified further below.

Besides the multithreading interface set specified in this section, systems must also provide enhancements to the system services provided by libc. These enhancements are, for the most part, transparent to the application. However, in a few cases, these enhancements manifest themselves as new entry points in libc. This is necessary as the manner in which the interface was previously defined was non reentrant, and made it impossible to express a multithread-safe implementation. For the most part, these new reentrant routines have the name of the old routine with the string “_r” appended. All of these entry points are part of libc. Those whose “root names” are also present in libc also reside there. All, with the exception of `___errno`, have synonyms. See the previous section “C Library” for additional information.

For an application to use libthread correctly, it must specify a reference to libthread in a **DT_NEEDED** entry prior to a **DT_NEEDED** entry which specifies libc. libthread redefines the semantics of a number of entry points normally provided by libc, primarily to support the correct management of signals in a multithreaded program.

Library Contents

Table 6-25 libthread contents

cond_broadcast		sema_post		thr_setconcurrency
cond_destroy		sema_trywait		thr_setprio
cond_init		sema_wait		thr_setspecific
cond_signal		setcontext		thr_sigsetmask
cond_timedwait		sigaction		thr_suspend
cond_wait		siglongjmp	2.4	thr_yield
creat64	L(E), 2.4	sigprocmask		
fork1		sigsetjmp	2.4	L - 32-bit ABI Large file support routine.
mutex_destroy		sigsuspend	2.4	
mutex_init		sigwait		E - EXPERIMENTAL interfaces.
mutex_lock		sleep		
mutex_trylock		thr_continue		
mutex_unlock		thr_create		2.4 - Interfaces added in SCD2.4
open64	L(E), 2.4	thr_exit		
rw_rdlock		thr_getconcurrency		
rw_tryrdlock		thr_getprio		
rw_trywrlock		thr_getspecific		
rw_unlock		thr_join		
rw_wrlock		thr_keycreate		
rwlock_destroy		thr_kill		
rwlock_init		thr_main		
sema_destroy		thr_min_stack		
sema_init		thr_self		

Structures and Manifest Constants

Figure 6-61 Manifest Constants and Data Types from <errno.h>

```

/* When _REENTRANT is defined, a multithreaded application is being constructed. */
#ifdef _REENTRANT
#define errno (*__errno())
#else
extern int errno;
#endif

```

Figure 6-62 Manifest Constants and Data Types from <synch.h>

```
#define USYNC_THREAD      0                /* private to a process */
#define USYNC_PROCESS    1                /* shared by processes */
typedef struct { /* unspecified, but sizeof(mutex_t) is 24. */} mutex_t;
typedef struct { /* unspecified, but sizeof(cond_t) is 16. */} cond_t;
typedef struct { /* unspecified, but sizeof(sema_t) is 48. */} sema_t;
typedef struct { /* unspecified, but sizeof(rwlock_t) is 64. */} rwlock_t;
```

Figure 6-63 Manifest Constants and Data Types from <thread.h>

```
typedef unsigned int thread_t;
typedef unsigned int thread_key_t;
#define THR_BOUND          0x00000001
#define THR_NEW_LWP       0x00000002
#define THR_DETACHED      0x00000040
#define THR_SUSPENDED     0x00000080
#define THR_DAEMON        0x00000100
```

libucb - University of California at Berkeley Compatibility Library

Overview

The interfaces specified in this section provide applications with compatibility to interfaces initially developed at the University of California at Berkeley.

The interface set described here resides entirely in the REQUIRED dynamic library: */usr/lib/libucb.so.1*. Interface members of this library, listed in the table below, are REQUIRED unless explicitly noted otherwise.

Note: new applications should not be using this library.

SCD Extension to the System V ABI

The SCD requires a new library: */usr/lib/libucb.so.1*, which has functions not specified by the gABI. These functions are not defined in the SVID. The semantics for these additional function definitions are available either in the *SCD Interface Semantics* document or XPG4.2 (as noted below).

Library Contents

Table 6-26 libucb contents

_longjmp	E	getwd	E	scandir64	2.4,L,E	sprintf	E
_setjmp	E	index	E	sethostname	E	srand	E
alphasort	E	initstate	E	setjmp	E	srandom	E
alphasort64	2.4,L,E	killpg	E	setlinebuf	E	times	E
bcmp		longjmp	E	setpriority	E	ualarm	E
bcopy	E	mctl	E	setregid	E	usleep	E
bzero	E	nice	E	setreuid	E	vsprintf	E
fopen	E	psignal	E	setstate		wait3	E
freopen	E	rand	E	settimeofday			
ftime		random		sigblock	E	L - 32-bit ABI Large file support routine.	
getdtablesize	E	re_comp		siginterrupt		E - EXPERIMENTAL interface.	
gethostid		re_exec		signal	E		
gethostname	E	readdir64	2.4,L,E	sigpause	E	2.4 - Interfaces added in SCD2.4	
getpagesize		reboot	E	sigsetmask	E		
getpriority	E	rindex	E	sigstack			
getrusage		scandir	E	sigvec	E		
gettimeofday	E	setbuffer	E	sleep	E		

libw - Wide Character Support Library

Overview

The interfaces specified in this section provide application support for wide (2-byte) characters.

The interface set described here resides entirely in the REQUIRED dynamic library: */usr/lib/libw.so.1*. Interface members of this library, listed in the table below, are REQUIRED unless explicitly noted otherwise.

SCD Extensions to the System V ABI

The SCD requires a new library: */usr/lib/libw.so.1*, which has functions not specified by the gABI. These functions are not defined in the SVID. The semantics for these additional function definitions are available in the SCD Interface Semantics document. Structures and manifest constants pertaining to these functions are as specified further below.

Library Contents

Table 6-27 libw contents

fgetc	(1) 2.4	iswgraph	(1) 2.4	wscol	2.4 (E)	wstod	2.4 (E)
fgetws	(1) 2.4	iswlower	(1) 2.4	wscoll	2.4 (E)	wstok	2.4 (E)
fputc	(1) 2.4	iswprint	(1) 2.4	wscopy	2.4 (E)	wstol	2.4 (E)
fputws	(1) 2.4	iswpunct	(1) 2.4	wscspn	2.4 (E)	wsxfrm	2.4 (E)
getwidth	2.4 (E)	iswspace	(1) 2.4	wsdup	2.4 (E)		
getws	2.4 (E)	iswupper	(1) 2.4	wslen	2.4 (E)	1 - Defined in XPG4.2	
isenglish	2.4 (E)	iswxdigit	(1) 2.4	wsncasecmp	2.4 (E)		
isideogram	2.4 (E)	putws	2.4 (E)	wsncat	2.4 (E)	2.4 - New interfaces added or changed in SCD2.4.	
isnumber	2.4 (E)	towlower	(1) 2.4	wsncmp	2.4 (E)		
isphonogram	2.4 (E)	towupper	(1) 2.4	wsncpy	2.4 (E)		
isspecial	2.4 (E)	ungetc	(1) 2.4	wspbrk	2.4 (E)	E - EXPERIMENTAL Interface	
iswalnum	(1) 2.4	wscasecmp	2.4 (E)	wsprintf	2.4 (E)		
iswalpha	(1) 2.4	wscat	2.4 (E)	wsrchr	2.4 (E)		
iswcntrl	(1) 2.4	wschr	2.4 (E)	wsscanf	2.4 (E)		
iswdigit	(1) 2.4	wscmp	2.4 (E)	wsspnp	2.4 (E)		

Structures and Manifest Constants

Figure 6-64 Manifest Constants and Data Types from <sys/euc.h>

```
typedef struct {
    short int _eucw1, _eucw2, _eucw3;
    short int _scrw1, _scrw2, _scrw3;
    short int _pcw;
    char _multibyte;
} eucwidth_t;
```

Figure 6-65 Manifest Constants and Data Types from <wchar.h>

```
typedef long wchar_t;
```

Figure 6-66 Manifest Constants and Data Types from <wctype.h>

```
typedef long wint_t;
```


Libraries (64-bit gABI) - EXPERIMENTAL

Overview

In addition to the common (32-bit ABI and 64-bit ABI) changes noted previously in this chapter, this section specifies additional generic changes for the 64-bit ABI. The 64-bit ABI interfaces are EXPERIMENTAL.

Miscellaneous Libraries Changes (64-bit gABI):

#	Facility	Location	Description
1	Network Services Library	gABI	<p>add to figure 6-11, page 6-12 the following functions:</p> <pre>xdr_hyper xdr_u_hyper xdr_int32 xdr_u_int32 xdr_long_double</pre>
2	Global Data Symbols	gABI	change in page 6-7 the type of <code>_numeric[2]</code> from “unsigned char” to “ <code>wchar_t</code> ”
3	Vendor Extensions	gABI	<p>Replace the last paragraph on page 6-8 (the one beginning “A symbol of the form” and ending “to the system library.”) with the following:</p> <p>Symbols with the prefix <code>_\${vendor.company}</code> provide operating system entries for the vendor named <i>company</i>. The system library does not have unadorned alternatives for these names. As an example, the “XYZ Computer Company” might use the prefix <code>_\${vendor.xyz}</code>.</p>

Libraries (64-bit psABI) - EXPERIMENTAL

Overview

A In addition to the common (32-bit ABI and 64-bit ABI) changes noted previously in this chapter, this section specifies additional SPARC processor-specific changes for the 64-bit ABI. The 64-bit ABI interfaces are EXPERIMENTAL.

Table 31: Library Logical and Reference Names

Library	Reference Name
(runtime linker)	<code>/usr/lib/sparcv9/ld.so.1</code>
libaio	<code>/usr/lib/sparcv9/libaio.so.1</code>
libc	<code>/usr/lib/sparcv9/libc.so.1</code>
libdl	<code>/usr/lib/sparcv9/libdl.so.1</code>
libelf	<code>/usr/lib/sparcv9/libelf.so.1</code>
libm	<code>/usr/lib/sparcv9/libm.so.1</code>
libnsl	<code>/usr/lib/sparcv9/libnsl.so.1</code>
libposix4	<code>/usr/lib/sparcv9/libposix4.so.1</code>
libpthread	<code>/usr/lib/sparcv9/libpthread.so.1</code>
libresolv	<code>/usr/lib/sparcv9/libresolv.so.1</code>
libsocket	<code>/usr/lib/sparcv9/libsocket.so.1</code>
libthread	<code>/usr/lib/sparcv9/libthread.so.1</code>
libw	<code>/usr/lib/sparcv9/libw.so.1</code>

Libc - C library

The Semantics for these additional function definitions are available in the SCD Interface Semantics document.

Table 6-32 libc SPARC Support Routines (64-bit ABI)

<code>__align_cpy_1</code>	2.4	<code>_Qp_div</code>	2.4	<code>_Qp_sqrt</code>	2.4	1 - See C Library Changes section at the beginning of this chapter.
<code>__align_cpy_2</code>	2.4	<code>_Qp_dtoq</code>	2.4	<code>_Qp_stoq</code>	2.4	
<code>__align_cpy_4</code>	2.4	<code>_Qp_feq</code>	2.4	<code>_Qp_sub</code>	2.4	
<code>__align_cpy_8</code>	2.4	<code>_Qp_fge</code>	2.4	<code>_Qp_uitoq</code>	2.4	
<code>__align_cpy_16</code>	2.4	<code>_Qp_fgt</code>	2.4	<code>_Qp_uptoq</code>	2.4	
<code>__dtou</code>	1,2.4	<code>_Qp_fne</code>	2.4	<code>_Qp_xtoq</code>	2.4	
<code>__dtoul</code>	2.4	<code>_Qpflt</code>	2.4			
<code>__dtoll</code>	2,2.4	<code>_Qp_fbe</code>	2.4			
<code>__dtoull</code>	2,2.4	<code>_Qp_itoq</code>	2.4			
<code>__ftoul</code>	2.4	<code>_Qp_mul</code>	2.4			
<code>__ftoll</code>	2,2.4	<code>_Qp_neg</code>	2.4			2 - Long long intrinsics support routines - routines supporting operations on a 64-bit integer ("long long") for both signed and unsigned quantities. Calling sequence for 64-bit integer arguments and return value is described in Low-Level System Information Changes in Chapter 3.
<code>__ftou</code>	1,2.4	<code>_Qp_qtod</code>	2.4			
<code>__ftoull</code>	2,2.4	<code>_Qp_qtoi</code>	2.4			
<code>__sparc_utrap_install</code>	2.4	<code>_Qp_qtos</code>	2.4			
<code>_Qp_add</code>	2.4	<code>_Qp_qtoui</code>	2.4			
<code>_Qp_cmp</code>	2.4	<code>_Qp_qtoux</code>	2.4			
<code>_Qp_cmpe</code>	2.4	<code>_Qp_qtox</code>	2.4			
						2.4 - New interfaces added to SCD2.4 or changed interfaces in SCD2.4.

Library Contents

Each library interface listed in this chapter for the 32-bit ABI is also supported in the 64-bit ABI with the following exceptions:

- 1) The large files support (transitional) interfaces provided in the 32-bit ABI (i.e. those file-accessing interfaces whose standard names have been qualified with a "64" suffix) are not present in the 64-bit ABI. The interfaces with unqualified names in the 64-bit ABI all operate on large files directly, since large files constitute the native files interface in the 64-bit ABI.

System Data Interfaces

Vendor Extensions

An ABI-conforming system vendor may add additional symbolic constants (represented in this chapter as ANSI C #define macros) to facilitate the use of vendor-specific services. The ABI does not define these symbolic constants or their values, and programs using them are not ABI-conforming. Nonetheless, the ABI defines an extension mechanism, providing a way to avoid conflict among the services from multiple vendors. This extension mechanism is as follows:

- Non-negative symbolic constant values are reserved to SPARC International.
- Negative symbolic constant values are reserved to vendors. Bits 30 through 15 of each symbolic constant value must contain the binary representation of the vendor's Vendor Identification Number obtained from SPARC International.

It is expected that vendors will use this extension mechanism to add, for example, new vendor-specific `_SC_symbolic` constants to `<unistd.h>`.

Structures and Manifest Constants

The following figures represent all library-related structures and manifest constants for the 64-bit ABI:

Figure 6-6: <errno.h>

```

#define ENAMETOOLONG    78
#define EOVERFLOW      79
#define ENOTUNIQ       80
#define EBADF          81
#define EREMCHG        82
#define ENOSYS         89
#define ELOOP          90
#define ERESTART       91
#define ESTRPIPE       92
#define ENOTEMPTY      93
#define EUSERS         94
#define ESTALE         151
extern int errno;

```

Figure 6-7: <fcntl.h>

```

typedef struct flock{
    short    l_type;
    short    l_whence;
    off_t    l_start;
    off_t    l_len;
    int      l_sysid;
    pid_t    l_pid;
    long     l_pad[4];
} flock_t;

```

Figure 6-12: <sys/ipc.h>

```

struct ipc_perm{
    uid_t    uid;
    gid_t    gid;
    uid_t    cuid;
    gid_t    cgid;
    mode_t   mode;
    uint_t   seq;
    key_t    key;
};

```

Figure 6-16: <math.h>

```

typedef union _h_val {
    unsigned int    i[2];
    unsigned long   l;
    double          d;
} _h_val;
external const _h_val  __huge_val;
#define HUGE_VAL       __huge_val.d;

```

Figure 6-19: <sys/msg.h>

```

struct msqid_ds {
    struct ipc_perm  msg_perm;
    struct msg       *msg_first;
    struct msg       *msg_last;
    unsigned long    msg_ctypes;
    unsigned long    msg_qnum;
    unsigned long    msgqbytes;
    pid_t            msglspid;
    pid_t            msglrpid;
    time_t           msg_stime;
    time_t           msg_rtime;
    time_t           msg_ctime;
    long             msg_pad4[4];
};

```

Figure 6-20: <sys/netconfig.h>

```

struct netconfig {
    char          *nc_netid;
    unsigned int  nc_semantics;
    unsigned int  nc_flag;
    char          *nc_protofmly;
    char          *nc_proto;
    char          *nc_device;
    unsigned int  nc_nlookups;
    char          **nc_lookups;
    unsigned int  nc_unused[8];
};

```

Figure 6-21: <netdir.h>

```

struct nd_addrlist {int n_cnt; struct netbuf *n_addrs;};
struct nd_hostservlist {int h_cnt; struct nd_hostserv *h_hostservs;};
struct nd_hostserv {char *h_host; char *h_serv;};
#define ND_BADARG -2
#define ND_NOMEM -1
#define ND_OK 0
#define ND_NOHOST 1
#define ND_NOSERV 2
#define ND_NOSYM 3
#define ND_OPEN 4
#define ND_ACCESS 5
#define ND_UKNWN 6
#define ND_NOCTRL 7
#define ND_FAILCTRL 8
#define ND_SYSTEM 9
#define ND_HOSTSERV 0
#define ND_HOSTSERVLIST 1
#define ND_ADDR 2
#define ND_ADDRLIST 3
#define HOST_SELF "\\1"
#define HOST_ANY "\\2"
#define HOST_BROADCAST "\\3"
#define ND_SET_BROADCAST 1
#define ND_SET_RESERVEDPORT 2
#define ND_CHECK_RESERVEDPORT 3
#define ND_MERGEADDR 4

```

Figure 6-23: <sys/param.h>

```
#define CANBSIZ      256
#define NGROUPS_UMIN  0
#define MAXPATHLEN  1024
#define MAXSYMLINKS  20
#define MAXNAMELEN   256
#define NADDR        13
#define PIPE_MAX     5120
#define NBBY         8
#define NBPSCTR      512
```

Figure 6-27 <sys/resource.h>

```
typedef unsigned long rlim_t;
#define RLIM_INFINITY  (-31)
#define RLIM_SAVED_MAX  (-21)
#define RLIM_SAVED_CUR  (-11)
```

Figure 6-28: <rpc/rpc.h>

```

union des_block {
    struct {
        uint_t high;
        uint_t low;
    } key;
    char    c[8];
};
struct opaque_auth {
    enum_t oa_flavor;
    caddr_t oa_base;
    uint_t oa_length;
};
typedef struct __auth {
    struct opaque_auth    ah_cred;
    struct opaque_auth    ah_verf;
    union des_block       ah_key;
    struct auth_ops {
        void (*ah_nextverf)();
        int  (*ah_marshall)();
        int  (*ah_validate)();
        int  (*ah_refresh)();
        void (*ah_destroy)();
    } *ah_ops;
    caddr_t ah_private;
} AUTH;
struct authsys_parms {
    uint_t    aup_time;
    caddr_t   aup_machname;
    uid_t     aup_uid;
    gid_t     aup_gid;
    uint_t    aup_len;
    gid_t     *aup_gids;
};
extern struct opaque_auth __null_auth;
#define AUTH_NONE 0
struct rpc_err {
    enum clnt_statre_status;
    union {
        struct {
            int RE_errno;
            int RE_t_errno;
        } RE_err;
        enum auth_stat RE_why;
        struct {
            rpcvers_t low;
            rpcvers_t high;
        } RE_vers;
        struct {
            int s1;
            int s2;
        } RE_lb;
    }
} ru;
typedef struct __client {
    AUTH *cl_auth;
    struct clnt_ops {
        enum clnt_stat (*cl_call)();
        void (*cl_abort)();
        void (*cl_geterr)();
        bool_t (*cl_freeres)();
        void (*cl_destroy)();
        bool_t (*cl_control)();
        int (*cl_settimers)();
    } *cl_ops;
    caddr_t cl_private;
    char *cl_netid;
    char *cl_tp;
} CLIENT;
typedef uint_t    rpcprog_t;
typedef uint_t    rpcvers_t;
typedef uint_t    rpcproc_t;
struct xp_ops {
    bool_t (*xp_rcv)();
    enum xpstat (*xp_stat)();
    bool_t (*xp_getargs)();
    bool_t (*xp_reply)();
    bool_t (*xp_freeargs)();
    void (*xp_destroy)();
    bool_t (*xp_control)();
};

```

Figure 6-28: <rpc/rpc.h> (continued)

```

typedef struct __svcxpirt {
    int                xp_fd;
    ushort_t          xp_port;
    struct xp_ops      *xp_ops;
    int                xp_addrlen;
    char               *xp_tp;
    char               *xp_netid;
    struct netbuf      xp_ltaddr;
    struct netbuf      xp_rtaddr;
    char               xp_raddr[16];
    struct opaque_auth xp_verf;
    caddr_t            xp_p1;
    caddr_t            xp_p2;
    caddr_t            xp_p3;
    int                xp_type;
} SVCXPRT;
struct svc_req {
    rpcprog_t          rq_prog;
    rpcvers_t          rq_vers;
    rpcproc_t          rq_proc;
    struct opaque_auth rq_cred;
    caddr_t            rq_clntcred;
    struct __svcxpirt *rq_xprt;
};
#define FD_SETSIZE    65536
struct accepted_reply {
    struct opaque_auth ar_verf;
    enum accept_stat   ar_stat;
    union {
        struct {
            rpcvers_t  low;
            rpcvers_t  high;
        } AR_versions;
        struct {
            caddr_t     where;
            xdrproc_t    proc;
        } AR_results;
    } ru;
};
struct rejected_reply {
    enum reject_stat   rj_stat;
    union {
        struct {
            rpcvers_t  low;
            rpcvers_t  high;
        } RJ_versions;
        enum auth_stat  RJ_why;
    } ru;
};
struct call_body {
    rpcvers_t          cb_rpcvers;
    rpcprog_t          cb_prog;
    rpcvers_t          cb_vers;
    rpcproc_t          cb_proc;
    struct opaque_auth cb_cred;
    struct opaque_auth cb_verf;
};
struct rpc_msg {
    uint_t             rm_xid;
    enum msg_type      rm_direction;
    union {
        struct call_body  RM_cmb;
        struct reply_body RM_rmd;
    } ru;
};
struct rpcb {
    rpcprog_t          r_prog;
    rpcvers_t          r_vers;
    char               *r_netid;
    char               *r_addr;
    char               *r_owner;
};
struct authdes_fullname {
    char               *name;
    des_block          key;
    uint_t             window;
};
struct authdes_cred {
    enum               authdes_namekind    adc_namekind;
    struct             authdes_fullname    adc_fullname;
    uint_t             adc_nickname;
};

```

Figure 6-28: <rpc/rpc.h> (continued)

```

typedef struct XDR {
    enum xdr_op
    struct xdr_ops {
        bool_t      (*x_getlong)(struct XDR *, long *);
        bool_t      (*x_putlong)(struct XDR *, long *);
        bool_t      (*x_getbytes)(struct XDR *, caddr_t, int);
        bool_t      (*x_putbytes)(struct XDR *, caddr_t, int);
        uint_t      (*x_getpostn)(struct XDR *);
        bool_t      (*x_setpostn)(struct XDR *, uint_t);
        int *       (*x_inline)(struct XDR *, int);
        void        (*x_destroy)(struct XDR *);
        bool_t      (*x_control)(struct XDR *, int, void *);
        bool_t      (*x_getint32)(struct XDR *, int32_t *);
        bool_t      (*x_putint32)(struct XDR *, int32_t *);
    } *x_ops;
    caddr_t      x_public;
    caddr_t      x_private;
    caddr_t      x_base;
    int          x_handy;
} XDR;

typedef bool_t      (*xdrproc_t)();

```

Figure 6-30: <sys/sem.h>

```

struct semid_ds {
    struct ipc_perm      sem_perm;
    struct sem           *sem_base;
    ushort_t            sem_nsems;
    time_t              sem_otime;
    time_t              sem_ctime;
    int                 sem_binary;
    long                sem_pad3[3];
};

```

Figure 6-31: <setjmp.h>

```

#define      _JBLEN      12
#define      _SIGJBLEN  19
typedef long jmp_buf[_JBLEN];
typedef long sigjmp_buf[_SIGJBLEN];

```

Figure 6-32: <sys/shm.h>

```

struct shmid_ds{
    struct ipc_perm      shm_perm;
    size_t              shm_segsz;
    void                *shm_amp;
    ushort_t            shm_lkcnt;
    pid_t               shm_lpid;
    pid_t               shm_cpid;
    unsigned long       shm_nattch;
    ulong_t             shm_cnattch;
    time_t              shm_atime;
    time_t              shm_dtime;
    time_t              shm_ctime;
    char                shm_pad4[4];
};

```

Figure 6-33: <signal.h>

```

#define SS_DISABLE          0x00000002
struct sigaltstack {
    void            *ss_sp;
    size_t         ss_size;
    int            ss_flags;
} stack_t;

struct sigstack {
    void            *ss_sp;
    int            ss_onstack;
};

typedef struct {
    unsigned int    sigbits[4]
} sigset_t;

struct sigaction {
    int            sa_flags;
    void            (*_handler)();
    sigset_t      sa_mask;
};

```

Figure 6-34: <sys/signinfo.h>

```

#define SI_MAXSZ          256
#define SI_PAD            ((SI_MAXSZ / sizeof (int)) - 4)
typedef struct timespec {time_t _tv_sec; long _tv_nsec;} timespec_t;
union __sigval { /* should be defined */;

typedef struct {
    int            si_signo;
    int            si_code;
    int            si_errno;
    int            si_pad;
    union {
        int        __pad[SI_PAD];
        struct {
            pid_t    __pid;
            union {
                struct {uid_t __uid; union __sigval __value;} __kill;
                struct {clock_t __utime; int __status; clock_t __stime;} __cld;
            } __pdata;
        } __proc;
        struct {void *__addr; int __trapno; caddr_t __pc; } __fault;
        struct {int __fd; long __band;} __file;
        struct {
            caddr_t    __faddr;
            timestruc_t __tstamp;
            short      __syscall;
            char       __nsysarg;
            char       __fault;
            long       __sysarg[8];
            int        __mstate[10];
        } __prof;
    } __data;
} siginfo_t;

```

Figure 6-35: <sys/stat.h>

```
#define _ST_FSTYPSZ      16
struct      stat {
    dev_t      st_dev;
    ino_t      st_ino;
    mode_t     st_mode;
    nlink_t    st_nlink;
    uid_t      st_uid;
    gid_t      st_gid;
    dev_t      st_rdev;
    off_t      st_size;
    timestruc_t st_atim;
    timestruc_t st_mtim;
    timestruc_t st_ctim;
    int        st_blksize;
    long       st_blocks;
    char       st_fstype[_ST_FSTYPSZ];
};
#define st_mtime      st_mtim.tv_sec
#define st_ctime      st_ctim.tv_sec
#define st_atime      st_atim.tv_sec
```

Figure 6-36: <sys/statvfs.h>

```
typedef struct statvfs {
    unsigned long    f_bsize;
    unsigned long    f_frsize;
    ulong_t          f_blocks;
    ulong_t          f_bfree;
    ulong_t          f_bavail;
    ulong_t          f_files;
    ulong_t          f_ffree;
    ulong_t          f_favail;
    unsigned long    f_fsid;
    char             f_basetype[_FSTYPSZ];
    unsigned long    f_flag;
    unsigned long    f_namemax;
    char             f_fstr[32];
} statvfs_t;
```

Figure 6-38: <stddef.h>

```
#define NULL          0
typedef long          ptrdiff_t;
typedef unsigned long size_t;
typedef int           wchar_t;
```


Figure 6-39: <stdio.h>

```

typedef long          fpos_t;
#define _NFILE        20
#define NULL          0
#define BUFSIZ        1024
#define _IOFBF        0000
#define _IOLBF        0100
#define _IONBF        0004
#define EOF           (-1)
#define FOPEN_MAX     _NFILE
#define FILENAME_MAX  1024
#define SEEK_SET      0
#define SEEK_CUR      1
#define SEEK_END      2
#define TMP_MAX       17576
#define stdin         (&_iob[0])
#define stdout        (&_iob[1])
#define stderr        (&_iob[2])
#define L_ctermid     9
#define L_cuserid     9
#define P_tmpdir      "/var/tmp/"
#define L_tmpnam      25

typedef struct {
    long          __impl[16];
} FILE;

```

Figure 6-41: <stropts.h>

```

struct strbuf {
    int          maxlen;
    int          len;
    caddr_t      buf;
};

struct strpeek {
    struct strbuf  ctlbuf;
    struct strbuf  databuf;
    int flags;
};

struct strfdinsert {
    struct strbuf  ctlbuf;
    struct strbuf  databuf;
    int flags;
    int fildes;
    int offset;
};

struct strrecvfd {
    int          fd;
    uid_t        uid;
    gid_t        gid;
    char         fill[8];
};

```

Figure 6-42: <termios.h>

```

typedef unsigned int  tcflag_t;
typedef unsigned char cc_t;
typedef unsigned int  speed_t;

```

Figure 6-47: <sys/time.h>

```

struct timeval {
    time_t      tv_sec;
    long        tv_usec;
};

struct timezone {
    int         tz_minuteswest;
    int         tz_dsttime;
};

typedef struct timespec {
    time_t      tv_sec;
    long        tv_nsec;
} timespec_t;

typedef struct timespec timestruc_t;

```

Figure 6-54: <tiuser.h>

```

struct t_bind {
    struct netbuf  addr;
    uint_t        qlen;
};

struct t_info {
    int           addr;
    int           options;
    int           tsdu;
    int           etsdu;
    int           connect;
    int           discon;
    int           servtype;
};

struct t_optmgmt {
    struct netbuf  opt;
    unsigned int  flags;
};

struct t_uderr {
    struct netbuf  addr;
    struct netbuf  opt;
    int           error;
};

```

Figure 6-59: <sys/types.h>

```

typedef unsigned char    uchar_t;
typedef unsigned short   ushort_t;
typedef unsigned int     uint_t;
typedef unsigned long    ulong_t;
typedef char*           caddr_t;
typedef long            daddr_t;
typedef short           sysid_t;
typedef long            off_t;
typedef ulong_t         ino_t;
typedef int             id_t;
typedef uint_t          major_t;
typedef uint_t          minor_t;
typedef uint_t          mode_t;
typedef int             key_t;
typedef int             uid_t;
typedef uid_t           gid_t;
typedef uint_t          nlink_t;
typedef ulong_t         dev_t;
typedef int             pid_t;
typedef ulong_t         size_t;
typedef long            time_t;
typedef long            clock_t;

```

Figure 6-60: <ucontext.h>

```

#include <sys/regset.h>
typedef struct ucontext {
    uint_t                uc_flags;
    ucontext_t            *uc_link;
    sigset_t              uc_sigmask;
    stack_t               uc_stack;
    mcontext_t            uc_mcontext;
    long                  uc_filler[4];
} ucontext_t;
#define UC_ASR                040

```

Figure 6-61: <uio.h>

```

typedef struct iovec {
    caddr_t               iov_base;
    size_t                iov_len;
} iovec_t;

```

Figure 6-63: <unistd.h>

```

#define _POSIX_VERSION          *
#define _XOPEN_VERSION         *
#define _SC_ARG_MAX            1
#define _SC_CHILD_MAX         2
#define _SC_CLK_TCK           3
#define _SC_NGROUPS_MAX       4
#define _SC_OPEN_MAX          5
#define _SC_JOB_CONTROL        6
#define _SC_SAVED_IDS         7
#define _SC_VERSION           8
#define _SC_PASS_MAX          9
#define _SC_LOGNAME_MAX       10
#define _SC_PAGESIZE          11
#define _SC_XOPEN_VERSION     12
#define _PC_LINK_MAX          1
#define _PC_MAX_CANON         2
#define _PC_MAX_INPUT         3
#define _PC_NAME_MAX          4
#define _PC_PATH_MAX          5
#define _PC_PIPE_BUF          6
#define _PC_NO_TRUNC          7
#define _PC_VDISABLE          8
#define _PC_CHOWN_RESTRICTED  9
#define STDIN_FILENO          0
#define STDOUT_FILENO         1
#define STDERR_FILENO         2

```

Figure 6-140: <inttypes.h>

```

typedef signed char          int8_t;
typedef short               int16_t;
typedef int                 int32_t;
typedef long                int64_t;
typedef unsigned char       uint8_t;
typedef unsigned short      uint16_t;
typedef unsigned int        uint32_t;
typedef unsigned long       uint64_t;
typedef long                intptr_t;
typedef unsigned long       uintptr_t;

```

Figure 6-141: <sys/regset.h>

```

typedef long          greg_t;
#define _NGREG        21
typedef greg_t        gregset_t[_NGREG];
#define REG_CCR       0
#define REG_PC        1
#define REG_nPC      2
#define REG_Y         3
#define REG_G1        4
#define REG_G2        5
#define REG_G3        6
#define REG_G4        7
#define REG_G5        8
#define REG_G6        9
#define REG_G7        10
#define REG_O0        11
#define REG_O1        12
#define REG_O2        13
#define REG_O3        14
#define REG_O4        15
#define REG_O5        16
#define REG_O6        17
#define REG_O7        18
#define REG_ASI       19
#define REG_FPRS      20
#define REG_SP        REG_O6
#define REG_R0        REG_O0
#define REG_R1        REG_O1

struct rwindow {
    greg_t          rw_local[8];
    greg_t          rw_in[8];
};

typedef struct {
    int             wbcnt;
    greg_t          *spbuf[SPARC_MAXREGWINDOWS];
    struct window   wbuf[SPARC_MAXREGWINDOW];
} gwindows_t;

struct fpq {
    unsigned int    *fpq_addr;
    unsigned int    *fpq_instr;
};

struct fpu {
    union {
        uint_t      fpu_regs[32];
        uint64_t     fpu_dregs[32];
        long double  fpu_qregs[16];
    } fpu_fr;
    struct fq        *fpu_q;
    uint64_t         fpu_fsr;
    uint8_t          fpu_qcnt;
    uint8_t          fpu_q_entsize;
    uint8_t          fpu_en;
};

typedef struct fpu fpregset_t;

typedef greg_t asrset_t[16]; /* %asr16 -> %asr31 */

typedef struct {
    uint_t          xrs_id;
    caddr_t         xrs_ptr;
} xrs_t;

typedef struct {
    gregset_t       gregs;
    gwindows_t      *gwins;
    fpregset_t      fpregs;
    xrs_t           xrs;
    asrset_t        asrs;
    long            filler[4];
} mcontext_t;

#define SPARC_MAXREGWINDOW 31

```

Figure 6-142: <tpicommon.h>

```

struct opthdr {
    int         level;
    int         name;
    int         len;
};

```

Figure 6-143: <sys/utrap.h>

```

#define UT_INSTRUCTION_EXCEPTION      1
#define UT_INSTRUCTION_ERROR          2
#define UT_INSTRUCTION_PROTECTION    3
#define UT_ILLTRAP_INSTRUCTION       4
#define UT_ILLEGAL_INSTRUCTION       5
#define UT_PRIVILEGED_OPCODE         6
#define UT_FP_DISABLED               7
#define UT_FP_EXCEPTION_IEEE_754     8
#define UT_FP_EXCEPTION_OTHER        9
#define UT_TAG_OVERVIEW              10
#define UT_DIVISION_BY_ZERO          11
#define UT_DATA_EXCEPTION            12
#define UT_DATA_ERROR                13
#define UT_DATA_PROTECTION           14
#define UT_MEM_ADDRESS_NOT_ALIGNED   15
#define UT_PRIVILEGED_ACTION         16
#define UT_ASYNC_DATA_ERROR          17
#define UT_TRAP_INSTRUCTION_16       18
#define UT_TRAP_INSTRUCTION_17       19
#define UT_TRAP_INSTRUCTION_18       20
#define UT_TRAP_INSTRUCTION_19       21
#define UT_TRAP_INSTRUCTION_20       22
#define UT_TRAP_INSTRUCTION_21       23
#define UT_TRAP_INSTRUCTION_22       24
#define UT_TRAP_INSTRUCTION_23       25
#define UT_TRAP_INSTRUCTION_24       26
#define UT_TRAP_INSTRUCTION_25       27
#define UT_TRAP_INSTRUCTION_26       28
#define UT_TRAP_INSTRUCTION_27       29
#define UT_TRAP_INSTRUCTION_28       30
#define UT_TRAP_INSTRUCTION_29       31
#define UT_TRAP_INSTRUCTION_30       32
#define UT_TRAP_INSTRUCTION_31       33

#define UTH_NOCHANGE ((utrap_handler_t)(-1))

typedef int utrap_entry_t;
typedef void *utrap_handler_t;

__sparc_utrap_install(utrap_entry_t type,
    utrap_handler_t new_precise, utrap_handler_t new_deferred,
    utrap_handler_t *old_precise, utrap_handler_t *old_deferred);

```

Figure 6-137: <netinet/in.h>

```

typedef uint16_t          in_port_t;
typedef uint_t           in_addr_t;
typedef uint_t           ipaddr_t;
typedef unsigned short   sa_family_t;
#define INADDR_ANY       (ipaddr_t)0x00000000U
#define INADDR_LOOPBACK (ipaddr_t)0x7f000001U
#define INADDR_BROADCAST (ipaddr_t)0xffffffffU
struct in_addr {
    union {
        struct { uint8_t s_b1, s_b2, s_b3, s_b4; } _S_un_b;
        struct { uint16_t s_w1, s_w2; } _S_un_w;
        in_addr_t _S_addr;
    } _S_un;
};
#define IN_CLASSA(i)      (((ipaddr_t)(i) & 0x80000000U) == 0)
#define IN_CLASSA_NET    (ipaddr_t)0xff000000U
#define IN_CLASSA_NSHIFT 24
#define IN_CLASSA_HOST   (ipaddr_t)0x00ffffffU
#define IN_CLASSA_MAX    128
#define IN_CLASSB(i)     (((ipaddr_t)(i) & 0xc0000000U) == 0x80000000U)
#define IN_CLASSB_NET    (ipaddr_t)0xffff0000U
#define IN_CLASSB_NSHIFT 16
#define IN_CLASSB_HOST   (ipaddr_t)0x0000ffffU
#define IN_CLASSB_MAX    65536
#define IN_CLASSC(i)     (((ipaddr_t)(i) & 0xe0000000U) == 0xc0000000U)
#define IN_CLASSC_NET    (ipaddr_t)0xffff0000U
#define IN_CLASSC_NSHIFT 8
#define IN_CLASSC_HOST   (ipaddr_t)0x00000fffU
#define IN_CLASSD(i)     (((ipaddr_t)(i) & 0xf0000000U) == 0xe0000000U)
#define IN_CLASSD_NET    (ipaddr_t)0xf0000000U
#define IN_CLASSD_NSHIFT 28
#define IN_CLASSD_HOST   (ipaddr_t)0x0ffffffU
#define IN_MULTICAST(i)  IN_CLASSD(i)
#define IN_EXPERIMENTAL(i) (((ipaddr_t)(i) & 0xe0000000U) == 0xe0000000U)
#define IN_BADCLASS(i)   (((ipaddr_t)(i) & 0xf0000000U) == 0xf0000000U)
#define INADDR_ANY       (ipaddr_t)0x00000000U
#define INADDR_LOOPBACK (ipaddr_t)0x7f000001U
#define INADDR_BROADCAST (ipaddr_t)0xffffffffU
#define INADDR_UNSPEC_GROUP (ipaddr_t)0xe0000000U
#define INADDR_ALLHOSTS_GROUP (ipaddr_t)0xe0000001U
#define INADDR_ALLRTRS_GROUP (ipaddr_t)0xe0000002U
#define INADDR_MAX_LOCAL_GROUP (ipaddr_t)0xe00000ffU

```

CHAPTER 7: Formats and Protocols

SCD

2.4

Formats and Protocols

Introduction

This chapter is split into a common section followed by a 64-bit gABI section. The common section applies to both the 32-bit ABI and 64-bit ABI, except where explicitly noted otherwise. Archive file formats, networking protocols, and the terminfo data base format may be found in Chapter 7 of the *System V Application Binary Interface*.

Formats and Protocols Changes

The following are changes to the *System V ABI*, the *System V ABI SPARC Processor Supplement*, and the *System V Interface Definition* as reported to SPARC International.

#	Facility	Location	Description
1	rpcbind Operation	gABI	Change - page 7-38-The reference to IP in the first paragraph is ambiguous -- port 111 is used for IP-carried transports (rather than IP itself).
2	terminfo	gABI	on page 7-7 and 7-8, the text: . . . terminal capabilities are stored here in the order in which that are listed under the . . . should read: . . terminal capabilities are stored here in the order in which they are listed under the . .

Interconnecting SCD Conforming Systems

Overview

This section contains the REQUIRED internetworking interfaces available to applications running on an SCD conforming system. Note that the networking ABI is defined by the TLI interfaces described in section BA_LIB of the *System V Interface Definition (Third Edition), Volume I*. This chapter adds to that definition by specifying that there shall be present in all SCD complying systems an Internet Protocol Suite (IPS) transport provider that is accessible through TLI. Also added are the commands, which exist in `/usr/bin`, and their associated daemons, which exist in `/usr/sbin`.

Transport Providers

All SPARC-compliant systems will provide a transport provider interface for each of the IP protocols, TCP, UDP, ICMP, and ARP. The device names for these transport provider interfaces must be `/dev/tcp`, `/dev/udp`, `/dev/icmp`, and `/dev/arp` respectively. Additionally, shared objects will be present to convert IP format universal addresses into the necessary internal format needed by the TLI interfaces. These interfaces are previously defined in the Network Services Library section of chapter 6 (Libraries).

Additional Interfaces

The interfaces listed below in Table 7-1 show the additional commands, protocols, and service daemons that are included to ensure inter-operability between SCD conforming systems. The table includes three columns, the command name which is invoked, the Request for Comments (RFC) number for the protocol specification as maintained by the Internet Engineering Task Force, and a short description of the feature provided.

Table 7-2 shows the “well-known” port numbers as derived from RFC 1700 that SCD conforming systems are REQUIRED to provide for supported services.

Table 7-1: Required Commands

Command	RFC	Description
rlogin	BSDNET	Remote terminal services (BSD)
rsh	BSDNET	Remote user shell (BSD)
rcp	BSDNET	Remote file copy (BSD)
rwho	BSDNET	Remote user information service (BSD)
rdate	BSDNET	Remote uptime statistics (BSD)
talk	BSDNET	Remote chat utility (BSD)
finger	rfc1288	Information server for logged on users
telnet	<many>	Interactive terminal services
ftp	rfc959	File transfer protocol
arp	rfc826	Address Resolution Protocol

Table 7-2: Well-Known Port Numbers

Keyword	Description	TCP Port Number	UDP Port Number
tcpmux	rfc1078	1	
echo	Echo	7	7
discard	Discard	9	9
systat	Active Users	11	11
daytime	Daytime	13	13
netstat	Who is up or NETSTAT	15	15
chargen	Character Generator	19	19
ftp-data	File Transfer Protocol (Data)	20	
ftp	File Transfer Protocol	21	
telnet	Terminal Connection	23	
smtp	Simple Mail Transport Protocol	25	
time	Time	37	37
name	Host Name Server	42	42
nickname	Who Is	43	43
domain	Domain Name Server	53	53
tftp	Trivial File Transfer	69	69
	Any private RJE service	77	77
finger	Finger	79	79
supdup	SUPDUP Protocol	95	
hostname	NIC Host Name Server	101	
iso-tsap	ISO-TSAP	102	
uucp-path	UUCP Path Service	117	
ntp	Network Time Protocol	123	123
x	X Window Service	6000+Display Number	

Format and Protocols (64-bit gABI) - EXPERIMENTAL

Format and Protocols changes:

#	Facility	Location	Description
1	XDR	gABI	page 7-14, add new data type "Quad Precision Floating-point", after the description of the "Double-precision Floating-point" as follows:

Quad-precision Floating-point

XDR defines the encoding for the quad-precision floating-point data type "long double" (128 bits or 16 bytes). The encoding used is a logical extension to the IEEE standard for single and double precision encoding. XDR encodes the following three fields, which describe the quad-precision floating-point number:

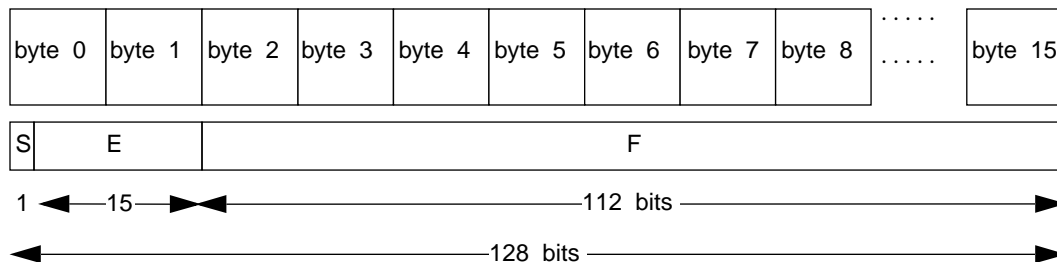
- S:** The sign of the number. Values 0 and 1 represent positive and negative, respectively. One bit.
- E:** The exponent of the number, base 2. 15 bits are devoted to this field. The exponent is biased by 16383.
- F:** The fractional part of the number's mantissa, base 2. 112 bits are devoted to this field.

Therefore, the floating-point number is described by:

$$(-1)^{S} * 2^{(E-Bias)} * 1.F$$

It is declared as follows:

Quad-Precision Floating-point



The most and least significant bits of a quad-precision floating-point number are 0 and 127. The beginning bit (and the most significant bit) offsets of S, E, and F are 0, 1, and 16, respectively. Note that these numbers refer to the mathematical positions of the bits, and NOT to their actual physical locations (which vary from medium to medium).

Even though quad-precision floating-point is not yet a part of the IEEE standard, the IEEE 754 specifications should be consulted concerning the encoding for signed zero, signed infinity (overflow), and denormalized numbers (underflow). According to IEEE specifications, the "NaN" (not a number) is system dependent.

CHAPTER 8: System Commands

SCD
2.4

System Commands

Introduction

This chapter is common to both the 32-bit ABI and 64-bit ABI, unless explicitly noted otherwise. 32-bit ABI commands are REQUIRED, unless explicitly noted otherwise. 64-bit ABI commands are EXPERIMENTAL.

This chapter contains the commands for application programs as listed in the *System V Application Binary Interface (Third Edition)*, and described in the *System V Interface Definition, (Third Edition)*.

Table 8-1. Commands for Application Programs

ar 2.3	false	pwd *	uucp
awk 2.4 (L)	find †† 2.4 (L)	rm 2.4 (L)	uulog
basename 2.3	fmtmsg	rmdir 2.4 (L)	uustat
cat	gencat 2.3	sed 2.4 (L)	uux
cd†	gettxt	sh ††, 2.4 (L)	vi
chgrp 2.4 (L)	grep	sleep	wait†
chmod 2.4 (L)	id	sort	wc 2.3 (L)
chown 2.4 (L)	kill	stty	who ††
cmp 2.4 (L)	line	su	
cp 2.4 (L)	ln 2.4 (L)	sum 2.3 (L)	
cpio	logname	tail 2.4 (L)	
compress 2.3 (L)	lp	tee 2.4 (L)	
date	ls 2.4 (L)	test *, 2.4 (L)	
dirname 2.3	make 2.3	touch 2.4 (L)	
dd 2.4 (L)	mkdir 2.4 (L)	tr 2.4 (L)	
df	mv 2.4 (L)	true	
echo *	passwd	tty	
ed	pg	umask †	
ex ††	pr ††	uname	
expr	priocntl	uncompress 2.3 (L)	

* These commands marked are also built into the standard UNIX system shell, sh.

† These commands are only available as commands built-in to the UNIX system shell, sh.

†† - See system commands changes that follow.

L- REQUIRED 32-bit ABI Large File aware utility.

2.3- New interfaces added to SCD2.3.

2.4- New interfaces added to SCD2.4.

Figure 8-1 : tar.h

```
#define TMAGIC          "ustar"
#define TMAGLEN        6
#define TVERSION       "00"
#define TVERSLEN       2
#define REGTYPE        '0'
#define AREGTYPE       '\0'
#define LNKTYPE        '1'
#define SYMTYPE        '2'
#define CHRTYPE        '3'
#define BLKTYPE        '4'
#define DIRTYPE        '5'
#define FIFOTYPE       '6'
#define CONTTYPE       '7'
#define TSUID          04000
#define TSGID          02000
#define TSVTX         01000
#define TUREAD         00400
#define TUWRITE        00200
#define TUEXEC         00100
#define TGREAD         00040
#define TGWRITE        00020
#define TGEXEC         00010
#define TOREAD         00004
#define TOWRITE        00002
#define TOEXEC         00001
```

System Commands Changes

The following are changes to the basic system commands (detailed in the *System V Application Binary Interface*), as reported to SPARC International.

#	Facility	Location	Description
1	ex(BU_CMD)	SVID, Vol. 2	Change - The SVID states that the "ed compatible" option of ex causes the g suffix on substitute commands to be remembered, and toggled by repeating the suffix. Omitted from this description is the fact that this behavior is applicable only to the "&" form of substitute commands.

System Commands Changes (continued)

#	Facility	Location	Description
1	<i>ex</i> (BU_CMD)	SVID, Vol. 2	Change - The “c” command should be defined as “Enters input mode; the input text replaces the specified lines. The last input line becomes the current line; if no lines are input the line before the deleted line(s) becomes the current line.”
1	<i>ex</i> (BU_CMD)	SVID, Vol. 2	Change - The “m” command description must be changed to note that the current line becomes the last of the moved lines, rather than the first.
2	<i>find</i> (BU_CMD)	SVID, Vol. 2	Change the descriptions of -atime, -mtime, and -ctime from “in n days” to “n days ago.”
3	<i>pr</i> (BU_CMD)	SVID, Vol. 2	Change - The SVID says that using -m with the -column option will cause the -m option to override the -column option. This does not match current practice; using these two options together will be treated as an error.
3	<i>pr</i> (BU_CMD)	SVID, Vol. 2	Change - Comments about truncating lines in the text of the description and in the options are incorrect with respect to single column output: existing practice and P1003.2 is that truncation is not applied to single column output. The “note” in the description of the -w option is to be applicable to multi-column output only. In the description change the second paragraph to read: “By default, in multi-column mode, columns are...”
4	<i>sh</i> (BU_CMD)	SVID, Vol. 2	Changed - In the section marked “Input/Output” the description of “<<[-]word” states: “... \ must be used to quote the characters \, \$, ‘, and the first character of word” should be changed to read “... \ must be used to quote the characters \, \$, and “”. This matches both existing practice and P1003.2.
5	<i>who</i> (AU_CMD)	SVID, Vol. 2	Change the description of the -T and -a option to “The -T and -a options to who are unspecified and cannot be relied on to be portable.” <i>Rationale:</i> On investigation, these options were found to differ on various SPARC implementations. The -a option is an aggregate option; rather than using this option, for SCD 2.x portability an application should use the specific individual options to who that the application requires. Rather than using the -T option, an application should use either the -s or -u option for SCD 2.x portability.
6	<i>sh</i> (BU_CMD)	gABI	On page 8-1 at the bottom of the page, the text:] . . . UNIX system shell (<i>sh</i> (BU_CMD)). should read:] UNIX system shell <i>sh</i> (BU_CMD).

CHAPTER 9: Execution Environment

SCD
2.4

Execution Environment

Introduction

This chapter is common to both the 32-bit ABI and 64-bit ABI, unless explicitly noted otherwise.

All information regarding File System Structure and Contents may be found in Chapter 9 of the *System V Application Binary Interface (Third Edition)*.

Execution Environment Changes

The following are changes to the *System V Application Binary Interface (Third Edition)*, the *System V Application Binary Interface - SPARC Processor Supplement (Third Edition)*, and the *System V Interface Definition (Third Edition)* as reported to SPARC International.

#	Facility	Location	Description
1	Root sub-tree - <i>/dev</i>	gABI	<p>Change Figure 9-1 page 9-4, Required device files are: <i>/dev/tty</i>, <i>/dev/null</i>, <i>/dev/console</i>, <i>/dev/zero</i>.</p> <p>The device: <i>/dev/lpx</i> should be removed from this figure; <i>/dev/lpX</i> is not an SCD required device. Similarly the sub-directories: <i>/dev/rmt</i> and <i>/dev/mt</i> should be removed from this figure; these are not SCD required <i>/dev/</i> sub-directories.</p> <p>The definition of <i>/dev/zero</i> can be found in the SCD 2.4 Interface Semantics.</p>



CHAPTER 10: Windowing and Terminal Interfaces

SCD
2.4

Windowing and Terminal Interface

Introduction

This chapter is common to both the 32-bit ABI and 64-bit ABI, except that the 64-bit ABI is EXPERIMENTAL.

This chapter contains the following major sections:

- 1- The X library (libX11)
- 2- The Motif 1.2 widget set library (libMrm, libXm)
- 3- The X Toolkit (libXt)
- 4- The Open Look Widget set (libXol)
- 5- The X extensions library (libEext)

The following table identifies the actual version numbers and reference names for windowing and terminal shared objects on a SPARC system:

Library	Reference Name
libMrm	/usr/lib/libMrm.so.1.2 (deprecated) /usr/lib/libMrm.so.3 (deprecated) /usr/dt/lib/libMrm.so.1.2 (deprecated) /usr/dt/lib/libMrm.so.3
libX11	/usr/lib/libX11.so.5 /usr/lib/libX11.so.4 (deprecated)
libXext	/usr/lib/libXext.so.0
libXol	/usr/lib/libXol.so.3
libXm	/usr/lib/libXm.so.1.2 (deprecated) /usr/lib/libXm.so.3 (deprecated) /usr/dt/lib/libXm.so.1.2 (deprecated) /usr/dt/lib/libXm.so.3
libXt	/usr/lib/libXt.so.5 /usr/lib/libXt.so.4 (deprecated)

The X Library

Overview

This section identifies binary interfaces for libX, which are defined in *The X Window System (Third Edition)* by Robert W. Scheifler and James Gettys (Digital Press, ISBN 1-55558-088-2).

In addition, all SCD 2.4 systems will support the mechanisms and conventions as specified in the *Inter-Client Communications Convention Manual (ICCCM)* in *The X Window System (Third Edition)* by Robert W. Scheifler and James Gettys (Digital Press, ISBN 1-55558-088-2).

The libX Interfaces

The interfaces listed below in Table 10-1 have been included in SCD 2.4 because they are REQUIRED to be present on all compliant systems, in the dynamic libraries: /usr/lib/libX11.so.4 and /usr/lib/libX11.so.5. Table 10-2 contains the exported data which are also REQUIRED to be present in /usr/lib/libX11.so.4 and /usr/lib/libX11.so.5. The format of these entries is: data[size].

Since the X Version 11, Release 5 specification is a proper superset of the X Version 11, Release 4 specification, application developers can link to either /usr/lib/libX11.so.4 or /usr/lib/libX11.so.5 to access X Version 11, Release 4 and Release 5 interfaces. See the SPARC Compliance Definition 2.0, 2.1, 2.2, 2.3, or 2.4 for a list of X11R4 components.

Figures 10-1 through 10-7 detail the manifest constants and visible data structures associated with the X library.

Table 10-1: Contents of libX (1 of 4)

XActivateScreenSaver	XcmsCIELabQueryMinL	XConvertSelection
XAddExtension	XcmsCIELabToCIEXYZ	XCopyArea
XAddHost	XcmsCIELabWhiteShiftColors	XCopyColormapAndFree
XAddHosts	XcmsCIELuvClipL	XCopyGC
XAddPixel	XcmsCIELuvClipLuv	XCopyPlane
XAddToExtensionList	XcmsCIELuvClipuv	XCreateBitmapFromData
XAddToSaveSet	XcmsCIELuvQueryMaxC	XCreateColormap
XAllocClassHint	XcmsCIELuvQueryMaxL	XCreateFontCursor
XAllocColor	XcmsCIELuvQueryMaxLC	XCreateFontSet
XAllocColorCells	XcmsCIELuvQueryMinL	XCreateGC
XAllocColorPlanes	XcmsCIELuvToCIEuvY	XCreateGlyphCursor
XAllocIconSize	XcmsCIELuvWhiteShiftColors	XCreateIC
XAllocNamedColor	XcmsCIEuvYToCIELuv	XCreateImage
XAllocSizeHints	XcmsCIEuvYToCIEXYZ	XCreatePixmap
XAllocStandardColormap	XcmsCIEuvYToTekHVC	XCreatePixmapCursor
XAllocWMHints	XcmsCIExyYToCIEXYZ	XCreatePixmapFromBitmapData
XAllowEvents	XcmsCIEXYZToCIELab	XCreateRegion
XAllPlanes	XcmsCIEXYZToCIEuvY	XCreateSimpleWindow
XAutoRepeatOff	XcmsCIEXYZToCIExyY	XCreateWindow
XAutoRepeatOn	XcmsCIEXYZToRGBi	XDefaultColormap
XBaseFontNameListOfFontSet	XcmsClientWhitePointOfCCC	XDefaultColormapOfScreen
XBell	XcmsConvertColors	XDefaultDepth
XBitmapBitOrder	XcmsCreateCCC	XDefaultDepthOfScreen
XBitmapPad	XcmsDefaultCCC	XDefaultGC
XBitmapUnit	XcmsDisplayOfCCC	XDefaultGCOfScreen
XBlackPixel	XcmsFormatOfPrefix	XDefaultRootWindow
XBlackPixelOfScreen	XcmsFreeCCC	XDefaultScreen
XCellsOfScreen	XcmsLookupColor	XDefaultScreenOfDisplay
XChangeActivePointerGrab	XcmsPrefixOfFormat	XDefaultString
XChangeGC	XcmsQueryBlack	XDefaultVisual
XChangeKeyboardControl	XcmsQueryBlue	
XChangeKeyboardMapping	XcmsQueryColor	
XChangePointerControl	XcmsQueryColors	
XChangeProperty	XcmsQueryGreen	
XChangeSaveSet	XcmsQueryRed	
XChangeWindowAttributes	XcmsQueryWhite	
XCheckIfEvent	XcmsRGBiToCIEXYZ	
XCheckMaskEvent	XcmsRGBiToRGB	
XCheckTypedEvent	XcmsRGBToRGBi	
XCheckTypedWindowEvent	XcmsScreenNumberOfCCC	
XCheckWindowEvent	XcmsScreenWhitePointOfCCC	
XCirculateSubwindows	XcmsSetCompressionProc	
XCirculateSubwindowsDown	XcmsSetWhiteAdjustProc	
XCirculateSubwindowsUp	XcmsSetWhitePoint	
XClearArea	XcmsStoreColor	
XClearWindow	XcmsStoreColors	
XClipBox	XcmsTekHVCClipC	
XCloseDisplay	XcmsTekHVCClipV	
XCloseIM	XcmsTekHVCClipVC	
XcmsAddColorSpace	XcmsTekHVCQueryMaxC	
XcmsAddFunctionSet	XcmsTekHVCQueryMaxV	
XcmsAllocColor	XcmsTekHVCQueryMaxVC	
XcmsAllocNamedColor	XcmsTekHVCQueryMaxVSamples	
XcmsCCCOFColormap	XcmsTekHVCQueryMinV	
XcmsCIELabClipab	XcmsTekHVCToCIEuvY	
XcmsCIELabClipL	XcmsTekHVCWhiteShiftColors	
XcmsCIELabClipLab	XcmsVisualOfCCC	
XcmsCIELabQueryMaxC	XConfigureWindow	
XcmsCIELabQueryMaxL	XConnectionNumber	
XcmsCIELabQueryMaxLC	XContextDependentDrawing	

Table 10-1. Contents of libX (2 of 4)

XDefaultVisualOfScreen	XFetchBuffer	XGetSubImage
XDefineCursor	XFetchBytes	XGetTextProperty
XDeleteContext	XFetchName	XGetTransientForHint
XDeleteModifiermapEntry	XFillArc	XGetVisualInfo
XDeleteProperty	XFillArcs	XGetWindowAttributes
XDestroyIC	XFillPolygon	XGetWindowProperty
XDestroyImage	XFillRectangle	XGetWMLClientMachine
XDestroyRegion	XFillRectangles	XGetWMCOLormapWindows
XDestroySubwindows	XFilterEvent	XGetWMHints
XDestroyWindow	XFindContext	XGetWMLIconName
XDisableAccessControl	XFindOnExtensionList	XGetWMName
XDisplayCells	XFlush	XGetWMNormalHints
XDisplayHeight	XFlushGC	XGetWMProtocols
XDisplayHeightMM	XFontsOfFontSet	XGetWMSizeHints
XDisplayKeycodes	XForceScreenSaver	XGrabButton
XDisplayMotionBufferSize	XFree	XGrabKey
XDisplayName	XFreeColormap	XGrabKeyboard
XDisplayOfIM	XFreeColors	XGrabPointer
XDisplayOfScreen	XFreeCursor	XGrabServer
XDisplayPlanes	XFreeExtensionList	XHeightMMOfScreen
XDisplayString	XFreeFont	XHeightOfScreen
XDisplayWidth	XFreeFontInfo	XIconifyWindow
XDisplayWidthMM	XFreeFontNames	XIfEvent
XDoesBackingStore	XFreeFontPath	XImageByteOrder
XDoesSaveUnders	XFreeFontSet	XIMOfIC
XDrawArc	XFreeGC	XInitExtension
XDrawArcs	XFreeModifiermap	XInsertModifiermapEntry
XDrawImageString	XFreePixmap	XInstallColormap
XDrawImageString16	XFreeStringList	XInternAtom
XDrawLine	XGCContextFromGC	XIntersectRegion
XDrawLines	XGeometry	
XDrawPoint	XGetAtomName	
XDrawPoints	XGetClassHint	
XDrawRectangle	XGetCommand	
XDrawRectangles	XGetDefault	
XDrawSegments	XGetErrorDatabaseText	
XDrawString	XGetErrorText	
XDrawString16	XGetFontPath	
XDrawText	XGetFontProperty	
XDrawText16	XGetGCValues	
XEHeadOfExtensionList	XGetGeometry	
XEmptyRegion	XGetIconName	
XEnableAccessControl	XGetIconSizes	
XEqualRegion	XGetICValues	
XESetCloseDisplay	XGetImage	
XESetCopyGC	XGetIMValues	
XESetCreateFont	XGetInputFocus	
XESetCreateGC	XGetKeyboardControl	
XESetError	XGetKeyboardMapping	
XESetErrorString	XGetModifierMapping	
XESetEventToWire	XGetMotionEvents	
XESetFlushGC	XGetNormalHints	
XESetFreeFont	XGetPixel	
XESetFreeGC	XGetPointerControl	
XESetPrintErrorValues	XGetPointerMapping	
XESetWireToError	XGetRGCColormaps	
XESetWireToEvent	XGetScreenSaver	
XEventMaskOfScreen	XGetSelectionOwner	
XEventsQueued	XGetSizeHints	
XExtentsOfFontSet	XGetStandardColormap	

Table 10-1. Contents of libX (3 of 4)

XKeycodeToKeysym	XPutBackEvent	XRootWindow
XKeysymToKeycode	XPutImage	XRootWindowOfScreen
XKeysymToString	XPutPixel	XRotateBuffers
XKillClient	XQLength	XRotateWindowProperties
XLastKnownRequestProcessed	XQueryBestCursor	XSaveContext
XListDepths	XQueryBestSize	XScreenCount
XListExtensions	XQueryBestStipple	XScreenNumberOfScreen
XListFonts	XQueryBestTile	XScreenOfDisplay
XListFontsWithInfo	XQueryColor	XScreenResourceString
XListHosts	XQueryColors	XSelectInput
XListInstalledColormaps	XQueryExtension	XSendEvent
XListPixmapFormats	XQueryFont	XServerVendor
XListProperties	XQueryKeymap	XSetAccessControl
XLoadFont	XQueryPointer	XSetAfterFunction
XLoadQueryFont	XQueryTextExtents	XSetArcMode
XLocaleOfFontSet	XQueryTextExtents16	XSetBackground
XLocaleOfIM	XQueryTree	XSetClassHint
XLookupColor	XRaiseWindow	XSetClipMask
XLookupKeysym	XReadBitmapFile	XSetClipOrigin
XLookupString	XRebindKeysym	XSetClipRectangles
XLowerWindow	XRecolorCursor	XSetCloseDownMode
XMapRaised	XReconfigureWMWindow	XSetCommand
XMapSubwindows	XRectInRegion	XSetDashes
XMapWindow	XRefreshKeyboardMapping	XSetErrorHandler
XMaskEvent	XRemoveFromSaveSet	XSetFillRule
XMatchVisualInfo	XRemoveHost	XSetFillStyle
XMaxCmapsOfScreen	XRemoveHosts	XSetFont
XMaxRequestSize	XReparentWindow	XSetFontPath
XmbDrawImageString	XResetScreenSaver	XSetForeground
XmbDrawString	XResizeWindow	XSetFunction
XmbDrawText	XResourceManagerString	
XmbLookupString	XRestackWindows	
XmbResetIC	XrmCombineDatabase	
XmbSetWMPProperties	XrmCombineFileDatabase	
XmbTextEscapement	XrmDestroyDatabase	
XmbTextExtents	XrmEnumerateDatabase	
XmbTextListToTextProperty	XrmGetDatabase	
XmbTextPerCharExtents	XrmGetFileDatabase	
XmbTextPropertyToTextList	XrmGetResource	
XMinCmapsOfScreen	XrmGetStringDatabase	
XMoveResizeWindow	XrmInitialize	
XMoveWindow	XrmLocaleOfDatabase	
XNewModifiermap	XrmMergeDatabases	
XNextEvent	XrmParseCommand	
XNextRequest	XrmPermStringToQuark	
XNoOp	XrmPutFileDatabase	
XOffsetRegion	XrmPutLineResource	
XOpenDisplay	XrmPutResource	
XOpenIM	XrmPutStringResource	
XParseColor	XrmQGetResource	
XParseGeometry	XrmQGetSearchList	
XPeekEvent	XrmQGetSearchResource	
XPeekIfEvent	XrmQPutResource	
XPending	XrmQPutStringResource	
Xpermalloc	XrmQuarkToString	
XPlanesOfScreen	XrmSetDatabase	
XPointInRegion	XrmStringToBindingQuarkList	
XPolygonRegion	XrmStringToQuark	
XProtocolRevision	XrmStringToQuarkList	
XProtocolVersion	XrmUniqueQuark	

Table 10-1. Contents of libX (4 of 4)

XSetGraphicsExposures	XStoreBuffer	XwcTextExtents
XSetICFocus	XStoreBytes	XwcTextListToTextProperty
XSetIconName	XStoreColor	XwcTextPerCharExtents
XSetIconSizes	XStoreColors	XwcTextPropertyToTextList
XSetICValues	XStoreName	XWhitePixel
XSetInputFocus	XStoreNamedColor	XWhitePixelOfScreen
XSetIOErrorHandler	XStringListToTextProperty	XWidthMMOfScreen
XSetLineAttributes	XStringToKeysym	XWidthOfScreen
XSetLocaleModifiers	XSubImage	XWindowEvent
XSetModifierMapping	XSubtractRegion	XWithdrawWindow
XSetNormalHints	XSupportsLocale	XWMGeometry
XSetPlaneMask	XSync	XWriteBitmapFile
XSetPointerMapping	XSynchronize	XXorRegion
XSetRegion	XTextExtents	_XAllocScratch ¹
XSetRGBColormaps	XTextExtents16	_XFlush ¹
XSetScreenSaver	XTextPropertyToStringList	_XFlushGCCache ¹
XSetSelectionOwner	XTextWidth	_XRead ¹
XSetSizeHints	XTextWidth16	_XReadPad ¹
XSetStandardColormap	XTranslateCoordinates	_XReply ¹
XSetStandardProperties	XUndefineCursor	_XSend ¹
XSetState	XUngrabButton	_XSetLastRequestRead ¹
XSetStipple	XUngrabKey	1 - New in SCD 2.3
XSetSubwindowMode	XUngrabKeyboard	
XSetTextProperty	XUngrabPointer	
XSetTile	XUngrabServer	
XSetTransientForHint	XUninstallColormap	
XSetTSTOrigin	XUnionRectWithRegion	
XSetWindowBackground	XUnionRegion	
XSetWindowBackgroundPixmap	XUnloadFont	
XSetWindowBorder	XUnmapSubwindows	
XSetWindowBorderPixmap	XUnmapWindow	
XSetWindowBorderWidth	XUnsetICFocus	
XSetWindowColormap	XVaCreateNestedList	
XSetWMCClientMachine	XVendorRelease	
XSetWMCOLormapWindows	XVisualIDFromVisual	
XSetWMHints	XWarpPointer	
XSetWMIconName	XwcDrawImageString	
XSetWMName	XwcDrawString	
XSetWMNormalHints	XwcDrawText	
XSetWMProperties	XwcFreeStringList	
XSetWMProtocols	XwcLookupString	
XSetWMSizeHints	XwcResetIC	
XShrinkRegion	XwcTextEscapement	

Table 10-2: Exported Data for libX.

XcmsCIELabColorSpace[0x18]
 XcmsCIELuvColorSpace[0x18]
 XcmsCIEuvYColorSpace[0x18]
 XcmsCIExyYColorSpace[0x18]
 XcmsCIEXYZColorSpace[0x18]
 XcmsLinearRGBFunctionSet[0x18]
 XcmsRGBColorSpace[0x18]
 XcmsRGBiColorSpace[0x18]
 XcmsTekHVCCColorSpace[0x18]
 XcmsUNDEFINEDColorSpace[0x18]
 _Xdebug[0x4]

Unsafe Macros

Ordinarily, this document only specifies the system resources available for use by applications on all SPARC compliant systems and makes no comment regarding the programming language or API used by application programmers for building applications. But SPARC International recognizes that many SPARC applications will be written in the C programming language and are likely to use the API specified by the X Consortium. Some of the data structures defined as part of the X ABI, such as struct Display, struct Screen, and struct XImage, are intended to be opaque to the application; that is, the application isn't supposed to contain any knowledge of the size or layout of the data structures.

Some of the macros defined by the X Consortium as part of the X API violate this assumption for opaque data. Below is a table of macros from the X API which are considered by SPARC International to be unsafe; that is, they cause knowledge about the size and/or layout of opaque data structures to be embedded in applications. Embedding this information in an application may prevent the application from being binary compatible with future versions of X which use a different size or layout for these opaque data structures.

Fortunately, each of the unsafe macros has a counterpart in the X library. Table 10-3 below lists each of the unsafe macros and its safe function counterpart from the X library.

Table 10-3: ABI Unsafe Macros

Unsafe Macro	Equivalent X Function
BitmapPad	XBitmapPad
BitmapUnit	XBitmapUnit
BlackPixel	XBlackPixel
BlackPixelOfScreen	XBlackPixelOfScreen
CellsOfScreen	XCCellsOfScreen
ClientWhitePointOfCCC	XClientWhitePointOfCCC
ConnectionNumber	XConnectionNumber
DefaultColormap	XDefaultColormap
DefaultColormapOfScreen	XDefaultColormapOfScreen
DefaultDepth	XDefaultDepth
DefaultDepthOfScreen	XDefaultDepthOfScreen
DefaultGC	XDefaultGC
DefaultGCOfScreen	XDefaultGCOfScreen
DefaultRootWindow	XDefaultRootWindow
DefaultScreen	XDefaultScreen
DefaultScreenOfDisplay	XDefaultScreenOfDisplay
DefaultVisual	XDefaultVisual
DefaultVisualOfScreen	XDefaultVisualOfScreen
DisplayCells	XDisplayCells
DisplayHeight	XDisplayHeight
DisplayHeightMM	XDisplayHeightMM
DisplayOfCCC	XDisplayOfCCC
DisplayOfScreen	XDisplayOfScreen
DisplayPlanes	XDisplayPlanes
DisplayString	XDisplayString

Table 10-3: ABI Unsafe Macros

Unsafe Macro	Equivalent X Function
DisplayWidth	XDisplayWidth
DisplayWidthMM	XDisplayWidthMM
DoesBackingStore	XDoesBackingStore
DoesSaveUnders	XDoesSaveUnders
EventMaskOfScreen	XEventMaskOfScreen
HeightMMOfScreen	XHeightMMOfScreen
HeightOfScreen	XHeightOfScreen
ImageByteOrder	XImageByteOrder
LastKnownRequestProcessed	XLastKnownRequestProcessed
MaxCmapsOfScreen	XMaxCmapsOfScreen
MinCmapsOfScreen	XMinCmapsOfScreen
NextRequest	XNextRequest
PlanesOfScreen	XPlanesOfScreen
ProtocolRevision	XProtocolRevision
ProtocolVersion	XProtocolVersion
QLength	XQLength
RootWindow	XRootWindow
RootWindowOfScreen	XRootWindowOfScreen
ScreenCount	XScreenCount
ScreenNumberOfCCC	XScreenNumberOfCCC
ScreenOfDisplay	XScreenOfDisplay
ScreenWhiteOfCCC	XScreenWhiteOfCCC
ServerVendor	XServerVendor
VendorRelease	XVendorRelease
VisualOfCCC	XVisualOfCCC
WhitePixel	XWhitePixel
WhitePixelOfScreen	XWhitePixelOfScreen
WidthMMOfScreen	XWidthMMOfScreen
WidthOfScreen	XWidthOfScreen

Following are the definitions of manifest constants and data types needed by applications to interface to the Xlib functions listed in Table 10-1.

Though the SPARC Compliance Definition specifies an Application Binary Interface (ABI) rather than an Application Program Interface (API) the manifest constants and data type definitions are broken up into different tables based on which header files a programmer would ordinarily expect to find the definitions for two reasons:

- it makes the document more informative for the programmer who is trying to meet the standard, and
- it is expected that this will make the document easier to edit and review.

The header files these definitions are taken from are

- `<X11/Xlib.h>`,
- `<X11/X.h>`,
- `<X11/Xresource.h>`,
- `<X11/Xutil.h>`,
- `<X11/Xcms.h>`, and
- `<X11/keysymdef.h>`.

All header definitions are based on X, version 11, release 5 from the MIT X Consortium.

X Library Changes

The following are changes to the System V Application Binary Interface, SPARC processor supplement, Third Edition.

#	Facility	Location	Description
1	<X11/X.h>	psABI	<p>Addition - On page 6-81, in Figure 6-83 add the following constants:</p> <pre> #define X_PROTOCOL 11 #define X_PROTOCOL_REVISION 0 #define ShiftMapIndex 0 #define LockMapIndex 1 #define ControlMapIndex 2 #define Mod1MapIndex 3 #define Mod2MapIndex 4 #define Mod3MapIndex 5 #define Mod4MapIndex 6 #define Mod5MapIndex 7 #define FirstExtensionError 128 #define LastExtensionError 255 #define GCLastBit 22 #define FontChange 255 #define DisableScreenSaver 0 #define DisableScreenInterval 0 #define HostInsert 0 #define HostDelete 1 </pre>
2	<X11/X.h>	psABI	<p>Addition - On page 6-81, in Figure 6-83 add the following type definition: typedef unsigned long Mask;</p>
3	<X11/Xcms.h>	psABI	<p>Addition - On page 6-99, in Figure 6-101 make the following addition:</p> <pre> typedef struct _XcmsCCC { Display *dpy; int screenNumber; Visual *visual; XcmsColor clientWhitePt; XcmsCompressionProc gamutCompProc; XPointer gamutCompClientData; XcmsWhiteAdjustProc whitePtAdjProc; XPointer whitePtAdjClientData; XcmsPerScrnInfo *pPerScrnInfo; } XcmsCCCRec; </pre>
4	<X11/Xcms.h>	psABI	<p>Change - On page 6-99, in Figure 6-101 make the following change:</p> <pre> Replace typedef void *XcmsCCC; With typedef struct _XcmsCCC *XcmsCCC; </pre>
5	<X11/Xcms.h>	psABI	<p>Addition - On page 6-96, in Figure 6-98 add the following type definitions:</p> <pre> typedef Status (*XcmsWhiteAdjustProc)(); typedef Status (*XcmsScreenInitProc)(); typedef void (*XcmsScreenFreeProc)(); </pre>

X Library Changes (continued)

#	Facility	Location	Description
6	<X11/Xlib.h>	psABI	<p>Change - On page 6-101, in Figure 6-104 make the following changes:</p> <p>Replace <code>typedef void XExtData;</code> With</p> <pre>typedef struct _XExtData { int number; struct _XExtData *next; int (*free_private)(); XPointer private_data; } XExtData;</pre> <p>Replace <code>typedef void XExtCodes;</code> With</p> <pre>typedef struct { int extension; int major_opcode; int first_event; int first_error; } XExtCodes;</pre>
7	<X11/Xlib.h>	psABI	<p>Change - On page 6-102, in Figure 6-105 make the following changes:</p> <p>Replace <code>typedef void *GC;</code> With <code>typedef struct _XGC *GC;</code></p> <p>Replace <code>typedef struct _dummy Visual;</code> With <code>typedef struct Visual;</code></p>
8	<X11/Xlib.h>	psABI	<p>Addition - On page 6-102, in Figure 6-105 add the following type definition: <code>typedef struct Depth;</code></p>
9	<X11/Xlib.h>	psABI	<p>Change - On page 6-103, in Figure 6-106 make the following change:</p> <p>Replace <code>typedef struct _dummy Screen;</code> With <code>typedef struct _Screen Screen;</code></p>
10	<X11/Xlib.h>	psABI	<p>Change - On page 6-107, in Figure 6-110 make the following change:</p> <p>Replace <code>typedef struct _dummy Display;</code> With <code>typedef struct _XDisplay Display;</code></p>
11	<X11/Xlib.h>	psABI	<p>Change - On page 6-121, in Figure 6-124 make the following change:</p> <p>Replace <code>typedef struct _dummy XFontSet;</code> With <code>typedef struct _XFontSet *XFontSet;</code></p>
12	<X11/Xresource.h>	psABI	<p>Addition - On page 6-127, in Figure 6-130 add the following constant:</p> <pre>#define NULLSTRING ((XrmString) 0)</pre>
13	<X11/Xresource.h>	psABI	<p>Addition - On page 6-127, in Figure 6-130 add the following type definitions:</p> <pre>typedef char *XrmString; typedef struct _XrmHashBucketRec XrmHashBucket; typedef struct _XrmHashBucketRec *XrmDatabase;</pre>

X Library Changes (continued)

#	Facility	Location	Description
14	<X11/Xlib.h>	psABI	<p>Change - On page 6-122, in Figure 6-125 make the following changes:</p> <p>Replace</p> <pre>typedef struct { char *chars; int nchars; int delta; XFontSet font_set; } XmbTextItem;</pre> <p>With</p> <pre>typedef struct { char *chars; int nchars; int delta; XFontSet font_set; } XmbTextItem;</pre> <p>Replace With</p> <pre>typedef void *XIM; typedef struct _XIM *XIM;</pre> <p>Replace With</p> <pre>typedef void *XIC; typedef struct _XIC *XIC;</pre> <p>Replace With</p> <pre>typedef void *XIC; typedef struct _XIC *XIC;</pre>
15	<X11/Xutil.h>	psABI	<p>Change - On page 6-132, in Figure 6-135 make the following changes:</p> <p>Replace With:</p> <pre>typedef void *Region; typedef struct _XRegion *Region;</pre>

Figure 10-1: Manifest Constants from <X11/keysymdef.h>

```

#define XK_VoidSymbol          0xFFFFF
#define XK_BackSpace          0xFF08
#define XK_Tab                 0xFF09
#define XK_Linefeed           0xFF0A
#define XK_Clear               0xFF0B
#define XK_Return              0xFF0D
#define XK_Pause               0xFF13
#define XK_Scroll_Lock        0xFF14
#define XK_Escape              0xFF1B
#define XK_Delete              0xFFFF
#define XK_Multi_key           0xFF20
#define XK_Kanji                0xFF21
#define XK_Muhenkan             0xFF22
#define XK_Henkan_Mode         0xFF23
#define XK_Henkan               0xFF23
#define XK_Romaji               0xFF24
#define XK_Hiragana             0xFF25
#define XK_Katakana             0xFF26
#define XK_Hiragana_Katakana   0xFF27
#define XK_Zenkaku              0xFF28
#define XK_Hankaku              0xFF29
#define XK_Zenkaku_Hankaku     0xFF2A
#define XK_Touroku              0xFF2B
#define XK_Massyo               0xFF2C
#define XK_Kana_Lock           0xFF2D
#define XK_Kana_Shift          0xFF2E
#define XK_Eisu_Shift          0xFF2F
#define XK_Eisu_toggle         0xFF30
#define XK_Home                 0xFF50
#define XK_Left                 0xFF51
#define XK_Up                   0xFF52
#define XK_Right                0xFF53
#define XK_Down                 0xFF54
#define XK_Prior                0xFF55
#define XK_Next                 0xFF56
#define XK_End                  0xFF57
#define XK_Begin                0xFF58
#define XK_Select               0xFF60
#define XK_Print                0xFF61
#define XK_Execute              0xFF62
#define XK_Insert               0xFF63
#define XK_Undo                 0xFF65
#define XK_Redo                 0xFF66
#define XK_Menu                 0xFF67
#define XK_Find                 0xFF68
#define XK_Cancel               0xFF69
#define XK_Help                 0xFF6A
#define XK_Break                0xFF6B
#define XK_Mode_switch          0xFF7E
#define XK_script_switch        0xFF7E
#define XK_Num_Lock             0xFF7F
#define XK_KP_Space             0xFF80
#define XK_KP_Tab               0xFF89
#define XK_KP_Enter             0xFF8D
#define XK_KP_F1                0xFF91
#define XK_KP_F2                0xFF92
#define XK_KP_F3                0xFF93
#define XK_KP_F4                0xFF94
#define XK_KP_Equal             0xFFBD
#define XK_KP_Multiply          0xFFAA
#define XK_KP_Add               0xFFAB
#define XK_KP_Separator         0xFFAC
#define XK_KP_Subtract          0xFFAD
#define XK_KP_Decimal           0xFFAE
#define XK_KP_Divide            0xFFAF
#define XK_KP_0                 0xFFB0
#define XK_KP_1                 0xFFB1
#define XK_KP_2                 0xFFB2
#define XK_KP_3                 0xFFB3
#define XK_KP_4                 0xFFB4
#define XK_KP_5                 0xFFB5
#define XK_KP_6                 0xFFB6
#define XK_KP_7                 0xFFB7
#define XK_KP_8                 0xFFB8
#define XK_KP_9                 0xFFB9
#define XK_F1                   0xFFBE
#define XK_F2                   0xFFBF

```

```
#define XK_F3                0xFFC0
#define XK_F4                0xFFC1
#define XK_F5                0xFFC2
#define XK_F6                0xFFC3
#define XK_F7                0xFFC4
#define XK_F8                0xFFC5
#define XK_F9                0xFFC6
#define XK_F10               0xFFC7
#define XK_F11               0xFFC8
#define XK_L1                0xFFC8
#define XK_F12               0xFFC9
#define XK_L2                0xFFC9
#define XK_F13               0xFFCA
#define XK_L3                0xFFCA
#define XK_F14               0xFFCB
#define XK_L4                0xFFCB
#define XK_F15               0xFFCC
#define XK_L5                0xFFCC
#define XK_F16               0xFFCD
#define XK_L6                0xFFCD
#define XK_F17               0xFFCE
#define XK_L7                0xFFCE
#define XK_F18               0xFFCF
#define XK_L8                0xFFCF
#define XK_F19               0xFFD0
#define XK_L9                0xFFD0
#define XK_F20               0xFFD1
#define XK_L10               0xFFD1
#define XK_F21               0xFFD2
#define XK_R1                0xFFD2
#define XK_F22               0xFFD3
#define XK_R2                0xFFD3
#define XK_F23               0xFFD4
#define XK_R3                0xFFD4
#define XK_F24               0xFFD5
#define XK_R4                0xFFD5
#define XK_F25               0xFFD6
#define XK_R5                0xFFD6
#define XK_F26               0xFFD7
#define XK_R6                0xFFD7
#define XK_F27               0xFFD8
#define XK_R7                0xFFD8
#define XK_F28               0xFFD9
#define XK_R8                0xFFD9
#define XK_F29               0xFFDA
#define XK_R9                0xFFDA
#define XK_F30               0xFFDB
#define XK_R10               0xFFDB
#define XK_F31               0xFFDC
#define XK_R11               0xFFDC
#define XK_F32               0xFFDD
#define XK_R12               0xFFDD
#define XK_F33               0xFFDE
#define XK_R13               0xFFDE
#define XK_F34               0xFFDF
#define XK_R14               0xFFDF
#define XK_F35               0xFFE0
#define XK_R15               0xFFE0

/* Modifiers */
#define XK_Shift_L           0xFFE1
#define XK_Shift_R           0xFFE2
#define XK_Control_L         0xFFE3
#define XK_Control_R         0xFFE4
#define XK_Caps_Lock        0xFFE5
#define XK_Shift_Lock       0xFFE6
#define XK_Meta_L            0xFFE7
#define XK_Meta_R            0xFFE8
#define XK_Alt_L             0xFFE9
#define XK_Alt_R             0xFFEA
#define XK_Super_L           0xFFEB
#define XK_Super_R           0xFFEC
#define XK_Hyper_L           0xFFED
#define XK_Hyper_R           0xFFEE

#endif /* XK_MISCELLANY */

/* Latin 1 Byte 3 = 0 */
#ifndef XK_LATIN1

#define XK_space             0x020
```



```

#define XK_exclam           0x021
#define XK_quotedbl        0x022
#define XK_numbersign      0x023
#define XK_dollar          0x024
#define XK_percent         0x025
#define XK_ampersand       0x026
#define XK_apostrophe      0x027
#define XK_parenleft       0x028
#define XK_parenright      0x029
#define XK_asterisk        0x02a
#define XK_plus            0x02b
#define XK_comma           0x02c
#define XK_minus           0x02d
#define XK_period          0x02e
#define XK_slash           0x02f
#define XK_0               0x030
#define XK_1               0x031
#define XK_2               0x032
#define XK_3               0x033
#define XK_4               0x034
#define XK_5               0x035
#define XK_6               0x036
#define XK_7               0x037
#define XK_8               0x038
#define XK_9               0x039
#define XK_colon           0x03a
#define XK_semicolon       0x03b
#define XK_less            0x03c
#define XK_equal           0x03d
#define XK_greater         0x03e
#define XK_question        0x03f
#define XK_at              0x040
#define XK_A               0x041
#define XK_B               0x042
#define XK_C               0x043
#define XK_D               0x044
#define XK_E               0x045
#define XK_F               0x046
#define XK_G               0x047
#define XK_H               0x048
#define XK_I               0x049
#define XK_J               0x04a
#define XK_K               0x04b
#define XK_L               0x04c
#define XK_M               0x04d
#define XK_N               0x04e
#define XK_O               0x04f
#define XK_P               0x050
#define XK_Q               0x051
#define XK_R               0x052
#define XK_S               0x053
#define XK_T               0x054
#define XK_U               0x055
#define XK_V               0x056
#define XK_W               0x057
#define XK_X               0x058
#define XK_Y               0x059
#define XK_Z               0x05a
#define XK_bracketleft     0x05b
#define XK_backslash       0x05c
#define XK_bracketright    0x05d
#define XK_asciicircum     0x05e
#define XK_underscore      0x05f
#define XK_grave           0x060
#define XK_a               0x061
#define XK_b               0x062
#define XK_c               0x063
#define XK_d               0x064
#define XK_e               0x065
#define XK_f               0x066
#define XK_g               0x067
#define XK_h               0x068
#define XK_i               0x069
#define XK_j               0x06a
#define XK_k               0x06b
#define XK_l               0x06c
#define XK_m               0x06d
#define XK_n               0x06e
#define XK_o               0x06f
#define XK_p               0x070
#define XK_q               0x071

```

```
#define XK_r                0x072
#define XK_s                0x073
#define XK_t                0x074
#define XK_u                0x075
#define XK_v                0x076
#define XK_w                0x077
#define XK_x                0x078
#define XK_y                0x079
#define XK_z                0x07a
#define XK_braceleft       0x07b
#define XK_bar              0x07c
#define XK_braceright     0x07d
#define XK_asciitilde      0x07e
#define XK_nobreakspace    0x0a0
#define XK_exclamdown     0x0a1
#define XK_cent            0x0a2
#define XK_sterling        0x0a3
#define XK_currency        0x0a4
#define XK_yen             0x0a5
#define XK_brokenbar      0x0a6
#define XK_section         0x0a7
#define XK_diaeresis       0x0a8
#define XK_copyright       0x0a9
#define XK_ordfeminine     0x0aa
#define XK_guillemotleft   0x0ab
#define XK_notsign         0x0ac
#define XK_hyphen          0x0ad
#define XK_registered      0x0ae
#define XK_macron           0x0af
#define XK_degree          0x0b0
#define XK_plusminus       0x0b1
#define XK_twosuperior     0x0b2
#define XK_threesuperior  0x0b3
#define XK_acute           0x0b4
#define XK_mu              0x0b5
#define XK_paragraph       0x0b6
#define XK_periodcentered  0x0b7
#define XK_cedilla         0x0b8
#define XK_onesuperior    0x0b9
#define XK_masculine       0x0ba
#define XK_guillemotright  0x0bb
#define XK_onequarter      0x0bc
#define XK_onehalf         0x0bd
#define XK_threequarters   0x0be
#define XK_questiondown    0x0bf
#define XK_Agrave          0x0c0
#define XK_Aacute          0x0c1
#define XK_Acircumflex     0x0c2
#define XK_Atilde          0x0c3
#define XK_Adiaeresis      0x0c4
#define XK_Aring           0x0c5
#define XK_AE              0x0c6
#define XK_Ccedilla        0x0c7
#define XK_Egrave          0x0c8
#define XK_Eacute          0x0c9
#define XK_Ecircumflex     0x0ca
#define XK_Ediaeresis      0x0cb
#define XK_Igrave          0x0cc
#define XK_Iacute          0x0cd
#define XK_Icircumflex     0x0ce
#define XK_Idiaeresis      0x0cf
#define XK_ETH              0x0d0
#define XK_Ntilde          0x0d1
#define XK_Ograve          0x0d2
#define XK_Oacute          0x0d3
#define XK_Ocircumflex     0x0d4
#define XK_Otilde          0x0d5
#define XK_Odiaeresis      0x0d6
#define XK_multiply        0x0d7
#define XK_Ooblique        0x0d8
#define XK_Ugrave          0x0d9
#define XK_Uacute          0x0da
#define XK_Ucircumflex     0x0db
#define XK_Udiaeresis      0x0dc
#define XK_Yacute          0x0dd
#define XK_THORN           0x0de
#define XK_ssharp          0x0df
#define XK_agrave          0x0e0
#define XK_aacute          0x0e1
#define XK_acircumflex     0x0e2
#define XK_atilde          0x0e3
```

```

#define XK_adiaeresis          0x0e4
#define XK_aring               0x0e5
#define XK_ae                  0x0e6
#define XK_ccedilla           0x0e7
#define XK_egrave              0x0e8
#define XK_eacute             0x0e9
#define XK_ecircumflex        0x0ea
#define XK_ediaeresis         0x0eb
#define XK_igrave             0x0ec
#define XK_iacute             0x0ed
#define XK_icircumflex        0x0ee
#define XK_idiaeresis         0x0ef
#define XK_eth                 0x0f0
#define XK_ntilde             0x0f1
#define XK_ograve             0x0f2
#define XK_oacute             0x0f3
#define XK_ocircumflex        0x0f4
#define XK_otilde             0x0f5
#define XK_odiaeresis         0x0f6
#define XK_division           0x0f7
#define XK_oslash             0x0f8
#define XK_ugrave            0x0f9
#define XK_uacute            0x0fa
#define XK_ucircumflex        0x0fb
#define XK_udiaeresis         0x0fc
#define XK_yacute            0x0fd
#define XK_thorn              0x0fe
#define XK_ydiaeresis         0x0ff
#endif /* XK_LATIN1 */

/* Latin 2 Byte 3 = 1 */
#ifndef XK_LATIN2
#define XK_Aogonek             0x1a1
#define XK_breve              0x1a2
#define XK_Lstroke            0x1a3
#define XK_Lcaron             0x1a5
#define XK_Sacute             0x1a6
#define XK_Scaron             0x1a9
#define XK_Scedilla           0x1aa
#define XK_Tcaron             0x1ab
#define XK_Zacute             0x1ac
#define XK_Zcaron             0x1ae
#define XK_Zabovedot         0x1af
#define XK_aogonek           0x1b1
#define XK_ogonek            0x1b2
#define XK_lstroke           0x1b3
#define XK_lcaron            0x1b5
#define XK_sacute            0x1b6
#define XK_caron             0x1b7
#define XK_scaron            0x1b9
#define XK_scedilla          0x1ba
#define XK_tcaron            0x1bb
#define XK_zacute            0x1bc
#define XK_doubleacute        0x1bd
#define XK_zcaron            0x1be
#define XK_zabovedot         0x1bf
#define XK_Racute            0x1c0
#define XK_Abreve            0x1c3
#define XK_Lacute            0x1c5
#define XK_Cacute            0x1c6
#define XK_Ccaron            0x1c8
#define XK_Eogonek           0x1ca
#define XK_Ecaron            0x1cc
#define XK_Dcaron            0x1cf
#define XK_Dstroke           0x1d0
#define XK_Nacute            0x1d1
#define XK_Ncaron            0x1d2
#define XK_Odoubleacute       0x1d5
#define XK_Rcaron            0x1d8
#define XK_Uring             0x1d9
#define XK_Udoubleacute       0x1db
#define XK_Tcedilla           0x1de
#define XK_racute            0x1e0
#define XK_abreve            0x1e3
#define XK_lacute            0x1e5
#define XK_cacute            0x1e6
#define XK_ccaron            0x1e8
#define XK_eogonek           0x1ea
#define XK_ecaron            0x1ec
#define XK_dcaron            0x1ef
#define XK_dstroke           0x1f0

```

```

#define XK_nacute                0x1f1
#define XK_ncaron                0x1f2
#define XK_odoubleacute          0x1f5
#define XK_udoubleacute          0x1fb
#define XK_rcaron                0x1f8
#define XK_uring                 0x1f9
#define XK_tcedilla              0x1fe
#define XK_abovedot              0x1ff
#endif /* XK_LATIN2 */

/* Latin 3 Byte 3 = 2 */
#ifdef XK_LATIN3
#define XK_Hstroke                0x2a1
#define XK_Hcircumflex           0x2a6
#define XK_Iabovedot             0x2a9
#define XK_Gbreve                0x2ab
#define XK_Jcircumflex           0x2ac
#define XK_hstroke               0x2b1
#define XK_hcircumflex           0x2b6
#define XK_idotless              0x2b9
#define XK_gbreve                0x2bb
#define XK_jcircumflex           0x2bc
#define XK_Cabovedot             0x2c5
#define XK_Ccircumflex           0x2c6
#define XK_Gabovedot             0x2d5
#define XK_Gcircumflex           0x2d8
#define XK_Ubreve                0x2dd
#define XK_Scircumflex           0x2de
#define XK_cabovedot             0x2e5
#define XK_ccircumflex           0x2e6
#define XK_gabovedot             0x2f5
#define XK_gcircumflex           0x2f8
#define XK_ubreve                0x2fd
#define XK_scircumflex           0x2fe
#endif /* XK_LATIN3 */

/* Latin 4 Byte 3 = 3 */
#ifdef XK_LATIN4
#define XK_kra                    0x3a2
#define XK_Rcedilla              0x3a3
#define XK_Itilde                0x3a5
#define XK_Lcedilla              0x3a6
#define XK_Emacron               0x3aa
#define XK_Gcedilla              0x3ab
#define XK_Tslash                0x3ac
#define XK_rcedilla              0x3b3
#define XK_ityilde               0x3b5
#define XK_lcedilla              0x3b6
#define XK_emacron               0x3ba
#define XK_gcedilla              0x3bb
#define XK_tslash                0x3bc
#define XK_ENG                   0x3bd
#define XK_eng                   0x3bf
#define XK_Amacron               0x3c0
#define XK_logonek               0x3c7
#define XK_Eabovedot             0x3cc
#define XK_Imacron               0x3cf
#define XK_Ncedilla              0x3d1
#define XK_Omacron               0x3d2
#define XK_Kcedilla              0x3d3
#define XK_Uogonek               0x3d9
#define XK_Utilde                0x3dd
#define XK_Umacron               0x3de
#define XK_amacron               0x3e0
#define XK_iogonek               0x3e7
#define XK_eabovedot             0x3ec
#define XK_imacron               0x3ef
#define XK_ncedilla              0x3f1
#define XK_omacron               0x3f2
#define XK_kcedilla              0x3f3
#define XK_uogonek               0x3f9
#define XK_utilde                0x3fd
#define XK_umacron               0x3fe
#endif /* XK_LATIN4 */

/* Katakana Byte 3 = 4 */
#ifdef XK_KATAKANA
#define XK_overline               0x47e
#define XK_kana_fullstop         0x4a1
#define XK_kana_openingbracket   0x4a2
#define XK_kana_closingbracket   0x4a3

```

```

#define XK_kana_comma          0x4a4
#define XK_kana_conjunctive    0x4a5
#define XK_kana_WO             0x4a6
#define XK_kana_a              0x4a7
#define XK_kana_i              0x4a8
#define XK_kana_u              0x4a9
#define XK_kana_e              0x4aa
#define XK_kana_o              0x4ab
#define XK_kana_ya             0x4ac
#define XK_kana_yu             0x4ad
#define XK_kana_yo             0x4ae
#define XK_kana_tsu            0x4af
#define XK_prolongedsound      0x4b0
#define XK_kana_A              0x4b1
#define XK_kana_I              0x4b2
#define XK_kana_U              0x4b3
#define XK_kana_E              0x4b4
#define XK_kana_O              0x4b5
#define XK_kana_KA             0x4b6
#define XK_kana_KI             0x4b7
#define XK_kana_KU             0x4b8
#define XK_kana_KE             0x4b9
#define XK_kana_KO             0x4ba
#define XK_kana_SA             0x4bb
#define XK_kana_SHI            0x4bc
#define XK_kana_SU             0x4bd
#define XK_kana_SE             0x4be
#define XK_kana_SO             0x4bf
#define XK_kana_TA             0x4c0
#define XK_kana_CHI            0x4c1
#define XK_kana_TSU            0x4c2
#define XK_kana_TE             0x4c3
#define XK_kana_TO             0x4c4
#define XK_kana_NA             0x4c5
#define XK_kana_NI             0x4c6
#define XK_kana_NU             0x4c7
#define XK_kana_NE             0x4c8
#define XK_kana_NO             0x4c9
#define XK_kana_HA             0x4ca
#define XK_kana_HI             0x4cb
#define XK_kana_FU             0x4cc
#define XK_kana_HE             0x4cd
#define XK_kana_HO             0x4ce
#define XK_kana_MA             0x4cf
#define XK_kana_MI             0x4d0
#define XK_kana_MU             0x4d1
#define XK_kana_ME             0x4d2
#define XK_kana_MO             0x4d3
#define XK_kana_YA             0x4d4
#define XK_kana_YU             0x4d5
#define XK_kana_YO             0x4d6
#define XK_kana_RA             0x4d7
#define XK_kana_RI             0x4d8
#define XK_kana_RU             0x4d9
#define XK_kana_RE             0x4da
#define XK_kana_RO             0x4db
#define XK_kana_WA             0x4dc
#define XK_kana_N              0x4dd
#define XK_voicedsound         0x4de
#define XK_semivoicedsound     0x4df
#define XK_kana_switch         0xFF7E
#endif /* XK_KATAKANA */

/* Arabic Byte 3 = 5 */
#ifndef XK_ARABIC
#define XK_Arabic_comma          0x5ac
#define XK_Arabic_semicolon     0x5bb
#define XK_Arabic_question_mark 0x5bf
#define XK_Arabic_hamza         0x5c1
#define XK_Arabic_maddaonalef   0x5c2
#define XK_Arabic_hamzaonalef   0x5c3
#define XK_Arabic_hamzaonwaw    0x5c4
#define XK_Arabic_hamzaunderalef 0x5c5
#define XK_Arabic_hamzaonyeh    0x5c6
#define XK_Arabic_alef           0x5c7
#define XK_Arabic_beh            0x5c8
#define XK_Arabic_tehmarbuta    0x5c9
#define XK_Arabic_teh           0x5ca
#define XK_Arabic_theh           0x5cb
#define XK_Arabic_jeem          0x5cc
#define XK_Arabic_hah           0x5cd

```

```

#define XK_Arabic_khah          0x5ce
#define XK_Arabic_dal          0x5cf
#define XK_Arabic_thal        0x5d0
#define XK_Arabic_ra          0x5d1
#define XK_Arabic_zain        0x5d2
#define XK_Arabic_seen        0x5d3
#define XK_Arabic_sheen       0x5d4
#define XK_Arabic_sad         0x5d5
#define XK_Arabic_dad         0x5d6
#define XK_Arabic_tah         0x5d7
#define XK_Arabic_zah         0x5d8
#define XK_Arabic_ain         0x5d9
#define XK_Arabic_ghain       0x5da
#define XK_Arabic_tatweel     0x5e0
#define XK_Arabic_feh         0x5e1
#define XK_Arabic_qaf         0x5e2
#define XK_Arabic_kaf         0x5e3
#define XK_Arabic_lam         0x5e4
#define XK_Arabic_meem        0x5e5
#define XK_Arabic_noon        0x5e6
#define XK_Arabic_ha          0x5e7
#define XK_Arabic_waw         0x5e8
#define XK_Arabic_alefmakura  0x5e9
#define XK_Arabic_yeh         0x5ea
#define XK_Arabic_fathatan    0x5eb
#define XK_Arabic_dammatan    0x5ec
#define XK_Arabic_kasratan    0x5ed
#define XK_Arabic_fatha       0x5ee
#define XK_Arabic_damma       0x5ef
#define XK_Arabic_kasra       0x5f0
#define XK_Arabic_shadda      0x5f1
#define XK_Arabic_sukun       0x5f2
#define XK_Arabic_switch      0xFF7E
#endif /* XK_ARABIC */

/* Cyrillic Byte 3 = 6 */
#ifndef XK_CYRILLIC
#define XK_Serbian_dje         0x6a1
#define XK_Macedonia_gje      0x6a2
#define XK_Cyrillic_io        0x6a3
#define XK_Ukrainian_je       0x6a4
#define XK_Macedonia_dse      0x6a5
#define XK_Ukrainian_i        0x6a6
#define XK_Ukrainian_yi       0x6a7
#define XK_Cyrillic_je        0x6a8
#define XK_Cyrillic_lje       0x6a9
#define XK_Cyrillic_nje       0x6aa
#define XK_Serbian_tshe       0x6ab
#define XK_Macedonia_kje      0x6ac
#define XK_Byelorussian_shortu 0x6ae
#define XK_Cyrillic_dzhe      0x6af
#define XK_numerosign         0x6b0
#define XK_Serbian_DJE        0x6b1
#define XK_Macedonia_GJE      0x6b2
#define XK_Cyrillic_IO        0x6b3
#define XK_Ukrainian_IE       0x6b4
#define XK_Macedonia_DSE      0x6b5
#define XK_Ukrainian_I        0x6b6
#define XK_Ukrainian_YI       0x6b7
#define XK_Cyrillic_JE        0x6b8
#define XK_Cyrillic_LJE       0x6b9
#define XK_Cyrillic_NJE       0x6ba
#define XK_Serbian_TSHE       0x6bb
#define XK_Macedonia_KJE      0x6bc
#define XK_Byelorussian_SHORTU 0x6be
#define XK_Cyrillic_DZHE      0x6bf
#define XK_Cyrillic_yu        0x6c0
#define XK_Cyrillic_a         0x6c1
#define XK_Cyrillic_be        0x6c2
#define XK_Cyrillic_tse       0x6c3
#define XK_Cyrillic_de        0x6c4
#define XK_Cyrillic_ie        0x6c5
#define XK_Cyrillic_ef        0x6c6
#define XK_Cyrillic_ghe       0x6c7
#define XK_Cyrillic_ha        0x6c8
#define XK_Cyrillic_i         0x6c9
#define XK_Cyrillic_shorti    0x6ca
#define XK_Cyrillic_ka        0x6cb
#define XK_Cyrillic_el        0x6cc
#define XK_Cyrillic_em        0x6cd
#define XK_Cyrillic_en        0x6ce

```

```

#define XK_Cyrillic_o          0x6cf
#define XK_Cyrillic_pe        0x6d0
#define XK_Cyrillic_ya        0x6d1
#define XK_Cyrillic_er        0x6d2
#define XK_Cyrillic_es        0x6d3
#define XK_Cyrillic_te        0x6d4
#define XK_Cyrillic_u         0x6d5
#define XK_Cyrillic_zhe       0x6d6
#define XK_Cyrillic_ve        0x6d7
#define XK_Cyrillic_softsign  0x6d8
#define XK_Cyrillic_yeru      0x6d9
#define XK_Cyrillic_ze        0x6da
#define XK_Cyrillic_sha       0x6db
#define XK_Cyrillic_e         0x6dc
#define XK_Cyrillic_shcha     0x6dd
#define XK_Cyrillic_che       0x6de
#define XK_Cyrillic_hardsign  0x6df
#define XK_Cyrillic_YU        0x6e0
#define XK_Cyrillic_A         0x6e1
#define XK_Cyrillic_BE        0x6e2
#define XK_Cyrillic_TSE       0x6e3
#define XK_Cyrillic_DE        0x6e4
#define XK_Cyrillic_IE        0x6e5
#define XK_Cyrillic_EF        0x6e6
#define XK_Cyrillic_GHE       0x6e7
#define XK_Cyrillic_HA        0x6e8
#define XK_Cyrillic_I         0x6e9
#define XK_Cyrillic_SHORTI    0x6ea
#define XK_Cyrillic_KA        0x6eb
#define XK_Cyrillic_EL        0x6ec
#define XK_Cyrillic_EM        0x6ed
#define XK_Cyrillic_EN        0x6ee
#define XK_Cyrillic_O         0x6ef
#define XK_Cyrillic_PE        0x6f0
#define XK_Cyrillic_YA        0x6f1
#define XK_Cyrillic_ER        0x6f2
#define XK_Cyrillic_ES        0x6f3
#define XK_Cyrillic_TE        0x6f4
#define XK_Cyrillic_U         0x6f5
#define XK_Cyrillic_ZHE       0x6f6
#define XK_Cyrillic_VE        0x6f7
#define XK_Cyrillic_SOFTSIGN  0x6f8
#define XK_Cyrillic_YERU     0x6f9
#define XK_Cyrillic_ZE        0x6fa
#define XK_Cyrillic_SHA       0x6fb
#define XK_Cyrillic_E         0x6fc
#define XK_Cyrillic_SHCHA     0x6fd
#define XK_Cyrillic_CHE       0x6fe
#define XK_Cyrillic_HARDSIGN  0x6ff
#endif /* XK_CYRILLIC */

/* Greek* Byte 3 = 7 */
#ifndef XK_GREEK
#define XK_Greek_ALPHAaccent  0x7a1
#define XK_Greek_EPSILONaccent 0x7a2
#define XK_Greek_ETAaccent    0x7a3
#define XK_Greek_IOTAaccent   0x7a4
#define XK_Greek_IOTAdieresis 0x7a5
#define XK_Greek_OMICRONaccent 0x7a7
#define XK_Greek_UPSILONaccent 0x7a8
#define XK_Greek_UPSILONdieresis 0x7a9
#define XK_Greek_OMEGAaccent  0x7ab
#define XK_Greek_accentdieresis 0x7ae
#define XK_Greek_horizbar     0x7af
#define XK_Greek_alphaaccent  0x7b1
#define XK_Greek_epsilonaccent 0x7b2
#define XK_Greek_etaaccent    0x7b3
#define XK_Greek_iotaaccent   0x7b4
#define XK_Greek_iotadieresis 0x7b5
#define XK_Greek_iotaaccentdieresis 0x7b6
#define XK_Greek_omicronaccent 0x7b7
#define XK_Greek_upsilonaccent 0x7b8
#define XK_Greek_upsilondieresis 0x7b9
#define XK_Greek_upsilonaccentdieresis 0x7ba
#define XK_Greek_omegaaccent  0x7bb
#define XK_Greek_ALPHA        0x7c1
#define XK_Greek_BETA         0x7c2
#define XK_Greek_GAMMA        0x7c3
#define XK_Greek_DELTA        0x7c4
#define XK_Greek_EPSILON      0x7c5
#define XK_Greek_ZETA         0x7c6

```

```

#define XK_Greek_ETA           0x7c7
#define XK_Greek_THETA        0x7c8
#define XK_Greek_IOTA         0x7c9
#define XK_Greek_KAPPA        0x7ca
#define XK_Greek_LAMDA        0x7cb
#define XK_Greek_LAMBDA       0x7cb
#define XK_Greek_MU           0x7cc
#define XK_Greek_NU           0x7cd
#define XK_Greek_XI           0x7ce
#define XK_Greek_OMICRON      0x7cf
#define XK_Greek_PI           0x7d0
#define XK_Greek_RHO          0x7d1
#define XK_Greek_SIGMA        0x7d2
#define XK_Greek_TAU          0x7d4
#define XK_Greek_UPSILON      0x7d5
#define XK_Greek_PHI          0x7d6
#define XK_Greek_CHI          0x7d7
#define XK_Greek_PSI          0x7d8
#define XK_Greek_OMEGA        0x7d9
#define XK_Greek_alpha        0x7e1
#define XK_Greek_beta         0x7e2
#define XK_Greek_gamma        0x7e3
#define XK_Greek_delta        0x7e4
#define XK_Greek_epsilon      0x7e5
#define XK_Greek_zeta         0x7e6
#define XK_Greek_eta          0x7e7
#define XK_Greek_theta        0x7e8
#define XK_Greek_iota         0x7e9
#define XK_Greek_kappa        0x7ea
#define XK_Greek_lamda        0x7eb
#define XK_Greek_lambda       0x7eb
#define XK_Greek_mu           0x7ec
#define XK_Greek_nu           0x7ed
#define XK_Greek_xi           0x7ee
#define XK_Greek_omicron      0x7ef
#define XK_Greek_pi           0x7f0
#define XK_Greek_rho          0x7f1
#define XK_Greek_sigma        0x7f2
#define XK_Greek_finalsmallsigma 0x7f3
#define XK_Greek_tau          0x7f4
#define XK_Greek_upsilon      0x7f5
#define XK_Greek_phi          0x7f6
#define XK_Greek_chi          0x7f7
#define XK_Greek_psi          0x7f8
#define XK_Greek_omega        0x7f9
#define XK_Greek_switch       0xFF7E
#endif /* XK_GREEK */

/*TechnicalByte 3 = 8 */
#ifndef XK_TECHNICAL
#define XK_leftradical         0x8a1
#define XK_topleftradical     0x8a2
#define XK_horizconnector     0x8a3
#define XK_topintegral         0x8a4
#define XK_botintegral        0x8a5
#define XK_vertconnector      0x8a6
#define XK_topleftsqbracket   0x8a7
#define XK_botleftsqbracket   0x8a8
#define XK_toprightsqbracket  0x8a9
#define XK_botrightsqbracket  0x8aa
#define XK_topleftparens      0x8ab
#define XK_botleftparens     0x8ac
#define XK_toprightparens    0x8ad
#define XK_botrightparens    0x8ae
#define XK_leftmiddlecurlybrace 0x8af
#define XK_rightmiddlecurlybrace 0x8b0
#define XK_topleftsummation   0x8b1
#define XK_botleftsummation   0x8b2
#define XK_topvertsummationconnector 0x8b3
#define XK_botvertsummationconnector 0x8b4
#define XK_toprightsummation  0x8b5
#define XK_botrightsummation  0x8b6
#define XK_rightmiddlesummation 0x8b7
#define XK_lessthanequal      0x8bc
#define XK_notequal           0x8bd
#define XK_greaterequal       0x8be
#define XK_integral           0x8bf
#define XK_therefore          0x8c0
#define XK_variation          0x8c1
#define XK_infinity           0x8c2
#define XK_nabla              0x8c5

```



```
#define XK_approximate      0x8c8
#define XK_similarequal    0x8c9
#define XK_ifonlyif       0x8cd
#define XK_implies        0x8ce
#define XK_identical      0x8cf
#define XK_radical        0x8d6
#define XK_includedin     0x8da
#define XK_includes       0x8db
#define XK_intersection    0x8dc
#define XK_union          0x8dd
#define XK_logicaland     0x8de
#define XK_logicalor      0x8df
#define XK_partialderivative 0x8ef
#define XK_function       0x8f6
#define XK_leftarrow      0x8fb
#define XK_uparrow        0x8fc
#define XK_rightarrow     0x8fd
#define XK_downarrow      0x8fe
#endif /* XK_TECHNICAL */
```

```
/* Special Byte 3 = 9 */
#ifdef XK_SPECIAL
#define XK_blank           0x9df
#define XK_soliddiamond   0x9e0
#define XK_checkerboard   0x9e1
#define XK_ht             0x9e2
#define XK_ff             0x9e3
#define XK_cr             0x9e4
#define XK_lf             0x9e5
#define XK_nl            0x9e8
#define XK_vt            0x9e9
#define XK_lowrightcorner 0x9ea
#define XK_uprightcorner  0x9eb
#define XK_upleftcorner   0x9ec
#define XK_lowleftcorner  0x9ed
#define XK_crossinglines  0x9ee
#define XK_horizlinescan1 0x9ef
#define XK_horizlinescan3 0x9f0
#define XK_horizlinescan5 0x9f1
#define XK_horizlinescan7 0x9f2
#define XK_horizlinescan9 0x9f3
#define XK_left           0x9f4
#define XK_right          0x9f5
#define XK_bott           0x9f6
#define XK_topt           0x9f7
#define XK_vertbar        0x9f8
#endif /* XK_SPECIAL */
```

```
/* Publishing Byte 3 = a */
#ifdef XK_PUBLISHING
#define XK_emspace        0xaa1
#define XK_enspace        0xaa2
#define XK_em3space       0xaa3
#define XK_em4space       0xaa4
#define XK_digitspace     0xaa5
#define XK_punctspace     0xaa6
#define XK_thinspace      0xaa7
#define XK_hairspace      0xaa8
#define XK_emdash         0xaa9
#define XK_endash         0xaaa
#define XK_signifblank    0xaac
#define XK_ellipsis       0xaae
#define XK_doubbaselinedot 0xaaf
#define XK_onethird       0xab0
#define XK_twothirds     0xab1
#define XK_onefifth       0xab2
#define XK_twofifths     0xab3
#define XK_threefifths   0xab4
#define XK_fourfifths    0xab5
#define XK_onesixth      0xab6
#define XK_fivesixths    0xab7
#define XK_careof         0xab8
#define XK_figdash        0xabb
#define XK_leftanglebracket 0xabc
#define XK_decimalpoint   0xabd
#define XK_rightanglebracket 0abe
#define XK_marker         0abf
#define XK_oneeighth     0xac3
#define XK_threeeighths   0xac4
#define XK_fiveeighths    0xac5
#define XK_seveneighths   0xac6
```

```

#define XK_trademark                0xac9
#define XK_signaturemark            0xaca
#define XK_trademarkincircle        0xacb
#define XK_leftpentriangle         0xacc
#define XK_rightpentriangle        0xacd
#define XK_emopencircle            0xace
#define XK_emopenrectangle         0xacf
#define XK_leftsinglequotemark     0xad0
#define XK_rightsinglequotemark    0xad1
#define XK_leftdoublequotemark    0xad2
#define XK_rightdoublequotemark   0xad3
#define XK_prescription             0xad4
#define XK_minutes                 0xad6
#define XK_seconds                 0xad7
#define XK_latincross              0xad9
#define XK_hexagram                0xada
#define XK_filledrectbullet        0xadb
#define XK_filledlefttribullet     0xadc
#define XK_filledrighttribullet    0xadd
#define XK_emfilledcircle         0xade
#define XK_emfilledrect           0xadf
#define XK_enopencirclebullet     0xae0
#define XK_enopensquarebullet     0xae1
#define XK_openrectbullet         0xae2
#define XK_opentribulletup        0xae3
#define XK_opentribulletdown      0xae4
#define XK_openstar                0xae5
#define XK_enfilledcircbullet     0xae6
#define XK_enfilledsqbullet       0xae7
#define XK_filledtribulletup      0xae8
#define XK_filledtribulletdown    0xae9
#define XK_leftpointer            0xaea
#define XK_rightpointer           0xaeb
#define XK_club                   0xaec
#define XK_diamond                0xaed
#define XK_heart                  0xaee
#define XK_maltesecross           0xaf0
#define XK_dagger                 0xaf1
#define XK_doubledagger           0xaf2
#define XK_checkmark              0xaf3
#define XK_ballotcross            0xaf4
#define XK_musicalsharp           0xaf5
#define XK_musicalflat           0xaf6
#define XK_malesymbol             0xaf7
#define XK_femalesymbol           0xaf8
#define XK_telephone              0xaf9
#define XK_telephonerecorder      0xafa
#define XK_phonographcopyright    0xafb
#define XK_caret                  0xafc
#define XK_singlelowquotemark     0xafd
#define XK_doublelowquotemark     0xafe
#define XK_cursor                 0xaff
#endif /* XK_PUBLISHING */

/* APL Byte 3 = b */
#ifdef XK_APL
#define XK_leftcaret              0xba3
#define XK_rightcaret            0xba6
#define XK_downcaret             0xba8
#define XK_upcaret               0xba9
#define XK_overbar               0xbc0
#define XK_downtack              0xbc2
#define XK_upshoe                0xbc3
#define XK_downstile             0xbc4
#define XK_underbar              0xbc6
#define XK_jot                   0xbca
#define XK_quad                  0xbcc
#define XK_uptack                 0xbce
#define XK_circle                0xbcf
#define XK_upstile                0xbd3
#define XK_downshoe              0xbd6
#define XK_rightshoe             0xbd8
#define XK_leftshoe              0xbda
#define XK_lefttack               0xbdc
#define XK_righttack             0xbfc
#endif /* XK_APL */

/* Hebrew Byte 3 = c */
#ifdef XK_HEBREW
#define XK_hebrew_doublelowline   0xcdf
#define XK_hebrew_aleph           0xce0

```

```
#define XK_hebrew_bet          0xce1
#define XK_hebrew_gimel       0xce2
#define XK_hebrew_dalet       0xce3
#define XK_hebrew_he          0xce4
#define XK_hebrew_waw         0xce5
#define XK_hebrew_zain        0xce6
#define XK_hebrew_chet        0xce7
#define XK_hebrew_tet         0xce8
#define XK_hebrew_yod         0xce9
#define XK_hebrew_finalkaph    0xcea
#define XK_hebrew_kaph        0xceb
#define XK_hebrew_lamed       0xcec
#define XK_hebrew_finalmem     0xced
#define XK_hebrew_mem         0xcee
#define XK_hebrew_finalnun     0xcef
#define XK_hebrew_nun         0xcf0
#define XK_hebrew_samech      0xcf1
#define XK_hebrew_ayin        0xcf2
#define XK_hebrew_finalpe     0xcf3
#define XK_hebrew_pe          0xcf4
#define XK_hebrew_finalzade    0xcf5
#define XK_hebrew_zade        0xcf6
#define XK_hebrew_qoph        0xcf7
#define XK_hebrew_resh        0xcf8
#define XK_hebrew_shin        0xcf9
#define XK_hebrew_taw         0xcfa
#define XK_Hebrew_switch      0xFF7E
```

The X Extension Library

Overview

This chapter identifies binary interfaces for libXext, which are defined in the document entitled “X11 Nonrectangular Window Shape Extension” by Keith Packard (copyright X Consortium).

The Extension Library Interfaces

The interfaces listed below in Table 10-4 have been included in SCD2.4 because they are REQUIRED to be present on all compliant systems, in the dynamic library */usr/lib/libXext.so.0*. Note that for this release of the SCD the interfaces exported by this library are restricted only to those concerned with the X11 Nonrectangular Window Shape Extension.

Table 10-4: Contents of libXext

XShapeCombineMask
 XShapeCombineRectangles
 XShapeCombineRegion
 XShapeCombineShape
 XShapeGetRectangles
 XShapeInputSelected
 XShapeOffsetShape
 XShapeQueryExtension
 XShapeQueryExtents
 XShapeQueryVersion
 XShapeSelectInput

Figure 10-2 details the manifest constants associated with the Extension library.

Figure 10-2: Manifest Constants from <X11/extensions/shape.h>

```
#define ShapeSet           0
#define ShapeUnion        1
#define ShapeIntersect    2
#define ShapeSubtract     3
#define ShapeInvert       4
#define ShapeBounding     0
#define ShapeClip         1
#define ShapeNotifyMask   (1L << 0)
#define ShapeNotify       0
```

The X Toolkit

Overview

This chapter identifies binary interfaces for `libXt`, which are defined in the document *X Toolkit Intrinsics - C Language Interface* by Joel McCormack, Paul Asente, and Ralph R. Swick which is distributed by the X Consortium with X Version 11, Release 5.

In addition, all SCD 2.4 systems will support the X 11 Release 5 Protocol, as defined in *The X Window System (Third Edition)* by Robert W. Scheifler and James Gettys (Digital Press, ISBN 1-55558-088-2).

Finally, all SCD 2.4 systems will support the mechanisms and conventions as specified in the *Inter-Client Communications Convention Manual (ICCCM)* in *The X Window System (Third Edition)* by Robert W. Scheifler and James Gettys (Digital Press, ISBN 1-55558-088-2).

The libXt Interfaces

The interfaces listed below in Table 10-5 and Table 10-6 have been included in SCD 2.4 because they are REQUIRED to be present on all compliant systems, in the dynamic libraries: `/usr/lib/libXt.so.4` and `/usr/lib/libXt.so.5`.

Table 10-7 contains a list of unsafe macros. These macros should be avoided by application programmers which are trying to build portable SPARC applications.

Table 10-8 contains the exported data which are also REQUIRED to be present in `libXt.so.4` and `libXt.so.5`. The format of these entries is: `data[size]`. Data without a size are opaque.

Since the X Version 11, Release 5 specification is a proper superset of the X Version 11, Release 4 specification, application developers can link to either `/usr/lib/libXt.so.4` or `/usr/lib/libXt.so.5` to access X Version 11, Release 4 and Release 5 interfaces. See the SPARC Compliance Definition 2.0, 2.1, 2.2, 2.3 or 2.4 for a list of X11R4 components.

Figures 10-10 through 10-18 detail the manifest constants and visible data structures associated with the X Toolkit library.

Figures 10-19 through 10-26 detail the manifest constants and visible data structures needed by widget programmers. Only widget programmers should use the information in these figures.

Table 10-5. Contents of libXt, Part 1 of 2

XtAddCallback	XtCvtIntToUnsignedChar	XtIsSensitive
XtAddCallbacks	XtCvtStringToAcceleratorTable	XtIsShell
XtAddEventHandler	XtCvtStringToAtom	XtIsSubclass
XtAddExposureToRegion	XtCvtStringToBool	XtIsTopLevelShell
XtAddGrab	XtCvtStringToBoolean	XtIsTransientShell
XtAddRawEventHandler	XtCvtStringToCursor	XtIsVendorShell
XtAllocateGC	XtCvtStringToDimension	XtIsWidget
XtAppAddActionHook	XtCvtStringToDisplay	XtIsWMShell
XtAppAddActions	XtCvtStringToFile	XtKeysymToKeycodeList
XtAppAddInput	XtCvtStringToFloat	XtLastTimestampProcessed
XtAppAddTimeout	XtCvtStringToFont	XtMakeGeometryRequest
XtAppAddWorkProc	XtCvtStringToFontSet	XtMakeResizeRequest
XtAppCreateShell	XtCvtStringToFontStruct	XtMalloc
XtAppError	XtCvtStringToInitialState	XtManageChild
XtAppErrorMsg	XtCvtStringToInt	XtManageChildren
XtAppGetErrorDatabase	XtCvtStringToPixel	XtMapWidget
XtAppGetErrorDatabaseText	XtCvtStringToShort	XtMenuPopupAction
XtAppGetSelectionTimeout	XtCvtStringToTranslationTable	XtMergeArgLists
XtAppInitialize	XtCvtStringToUnsignedChar	XtMoveWidget
XtAppMainLoop	XtCvtStringToVisual	XtName
XtAppNextEvent	XtDatabase	XtNameToWidget
XtAppPeekEvent	XtDestroyApplicationContext	XtNewString
XtAppPending	XtDestroyWidget	XtOpenDisplay
XtAppProcessEvent	XtDisownSelection	XtOverrideTranslations
XtAppReleaseCacheRefs	XtDispatchEvent	XtOwnSelection
XtAppSetErrorHandler	XtDisplay	XtOwnSelectionIncremental
XtAppSetErrorMsgHandler	XtDisplayInitialize	XtParent
XtAppSetFallbackResources	XtDisplayOfObject	XtParseAcceleratorTable
XtAppSetSelectionTimeout	XtDisplayStringConversionWarning	XtParseTranslationTable
XtAppSetTypeConverter	XtDisplayToApplicationContext	XtPopdown
XtAppSetWarningHandler	XtFindFile	XtPopup
XtAppSetWarningMsgHandler	XtFree	XtPopupSpringLoaded
XtAppWarning	XtGetActionKeysym	XtQueryGeometry
XtAppWarningMsg	XtGetActionList	XtRealizeWidget
XtAugmentTranslations	XtGetApplicationNameAndClass	XtRealloc
XtBuildEventMask	XtGetApplicationResources	XtRegisterCaseConverter
XtCallAcceptFocus	XtGetConstraintResourceList	XtRegisterGrabAction
XtCallActionProc	XtGetGC	XtReleaseGC
XtCallbackExclusive	XtGetKeysymTable	XtRemoveActionHook
XtCallbackNone	XtGetMultiClickTime	XtRemoveAllCallbacks
XtCallbackNonexclusive	XtGetResourceList	XtRemoveCallback
XtCallbackPopdown	XtGetSelectionRequest	XtRemoveCallbacks
XtCallbackReleaseCacheRef	XtGetSelectionValue	XtRemoveEventHandler
XtCallbackReleaseCacheRefList	XtGetSelectionValueIncremental	XtRemoveGrab
XtCallCallbackList	XtGetSelectionValues	XtRemoveInput
XtCallCallbacks	XtGetSelectionValuesIncremental	XtRemoveRawEventHandler
XtCallConverter	XtGetSubresources	XtRemoveTimeout
XtCallloc	XtGetSubvalues	XtRemoveWorkProc
XtClass	XtGetValues	XtResizeWidget
XtCloseDisplay	XtGrabButton	XtResizeWindow
XtConfigureWidget	XtGrabKey	XtResolvePathname
XtConvertAndStore	XtGrabKeyboard	XtScreen
XtConvertCase	XtGrabPointer	XtScreenDatabase
XtCreateApplicationContext	XtHasCallbacks	XtScreenOfObject
XtCreateManagedWidget	XtInitializeWidgetClass	XtSetKeyboardFocus
XtCreatePopupShell	XtInsertEventHandler	XtSetKeyTranslator
XtCreateWidget	XtInsertRawEventHandler	XtSetLanguageProc
XtCreateWindow	XtInstallAccelerators	XtSetMappedWhenManaged
XtCvtColorToPixel	XtInstallAllAccelerators	XtSetMultiClickTime
XtCvtIntToBool	XtIsApplicationShell	XtSetSensitive
XtCvtIntToBoolean	XtIsComposite	XtSetSubvalues
XtCvtIntToColor	XtIsConstraint	XtSetTypeConverter
XtCvtIntToFloat	XtIsManaged	XtSetValues
XtCvtIntToFont	XtIsObject	XtSetWMColormapWindows
XtCvtIntToPixel	XtIsOverrideShell	XtSuperclass
XtCvtIntToPixmap	XtIsRealized	XtToolkitInitialize
XtCvtIntToShort	XtIsRectObj	XtTranslateCoords

Table 10-5. Contents of `libXt`, Part 2 of 2

XtTranslateKey	XtUnrealizeWidget	XtVaGetValues
XtTranslateKeycode	XtVaAppCreateShell	XtVaSetSubvalues
XtUngrabButton	XtVaAppInitialize	XtVaSetValues
XtUngrabKey	XtVaCreateArgsList	XtWidgetToApplicationContext
XtUngrabKeyboard	XtVaCreateManagedWidget	XtWindow
XtUngrabPointer	XtVaCreatePopupShell	XtWindowOfObject
XtUninstallTranslations	XtVaCreateWidget	XtWindowToWidget
XtUnmanageChild	XtVaGetApplicationResources	
XtUnmanageChildren	XtVaGetSubresources	_XTInherit
XtUnmapWidget	XtVaGetSubvalues	

Deprecated X Toolkit Functions

Table10-6 is a list of Xt functions which are now DEPRECATED but continue to be supported for the sake of old applications. The X Consortium defines these functions as obsolete. When MIT stops shipping these functions as part of the X11 sample implementation these functions may be removed from the SCD. Application developers are discouraged from using these functions in new applications. The effective date of DEPRECATION is November 1st, 1993. These function interfaces may be removed from the SCD as early as November 1st, 1996

Table10-6: Deprecated libXt Functions

Obsolete Function	Superseded By
XtAddActions	XtAppAddActions
XtAddConverter	XtSetTypeConverter
XtAddInput	XtAppAddInput
XtAddTimeOut	XtAppAddTimeOut
XtAddWorkProc	XtAppAddWorkProc2
XtAppAddConverter	XtAppSetTypeConverter
XtConvert	XtConvertAnStore
XtCreateApplicationShell	XtAppCreateShell
XtDestroyGC	XtReleaseGC
XtDirectConvert	XtCallConverter
XtError	XtAppError
XtErrorMsg	XtAppErrorMsg
XtGetErrorDatabase	XtAppGetErrorDatabase
XtGetErrorDatabaseText	XtAppGetErrorDatabaseText
XtGetSelectionTimeout	XtAppGetSelectionTimeout
XtInitialize	XtAppInitialize
XtMainLoop	XtAppMainLoop
XtNextEvent	XtAppNextEvent
XtPeekEvent	XtAppPeekEvent
XtPending	XtAppPending
XtProcessEvent	XtAppProcessEvent
XtSetErrorHandler	XtAppSetErrorHandler
XtSetErrorMsgHandler	XtAppSetErrorMsgHandler
XtSetSelectionTimeout	XtAppSetSelectionTimeout
XtSetWarningHandler	XtAppSetWarningHandler
XtSetWarningMsgHandler	XtAppSetWarningMsgHandler
XtStringConversionWarning	XtDisplayStringConversionWarning
XtWarning	XtAppWarning
XtWarningMsg	XtAppWarningMsg

Unsafe Macros

Ordinarily, this document only specifies the system resources available for use by applications on all SPARC compliant systems and makes no comment regarding the programming language or API used by application programmers for building applications. But SPARC International recognizes that many SPARC applications will be written in the C programming language and are likely to use the API specified by the X Consortium. Some of the macros defined by the X Consortium for the X Toolkit access symbols which are not defined to be part of the ABI.

All of these macros are defined by the `<X11/Intrinsic.h>` header file. Fortunately, each of these macros have ABI compliant functions which can be used in their place. Table 10-7 has a list of these macros. The ABI compliant functions have the same name as the the unsafe macros. This means that C programmers that wish to use the functions, rather than the macros, must `"#undef"` the macros in their source code after the point where the source code includes `<X11/Intrinsic.h>`.

Table 10-7. Unsafe Macros

XtIsApplicationShell
XtIsComposite
XtIsConstraint
XtIsOverrideShell
XtIsRectObj
XtIsShell
XtIsTopLevelShell
XtIsTransientShell
XtIsVendorShell
XtIsWidget
XtIsWMShell

Table 10-8. Exported Data for libXt.

applicationShellClassRec[0x9c]
applicationShellWidgetClass[0x4]
colorConvertArgs[0x18]
compositeClassRec[0x88]
compositeWidgetClass[0x4]
constraintClassRec[0xa4]
constraintWidgetClass[0x4]
coreWidgetClass[0x4]
objectClass[0x4]
objectClassRec[0x74]
overrideShellClassRec[0x90]
overrideShellWidgetClass[0x4]
rectObjClass[0x4]
rectObjClassRec[0x74]
screenConvertArg[0xc]
shellClassRec[0x8c]
shellWidgetClass[0x4]
topLevelShellClassRec[0x98]
topLevelShellWidgetClass[0x4]
transientShellClassRec[0x98]
transientShellWidgetClass[0x4]
vendorShellClassRec[0x94]
vendorShellWidgetClass[0x4]
widgetClass[0x4]
widgetClassRec[0x74]
wmShellClassRec[0x90]
wmShellWidgetClass[0x4]
XtCxtToolkitError[0x4]
_XtInheritTranslations¹
XtShellStrings²
XtStrings³

-
1. `_XtInheritTranslations` is only for use by widget programmers.
 2. `XtShellStrings` and `XtStrings` are reserved for use by the X Toolkit Library but applications which are intended to be portable at the binary level must refrain from accessing these global symbols. The definition of these global data may change in incompatible ways in future releases of X.
 3. See note number 3.

Figure 10-3: Manifest Constants and Data Types from <X11/Composite.h>

```
typedef struct _CompositeClassRec *CompositeWidgetClass;
typedef Cardinal (*XtOrderProc)();
```

Figure 10-4: Manifest Constants and Data Types from <X11/Constraint.h>

```
typedef struct _ConstraintClassRec *ConstraintWidgetClass;
```

Figure 10-5: Manifest Constants and Data Types from <X11/Core.h>

```
typedef struct _WidgetClassRec *CoreWidgetClass;
typedef struct _WidgetRec *CoreWidget
```

Figure 10-6: Manifest Constants and Data Types from <X11/Intrinsic.h>

```
#define XtSpecificationRelease      5
typedef char                       *String;
typedef struct _WidgetRec          *Widget;
typedef Widget                    *WidgetList;
typedef struct _WidgetClassRec     *WidgetClass;
typedef struct _CompositeRec       *CompositeWidget;
typedef struct _XtActionsRec       *XtActionList;
typedef struct _XtEventRec         *XtEventTable;
typedef struct _XtBoundAccActionRec *XtBoundAccActions;
typedef struct _XtAppStruct        *XtAppContext;
typedef unsigned long              XtValueMask;
typedef unsigned long              XtIntervalId;
typedef unsigned long              XtInputId;
typedef unsigned long              XtWorkProcId;
typedef unsigned int               XtGeometryMask;
typedef unsigned long              XtGCMask; /* Mask of values that are used by widget*/
typedef unsigned long              Pixel; /* Index into colormap */
typedef int                        XtCacheType;
#define XtCacheNone                0x001
#define XtCacheAll                  0x002
#define XtCacheByDisplay            0x003
#define XtCacheRefCount              0x100
typedef char                       Boolean;
typedef long                       XtArgVal;
typedef unsigned char              XtEnum;
typedef unsigned int               Cardinal;
typedef unsigned short             Dimension; /* Size in pixels */
typedef short                      Position; /* Offset from 0 coordinate */
typedef char*                      XtPointer;
typedef XtPointer                  Opaque;
typedef struct _TranslationData     *XtTranslations;
typedef struct _TranslationData     *XtAccelerators;
typedef unsigned int               Modifiers;
typedef void                       *XtActionProc();
typedef XtActionProc* XtBoundActions;
typedef struct _XtActionsRec {String string;XtActionProc proc; XtActionsRec;
typedef enum {XtAddress,XtBaseOffset,XtImmediate,
              XtResourceString,XtResourceQuark, XtWidgetBaseOffset,XtProcedureArg} XtAddressMode;
typedef struct {XtAddressMode address_mode;XtPointer address_id;Cardinal size;} XtConvertArgRec, *XtConvertArgList;
typedef void (*XtConvertArgProc)();
typedef struct {
    XtGeometryMask request_mode;
    Position x, y;
    Dimension width, height, border_width;
    Widget sibling;
    int stack_mode; /* Above, Below, TopIf, BottomIf, Opposite, DontChange */
} XtWidgetGeometry;
#define XtCWQueryOnly (1 << 7)
#define XtISMDontChange 5
typedef void (*XtConverter)(); /* obsolete */
typedef Boolean (*XtTypeConverter)();
typedef void (*XtDestructor)();
typedef Opaque XtCacheRef;
typedef Opaque XtActionHookId;
typedef void (*XtActionHookProc)();
typedef void (*XtKeyProc)();
typedef void (*XtCaseProc)();
typedef void (*XtEventHandler)();
```

```

typedef unsigned long EventMask;
typedef enum { XtListHead, XtListTail } XtListPosition;
typedef unsigned long XtInputMask;

#define XtInputNoneMask    0L
#define XtInputReadMask    (1L<<0)
#define XtInputWriteMask   (1L<<1)
#define XtInputExceptMask  (1L<<2)

typedef void (*XtTimerCallbackProc)();
typedef void (*XtInputCallbackProc)();

typedef struct { String name; XtArgVal value;} Arg, *ArgList;
typedef XtPointer XtVarArgsList;
typedef void (*XtCallbackProc)();

typedef struct _XtCallbackRec {
    XtCallbackProc callback;
    XtPointer closure;
} XtCallbackRec, *XtCallbackList;

typedef enum {
    XtCallbackNoList,
    XtCallbackHasNone,
    XtCallbackHasSome
} XtCallbackStatus;

typedef enum {
    XtGeometryYes, /* Request accepted. */
    XtGeometryNo, /* Request denied. */
    XtGeometryAlmost, /* Request denied, but willing to take replyBox. */
    XtGeometryDone /* Request accepted and done. */
} XtGeometryResult;

typedef enum {XtGrabNone, XtGrabNonexclusive, XtGrabExclusive} XtGrabKind;

typedef struct {
    Widget shell_widget;
    Widget enable_widget;
} XtPopdownIDRec, *XtPopdownID;

typedef struct _XtResource {
    String resource_name; /* Resource name */
    String resource_class; /* Resource class */
    String resource_type; /* Representation type desired */
    Cardinal resource_size; /* Size in bytes of representation */
    Cardinal resource_offset; /* Offset from base to put resource value */
    String default_type; /* representation type of specified default */
    XtPointer default_addr; /* Address of default resource */
} XtResource, *XtResourceList;
typedef void (*XtResourceDefaultProc)();
typedef String (*XtLanguageProc)();
typedef void (*XtErrorMsgHandler)();
typedef void (*XtErrorHandler)();
typedef void (*XtCreatePopupChildProc)();
typedef Boolean (*XtWorkProc)();
typedef struct { char match; String substitution;} SubstitutionRec, *Substitution;
typedef Boolean (*XtFilePredicate)();
typedef XtPointer XtRequestId;
typedef Boolean (*XtConvertSelectionProc)();
typedef void (*XtLoseSelectionProc)();
typedef void (*XtSelectionDoneProc)();
typedef void (*XtSelectionCallbackProc)();
typedef void (*XtLoseSelectionIncrProc)();
typedef void (*XtSelectionDoneIncrProc)();
typedef Boolean (*XtConvertSelectionIncrProc)();
typedef void (*XtCancelConvertSelectionProc)();
#define XtAllEvents ((EventMask) -1L)
#define XtIMXEvent 1
#define XtIMTimer 2
#define XtIMAlternateInput 4
#define XtIMAll (XtIMXEvent | XtIMTimer | XtIMAlternateInput)
#define XtVaNestedList "XtVaNestedList"
#define XtVaTypedArg "XtVaTypedArg"
#define XtUnspecifiedPixmap ((Pixmap)2)
#define XtUnspecifiedShellInt (-1)
#define XtUnspecifiedWindow ((Window)2)
#define XtUnspecifiedWindowGroup ((Window)3)
#define XtDefaultForeground "XtDefaultForeground"
#define XtDefaultBackground "XtDefaultBackground"
#define XtDefaultFont "XtDefaultFont"

```

```
#define XtDefaultFontSet    "XtDefaultFontSet"
#define XT_CONVERT_FAIL (Atom)0x80000001
```

Figure 10-7: Manifest Constants and Data Types from <X11/Object.h>

```
typedef struct _ObjectRec *Object;
typedef struct _ObjectClassRec *ObjectClass;
```

Figure 10-8: Manifest Constants and Data Types from <X11/RectObj.h>

```
typedef struct _RectObjRec *RectObj;
typedef struct _RectObjClassRec *RectObjClass;
```

Figure 10-9: Manifest Constants and Data Types from <X11/Shell.h>

```
/* Shell Widget */
/* Shell-specific resources names, classes, and a representation type. */
/* The string definitions are automatically generated. */

#define XtNiconName "iconName"
#define XtCIconName "IconName"
#define XtNiconPixmap "iconPixmap"
#define XtCIconPixmap "IconPixmap"
#define XtNiconWindow "iconWindow"
#define XtCIconWindow "IconWindow"
#define XtNiconMask "iconMask"
#define XtCIconMask "IconMask"
#define XtNwindowGroup "windowGroup"
#define XtCWindowGroup "WindowGroup"
#define XtNvisual "visual"
#define XtCVisual "Visual"
#define XtNtitleEncoding "titleEncoding"
#define XtCTitleEncoding "TitleEncoding"
#define XtNsaveUnder "saveUnder"
#define XtCSaveUnder "SaveUnder"
#define XtNtransient "transient"
#define XtCTransient "Transient"
#define XtNoverrideRedirect "overrideRedirect"
#define XtCOverrideRedirect "OverrideRedirect"
#define XtNtransientFor "transientFor"
#define XtCTransientFor "TransientFor"
#define XtNiconNameEncoding "iconNameEncoding"
#define XtCIconNameEncoding "IconNameEncoding"
#define XtNallowShellResize "allowShellResize"
#define XtCAllowShellResize "AllowShellResize"
#define XtNcreatePopupChildProc "createPopupChildProc"
#define XtCCreatePopupChildProc "CreatePopupChildProc"
#define XtNtitle "title"
#define XtCTitle "Title"
#define XtRAtom "Atom"
#define XtNargc "argc"
#define XtCArgc "Argc"
#define XtNargv "argv"
#define XtCArgv "Argv"
#define XtNiconX "iconX"
#define XtCIconX "IconX"
#define XtNiconY "iconY"
#define XtCIconY "IconY"
#define XtNinput "input"
#define XtCInput "Input"
#define XtNiconic "iconic"
#define XtCIconic "Iconic"
#define XtNinitialState "initialState"
#define XtCInitialState "InitialState"
#define XtNgeometry "geometry"
#define XtCGeometry "Geometry"
#define XtNbaseWidth "baseWidth"
#define XtCBaseWidth "BaseWidth"
#define XtNbaseHeight "baseHeight"
#define XtCBaseHeight "BaseHeight"
#define XtNwinGravity "winGravity"
#define XtCWinGravity "WinGravity"
#define XtNminWidth "minWidth"
#define XtCMinWidth "MinWidth"
#define XtNminHeight "minHeight"
```

```
#define XtCMinHeight "MinHeight"
#define XtNmaxWidth "maxWidth"
#define XtCMaxWidth "MaxWidth"
#define XtNmaxHeight "maxHeight"
#define XtCMaxHeight "MaxHeight"
#define XtNwidthInc "widthInc"
#define XtCWidthInc "WidthInc"
#define XtNheightInc "heightInc"
#define XtCHeightInc "HeightInc"
#define XtNminAspectY "minAspectY"
#define XtCMinAspectY "MinAspectY"
#define XtNmaxAspectY "maxAspectY"
#define XtCMaxAspectY "MaxAspectY"
#define XtNminAspectX "minAspectX"
#define XtCMinAspectX "MinAspectX"
#define XtNmaxAspectX "maxAspectX"
#define XtCMaxAspectX "MaxAspectX"
#define XtNwmTimeout "wmTimeout"
#define XtCwmTimeout "WmTimeout"
#define XtNwaitForWm "waitforwm"
#define XtCWaitForWm "Waitforwm"

/* Class record constants */

typedef struct _ShellClassRec *ShellWidgetClass;
typedef struct _OverrideShellClassRec *OverrideShellWidgetClass;
typedef struct _WMShellClassRec *WMShellWidgetClass;
typedef struct _TransientShellClassRec *TransientShellWidgetClass;
typedef struct _TopLevelShellClassRec *TopLevelShellWidgetClass;
typedef struct _ApplicationShellClassRec *ApplicationShellWidgetClass;
```

Figure 10-10: Manifest Constants and Data Types from <X11/StringDefs.h>

```
#define XtNaccelerators "accelerators"
#define XtNallowHoriz "allowHoriz"
#define XtNallowVert "allowVert"
#define XtNancestorSensitive "ancestorSensitive"
#define XtNbackground "background"
#define XtNbackgroundPixmap "backgroundPixmap"
#define XtNbitmap "bitmap"
#define XtNborderColor "borderColor"
#define XtNborder "borderColor"
#define XtNborderPixmap "borderPixmap"
#define XtNborderWidth "borderWidth"
#define XtNcallback "callback"
#define XtNchildren "children"
#define XtNcolormap "colormap"
#define XtNdepth "depth"
#define XtNdestroyCallback "destroyCallback"
#define XtNeditType "editType"
#define XtNfile "file"
#define XtNfont "font"
#define XtNforceBars "forceBars"
#define XtNforeground "foreground"
#define XtNfunction "function"
#define XtNheight "height"
#define XtNhighlight "highlight"
#define XtNhSpace "hSpace"
#define XtNindex "index"
#define XtNinitialResourcesPersistent "initialResourcesPersistent"
#define XtNinnerHeight "innerHeight"
#define XtNinnerWidth "innerWidth"
#define XtNinnerWindow "innerWindow"
#define XtNinsertPosition "insertPosition"
#define XtNinternalHeight "internalHeight"
#define XtNinternalWidth "internalWidth"
#define XtNjumpProc "jumpProc"
#define XtNjustify "justify"
#define XtNknobHeight "knobHeight"
#define XtNknobIndent "knobIndent"
#define XtNknobPixel "knobPixel"
#define XtNknobWidth "knobWidth"
#define XtNlabel "label"
#define XtNlength "length"
#define XtNlowerRight "lowerRight"
#define XtNmappedWhenManaged "mappedWhenManaged"
#define XtNmenuItem "menuItem"
#define XtNname "name"
```

```

#define XtNnotify "notify"
#define XtNnumChildren "numChildren"
#define XtNorientation "orientation"
#define XtNparameter "parameter"
#define XtNpixmap "pixmap"
#define XtNpopupCallback "popupCallback"
#define XtNpopdownCallback "popdownCallback"
#define XtNresize "resize"
#define XtNreverseVideo "reverseVideo"
#define XtNscreen "screen"
#define XtNscrollProc "scrollProc"
#define XtNscrollDCursor "scrollDCursor"
#define XtNscrollHCursor "scrollHCursor"
#define XtNscrollLCursor "scrollLCursor"
#define XtNscrollRCursor "scrollRCursor"
#define XtNscrollUCursor "scrollUCursor"
#define XtNscrollVCursor "scrollVCursor"
#define XtNselection "selection"
#define XtNselectionArray "selectionArray"
#define XtNsensitive "sensitive"
#define XtNshown "shown"
#define XtNspace "space"
#define XtNstring "string"
#define XtNtextOptions "textOptions"
#define XtNtextSink "textSink"
#define XtNtextSource "textSource"
#define XtNthickness "thickness"
#define XtNthumb "thumb"
#define XtNthumbProc "thumbProc"
#define XtNtop "top"
#define XtNtranslations "translations"
#define XtNunrealizeCallback "unrealizeCallback"
#define XtNupdate "update"
#define XtNuseBottom "useBottom"
#define XtNuseRight "useRight"
#define XtNvalue "value"
#define XtNvSpace "vSpace"
#define XtNwidth "width"
#define XtNwindow "window"
#define XtNx "x"
#define XtNy "y"
#define XtCAccelerators "Accelerators"
#define XtCBackground "Background"
#define XtCBitmap "Bitmap"
#define XtCBoolean "Boolean"
#define XtCBorderColor "BorderColor"
#define XtCBorderWidth "BorderWidth"
#define XtCCallback "Callback"
#define XtCColormap "Colormap"
#define XtCColor "Color"
#define XtCCursor "Cursor"
#define XtCDepth "Depth"
#define XtCEditType "EditType"
#define XtCEventBindings "EventBindings"
#define XtCFile "File"
#define XtCFont "Font"
#define XtCForeground "Foreground"
#define XtCFraction "Fraction"
#define XtCFunction "Function"
#define XtCHeight "Height"
#define XtCHSpace "HSpace"
#define XtCIndex "Index"
#define XtCInitialResourcesPersistent "InitialResourcesPersistent"
#define XtCInsertPosition "InsertPosition"
#define XtCInterval "Interval"
#define XtCJustify "Justify"
#define XtCKnobIndent "KnobIndent"
#define XtCKnobPixel "KnobPixel"
#define XtCLabel "Label"
#define XtCLength "Length"
#define XtCMappedWhenManaged "MappedWhenManaged"
#define XtCMargin "Margin"
#define XtCMenuItem "MenuItem"
#define XtCNotify "Notify"
#define XtCOrientation "Orientation"
#define XtCParameter "Parameter"
#define XtCPixmap "Pixmap"
#define XtCPosition "Position"
#define XtCReadOnly "ReadOnly"
#define XtCResize "Resize"
#define XtCReverseVideo "ReverseVideo"

```

```
#define XtCScreen "Screen"
#define XtCScrollProc "ScrollProc"
#define XtCScrollDCursor "ScrollDCursor"
#define XtCScrollHCursor "ScrollHCursor"
#define XtCScrollLCursor "ScrollLCursor"
#define XtCScrollRCursor "ScrollRCursor"
#define XtCScrollUCursor "ScrollUCursor"
#define XtCScrollVCursor "ScrollVCursor"
#define XtCSelection "Selection"
#define XtCSensitive "Sensitive"
#define XtCSelectionArray "SelectionArray"
#define XtCSpace "Space"
#define XtCString "String"
#define XtCTextOptions "TextOptions"
#define XtCTextPosition "TextPosition"
#define XtCTextSink "TextSink"
#define XtCTextSource "TextSource"
#define XtCThickness "Thickness"
#define XtCThumb "Thumb"
#define XtCTranslations "Translations"
#define XtCValue "Value"
#define XtCVSpace "VSpace"
#define XtCWidth "Width"
#define XtCWindow "Window"
#define XtCX "X"
#define XtCY "Y"
#define XtRAcceleratorTable "AcceleratorTable"
#define XtRAtom "Atom"
#define XtRBitmap "Bitmap"
#define XtRBool "Bool"
#define XtRBoolean "Boolean"
#define XtRCallback "Callback"
#define XtRCallProc "CallProc"
#define XtRCardinal "Cardinal"
#define XtRColor "Color"
#define XtRColormap "Colormap"
#define XtRCursor "Cursor"
#define XtRDimension "Dimension"
#define XtRDisplay "Display"
#define XtREditMode "EditMode"
#define XtREnum "Enum"
#define XtRFile "File"
#define XtRFloat "Float"
#define XtRFont "Font"
#define XtRFontStruct "FontStruct"
#define XtRFunction "Function"
#define XtRGeometry "Geometry"
#define XtRImmediate "Immediate"
#define XtRInitialState "InitialState"
#define XtRInt "Int"
#define XtRJustify "Justify"
#define XtRLongBoolean "Bool"
#define XtRObject "Object"
#define XtROrientation "Orientation"
#define XtRPixel "Pixel"
#define XtRPixmap "Pixmap"
#define XtRPointer "Pointer"
#define XtRPosition "Position"
#define XtRScreen "Screen"
#define XtRShort "Short"
#define XtRString "String"
#define XtRStringArray "StringArray"
#define XtRStringTable "StringTable"
#define XtRUnsignedChar "UnsignedChar"
#define XtRTranslationTable "TranslationTable"
#define XtRVisual "Visual"
#define XtRWidget "Widget"
#define XtRWidgetClass "WidgetClass"
#define XtRWidgetList "WidgetList"
#define XtRWindow "Window"
#define XtEoff "off"
#define XtEfalse "false"
#define XtEno "no"
#define XtEon "on"
#define XtEtrue "true"
#define XtEyes "yes"
#define XtEvertical "vertical"
#define XtEhorizontal "horizontal"
#define XtEtextRead "read"
#define XtEtextAppend "append"
#define XtEtextEdit "edit"
```



```
#define XtExtdefaultbackground "xtdefaultbackground"  
#define XtExtdefaultforeground "xtdefaultforeground"  
#define XtExtdefaultfont "xtdefaultfont"  
#define XtNfontSet "fontSet"  
#define XtRFontSet "FontSet"  
#define XtCFontSet "FontSet"
```

Figure 10-11: Manifest Constants and Data Types from <X11/Vendor.h>

```
typedef struct _VendorShellClassRec *VendorShellWidgetClass;
```

Subclassing Xt Widgets

Figure 10-12 through Figure 10-19 are intended to be used by widget programmers only. This information is included so widget programmers can subclass Xt widgets. This facilities are provided in the X Toolkit for the purpose of creating new widgets based on the Toolkit widgets. They are expected to be used only by code within widgets, not by code within ordinary applications. The structure of widgets should be opaque to the applications that use widgets but do not define them.

The subclassing of widgets is only supported for the Xt widgets. It is not supporting for either the OLIT widget set or the Motif widget set.

Figure 10-12: Manifest Constants and Data Types from <X11/CompositeP.h>

```
typedef struct _CompositePart {
    WidgetList children; /* array of ALL widget children */
    Cardinal num_children; /* total number of widget children */
    Cardinal num_slots; /* number of slots in children array */
    XtOrderProc insert_position; /* compute position of new child */
} CompositePart,*CompositePtr;
typedef struct _CompositeRec {CorePart core; CompositePart composite;} CompositeRec;
typedef struct _CompositeClassPart {
    XtGeometryHandler geometry_manager; /* geometry manager for children */
    XtWidgetProc change_managed; /* change managed state of child */
    XtWidgetProc insert_child; /* physically add child to parent */
    XtWidgetProc delete_child; /* physically remove child */
    XtPointer extension; /* pointer to extension record */
} CompositeClassPart,*CompositePartPtr;
typedef struct {
    XtPointer next_extension; /* 1st 4 mandated for all extension records */
    XrmQuark record_type; /* NULLQUARK; on CompositeClassPart */
    long version; /* must be XtCompositeExtensionVersion */
    Cardinal record_size; /* sizeof(CompositeClassExtensionRec) */
    Boolean accepts_objects;
} CompositeClassExtensionRec,*CompositeClassExtension;
typedef struct _CompositeClassRec { CoreClassPart core_class;CompositeClassPart composite_class;} CompositeClassRec;
#define XtCompositeExtensionVersion 1L
#define XtInheritGeometryManager ((XtGeometryHandler) _XtInherit)
#define XtInheritChangeManaged ((XtWidgetProc) _XtInherit)
#define XtInheritInsertChild ((XtWidgetProc) _XtInherit)
#define XtInheritDeleteChild ((XtWidgetProc) _XtInherit)
```

Figure 10-13: Manifest Constants and Data Types from <X11/ConstrainPh>

```
typedef struct _ConstraintPart { XtPointer mumble;} ConstraintPart;
typedef struct _ConstraintRec { CorePart core; CompositePart composite; ConstraintPart constraint;} ConstraintRec, *ConstraintWidget;
typedef struct _ConstraintClassPart {
    XtResourceList resources; /* constraint resource list */
    Cardinal num_resources; /* number of constraints in list */
    Cardinal constraint_size; /* size of constraint record */
    XtInitProc initialize; /* constraint initialization */
    XtWidgetProc destroy; /* constraint destroy proc */
    XtSetValuesFunc set_values; /* constraint set_values proc */
    XtPointer extension; /* pointer to extension record */
} ConstraintClassPart;
typedef struct {
    XtPointer next_extension; /* 1st 4 mandated for all extension records */
    XrmQuark record_type; /* NULLQUARK; on ConstraintClassPart */
    long version; /* must be XtConstraintExtensionVersion */
    Cardinal record_size; /* sizeof(ConstraintClassExtensionRec) */
    XtArgsProc get_values_hook;
} ConstraintClassExtensionRec, *ConstraintClassExtension;
typedef struct _ConstraintClassRec { CoreClassPart core_class; CompositeClassPart composite_class; ConstraintClassPart constraint_class;} ConstraintClassRec;
#define XtConstraintExtensionVersion 1L
```

Figure 10-14: Manifest Constants and Data Types from <X11/CorePh>

```
#define XtInheritTranslations ((String) &_XtInheritTranslations)
#define XtInheritRealize ((XtRealizeProc) _XtInherit)
#define XtInheritResize ((XtWidgetProc) _XtInherit)
#define XtInheritExpose ((XtExposeProc) _XtInherit)
#define XtInheritSetValuesAlmost ((XtAlmostProc) _XtInherit)
#define XtInheritAcceptFocus ((XtAcceptFocusProc) _XtInherit)
#define XtInheritQueryGeometry ((XtGeometryHandler) _XtInherit)
#define XtInheritDisplayAccelerator ((XtStringProc) _XtInherit)
typedef struct _CorePart {
    Widget self; /* pointer to widget itself */
    WidgetClass widget_class; /* pointer to Widget's ClassRec */
    Widget parent; /* parent widget */
    XrmName xrm_name; /* widget resource name quarkified */
    Boolean being_destroyed; /* marked for destroy */
    XtCallbackList destroy_callbacks; /* who to call when widget destroyed */
    XtPointer constraints; /* constraint record */
    Position x, y; /* window position */
    Dimension width, height; /* window dimensions */
    Dimension border_width; /* window border width */
    Boolean managed; /* is widget geometry managed? */
    Boolean sensitive; /* is widget sensitive to user events? */
    Boolean ancestor_sensitive; /* are all ancestors sensitive? */
    XtEventTable event_table; /* private to event dispatcher */
    XtTMRRec tm; /* translation management */
    XtTranslations accelerators; /* accelerator translations */
    Pixel border_pixel; /* window border pixel */
    Pixmap border_pixmap; /* window border pixmap or NULL */
    WidgetList popup_list; /* list of popups */
    Cardinal num_popups; /* how many popups */
    String name; /* widget resource name */
    Screen *screen; /* window's screen */
    Colormap colormap; /* colormap */
    Window window; /* window ID */
    Cardinal depth; /* number of planes in window */
    Pixel background_pixel; /* window background pixel */
    Pixmap background_pixmap; /* window background pixmap or NULL */
    Boolean visible; /* is window mapped and not occluded? */
    Boolean mapped_when_managed; /* map window if it's managed? */
} CorePart;
typedef struct _WidgetRec { CorePart core; } WidgetRec, CoreRec;
typedef struct _CoreClassPart {
    WidgetClass superclass; /* pointer to superclass ClassRec */
    String class_name; /* widget resource class name */
    Cardinal widget_size; /* size in bytes of widget record */
    XtProc class_initialize; /* class initialization proc */
    XtWidgetClassProc class_part_initialize; /* dynamic initialization */
    XtEnum class_initied; /* has class been initialized? */
    XtInitProc initialize; /* initialize subclass fields */
    XtArgsProc initialize_hook; /* notify that initialize called */
    XtRealizeProc realize; /* XCreateWindow for widget */
    XtActionList actions; /* widget semantics name to proc map */
    Cardinal num_actions; /* number of entries in actions */
}
```

```

XtResourceList resources; /* resources for subclass fields */
Cardinal num_resources; /* number of entries in resources */
XrmClass xrm_class; /* resource class quarkified */
Boolean compress_motion; /* compress MotionNotify for widget */
XtEnum compress_exposure; /* compress Expose events for widget*/
Boolean compress_enterleave; /* compress enter and leave events */
Boolean visible_interest; /* select for VisibilityNotify */
XtWidgetProc destroy; /* free data for subclass pointers */
XtWidgetProc resize; /* geom manager changed widget size */
XtExposeProc expose; /* redisplay window */
XtSetValuesFunc set_values; /* set subclass resource values */
XtArgsFunc set_values_hook; /* notify that set_values called */
XtAlmostProc set_values_almost; /* set_values got "Almost" geo reply */
XtArgsProc get_values_hook; /* notify that get_values called */
XtAcceptFocusProc accept_focus; /* assign input focus to widget */
XtVersionType version; /* version of intrinsics used */
XtPointer callback_private; /* list of callback offsets */
String tm_table; /* state machine */
XtGeometryHandler query_geometry; /* return preferred geometry */
XtStringProc display_accelerator; /* display your accelerator */
XtPointer extension; /* pointer to extension record */
} CoreClassPart;
typedef struct _WidgetClassRec { CoreClassPart core_class; } WidgetClassRec, CoreClassRec;
#define coreClassRec widgetClassRec

```

Figure 10-15: Manifest Constants and Data Types from <X11/IntrinsicP.h>

```

typedef struct {
    XrmQuark xrm_name; /* Resource name quark */
    XrmQuark xrm_class; /* Resource class quark */
    XrmQuark xrm_type; /* Resource representation type quark */
    Cardinal xrm_size; /* Size in bytes of representation */
    long int xrm_offset; /* -offset-1 */
    XrmQuark xrm_default_type; /* Default representation type quark */
    XtPointer xrm_default_addr; /* Default resource address */
} XrmResource, *XrmResourceList;
typedef unsigned long XtVersionType;
#define XT_VERSION 11
#define XT_REVISION 5
#define XtVersion (XT_VERSION * 1000 + XT_REVISION)
#define XtVersionDontCheck 0
typedef void (*XtProc)();
typedef void (*XtWidgetClassProc)();
typedef void (*XtWidgetProc)();
typedef Boolean (*XtAcceptFocusProc)();
typedef void (*XtArgsProc)();
typedef void (*XtInitProc)();
typedef Boolean (*XtSetValuesFunc)();
typedef Boolean (*XtArgsFunc)();
typedef void (*XtAlmostProc)();
typedef void (*XtExposeProc)();
#define XtExposeNoCompress ((XtEnum)False)
#define XtExposeCompressSeries ((XtEnum)True)
#define XtExposeCompressMultiple 2
#define XtExposeCompressMaximal 3
#define XtExposeGraphicsExpose 0x10
#define XtExposeGraphicsExposeMerged 0x20
#define XtExposeNoExpose 0x40
typedef void (*XtRealizeProc)();
typedef XtGeometryResult (*XtGeometryHandler)();
typedef void (*XtStringProc)();
typedef struct _XtTMRec {
    XtTranslations translations; /* private to Translation Manager */
    XtBoundActions proc_table; /* procedure bindings for actions */
    struct _XtStateRec *current_state; /* Translation Manager state ptr */
    unsigned long lastEventTime;
} XtTMRec, *XtTM;
extern Widget _XtWindowedAncestor( /* internal; implementation-dependent */);
extern void _XtInherit();
extern void XtCreateWindow();
extern void XtResizeWidget();
extern void XtMoveWidget();
extern void XtConfigureWidget();
extern void XtResizeWindow();

```

Figure 10-16: Data Types from <X11/ObjectP.h>

```

/*Object Instance Data Structures */
/* these fields match CorePart and can not be changed */

typedef struct _ObjectPart {
    Widget      self;          /* pointer to widget itself */
    WidgetClass widget_class; /* pointer to Widget's ClassRec */
    Widget      parent;       /* parent widget */
    XrmName     xrm_name;     /* widget resource name quarkified */
    Boolean     being_destroyed; /* marked for destroy */
    XtCallbackList destroy_callbacks; /* who to call when widget destroyed */
    XtPointer   constraints;  /* constraint record */
} ObjectPart;

typedef struct _ObjectRec {
    ObjectPart object;
} ObjectRec;

/*Object Class Data Structures */
/* these fields match CoreClassPart and can not be changed */
/* ideally these structures would only contain the fields required;
   but because the CoreClassPart cannot be changed at this late date
   extraneous fields are necessary to make the field offsets match */

typedef struct _ObjectClassPart {
    WidgetClass superclass; /* pointer to superclass ClassRec */
    String      class_name; /* widget resource class name */
    Cardinal    widget_size; /* size in bytes of widget record */
    XtProc      class_initialize; /* class initialization proc */
    XtWidgetClassProc class_part_initialize; /* dynamic initialization */
    XtEnum      class_initied; /* has class been initialized? */
    XtInitProc  initialize; /* initialize subclass fields */
    XtArgsProc  initialize_hook; /* notify that initialize called */
    XtProc      obj1; /* NULL */
    XtProc      obj2; /* NULL */
    Cardinal    obj3; /* NULL */
    XtResourceList resources; /* resources for subclass fields */
    Cardinal    num_resources; /* number of entries in resources */
    XrmClass     xrm_class; /* resource class quarkified */
    Boolean     obj4; /* NULL */
    Boolean     obj5; /* NULL */
    Boolean     obj6; /* NULL */
    Boolean     obj7; /* NULL */
    XtWidgetProc destroy; /* free data for subclass pointers */
    XtProc      obj8; /* NULL */
    XtProc      obj9; /* NULL */
    XtSetValuesFunc set_values; /* set subclass resource values */
    XtArgsFunc  set_values_hook; /* notify that set_values called */
    XtProc      obj10; /* NULL */
    XtArgsProc  get_values_hook; /* notify that get_values called */
    XtProc      obj11; /* NULL */
    XtVersionType version; /* version of intrinsics used */
    XtPointer   callback_private; /* list of callback offsets */
    String      obj12; /* NULL */
    XtProc      obj13; /* NULL */
    XtProc      obj14; /* NULL */
    XtPointer   extension; /* pointer to extension record */
} ObjectClassPart;

typedef struct _ObjectClassRec {
    ObjectClassPart object_class;
} ObjectClassRec;

```

Figure 10-17: Data Types from <X11/RectObjP.h>

```

/* Rectangle Object Instance Data Structures */
/* these fields match CorePart and can not be changed */

typedef struct _RectObjPart {
    Position    x, y; /* rectangle position */
    Dimension   width, height; /* rectangle dimensions */
    Dimension   border_width; /* rectangle border width */
    Boolean     managed; /* is widget geometry managed? */
    Boolean     sensitive; /* is widget sensitive to user events? */
    Boolean     ancestor_sensitive; /* are all ancestors sensitive? */
} RectObjPart;

typedef struct _RectObjRec {
    ObjectPart object;
    RectObjPart rectangle;
} RectObjRec;

```

```

/* Rectangle Object Class Data Structures */
/* these fields match CoreClassPart and can not be changed */
/* ideally these structures would only contain the fields required;
   but because the CoreClassPart cannot be changed at this late date
   extraneous fields are necessary to make the field offsets match */

typedef struct _RectObjClassPart {
    WidgetClass superclass; /* pointer to superclass ClassRec */
    String class_name; /* widget resource class name */
    Cardinal widget_size; /* size in bytes of widget record */
    XtProc class_initialize; /* class initialization proc */
    XtWidgetClassProc class_part_initialize; /* dynamic initialization */
    XtEnum class_inited; /* has class been initialized? */
    XtInitProc initialize; /* initialize subclass fields */
    XtArgsProc initialize_hook; /* notify that initialize called */
    XtProc rect1; /* NULL */
    XtPointer rect2; /* NULL */
    Cardinal rect3; /* NULL */
    XtResourceList resources; /* resources for subclass fields */
    Cardinal num_resources; /* number of entries in resources */
    XrmClass xrm_class; /* resource class quarkified */
    Boolean rect4; /* NULL */
    Boolean rect5; /* NULL */
    Boolean rect6; /* NULL */
    Boolean rect7; /* NULL */
    XtWidgetProc destroy; /* free data for subclass pointers */
    XtWidgetProc resize; /* geom manager changed widget size */
    XtExposeProc expose; /* redisplay rectangle */
    XtSetValuesFunc set_values; /* set subclass resource values */
    XtArgsFunc set_values_hook; /* notify that set_values called */
    XtAlmostProc set_values_almost; /* set values almost for geometry */
    XtArgsProc get_values_hook; /* notify that get_values called */
    XtProc rect9; /* NULL */
    XtVersionType version; /* version of intrinsics used */
    XtPointer callback_private; /* list of callback offsets */
    String rect10; /* NULL */
    XtGeometryHandler query_geometry; /* return preferred geometry */
    XtProc rect11; /* NULL */
    XtPointer extension; /* pointer to extension record */
} RectObjClassPart;

typedef struct _RectObjClassRec {
    RectObjClassPart rect_class;
} RectObjClassRec;

```

Figure 10-18: Manifest Constants and Data Types from <X11/ShellP.h>

```

/* Shell Widget Private Data */
/* New fields for the Shell widget class record */

typedef struct {XtPointer extension;} ShellClassPart;

typedef struct {
    XtPointer next_extension;
    XrmQuark record_type;
    long version;
    Cardinal record_size;
    XtGeometryHandler root_geometry_manager;
} ShellClassExtensionRec, *ShellClassExtension;

#define XtShellExtensionVersion 1L
#define XtInheritRootGeometryManager ((XtGeometryHandler)_XtInherit)

typedef struct _ShellClassRec {
    CoreClassPart core_class;
    CompositeClassPart composite_class;
    ShellClassPart shell_class;
} ShellClassRec;

/* New fields for the shell widget */
typedef struct {
    char *geometry;
    XtCreatePopupChildProc create_popup_child_proc;
    XtGrabKind grab_kind;
    Boolean spring_loaded;
    Boolean popped_up;
    Boolean allow_shell_resize;
}

```

```

        Boolean
#define _XtShellPositionValid          client_specified; /* re-using old name */
#define _XtShellNotReparented         (Boolean)(1<<0)
#define _XtShellPPositionOK           (Boolean)(1<<1)
#define _XtShellGeometryParsed       (Boolean)(1<<2)
        Boolean                         save_under;
        Boolean                         override_redirect;
        XtCallbackList                  popup_callback;
        XtCallbackList                  popdown_callback;
        Visual*                         visual;
} ShellPart;

typedef struct {
        CorePart                        core;
        CompositePart                   composite;
        ShellPart                       shell;
} ShellRec, *ShellWidget;

/* OverrideShell Widget Private Data */
/* New fields for the OverrideShell widget class record */
typedef struct { XtPointer extension; } OverrideShellClassPart;

typedef struct _OverrideShellClassRec {
        CoreClassPart   core_class;
        CompositeClassPart composite_class;
        ShellClassPart  shell_class;
        OverrideShellClassPart override_shell_class;
} OverrideShellClassRec;

/* No new fields for the override shell widget */
typedef struct {int frabjous;} OverrideShellPart;

typedef struct {
        CorePart      core;
        CompositePart composite;
        ShellPart     shell;
        OverrideShellPart override;
} OverrideShellRec, *OverrideShellWidget;
/* WMSHELL Widget Private Data , New fields for the WMSHELL widget class record */
typedef struct { XtPointer extension; } WMSHELLClassPart;

typedef struct _WMSHELLClassRec {
        CoreClassPart   core_class;
        CompositeClassPart composite_class;
        ShellClassPart  shell_class;
        WMSHELLClassPart wm_shell_class;
} WMSHELLClassRec;
/* New fields for the WM shell widget */
typedef struct {
        char      *title;
        int       wm_timeout;
        Boolean   wait_for_wm;
        Boolean   transient;
        Atom      wm_configure_denied, wm_moved;
        struct _OldXSizeHints { /* pre-R4 Xlib structure */
                long flags;
                int x, y;
                int width, height;
                int min_width, min_height;
                int max_width, max_height;
                int width_inc, height_inc;
                struct {int x; int y;} min_aspect, max_aspect;
        } size_hints;
        XWMHints  wm_hints;
        int base_width, base_height;
        int win_gravity;
        Atom title_encoding;
} WMSHELLPart;

typedef struct {
        CorePart      core;
        CompositePart composite;
        ShellPart     shell;
        WMSHELLPart  wm;
} WMSHELLRec, *WMSHELLWidget;

/* TransientShell Widget Private Data */
/* New fields for the TransientShell widget class record */
typedef struct { XtPointer extension; } TransientShellClassPart;

typedef struct _TransientShellClassRec {

```

```
CoreClassPart core_class;
CompositeClassPart composite_class;
ShellClassPart shell_class;
WMShellClassPart wm_shell_class;
VendorShellClassPart vendor_shell_class;
TransientShellClassPart transient_shell_class;
} TransientShellClassRec;

/* New fields for the transient shell widget */
typedef struct { Widget transient_for; } TransientShellPart;

typedef struct {
    CorePart core;
    CompositePart composite;
    ShellPart shell;
    WMShellPart wm;
    VendorShellPart vendor;
    TransientShellPart transient;
} TransientShellRec, *TransientShellWidget;

/* TopLevelShell Widget Private Data */
/* New fields for the TopLevelShell widget class record */
typedef struct { XtPointer extension; } TopLevelShellClassPart;

typedef struct _TopLevelShellClassRec {
    CoreClassPart core_class;
    CompositeClassPart composite_class;
    ShellClassPart shell_class;
    WMShellClassPart wm_shell_class;
    VendorShellClassPart vendor_shell_class;
    TopLevelShellClassPart top_level_shell_class;
} TopLevelShellClassRec;

/* New fields for the top level shell widget */
typedef struct {
    char *icon_name;
    Boolean iconic;
    Atom icon_name_encoding;
} TopLevelShellPart;

typedef struct {
    CorePart core;
    CompositePart composite;
    ShellPart shell;
    WMShellPart wm;
    VendorShellPart vendor;
    TopLevelShellPart topLevel;
} TopLevelShellRec, *TopLevelShellWidget;

/* ApplicationShell Widget Private Data */
/* New fields for the ApplicationShell widget class record */
typedef struct { XtPointer extension; } ApplicationShellClassPart;

typedef struct _ApplicationShellClassRec {
    CoreClassPart core_class;
    CompositeClassPart composite_class;
    ShellClassPart shell_class;
    WMShellClassPart wm_shell_class;
    VendorShellClassPart vendor_shell_class;
    TopLevelShellClassPart top_level_shell_class;
    ApplicationShellClassPart application_shell_class;
} ApplicationShellClassRec;

/* New fields for the application shell widget */
typedef struct {
    char *class;
    XrmClass xrm_class;
    int argc;
    char **argv;
} ApplicationShellPart;

typedef struct {
    CorePart core;
    CompositePart composite;
    ShellPart shell;
    WMShellPart wm;
    VendorShellPart vendor;
    TopLevelShellPart topLevel;
    ApplicationShellPart application;
}
```



```
} ApplicationShellRec, *ApplicationShellWidget;
```

Figure 10-19: Data Types from <X11/VendorPh>

```
/* New fields for the VendorShell widget class record */
typedef struct {
    XiPointer    extension; /* pointer to extension record */
} VendorShellClassPart;

typedef struct _VendorShellClassRec {
    CoreClassPart    core_class;
    CompositeClassPart composite_class;
    ShellClassPart    shell_class;
    WMShellClassPart  wm_shell_class;
    VendorShellClassPart vendor_shell_class;
} VendorShellClassRec;

/* New fields for the vendor shell widget. */
typedef struct {int vendor_specific;} VendorShellPart;

typedef struct {
    CorePart    core;
    CompositePart composite;
    ShellPart    shell;
    WMShellPart  wm;
    VendorShellPart vendor;
} VendorShellRec, *VendorShellWidget;
```

The OPEN LOOK Widget Set

Overview

This chapter identifies binary interfaces for libXol. A source description for the entry points and exported data structures so identified may be found in the *OLIT Reference Manual* (Sun Microsystems, Part No. 800-6055-10, Revision A.).

The libXol Interfaces

The names listed below in Tables 10-9 and 10-10 have been included in SCD 2.4 as the names for the OPEN LOOK Widget Set REQUIRED INTERFACES, and must to be present on all conformant systems through the reference name */usr/lib/libXol.so.3*. Note that Table 10-10 defines the size of exported data objects as a hexadecimal byte count described in square brackets after each name. However, this interface set is also DEPRECATED effective November 1st, 1993 and may be removed from this specification as early as November 1st, 1996. No new applications should be developed to use the OPEN LOOK Widget Set.

Figures 10-27 and 10-28 detail the manifest constants associated with libXol and its visible data structures, respectively. The sizes of certain global data are deliberately hidden from programmers. These sizes are left unspecified because there is no attempt made to support subclassing of OLIT widgets.

The names listed below in Tables 10-9 and 10-10 have been included in SCD 2.4 as the names for the OPEN LOOK Widget Set REQUIRED INTERFACES, and must be present on all conformant systems through the reference name */usr/lib/libXol.so.3*. However, this interface set is also DEPRECATED effective November 1st, 1993, and may be removed from this specification as early as November 1st, 1996. No new applications should be developed to use the OPEN LOOK Widget Set.

Table 10-10 defines the size of exported data objects as a hexadecimal byte count in square brackets after each name. Some of the sizes are given as question marks. Where the size of an item is a question mark, that symbol is a reserved symbol which must not be used by an SCD-conforming application. They are widget class records, and are only used in order to subclass widgets. The subclassing of OLIT widgets is not supported by the SCD.

Rationale:

The class records for OPEN LOOK widgets are of interest only for subclassing of these widgets. Programs that use these widgets without subclassing them need not and must not reference these symbols. Programs must also not use these symbols as external symbols for any other purpose because this will interfere with the Widget Library's use of them.

Programs that do subclass OPEN LOOK widgets do not conform to the SCD. This is because the authors of the SCD cannot guarantee upward compatibility of widgets from one release to the next. The size of each widget class record is subject to being changed by the owner(s) of the code. Widget subclass records depend on the size of the class record of the superordinate class. They will be rendered invalid at the binary interface level if the size of the superordinate class record changes.

Table 10-9. libXol Contents

AllocateBuffer	OIDragAndDrop	OITextEditReadSubString
AllocateTextBuffer	OIError	OITextEditRedraw
BackwardScanTextBuffer	OIGet50PercentGrey	OITextEditResize
CopyBuffer	OIGet75PercentGrey	OITextEditSetCursorPosition
CopyTextBufferBlock	OIGetApplicationResources	OITextEditTextBuffer
EndCurrentTextBufferWord	OIGetApplicationValues	OITextEditUpdate
ForwardScanTextBuffer	OIGetBeepVolume	OITextFieldCopyString
FreeBuffer	OIGetCurrentFocusWidget	OITextFieldGetString
FreeTextBuffer	OIGrabDragPointer	OIToolkitInitialize
GetOIBusyCursor	OIHasCallbacks	OIUgrabDragPointer
GetOIDuplicateCursor	OIHasFocus	OIUpdateDisplay
GetOIMoveCursor	OIInitialize	OIVaDisplayErrorMsg
GetOIPanCursor	OILayoutScrolledWindow	OIVaDisplayWarningMsg
GetOIQuestionCursor	OILListItemPointer	OIWMProtocolAction
GetOISWGeometries	OIMenuPopdown	OIWarning
GetOIStandardCursor	OIMenuPopup	PositionOfLine
GetOITargetCursor	OIMenuPost	PositionOfLocation
GetTextBufferBlock	OIMenuUnpost	PreviousLocation
GetTextBufferBuffer	OIMoveFocus	PreviousTextBufferWord
GetTextBufferChar	OIQueryAcceleratorDisplay	ReadFileIntoBuffer
GetTextBufferLine	OIQueryMnemonicDisplay	ReadFileIntoTextBuffer
GetTextBufferLocation	OIRegisterColorTupleListConverter	ReadStringIntoBuffer
GrowBuffer	OIRegisterHelp	ReadStringIntoTextBuffer
IncrementTextBufferLocation	OIRemoveCallback	RegisterTextBufferScanFunctions
InsertIntoBuffer	OISetErrorHandler	RegisterTextBufferUpdate
LastTextBufferLocation	OISetGaugeValue	RegisterTextBufferWordDefinition
LastTextBufferPosition	OISetInputFocus	ReplaceBlockInTextBuffer
LineOfPosition	OISetVaDisplayErrorMsgHandler	ReplaceCharInTextBuffer
LocationOfPosition	OISetVaDisplayWarningMsgHandler	SaveTextBuffer
LookupOIInputEvent	OISetWarningHandler	StartCurrentTextBufferWord
NextLocation	OITextEditClearBuffer	UnregisterTextBufferUpdate
NextTextBufferWord	OITextEditCopyBuffer	
OIAddCallback	OITextEditCopySelection	
OICallAcceptFocus	OITextEditGetCursorPosition	
OICallCallbacks	OITextEditGetLastPosition	
OICanAcceptFocus	OITextEditInsert	
OICategorySetPage	OITextEditPaste	

Table 10-10. Exported Data in libXol

abbrevMenuButtonClassRec[?]	footerPanelWidgetClass[0x4]	noticeShellWidgetClass[0x4]
abbrevMenuButtonWidgetClass[0x4]	formClassRec[?]	oblongButtonClassRec[?]
baseWindowShellClassRec[?]	formWidgetClass[0x4]	oblongButtonGadgetClass[0x4]
baseWindowShellWidgetClass[0x4]	gaugeClassRec[?]	oblongButtonGadgetClassRec[?]
bulletinBoardClassRec[?]	gaugeWidgetClass[0x4]	oblongButtonWidgetClass[0x4]
bulletinBoardWidgetClass[0x4]	helpClassRec[?]	popupWindowShellClassRec[?]
buttonClassRec[?]	helpWidgetClass[0x4]	popupWindowShellWidgetClass[0x4]
buttonGadgetClass[0x4]	listClassRec[?]	primitiveClassRec[?]
buttonGadgetClassRec[?]	listPaneClassRec[?]	primitiveWidgetClass[0x4]
buttonWidgetClass[0x4]	listPaneWidgetClass[0x4]	pushpinClassRec[?]
captionClassRec[?]	magClassRec[?]	pushpinWidgetClass[0x4]
captionWidgetClass[0x4]	magWidgetClass[0x4]	rectButtonClassRec[?]
categoryClassRec[?]	managerClassRec[?]	rectButtonWidgetClass[0x4]
categoryWidgetClass[0x4]	managerWidgetClass[0x4]	rubberTileClassRec[?]
checkBoxClassRec[?]	menuButtonClassRec[?]	rubberTileWidgetClass[0x4]
checkBoxWidgetClass[0x4]	menuButtonGadgetClass[0x4]	scrollbarClassRec[?]
controlAreaWidgetClass[0x4]	menuButtonGadgetClassRec[?]	scrollbarWidgetClass[0x4]
controlClassRec[?]	menuButtonWidgetClass[0x4]	scrolledWindowClassRec[?]
eventObjClass[0x4]	menuShellClassRec[?]	scrolledWindowWidgetClass[0x4]
eventObjClassRec[?]	menuShellWidgetClass[0x4]	scrollingListWidgetClass[0x4]
exclusivesClassRec[?]	nonexclusivesClassRec[?]	sliderClassRec[?]
exclusivesWidgetClass[0x4]	nonexclusivesWidgetClass[0x4]	sliderWidgetClass[0x4]
footerPanelClassRec[?]	noticeShellClassRec[?]	staticTextWidgetClass[0x4]

Figure 10-27. libXol Manifest Constants

#define OleditDone	0	#define OL_NOTICES	47	#define OL_VSBMENU	99
#define OleditError	1	#define OL_OBLONG	49	#define OL_NEXTAPP	101
#define OleditPosError	2	#define OL_OUT	50	#define OL_NEXTWINDOW	102
#define OleditReject	3	#define OL_OVERRIDE_PAIR	51	#define OL_PREVAPP	103
#define OL_ABSENT_PAIR	0	#define OL_PIXELS	52	#define OL_PREVWINDOW	104
#define OL_ALL	1	#define OL_POINTS	53	#define OL_WINDOWMENU	105
#define OL_ALWAYS	2	#define OL_POPUP	54	#define OL_WORKSPACEMENU	106
#define OL_ATOM_HELP	3	#define OL_PREVIOUS	55	#define OL_DEFAULTACTION	108
#define OL_BOTH	4	#define OL_PROG_DEFINED_SOURCE	56	#define OL_TOGGLEPUSHPIN	111
#define OL_BOTTOM	5	#define OL_RECTBUTTON	57	#define OL_IMMEDIATE	120
#define OL_BUTTONSTACK	6	#define OL_RIGHT	58	#define OL_CLICK_TO_TYPE	125
#define OL_CENTER	7	#define OL_ROWS	59	#define OL_REALESTATE	126
#define OL_CLASS_HELP	8	#define OL_SOURCE_FORM	60	#define OL_UNDERLINE	127
#define OL_COLUMNS	9	#define OL_SOURCE_PAIR	61	#define OL_HIGHLIGHT	128
#define OL_COPY_MASK_VALUE	10	#define OL_STAYUP	62	#define OL_INACTIVE	129
#define OL_COPY_SIZE	11	#define OL_STRING	63	#define OL_DISPLAY	130
#define OL_COPY_SOURCE_VALUE	12	#define OL_STRING_SOURCE	64	#define OL_PROC	131
#define OL_CURRENT	13	#define OL_TEXT_APPEND	65	#define OL_SIZE_PROC	132
#define OL_DEFAULT_PAIR	14	#define OL_TEXT_EDIT	66	#define OL_DRAW_PROC	133
#define OL_DISK_SOURCE	15	#define OL_TEXT_READ	67	#define OL_PINNED_MENU	134
#define OL_DISPLAY_FORM	16	#define OL_TOP	68	#define OL_PRESS_DRAG_MENU	135
#define OL_DOWN	17	#define OL_TRANSPARENT_SOURCE	69	#define OL_STAYUP_MENU	136
#define OL_EXISTING_SOURCE	18	#define OL_VERTICAL	70	#define OL_POINTER	137
#define OL_FIXEDCOLS	19	#define OL_WIDGET_HELP	73	#define OL_INPUTFOCUS	138
#define OL_FIXEDHEIGHT	20	#define OL_WINDOW_HELP	74	#define OL_QUIT	142
#define OL_FIXEDROWS	21	#define OL_WRAP_ANY	75	#define OL_DESTROY	143
#define OL_FIXEDWIDTH	22	#define OL_WRAP_WHITE_SPACE	76	#define OL_DISMISS	144
#define OL_HALFSTACK	29	#define OL_CONTINUOUS	77	#define OL_PRE	145
#define OL_HORIZONTAL	30	#define OL_GRANULARITY	78	#define OL_POST	146
#define OL_IMAGE	31	#define OL_RELEASE	79		
#define OL_IN	32	#define OL_TICKMARK	80		
#define OL_INDIRECT_SOURCE	33	#define OL_PERCENT	81		
#define OL_LABEL	34	#define OL_SLIDERVALUE	82		
#define OL_LEFT	35	#define OL_WT_BASE	83		
#define OL_MASK_PAIR	36	#define OL_WT_CMD	84		
#define OL_MAXIMIZE	37	#define OL_WT_NOTICE	85		
#define OL_MILLIMETERS	38	#define OL_WT_HELP	86		
#define OL_MINIMIZE	39	#define OL_WT_OTHER	87		
#define OL_NEVER	40	#define OL_SUCCESS	88		
#define OL_NEXT	41	#define OL_BAD_KEY	90		
#define OL_NONE	42	#define OL_MENU_FULL	91		
#define OL_NONEBOTTOM	43	#define OL_MENU_LIMITED	92		
#define OL_NONLEFT	44	#define OL_MENU_CANCEL	93		
#define OL_NONRIGHT	45	#define OL_MENUDEFAULT	96		
#define OL_NONETOP	46	#define OL_HSBMENU	98		

```
typedef enum {motionVerify, modVerify, leaveVerify} OIVerifyOpType;
typedef enum {OlslLeft, OlslRight} OIScanDirection;
typedef enum {OlslPositions, OlslWhiteSpace, OlslEOL, OlslParagraph, OlslLast} OIScanType;
typedef enum {NOTOPEN, READWRITE, READONLY, NEWFILE } TextFileStatus;
typedef enum {EDIT_FAILURE, EDIT_SUCCESS } EditResult;
typedef enum {SCAN_NOTFOUND, SCAN_WRAPPED, SCAN_FOUND, SCAN_INVALID } ScanResult;
typedef enum {SAVE_FAILURE, SAVE_SUCCESS } SaveResult;
```

```
#define TEXT_BUFFER_NOP (0)
#define TEXT_BUFFER_DELETE_START_LINE(1L<<0)
#define TEXT_BUFFER_DELETE_START_CHARS(1L<<1)
#define TEXT_BUFFER_DELETE_END_LINE(1L<<2)
#define TEXT_BUFFER_DELETE_END_CHARS(1L<<3)
#define TEXT_BUFFER_DELETE_JOIN_LINE(1L<<4)
#define TEXT_BUFFER_DELETE_SIMPLE(1L<<5)
#define TEXT_BUFFER_INSERT_SPLIT_LINE(1L<<6)
#define TEXT_BUFFER_INSERT_LINE(1L<<7)
#define TEXT_BUFFER_INSERT_CHARS(1L<<8)
#define CHANGE_BAR_WIDTH 3
#define CHANGE_BAR_HEIGHT 18
#define CHANGE_BAR_PAD 7
```

```
#define OL_DIM 1000
#define OL_NORMAL 1001
#define OL_PROPAGATE_TO_CONTROL_AREA 0x0001
#define OL_PROPAGATE_TO_CATEGORY 0x0002
#define OL_PROPAGATE (OL_PROPAGATE_TO_CONTROL_AREA | OL_PROPAGATE_TO_CATEGORY)
```

Figure 10-28. libXol Data Structures

```
typedef struct _CaptionClassRec CaptionWidgetClass;
typedef struct _CaptionRec *CaptionWidget;
typedef struct _CategoryClassRec *CategoryWidgetClass;
typedef struct _CategoryRec *CategoryWidget;
typedef struct _CheckBoxClassRec *CheckBoxWidgetClass;
typedef struct _CheckBoxRec *CheckBoxWidget;
typedef int ControlLayout;
typedef int OISameSize;
typedef struct _ControlClassRec *ControlAreaWidgetClass;
typedef struct _ControlRec *ControlAreaWidget;
typedef struct _EventObjClassRec *EventObjClass;
typedef struct _EventObjRec *EventObj;
typedef struct _ExclusivesClassRec *ExclusivesWidgetClass;
typedef struct _ExclusivesRec *ExclusivesWidget;
typedef struct _FooterPanelClassRec *FooterPanelWidgetClass;
typedef struct _FooterPanelRec *FooterPanelWidget;
typedef struct _FormClassRec *FormWidgetClass;
typedef struct _FormRec *FormWidget;
typedef struct _FormConstraintRec *FormConstraints;
typedef struct _SliderClassRec *GaugeWidgetClass;
typedef struct _SliderRec *GaugeWidget;
typedef struct _HelpClassRec *HelpWidgetClass;
typedef struct _HelpRec *HelpWidget;
typedef struct _ListPaneClassRec *ListPaneWidgetClass;
typedef struct _ListPaneRec *ListPaneWidget;
typedef struct _MagClassRec *MagWidgetClass;
typedef struct _MagRec *MagWidget;
typedef struct _ManagerClassRec *ManagerWidgetClass;
typedef struct _ManagerRec *ManagerWidget;
typedef struct _MenuShellClassRec *MenuShellWidgetClass;
typedef struct _MenuShellRec *MenuShellWidget;
typedef struct _MenuButtonClassRec *MenuButtonWidgetClass;
typedef struct _MenuButtonRec *MenuButtonWidget;
typedef struct _MenuButtonGadgetClassRec *MenuButtonGadgetClass;
typedef struct _MenuButtonGadgetRec *MenuButtonGadget;
typedef struct _NonexclusivesClassRec *NonexclusivesWidgetClass;
typedef struct _NonexclusivesRec *NonexclusivesWidget;
typedef struct _NoticeShellClassRec *NoticeShellWidgetClass;
typedef struct _NoticeShellRec *NoticeShellWidget;
typedef struct _OblongButtonClassRec *OblongButtonWidgetClass;
typedef struct _OblongButtonRec *OblongButtonWidget;
typedef struct _OblongButtonGadgetClassRec *OblongButtonGadgetClass;
typedef struct _OblongButtonGadgetRec *OblongButtonGadget;
typedef short OIDefine;
typedef unsigned long OIBitMask;
typedef struct _PopupWindowShellClassRec *PopupWindowShellWidgetClass;
typedef struct _PopupWindowShellRec *PopupWindowShellWidget;
typedef struct _PrimitiveClassRec *PrimitiveWidgetClass;
typedef struct _PrimitiveRec *PrimitiveWidget;
typedef struct _PushpinClassRec *PushpinWidgetClass;
typedef struct _PushpinRec *PushpinWidget;
typedef struct _RectButtonClassRec *RectButtonWidgetClass;
typedef struct _RectButtonRec *RectButtonWidget;
typedef struct _RubberTileClassRec *RubberTileWidgetClass;
typedef struct _RubberTileRec *RubberTileWidget;
typedef struct _ScrollbarClassRec *ScrollbarWidgetClass;
typedef struct _ScrollbarRec *ScrollbarWidget;
```

```

typedef struct OIScrollbarVerify {
    int                new_location;
    int                new_page;
    Boolean            ok;
    int                slidermin;
    int                slidermax;
    int                delta;
    Boolean            more_cb_pending;
} OIScrollbarVerify;
typedef struct _ScrolledWindowClassRec *ScrolledWindowWidgetClass;
typedef struct _ScrolledWindowRec *ScrolledWindowWidget;
typedef struct _OISWGeometries {
    Widget            sw;
    Widget            vsb;
    Widget            hsb;
    Dimension         bb_border_width;
    Dimension         vsb_width;
    Dimension         vsb_min_height;
    Dimension         hsb_height;
    Dimension         hsb_min_width;
    Dimension         sw_view_width;
    Dimension         sw_view_height;
    Dimension         bbc_width;
    Dimension         bbc_height;
    Dimension         bbc_real_width;
    Dimension         bbc_real_height;
    Boolean            force_hsb;
    Boolean            force_vsb;
} OISWGeometries;
typedef struct _OILListItem {
    OIDefine            label_type;
    XtPointer           label;
    XImage              *glyph;
    OIBitMask           attr;
    XtPointer           user_data;
    unsigned char       implementation_specific;
} OILListItem;
typedef struct _OILListToken *OILListToken;
typedef struct _OILListDelete { OILListToken *tokens; Cardinal num_tokens; } OILListDelete;
typedef struct _ListClassRec *ScrollingListWidgetClass;
typedef struct _ListRec *ScrollingListWidget;
typedef struct _SliderClassRec *SliderWidgetClass;
typedef struct _SliderRec *SliderWidget;
typedef struct OISliderVerify { int new_location; Boolean more_cb_pending; } OISliderVerify;
typedef struct _StaticTextClassRec *StaticTextWidgetClass;
typedef struct _StaticTextRec *StaticTextWidget;
typedef Dimension *TabTable;
typedef struct { OITextMarginHint hint; XRectangle *rect; } OITextMarginCallData, *OITextMarginCallDataPointer;
typedef struct {
    Boolean            ok;
    TextPosition       current_cursor;
    TextPosition       new_cursor;
    TextPosition       select_start;
    TextPosition       select_end;
} OITextMotionCallData, *OITextMotionCallDataPointer;
typedef struct {
    Booleanok;
    TextPosition       current_cursor;
    TextPosition       select_start;
    TextPosition       select_end;
    TextPosition       new_cursor;
    TextPosition       new_select_start;
    TextPosition       new_select_end;
    String             text;
    int                text_length;
} OITextModifyCallData, *OITextModifyCallDataPointer;

```

```

typedef struct {
    Boolean                requestor;
    TextPosition          new_cursor;
    TextPosition          new_select_start;
    TextPosition          new_select_end;
    String                inserted;
    String                deleted;
    TextLocation          delete_start;
    TextLocation          delete_end;
    TextLocation          insert_start;
    TextLocation          insert_end;
    TextPosition          cursor_position;
} OITextPostModifyCallData, *OITextPostModifyCallDataPointer;
typedef struct _TextEditClassRec
typedef struct _TextEditRec                *TextEditWidgetClass;
typedef struct _TextFieldClassRec          *TextFieldWidgetClass;
typedef struct _TextFieldRec                *TextFieldWidget;
typedef struct { String string; Boolean ok; OITextVerifyReason reason; } OITextFieldVerify, *OITextFieldVerifyPointer;
typedef char BufferElement;
typedef struct _Buffer { int size; int used; int esize; BufferElement *p; } Buffer;
typedef int                TextPosition;
typedef int                TextLine;
typedef int                TextPage;
typedef int                TextBlock;
typedef struct{ int size; int used; int esize; TextBlock *p; } BlockTable;
typedef struct { TextPage pageindex; unsigned long timestamp; } PageQueue;
typedef struct { TextPosition bytes; TextLine lines; TextPage qpos; BlockTable *dpos; } Page;
typedef struct { TextPage pageindex; Buffer *buffer; unsigned long userData; } Line;
typedef struct { int size; int used; int size; Page *p; } PageTable;
typedef struct { int size; int used; int esize; Line *p; } LineTable;
typedef struct _TextLocation { TextLine line; TextPosition offset; BufferElement *buffer; } TextLocation;
typedef int TextUndoHint;
typedef struct _TextUndoItem {
    String                string;
    TextLocation          start;
    TextLocation          end;
    TextUndoHint          hint;
} TextUndoItem;
typedef void (*TextUpdateFunction)();
typedef struct _TextUpdateCallback { TextUpdateFunction f; caddr_t d; } TextUpdateCallback;
typedef struct _TextBuffer {
    char                *filename;
    FILE                *tempfile;
    TextBlock           blockcnt;
    TextBlock           blocksize;
    LineTable           lines;
    PageTable           pages;
    BlockTable          *free_list;
    PageQueue           pqueue[PQLIMIT];
    TextPage            pagecount;
    TextPage            pageref;
    TextPage            curpageno;
    Buffer               *buffer;
    char                dirty;
    TextFileStatus      status;
    int                 refcount;
    TextUpdateCallback *update;
    TextUndoItem        deleted;
    TextUndoItem        insert;
} TextBuffer;

```


Motif 1.2 Widget Set

Overview

This chapter contains the interfaces to the Motif 1.2 user interface environment. The Motif 1.2 interfaces are represented through the libXm and libMrm libraries.

The Motif 1.2 Interface is comprised of a set of Graphical User Interface components that together make up a unique user interface environment. The components of this environment are: the toolkit, window manager, and user interface language. These user interface components are defined in the *OSF/Motif Programmer's Reference, Revision 1.2* (Open Software Foundation, Inc. 1992 - Prentice-Hall, ISBN 0-13-643115-1)

The Motif Interfaces

The interfaces listed below in Table 10-11, Table 10-12 and Table 10-13 have been included in SCD 2.4 because they are required to be present on all systems conforming to the SCD 2.4 REQUIRED interface definition for Motif 1.2, in the dynamic libraries `/usr/lib/libXm.so.1.2`, `/usr/lib/libXm.so.3`, `/usr/dt/lib/libXm.so.1.2`, and `/usr/dt/lib/libXm.so.3`; and `/usr/lib/libMrm.so.1.2`, `/usr/lib/libMrm.so.3`, `/usr/dt/lib/libMrm.so.1.2`, and `/usr/dt/lib/libMrm.so.3`."

Figure 10-29 through Figure 10-74 detail the manifest constants associated with libXm and libMrm and its visible data structures, respectively.

In addition to the interfaces listed in the tables all SCD 2.4 compliant systems that offer Motif must have the `uil(3)` command and the `mwm(3)` window manager as defined in the *OSF/Motif Programmer's Reference, Release 1.2* (Prentice-Hall, ISBN 0-13-643115-1).

Table 10-12 defines the size of exported data objects as a hexadecimal byte count in square brackets after each name. Some of the sizes are given as question marks. Where the size of an item is a question mark, that symbol is a reserved symbol which must not be used by an SCD-conforming application. They are widget class records, and are only used in order to subclass widgets. The subclassing of Motif widgets is not supported by the SCD.

Rationale:

The class records for Motif widgets are of interest only for subclassing of these widgets. Programs that use these widgets without subclassing them need not and must not reference these symbols. Programs must also not use these symbols as external symbols for any other purpose because this will interfere with the Widget Library's use of them.

Programs that do subclass Motif widgets do not conform to the SCD. This is because the authors of the SCD cannot guarantee upward compatibility of widgets from one release to the next. The size of each widget class record is subject to being changed by the owner(s) of the code. Widget subclass records depend on the size of the class record of the superordinate class. They will be rendered invalid at the binary interface level if the size of the superordinate class record changes.

Table 10-11: Contents of libXm (1 of 2)

XmActivateProtocol	XmCreatePulldownMenu	XmFontListGetNextFont
XmAddProtocolCallback	XmCreatePushButton	XmFontListInitFontContext
XmAddProtocols	XmCreatePushButtonGadget	XmFontListNextEntry
XmAddTabGroup	XmCreateQuestionDialog	XmFontListRemoveEntry
XmCascadeButtonGadgetHighlight	XmCreateRadioButton	XmGetAtomName
XmCascadeButtonHighlight	XmCreateRowColumn	XmGetColorCalculation
XmChangeColor	XmCreateScale	XmGetColors
XmClipboardCancelCopy	XmCreateScrollBar	XmGetDestination
XmClipboardCopy	XmCreateScrolledList	XmGetDragContext
XmClipboardCopyByName	XmCreateScrolledText	XmGetFocusWidget
XmClipboardEndCopy	XmCreateScrolledWindow	XmGetMenuCursor
XmClipboardEndRetrieve	XmCreateSelectionBox	XmGetPixmap
XmClipboardInquireCount	XmCreateSelectionDialog	XmGetPixmapByDepth
XmClipboardInquireFormat	XmCreateSeparator	XmGetPostedFromWidget
XmClipboardInquireLength	XmCreateSeparatorGadget	XmGetSecondaryResourceData
XmClipboardInquirePendingItems	XmCreateSimpleCheckBox	XmGetTabGroup
XmClipboardLock	XmCreateSimpleMenuBar	XmGetTearOffControl
XmClipboardRegisterFormat	XmCreateSimpleOptionMenu	XmGetVisibility
XmClipboardRetrieve	XmCreateSimplePopupMenu	XmGetXmDisplay
XmClipboardStartCopy	XmCreateSimplePulldownMenu	XmGetXmScreen
XmClipboardStartRetrieve	XmCreateSimpleRadioButton	XmInstallImage
XmClipboardUndoCopy	XmCreateTemplateDialog	XmInternAtom
XmClipboardUnlock	XmCreateText	XmIsMotifWMRunning
XmClipboardWithdrawFormat	XmCreateTextField	XmIsTraversable
XmCommandAppendValue	XmCreateToggleButton	XmListAddItem
XmCommandError	XmCreateToggleButtonGadget	XmListAddItems
XmCommandGetChild	XmCreateWarningDialog	XmListAddItemsUnselected
XmCommandSetValue	XmCreateWorkArea	XmListAddItemUnselected
XmConvertUnits	XmCreateWorkingDialog	XmListDeleteAllItems
XmCreateArrowButton	XmCvtCTToXmString	XmListDeleteItem
XmCreateArrowButtonGadget	XmCvtStringToUnitType	XmListDeleteItems
XmCreateBulletinBoard	XmCvtXmStringToCT	XmListDeleteItemsPos
XmCreateBulletinBoardDialog	XmDeactivateProtocol	XmListDeletePos
XmCreateCascadeButton	XmDestroyPixmap	XmListDeletePositions
XmCreateCascadeButtonGadget	XmDragCancel	XmListDeselectAllItems
XmCreateCommand	XmDragStart	XmListDeselectItem
XmCreateDialogShell	XmDropSiteConfigureStackingOrder	XmListDeselectPos
XmCreateDragIcon	XmDropSiteEndUpdate	XmListGetKbdItemPos
XmCreateDrawingArea	XmDropSiteQueryStackingOrder	XmListGetMatchPos
XmCreateDrawnButton	XmDropSiteRegister	XmListGetSelectedPos
XmCreateErrorDialog	XmDropSiteRetrieve	XmListItemExists
XmCreateFileSelectionBox	XmDropSiteStartUpdate	XmListItemPos
XmCreateFileSelectionDialog	XmDropSiteUnregister	XmListPosSelected
XmCreateForm	XmDropSiteUpdate	XmListPosToBounds
XmCreateFormDialog	XmDropTransferAdd	XmListReplaceItems
XmCreateFrame	XmDropTransferStart	XmListReplaceItemsPos
XmCreateInformationDialog	XmFileSelectionBoxGetChild	XmListReplaceItemsPosUnselected
XmCreateLabel	XmFileSelectionDoSearch	XmListReplaceItemsUnselected
XmCreateLabelGadget	XmFontListAdd	XmListReplacePositions
XmCreateList	XmFontListAppendEntry	XmListSelectItem
XmCreateMainWindow	XmFontListCopy	XmListSelectPos
XmCreateMenuBar	XmFontListCreate	XmListSetAddMode
XmCreateMenuShell	XmFontListEntryCreate	XmListSetBottomItem
XmCreateMessageBox	XmFontListEntryFree	XmListSetBottomPos
XmCreateMessageDialog	XmFontListEntryGetFont	XmListSetHorizPos
XmCreateOptionMenu	XmFontListEntryGetTag	XmListSetItem
XmCreatePanedWindow	XmFontListEntryLoad	XmListSetKbdItemPos
XmCreatePopupMenu	XmFontListFree	XmListSetPos
XmCreatePromptDialog	XmFontListFreeFontContext	XmListUpdateSelectedList

Table 10-11: Contents of libXm (2 of 2)

XmListYToPos	XmStringGetLtoR	XmTextGetBaseLine
XmMainWindowSep1	XmStringGetNextComponent	XmTextGetBaseline
XmMainWindowSep2	XmStringGetNextSegment	XmTextGetEditable
XmMainWindowSep3	XmStringHasSubstring	XmTextGetInsertionPosition
XmMainWindowSetAreas	XmStringHeight	XmTextGetLastPosition
XmMapSegmentEncoding	XmStringInitContext	XmTextGetMaxLength
XmMenuPosition	XmStringLength	XmTextGetSelection
XmMessageBoxGetChild	XmStringLineCount	XmTextGetSelectionPosition
XmOptionButtonGadget	XmStringNConcat	XmTextGetSelectionWcs
XmOptionLabelGadget	XmStringNCopy	XmTextGetSource
XmProcessTraversal	XmStringPeekNextComponent	XmTextGetString
XmRegisterSegmentEncoding	XmStringSegmentCreate	XmTextGetStringWcs
XmRemoveProtocolCallback	XmStringSeparatorCreate	XmTextGetSubstring
XmRemoveProtocols	XmStringWidth	XmTextGetSubstringWcs
XmRemoveTabGroup	XmTargetsAreCompatible	XmTextGetTopCharacter
XmRepTypeAddReverse	XmTextClearSelection	XmTextInsert
XmRepTypeGetId	XmTextCopy	XmTextInsertWcs
XmRepTypeGetNameList	XmTextCut	XmTextPaste
XmRepTypeGetRecord	XmTextDisableRedisplay	XmTextPosToXY
XmRepTypeGetRegistered	XmTextEnableRedisplay	XmTextRemove
XmRepTypeInstallTearOffModelConverter	XmTextFieldClearSelection	XmTextReplace
XmRepTypeRegister	XmTextFieldCopy	XmTextReplaceWcs
XmRepTypeValidValue	XmTextFieldCut	XmTextScroll
XmResolveAllPartOffsets	XmTextFieldGetBaseline	XmTextSetAddMode
XmResolvePartOffsets	XmTextFieldGetEditable	XmTextSetEditable
XmScaleGetValue	XmTextFieldGetInsertionPosition	XmTextSetHighlight
XmScaleSetValue	XmTextFieldGetLastPosition	XmTextSetInsertionPosition
XmScrollBarGetValues	XmTextFieldGetMaxLength	XmTextSetMaxLength
XmScrollBarSetValues	XmTextFieldGetSelection	XmTextSetSelection
XmScrolledWindowSetAreas	XmTextFieldGetSelectionPosition	XmTextSetSource
XmScrollVisible	XmTextFieldGetSelectionWcs	XmTextSetString
XmSelectionBoxGetChild	XmTextFieldGetString	XmTextSetStringWcs
XmSetColorCalculation	XmTextFieldGetStringWcs	XmTextSetTopCharacter
XmSetFontUnit	XmTextFieldGetSubstring	XmTextShowPosition
XmSetFontUnits	XmTextFieldGetSubstringWcs	XmTextXYToPos
XmSetMenuCursor	XmTextFieldInsert	XmToggleButtonGadgetGetState
XmSetProtocolHooks	XmTextFieldInsertWcs	XmToggleButtonGadgetSetState
XmStringBaseline	XmTextFieldPaste	XmToggleButtonGetState
XmStringByteCompare	XmTextFieldPosToXY	XmToggleButtonSetState
XmStringCompare	XmTextFieldRemove	XmTrackingEvent
XmStringConcat	XmTextFieldReplace	XmTrackingLocate
XmStringCopy	XmTextFieldReplaceWcs	XmTranslateKey
XmStringCreate	XmTextFieldSetAddMode	XmUninstallImage
XmStringCreateLocalized	XmTextFieldSetEditable	XmUpdateDisplay
XmStringCreateLtoR	XmTextFieldSetHighlight	XmVaCreateSimpleCheckBox
XmStringCreateSimple	XmTextFieldSetInsertionPosition	XmVaCreateSimpleMenuBar
XmStringDirectionCreate	XmTextFieldSetMaxLength	XmVaCreateSimpleOptionsMenu
XmStringDraw	XmTextFieldSetSelection	XmVaCreateSimplePopupMenu
XmStringDrawImage	XmTextFieldSetString	XmVaCreateSimplePulldownMenu
XmStringDrawUnderline	XmTextFieldSetStringWcs	XmVaCreateSimpleRadioBox
XmStringEmpty	XmTextFieldShowPosition	XmWidgetGetBaselines
XmStringExtent	XmTextFieldXYToPos	XmWidgetGetDisplayRect
XmStringFree	XmTextFindString	
XmStringFreeContext	XmTextFindStringWcs	

Table 10-12: Exported Data for Motif 1.2

vendorShellClassRec[?]	xmPushButtonGadgetClassRec[?]
vendorShellWidgetClass[0x4]	xmPushButtonGCacheObjClassRec[?]
xmArrowButtonClassRec[?]	xmPushButtonWidgetClass[0x4]
xmArrowButtonGadgetClass[0x4]	XmQmotif[0x4]
xmArrowButtonGadgetClassRec[?]	xmRowColumnClassRec[?]
xmArrowButtonWidgetClass[0x4]	xmRowColumnWidgetClass[0x4]
xmBulletinBoardClassRec[?]	xmSashClassRec[?]
xmBulletinBoardWidgetClass[0x4]	xmSashWidgetClass[0x4]
xmCascadeButtonClassRec[?]	xmScaleClassRec[?]
xmCascadeButtonGadgetClass[0x4]	xmScaleWidgetClass[0x4]
xmCascadeButtonGadgetClassRec[?]	xmScreenClass[0x4]
xmCascadeButtonGCacheObjClassRec[?]	xmScreenClassRec[?]
xmCascadeButtonWidgetClass[0x4]	xmScreenObjectClass[0x4]
xmCommandClassRec[?]	xmScrollBarClassRec[?]
xmCommandWidgetClass[0x4]	xmScrollBarWidgetClass[0x4]
xmDesktopClass[0x4]	xmScrolledWindowClassRec[?]
xmDesktopClassRec[?]	xmScrolledWindowWidgetClass[0x4]
xmDesktopObjectClass[0x4]	xmSelectionBoxClassRec[?]
xmDialogShellClassRec[?]	xmSelectionBoxWidgetClass[0x4]
xmDialogShellExtClassRec[?]	xmSeparatorClassRec[?]
xmDialogShellExtObjectClass[0x4]	xmSeparatorGadgetClass[0x4]
xmDialogShellWidgetClass[0x4]	xmSeparatorGadgetClassRec[?]
xmDisplayClass[0x4]	xmSeparatorGCacheObjClassRec[?]
xmDisplayClassRec[?]	xmSeparatorWidgetClass[0x4]
xmDisplayObjectClass[0x4]	xmShellExtClassRec[?]
xmDragContextClass[0x4]	xmShellExtObjectClass[0x4]
xmDragContextClassRec[?]	xmTearOffButtonClassRec[?]
xmDragIconClassRec[?]	xmTearOffButtonWidgetClass[0x4]
xmDragIconObjectClass[0x4]	xmTextClassRec[?]
xmDragOverShellClassRec[?]	xmTextFieldClassRec[?]
xmDragOverShellWidgetClass[0x4]	xmTextFieldWidgetClass[0x4]
xmDrawingAreaClassRec[?]	xmTextWidgetClass[0x4]
xmDrawingAreaWidgetClass[0x4]	xmToggleButtonClassRec[?]
xmDrawnButtonClassRec[?]	xmToggleButtonGadgetClass[0x4]
xmDrawnButtonWidgetClass[0x4]	xmToggleButtonGadgetClassRec[?]
xmDropSiteManagerClassRec[?]	xmToggleButtonGCacheObjClassRec[?]
xmDropSiteManagerObjectClass[0x4]	xmToggleButtonWidgetClass[0x4]
xmDropTransferClassRec[?]	xmVendorShellExtClassRec[?]
xmDropTransferObjectClass[0x4]	xmVendorShellExtObjectClass[0x4]
xmExtClassRec[?]	xmWorldClass[0x4]
xmExtObjectClass[0x4]	xmWorldClassRec[?]
xmFileSelectionBoxClassRec[?]	xmWorldObjectClass[0x4]
xmFileSelectionBoxWidgetClass[0x4]	
xmFormClassRec[?]	
xmFormWidgetClass[0x4]	
xmFrameClassRec[?]	
xmFrameWidgetClass[0x4]	
xmGadgetClass[0x4]	
xmGadgetClassRec[?]	
xmLabelClassRec[?]	
xmLabelGadgetClass[0x4]	
xmLabelGadgetClassRec[?]	
xmLabelGCacheObjClassRec[?]	
xmLabelWidgetClass[0x4]	
xmListClassRec[?]	
xmListWidgetClass[0x4]	
xmMainWindowClassRec[?]	
xmMainWindowWidgetClass[0x4]	
xmManagerClassRec[?]	
xmManagerWidgetClass[0x4]	
xmMenuShellClassRec[?]	
xmMenuShellWidgetClass[0x4]	
xmMessageBoxClassRec[?]	
xmMessageBoxWidgetClass[0x4]	
xmPanedWindowClassRec[?]	
xmPanedWindowWidgetClass[0x4]	
xmPrimitiveClassRec[?]	
xmPrimitiveWidgetClass[0x4]	
xmProtocolClassRec[?]	
xmProtocolObjectClass[0x4]	
xmPushButtonClassRec[?]	
xmPushButtonGadgetClass[0x4]	

Table 10-13: libMrm Contents

MrmCloseHierarchy
MrmFetchBitmapLiteral
MrmFetchColorLiteral
MrmFetchIconLiteral
MrmFetchLiteral
MrmFetchSetValues
MrmFetchWidget
MrmFetchWidgetOverride
MrmInitialize
MrmOpenHierarchy
MrmOpenHierarchyPerDisplay
MrmRegisterClass
MrmRegisterNames
MrmRegisterNamesInHierarchy

Figure 10-29. Motif 1.2 Data Structures from ArrowB.h

```
typedef struct _XmArrowButtonClassRec      * XmArrowButtonWidgetClass;
typedef struct _XmArrowButtonRec          * XmArrowButtonWidget;
```

Figure 10-30. Motif 1.2 Data Structures from ArrowBG.h

```
typedef struct _XmArrowButtonGadgetClassRec * XmArrowButtonGadgetClass;
typedef struct _XmArrowButtonGadgetRec     * XmArrowButtonGadget;
```

Figure 10-31. Motif 1.2 Data Structures from BulletinB.h

```
typedef struct _XmBulletinBoardClassRec    * XmBulletinBoardWidgetClass;
typedef struct _XmBulletinBoardRec        * XmBulletinBoardWidget;
```

Figure 10-32. Motif 1.2 Data Structures from CascadeB.h

```
typedef struct _XmCascadeButtonRec         * XmCascadeButtonWidget;
typedef struct _XmCascadeButtonClassRec    * XmCascadeButtonWidgetClass;
```

Figure 10-33. Motif 1.2 Data Structures from CascadeBG.h

```
typedef struct _XmCascadeButtonGadgetClassRec * XmCascadeButtonGadgetClass;
typedef struct _XmCascadeButtonGadgetRec     * XmCascadeButtonGadget;
typedef struct _XmCascadeButtonGCacheObjRec  * XmCascadeButtonGCacheObject;
```

Figure 10-34. Motif 1.2 Data Structures from Command.h

```
typedef struct _XmCommandClassRec          * XmCommandWidgetClass;
typedef struct _XmCommandRec              * XmCommandWidget;
```

Figure 10-35. Motif 1.2 Manifest Constants and Data Structures from CutPaste.h

```
#define XmClipboardFail          0
#define XmClipboardSuccess      1
#define XmClipboardTruncate     2
#define XmClipboardLocked       4
#define XmClipboardBadFormat    5
#define XmClipboardNoData       6
#define ClipboardFail           0
#define ClipboardSuccess        1
#define ClipboardTruncate      2
#define ClipboardLocked         4
#define ClipboardBadFormat     5
#define ClipboardNoData        6
typedef struct {long DataId; long PrivateId;} XmClipboardPendingRec, *XmClipboardPendingList;
```

Figure 10-36. Motif 1.2 Data Structures from DialogS.h

```
typedef struct _XmDialogShellClassRec * XmDialogShellWidgetClass;
typedef struct _XmDialogShellRec * XmDialogShellWidget;
```

Figure 10-37. Motif 1.2 Manifest Constants and Data Structures from Display.h

```
enum {
    XmDRAG_NONE,                XmDRAG_DROP_ONLY,
    XmDRAG_PREFER_PREREGISTER, XmDRAG_PREREGISTER,
    XmDRAG_PREFER_DYNAMIC,     XmDRAG_DYNAMIC,
    XmDRAG_PREFER_RECEIVER
};
typedef struct _XmDisplayRec *XmDisplay;
typedef struct _XmDisplayClassRec *XmDisplayClass;
```

Figure 10-38. Motif 1.2 Manifest Constants and Data Structures from DragC.h

```
#define XmDROP_MOVE                (1L << 0)
#define XmDROP_COPY                (1L << 1)
#define XmDROP_LINK                (1L << 2)
#define XmHELP                      2
typedef unsigned int               XmID;
#define _XA_MOTIF_DROP              "_MOTIF_DROP"
#define _XA_DRAG_FAILURE            "_MOTIF_DRAG_FAILURE"
#define _XA_DRAG_SUCCESS            "_MOTIF_DRAG_SUCCESS"
enum{ XmTOP_LEVEL_ENTER,          XmTOP_LEVEL_LEAVE,
      XmDRAG_MOTION,              XmDROP_SITE_ENTER,
      XmDROP_SITE_LEAVE,         XmDROP_START,
      XmDROP_FINISH,             XmDRAG_DROP_FINISH, XmOPERATION_CHANGED };
enum{ XmDROP,                    XmDROP_HELP, XmDROP_CANCEL, XmDROP_INTERRUPT };
#define XmDROP_NOOP 0L
enum{ XmBLEND_ALL, XmBLEND_STATE_SOURCE, XmBLEND_JUST_SOURCE, XmBLEND_NONE };
enum{ XmDROP_FAILURE,            XmDROP_SUCCESS };
enum{ XmCR_TOP_LEVEL_ENTER,      XmCR_TOP_LEVEL_LEAVE,
      XmCR_DRAG_MOTION,          XmCR_DROP_SITE_ENTER,
      XmCR_DROP_SITE_LEAVE,     XmCR_DROP_START,
      XmCR_DROP_FINISH,         XmCR_DRAG_DROP_FINISH,
      XmCR_OPERATION_CHANGED,   _XmNUMBER_DND_CB_REASONS };
typedef struct _XmDragContextClassRec *XmDragContextClass;
typedef struct _XmDragContextRec *XmDragContext;
typedef struct _XmAnyICCCallbackStruct{
    int reason;
    XEvent *event;
    Time timeStamp;
}XmAnyICCCallbackStruct, *XmAnyICCCallback;
typedef struct _XmTopLevelEnterCallbackStruct{
    int reason;
    XEvent *event;
    Time timeStamp;
    Screen *screen;
    Window window;
    Position x, y;
    unsigned char dragProtocolStyle;
    Atom iccHandle;
}XmTopLevelEnterCallbackStruct, *XmTopLevelEnterCallback;
typedef struct _XmTopLevelLeaveCallbackStruct{
    int reason;
    XEvent *event;
    Time timeStamp;
    Screen *screen;
    Window window;
}XmTopLevelLeaveCallbackStruct, *XmTopLevelLeaveCallback;

typedef struct _XmDropSiteEnterCallbackStruct{
    int reason;
    XEvent *event;
    Time timeStamp;
    unsigned char operation;
    unsigned char operations;
    unsigned char dropSiteStatus;
    Position x, y;
}XmDropSiteEnterCallbackStruct, *XmDropSiteEnterCallback;

typedef struct _XmDropSiteLeaveCallbackStruct{
    int reason;
```

```

XEvent      *event;
Time        timeStamp;
}XmDropSiteLeaveCallbackStruct, *XmDropSiteLeaveCallback;

typedef struct _XmDragMotionCallbackStruct{
    int      reason;
    XEvent   *event;
    Time     timeStamp;
    unsigned char  operation;
    unsigned char  operations;
    unsigned char  dropSiteStatus;
    Position    x, y;
}XmDragMotionCallbackStruct, *XmDragMotionCallback;

typedef struct _XmOperationChangedCallbackStruct{
    int      reason;
    XEvent   *event;
    Time     timeStamp;
    unsigned char  operation;
    unsigned char  operations;
    unsigned char  dropSiteStatus;
}XmOperationChangedCallbackStruct, *XmOperationChangedCallback;
typedef struct _XmDropStartCallbackStruct{
    int      reason;
    XEvent   *event;
    Time     timeStamp;
    unsigned char  operation;
    unsigned char  operations;
    unsigned char  dropSiteStatus;
    unsigned char  dropAction;
    Position    x, y;
    Window      window;
    Atom        iccHandle;
}XmDropStartCallbackStruct, *XmDropStartCallback;
typedef struct _XmDropFinishCallbackStruct{
    int      reason;
    XEvent   *event;
    Time     timeStamp;
    unsigned char  operation;
    unsigned char  operations;
    unsigned char  dropSiteStatus;
    unsigned char  dropAction;
    unsigned char  completionStatus;
}XmDropFinishCallbackStruct, *XmDropFinishCallback;
typedef struct _XmDragDropFinishCallbackStruct{int reason;XEvent *event; Time timeStamp; }XmDragDropFinishCallbackStruct, *XmDragDropFinishCallback;

```

Figure 10-39. Motif 1.2 Manifest Constants and Data Structures from DragIcon.h

```

enum {
    XmATTACH_NORTH_WEST,   XmATTACH_NORTH,
    XmATTACH_NORTH_EAST,  XmATTACH_EAST,
    XmATTACH_SOUTH_EAST,   XmATTACH_SOUTH,
    XmATTACH_SOUTH_WEST,  XmATTACH_WEST,
    XmATTACH_CENTER,      XmATTACH_HOT
};
typedef struct _XmDragIconRec *XmDragIconObject;
typedef struct _XmDragIconClassRec *XmDragIconObjectClass;

```

Figure 10-40. Motif 1.2 Data Structures from DragOverS.h

```

typedef struct _XmDragOverShellRec *XmDragOverShellWidget;
typedef struct _XmDragOverShellClassRec *XmDragOverShellWidgetClass;

```

Figure 10-41. Motif 1.2 Data Structures from DrawingA.h

```

typedef struct _XmDrawingAreaClassRec *XmDrawingAreaWidgetClass;
typedef struct _XmDrawingAreaRec *XmDrawingAreaWidget;

```

Figure 10-42. Motif 1.2 Data Structures from DrawnB.h

```

typedef struct _XmDrawnButtonClassRec *XmDrawnButtonWidgetClass;
typedef struct _XmDrawnButtonRec *XmDrawnButtonWidget;

```

Figure 10-43. Motif 1.2 Manifest Constants and Data Structures from DropSMgr.h

```

#define XmCR_DROP_SITE_LEAVE_MESSAGE 1
#define XmCR_DROP_SITE_ENTER_MESSAGE 2
#define XmCR_DROP_SITE_MOTION_MESSAGE 3
#define XmCR_DROP_MESSAGE 4
#define XmNO_DROP_SITE 1
#define XmINVALID_DROP_SITE 2
#define XmVALID_DROP_SITE 3
enum { XmDRAG_UNDER_NONE, XmDRAG_UNDER_PIXMAP,
      XmDRAG_UNDER_SHADOW_IN, XmDRAG_UNDER_SHADOW_OUT,
      XmDRAG_UNDER_HIGHLIGHT };
enum { XmDROP_SITE_SIMPLE, XmDROP_SITE_COMPOSITE,
      XmDROP_SITE_SIMPLE_CLIP_ONLY = 128,
      XmDROP_SITE_COMPOSITE_CLIP_ONLY };
enum { XmABOVE, XmBELOW };
enum { XmDROP_SITE_ACTIVE, XmDROP_SITE_INACTIVE };
typedef struct _XmDragProcCallbackStruct {
    int reason;
    XEvent * event;
    Time timeStamp;
    Widget dragContext;
    Position x, y;
    unsigned char dropSiteStatus;
    unsigned char operation;
    unsigned char operations;
    Boolean animate;
} XmDragProcCallbackStruct, * XmDragProcCallback;

typedef struct _XmDropProcCallbackStruct {
    int reason;
    XEvent * event;
    Time timeStamp;
    Widget dragContext;
    Position x, y;
    unsigned char dropSiteStatus;
    unsigned char operation;
    unsigned char operations;
    unsigned char dropAction;
} XmDropProcCallbackStruct, * XmDropProcCallback;

typedef struct _XmDropSiteVisualsRec {
    Pixel background;
    Pixel foreground;
    Pixel topShadowColor;
    Pixmap topShadowPixmap;
    Pixel bottomShadowColor;
    Pixmap bottomShadowPixmap;
    Dimension shadowThickness;
    Pixel highlightColor;
    Pixmap highlightPixmap;
    Dimension highlightThickness;
    Dimension borderWidth;
} XmDropSiteVisualsRec, * XmDropSiteVisuals;

/* DropSite Widget */

typedef struct _XmDropSiteManagerClassRec *XmDropSiteManagerObjectClass;
typedef struct _XmDropSiteManagerRec *XmDropSiteManagerObject;

```

Figure 10-44. Motif 1.2 Manifest Constants and Data Structures from DropTrans.h

```

#define XmTRANSFER_FAILURE 0
#define XmTRANSFER_SUCCESS 1

typedef struct _XmDropTransferClassRec * XmDropTransferObjectClass;
typedef struct _XmDropTransferRec * XmDropTransferObject;

typedef struct _XmDropTransferEntryRec {
    XtPointer client_data;
    Atom target;
} XmDropTransferEntryRec, * XmDropTransferEntry;

```

Figure 10-45. Motif 1.2 Data Structures from FileSB.h

```
typedef struct _XmFileSelectionBoxClassRec * XmFileSelectionBoxWidgetClass;
typedef struct _XmFileSelectionBoxRec * XmFileSelectionBoxWidget;
```

Figure 10-46. Motif 1.2 Data Structures from Form.h

```
typedef struct _XmFormClassRec * XmFormWidgetClass;
typedef struct _XmFormRec * XmFormWidget;
```

Figure 10-47. Motif 1.2 Data Structures from Frame.h

```
typedef struct _XmFrameClassRec * XmFrameWidgetClass;
typedef struct _XmFrameRec * XmFrameWidget;
```

Figure 10-48. Motif 1.2 Data Structures from Label.h

```
typedef struct _XmLabelClassRec * XmLabelWidgetClass;
typedef struct _XmLabelRec * XmLabelWidget;
```

Figure 10-49. Motif 1.2 Data Structures from LabelG.h

```
typedef struct _XmLabelGadgetClassRec * XmLabelGadgetClass;
typedef struct _XmLabelGadgetRec * XmLabelGadget;
typedef struct _XmLabelGCacheObjRec * XmLabelGCacheObject;
```

Figure 10-50. Motif 1.2 Manifest Constants and Data Structures from List.h

```
#define XmINITIAL 0
#define XmADDITION 1
#define XmMODIFICATION 2

typedef struct _XmListClassRec * XmListWidgetClass;
typedef struct _XmListRec * XmListWidget;
```

Figure 10-51. Motif 1.2 Data Structures from MainW.h

```
typedef struct _XmMainWindowClassRec * XmMainWindowWidgetClass;
typedef struct _XmMainWindowRec * XmMainWindowWidget;
```

Figure 10-52. Motif 1.2 Data Structures from MenuShell.h

```
typedef struct _XmMenuShellClassRec * XmMenuShellWidgetClass;
typedef struct _XmMenuShellWidgetRec * XmMenuShellWidget;
```

Figure 10-53. Motif 1.2 Data Structures from MessageB.h

```
typedef struct _XmMessageBoxClassRec * XmMessageBoxWidgetClass;
typedef struct _XmMessageBoxRec * XmMessageBoxWidget;
```

Figure 10-54. Motif 1.2 Manifest Constants and Data Structures from MrmPublic.h

```
#define MrmSUCCESS 1
#define MrmCREATE_NEW 3
#define MrmINDEX_RETRY 5 /* Retry on entering index required */
#define MrmINDEX_GT 7 /* Index orders greater-than entry */
#define MrmINDEX_LT 9 /* Index orders less-than entry */
#define MrmPARTIAL_SUCCESS 11 /* operation partly succeeded */
#define MrmFAILURE 0
#define MrmNOT_FOUND 2
#define MrmEXISTS 4
#define MrmNUL_GROUP 6
#define MrmNUL_TYPE 8
#define MrmWRONG_GROUP 10
#define MrmWRONG_TYPE 12
#define MrmOUT_OF_RANGE 14 /* Record number too big */
#define MrmBAD_RECORD 16 /* Record number wrong type */
#define MrmNULL_DATA 18 /* No data for entry */
#define MrmBAD_DATA_INDEX 20 /* Data index in RID out of range */
```

```

#define MrmBAD_ORDER 22 /* Bad ordering specifier */
#define MrmBAD_CONTEXT 24 /* Invalid Mrm context */
#define MrmNOT_VALID 26 /* Validation failure */
#define MrmBAD_BTREE 28 /* GT/LT pointer error in BTree */
#define MrmBAD_WIDGET_REC 30 /* Validation failure on widget record */
#define MrmBAD_CLASS_TYPE 32 /* Class type not a valid Mrmwc... value */
#define MrmNO_CLASS_NAME 34 /* User class name is null */
#define MrmTOO_MANY 36 /* Too many entries requested in some list */
#define MrmBAD_IF_MODULE 38 /* invalid interface module */
#define MrmNULL_DESC 40 /* Arglist or children descriptor null */
#define MrmOUT_OF_BOUNDS 42 /* Argument index out of arglist bounds */
#define MrmBAD_COMPRESS 44 /* Invalid compression code */
#define MrmBAD_ARG_TYPE 46 /* Invalid type, not in RGMrType... */
#define MrmNOT_IMP 48 /* Not yet implemented */
#define MrmNULL_INDEX 50 /* empty index string */
#define MrmBAD_KEY_TYPE 52 /* key must be MrmrIndex or MrmrRID */
#define MrmBAD_CALLBACK 54 /* Invalid callback descriptor */
#define MrmNULL_ROUTINE 56 /* Empty callback routine name string */
#define MrmVEC_TOO_BIG 58 /* too many elements in vector */
#define MrmBAD_HIERARCHY 60 /* invalid Mrm file hierarchy */
#define MrmBAD_CLASS_CODE 62 /* Class code not found in Mrmwc... */
#define MrmDISPLAY_NOT_OPENED 63 /* Display not yet created */
#define MrmEOF 64 /* End of file */
#define MrmUNRESOLVED_REFS 65 /* Unresolved widget refs in callback*/
#define MrmNcreateCallback "createCallback"
#define MrmCR_CREATE XmCR_CREATE
#define MrmwcUnknown 1
#define MrmRtypeMin 1
#define MrmRtypeInteger 1 /* int */
#define MrmRtypeBoolean 2
#define MrmRtypeChar8 3 /* a nul-terminated string */
#define MrmRtypeChar8Vector 4 /* a vector of char_8 strings */
#define MrmRtypeCString 5 /* a compound string (DDIS) */
#define MrmRtypeCStringVector 6 /* a vector of compound strings */
#define MrmRtypeFloat 7 /* 8 = TypeCompressed now unused */
#define MrmRtypeCallback 9 /* code for a callback descriptor */
#define MrmRtypePixmapImage 10 /* Pixmap in image form */
#define MrmRtypePixmapDDIF 11 /* Pixmap in DDIF form */
#define MrmRtypeResource 12 /* Mrm resource descriptor */
#define MrmRtypeNull 13 /* no value given */
#define MrmRtypeAddrName 14 /* nul-terminated string to be interpreted as runtime address */
#define MrmRtypeIconImage 15 /* icon image */
#define MrmRtypeFont 16 /* Mrm font structure */
#define MrmRtypeFontList 17 /* Mrm font list */
#define MrmRtypeColor 18 /* Mrm color descriptor */
#define MrmRtypeColorTable 19 /* Mrm color table */
#define MrmRtypeAny 20 /* Any is allowed in UID file */
#define MrmRtypeTransTable 21 /* Translation table (ASCIZ string) */
#define MrmRtypeClassRecName 22 /* class record name (ASCIZ string) */
#define MrmRtypeIntegerVector 23 /* a vector of integers */
#define MrmRtypeXBitmapFile 24 /* X bitmap file to make pixmap with */
#define MrmRtypeCountedVector 25 /* vector with associated count */
#define MrmRtypeKeysym 26 /* X keysym data type */
#define MrmRtypeSingleFloat 27 /* single float data type */
#define MrmRtypeWideCharacter 28 /* wide_character string type */
#define MrmRtypeFontSet 29
#define MrmRtypeMax 30
typedef short int MrmCode ; /* Used for codes, e.g. Mrmcr... */
typedef unsigned char MrmSCode ; /* Short code for small ranges */
typedef unsigned short int MrmOffset ; /* Used for offsets in records */
typedef short int MrmType ; /* Used for types, e.g. MrmrType... */
typedef unsigned short int MrmSize ; /* For size fields */
typedef short int MrmCount ; /* For counter fields */
typedef unsigned char MrmFlag ; /* flag fields */
typedef long int MrmResource_id ; /* Resource id in IDB files */
typedef short int MrmGroup ; /* For Mrm resource groups */
#define MrmMaxResourceSize 65535 /* (2)16 - 1 */
#define MrmOsOpenParamVersion 1
typedef struct { Cardinal version; char *default_fname;
    union { unsigned long related_name; Boolean clobber_flg; } nam_flg;
    Display *display;
} MrmOsOpenParam, *MrmOsOpenParamPtr ;
typedef struct MrmHierarchyDescStruct *MrmHierarchy;
typedef struct {String name ; XtPointer value ;} MRMRegisterArg, MrmRegisterArg, *MrmRegisterArglist ;
#define URMwcUnknown 1

```

Figure 10-55. Motif 1.2 Manifest Constants and Data Structures from MwmUtil.h

```

typedef struct {long flags; long functions; long decorations; int input_mode; long status;} MotifWmHints;

```

```

typedef MotifWmHints
#define MWM_HINTS_FUNCTIONS (1L << 0)
#define MWM_HINTS_DECORATIONS (1L << 1)
#define MWM_HINTS_INPUT_MODE (1L << 2)
#define MWM_HINTS_STATUS (1L << 3)
#define MWM_FUNC_ALL (1L << 0)
#define MWM_FUNC_RESIZE (1L << 1)
#define MWM_FUNC_MOVE (1L << 2)
#define MWM_FUNC_MINIMIZE (1L << 3)
#define MWM_FUNC_MAXIMIZE (1L << 4)
#define MWM_FUNC_CLOSE (1L << 5)
#define MWM_DECOR_ALL (1L << 0)
#define MWM_DECOR_BORDER (1L << 1)
#define MWM_DECOR_RESIZEH (1L << 2)
#define MWM_DECOR_TITLE (1L << 3)
#define MWM_DECOR_MENU (1L << 4)
#define MWM_DECOR_MINIMIZE (1L << 5)
#define MWM_DECOR_MAXIMIZE (1L << 6)
#define MWM_INPUT_MODELESS 0
#define MWM_INPUT_PRIMARY_APPLICATION_MODAL 1
#define MWM_INPUT_SYSTEM_MODAL 2
#define MWM_INPUT_FULL_APPLICATION_MODAL 3
#define MWM_TEAROFF_WINDOW (1L << 0)
#define MWM_INPUT_APPLICATION_MODAL MWM_INPUT_PRIMARY_APPLICATION_MODAL
typedef struct{ long flags; Window wm_window;}
typedef MotifWmInfo
#define MWM_INFO_STARTUP_STANDARD (1L << 0)
#define MWM_INFO_STARTUP_CUSTOM (1L << 1)
typedef struct{CARD32 flags; CARD32 functions; CARD32 decorations; INT32 inputMode; CARD32 status;} PropMotifWmHints;
typedef PropMotifWmHints
#define PROP_MOTIF_WM_HINTS_ELEMENTS 5
#define PROP_MWM_HINTS_ELEMENTS PROP_MOTIF_WM_HINTS_ELEMENTS
#define _XA_MOTIF_WM_HINTS "_XA_MOTIF_WM_HINTS"
#define _XA_MWM_HINTS "_XA_MOTIF_WM_HINTS"
#define _XA_MOTIF_WM_MESSAGES "_XA_MOTIF_WM_MESSAGES"
#define _XA_MWM_MESSAGES "_XA_MOTIF_WM_MESSAGES"
#define _XA_MOTIF_WM_OFFSET "_XA_MOTIF_WM_MESSAGES"
#define _XA_MOTIF_WM_MENU "_XA_MOTIF_WM_MESSAGES"
#define _XA_MWM_MENU "_XA_MOTIF_WM_MESSAGES"
typedef struct { CARD32 flags; CARD32 wmWindow;} PropMotifWmInfo;
typedef PropMotifWmInfo
#define PROP_MOTIF_WM_INFO_ELEMENTS 2
#define PROP_MWM_INFO_ELEMENTS PROP_MOTIF_WM_INFO_ELEMENTS
#define _XA_MOTIF_WM_INFO "_XA_MOTIF_WM_INFO"
#define _XA_MWM_INFO "_XA_MOTIF_WM_INFO"
#define _XA_MOTIF_BINDINGS "_XA_MOTIF_WM_INFO"

```

Figure 10-56. Motif 1.2 Data Structures from PanedW.h

```

typedef struct _XmPanedWindowClassRec *XmPanedWindowWidgetClass;
typedef struct _XmPanedWindowRec *XmPanedWindowWidget;

```

Figure 10-57. Motif 1.2 Data Structures from PushB.h

```

typedef struct _XmPushButtonClassRec *XmPushButtonWidgetClass;
typedef struct _XmPushButtonRec *XmPushButtonWidget;

```

Figure 10-58. Motif 1.2 Data Structures from PushBG.h

```

typedef struct _XmPushButtonGadgetClassRec *XmPushButtonGadgetClass;
typedef struct _XmPushButtonGadgetRec *XmPushButtonGadget;
typedef struct _XmPushButtonGCacheObjRec *XmPushButtonGCacheObject;

```

Figure 10-59. Motif 1.2 Manifest Constants and Data Structures from RepType.h

```

#define XmREP_TYPE_INVALID 0x1FFF
typedef unsigned short XmRepTypeId ;
typedef struct{
    String rep_type_name ;
    String *value_names ;
    unsigned char *values ;
    unsigned char num_values ;
    Boolean reverse_installed ;
    XmRepTypeId rep_type_id ;
}XmRepTypeEntryRec, *XmRepTypeEntry, XmRepTypeListRec, *XmRepTypeList ;

```

Figure 10-60. Motif 1.2 Data Structures from RowColumn.h

```
typedef struct _XmRowColumnClassRec * XmRowColumnWidgetClass;
typedef struct _XmRowColumnRec * XmRowColumnWidget;
```

Figure 10-61. Motif 1.2 Data Structures from Scale.h

```
/* fast XtIsSubclass define */
typedef struct _XmScaleClassRec * XmScaleWidgetClass;
typedef struct _XmScaleRec * XmScaleWidget;
```

Figure 10-62. Motif 1.2 Data Structures from Screen.h

```
typedef struct _XmScreenRec * XmScreen;
typedef struct _XmScreenClassRec * XmScreenClass;
```

Figure 10-63. Motif 1.2 Data Structures from ScrollBar.h

```
typedef struct _XmScrollBarClassRec * XmScrollBarWidgetClass;
typedef struct _XmScrollBarRec * XmScrollBarWidget;
```

Figure 10-64. Motif 1.2 Data Structures from ScrolledW.h

```
typedef struct _XmScrolledWindowClassRec * XmScrolledWindowWidgetClass;
typedef struct _XmScrolledWindowRec * XmScrolledWindowWidget;
```

Figure 10-65. Motif 1.2 Data Structures from SelectioB.h

```
typedef struct _XmSelectionBoxClassRec * XmSelectionBoxWidgetClass;
typedef struct _XmSelectionBoxRec * XmSelectionBoxWidget;
```

Figure 10-66. Motif 1.2 Data Structures from SeparatoG.h

```
typedef struct _XmSeparatorGadgetClassRec * XmSeparatorGadgetClass;
typedef struct _XmSeparatorGadgetRec * XmSeparatorGadget;
typedef struct _XmSeparatorGCacheObjRec * XmSeparatorGCacheObject;
```

Figure 10-67. Motif 1.2 Data Structures from Separator.h

```
typedef struct _XmSeparatorClassRec * XmSeparatorWidgetClass;
typedef struct _XmSeparatorRec * XmSeparatorWidget;
```

Figure 10-68. Motif 1.2 Data Structures from Text.h

```
typedef struct _XmTextSourceRec * XmTextSource;
typedef struct _XmTextClassRec * XmTextWidgetClass;
typedef struct _XmTextRec * XmTextWidget;
```

Figure 10-69. Motif 1.2 Data Structures from TextF.h

```
typedef struct _XmTextFieldClassRec * XmTextFieldWidgetClass;
typedef struct _XmTextFieldRec * XmTextFieldWidget;
```

Figure 10-70. Motif 1.2 Data Structures from ToggleB.h

```
typedef struct _XmToggleButtonClassRec * XmToggleButtonWidgetClass;
typedef struct _XmToggleButtonRec * XmToggleButtonWidget;
```

Figure 10-71. Motif 1.2 Data Structures from ToggleBG.h

```
typedef struct _XmToggleButtonGadgetClassRec * XmToggleButtonGadgetClass;
typedef struct _XmToggleButtonGadgetRec * XmToggleButtonGadget;
typedef struct _XmToggleButtonGCacheObjRec * XmToggleButtonGCacheObject;
```

Figure 10-72. Motif 1.2 Data Structures from VendorS.h

```
typedef struct _XmVendorShellRec *XmVendorShellWidget;
typedef struct _XmVendorShellClassRec *XmVendorShellWidgetClass;
```

Figure 10-73. Motif 1.2 Manifest Constants from VirtKeys.h

```
#define _OSF_Keysyms
#define osfXK_BackSpace          0x1004FF08
#define osfXK_Insert             0x1004FF63
#define osfXK_Delete            0x1004FFF0
#define osfXK_Copy              0x1004FF02
#define osfXK_Cut               0x1004FF03
#define osfXK_Paste             0x1004FF04
#define osfXK_AddMode           0x1004FF31
#define osfXK_PrimaryPaste      0x1004FF32
#define osfXK_QuickPaste       0x1004FF33
#define osfXK_PageLeft         0x1004FF40
#define osfXK_PageUp           0x1004FF41
#define osfXK_PageDown        0x1004FF42
#define osfXK_PageRight        0x1004FF43
#define osfXK_EndLine          0x1004FF57
#define osfXK_BeginLine        0x1004FF58
#define osfXK_Activate          0x1004FF44
#define osfXK_MenuBar          0x1004FF45
#define osfXK_Clear            0x1004FF0B
#define osfXK_Cancel           0x1004FF69
#define osfXK_Help             0x1004FF6A
#define osfXK_Menu             0x1004FF67
#define osfXK_Select           0x1004FF60
#define osfXK_Undo             0x1004FF65
#define osfXK_Left             0x1004FF51
#define osfXK_Up               0x1004FF52
#define osfXK_Right            0x1004FF53
#define osfXK_Down             0x1004FF54

Motif 1.2 Manifest Constants and Data Structures from Xm.h
#define XmUNSPECIFIED_PIXMAP    2
#define XmSTRING_OS_CHARSET     XmSTRING_ISO8859_1
#ifdef XmFALLBACK_CHARSET
#define XmFALLBACK_CHARSET     XmSTRING_ISO8859_1
#endif
#define XmDEFAULT_FONT          _XmSDEFAULT_FONT
#define XmDEFAULT_BACKGROUND    _XmSDEFAULT_BACKGROUND
#define XmDEFAULT_DARK_THRESHOLD 15
#define XmDEFAULT_LIGHT_THRESHOLD 77
#define XmDEFAULT_FOREGROUND_THRESHOLD 35
typedef enum{ XmFONT_IS_FONT, XmFONT_IS_FONTSET } XmFontType;
enum{ XmSTRING_DIRECTION_L_TO_R, XmSTRING_DIRECTION_R_TO_L };
#define XmSTRING_DIRECTION_DEFAULT ((XmStringDirection) 255)
typedef unsigned char           *XmString; /* opaque to outside */
typedef XmStringTable; /* opaque to outside */
typedef char                    *XmStringCharSet; /* Null term string */
typedef unsigned char           XmStringComponentType; /* component tags */
typedef unsigned char           XmStringDirection;
typedef struct _XmFontListRec    *XmFontListEntry; /* opaque to outside */
typedef struct _XmFontListRec    *XmFontList; /* opaque to outside */
typedef struct _XmStringContextRec *XmStringContext; /* opaque to outside */
typedef struct __XmStringRec      *XmString; /* opaque to outside */
typedef struct _XmStringContextRec *XmStringContext; /* opaque to outside */
typedef struct _XmFontListContextRec *XmFontContext; /* opaque to outside */
enum{ XmSTRING_COMPONENT_UNKNOWN, XmSTRING_COMPONENT_CHARSET,
      XmSTRING_COMPONENT_TEXT, XmSTRING_COMPONENT_DIRECTION,
      XmSTRING_COMPONENT_SEPARATOR, XmSTRING_COMPONENT_LOCALE_TEXT /* 6-125 reserved */ };
#define XmSTRING_COMPONENT_END ((XmStringComponentType) 126)
#define XmSTRING_COMPONENT_USER_BEGIN ((XmStringComponentType) 128) /* 128-255 are user tags */
#define XmSTRING_COMPONENT_USER_END ((XmStringComponentType) 255)
typedef struct _XmPrimitiveClassRec *XmPrimitiveWidgetClass;
typedef struct _XmPrimitiveRec *XmPrimitiveWidget;
typedef struct _XmGadgetClassRec *XmGadgetClass;
typedef struct _XmGadgetRec *XmGadget;
typedef struct _XmManagerClassRec *XmManagerWidgetClass;
typedef struct _XmManagerRec *XmManagerWidget;
enum{ XmCHANGE_ALL, XmCHANGE_NONE, XmCHANGE_WIDTH, XmCHANGE_HEIGHT };
enum{ XmPIXELS, Xm100TH_MILLIMETERS, Xm1000TH_INCHES, Xm100TH_POINTS, Xm100TH_FONT_UNITS };
enum{ XmDESTROY, XmUNMAP, XmDO_NOTHING };
enum{ XmEXPLICIT, XmPOINTER };
enum{ XmNONE, XmTAB_GROUP, XmSTICKY_TAB_GROUP, XmEXCLUSIVE_TAB_GROUP };
#define XmDYNAMIC_DEFAULT_TAB_GROUP ((XmNavigationType) 255)
enum{ /* XmNONE */ XmBELL = 1 };
```

```

enum{ XmNO_ORIENTATION,      XmVERTICAL, XmHORIZONTAL };
enum{ XmWORK_AREA, XmMENU_BAR, XmMENU_PULLDOWN, XmMENU_POPUP, XmMENU_OPTION };
enum{ XmNO_PACKING,          XmPACK_TIGHT, XmPACK_COLUMN,      XmPACK_NONE };
enum{ /* XmALIGNMENT_BASELINE_TOP, XmALIGNMENT_CENTER,
      XmALIGNMENT_BASELINE_BOTTOM, /* XmALIGNMENT_CONTENTS_TOP = 3,
      XmALIGNMENT_CONTENTS_BOTTOM  };
enum{ XmTEAR_OFF_ENABLED,    XmTEAR_OFF_DISABLED  };
enum{ XmUNPOST,             XmUNPOST_AND_REPLAY  };
enum{ XmLAST_POSITION = -1,  XmFIRST_POSITION   };
enum{ XmALIGNMENT_BEGINNING, XmALIGNMENT_CENTER,  XmALIGNMENT_END   };
enum{ XmALIGNMENT_BASELINE_TOP, /* XmALIGNMENT_CENTER, /*
      XmALIGNMENT_BASELINE_BOTTOM = 2, XmALIGNMENT_WIDGET_TOP,
      XmALIGNMENT_WIDGET_BOTTOM  };
enum{ XmFRAME_GENERIC_CHILD, XmFRAME_WORKAREA_CHILD,  XmFRAME_TITLE_CHILD  };
enum{ XmN_OF_MANY = 1,      XmONE_OF_MANY   };
enum{ XmATTACH_NONE,        XmATTACH_FORM,
      XmATTACH_OPPOSITE_FORM, XmATTACH_WIDGET,
      XmATTACH_OPPOSITE_WIDGET, XmATTACH_POSITION, XmATTACH_SELF  };
enum{ XmRESIZE_NONE,        XmRESIZE_GROW,  XmRESIZE_ANY  };
enum{ XmCR_NONE,           XmCR_HELP,
      XmCR_VALUE_CHANGED,  XmCR_INCREMENT,
      XmCR_DECREMENT,      XmCR_PAGE_INCREMENT,
      XmCR_PAGE_DECREMENT, XmCR_TO_TOP,
      XmCR_TO_BOTTOM,      XmCR_DRAG,
      XmCR_ACTIVATE,       XmCR_ARM,
      XmCR_DISARM,         XmCR_MAP = 16,
      XmCR_UNMAP,          XmCR_FOCUS,
      XmCR_LOSING_FOCUS,   XmCR_MODIFYING_TEXT_VALUE,
      XmCR_MOVING_INSERT_CURSOR, XmCR_EXECUTE,
      XmCR_SINGLE_SELECT,  XmCR_MULTIPLE_SELECT,
      XmCR_EXTENDED_SELECT, XmCR_BROWSE_SELECT,
      XmCR_DEFAULT_ACTION, XmCR_CLIPBOARD_DATA_REQUEST,
      XmCR_CLIPBOARD_DATA_DELETE, XmCR_CASCADING,
      XmCR_OK,              XmCR_CANCEL,
      XmCR_APPLY = 34,      XmCR_NO_MATCH,
      XmCR_COMMAND_ENTERED, XmCR_COMMAND_CHANGED,
      XmCR_EXPOSE,         XmCR_RESIZE,
      XmCR_INPUT,          XmCR_GAIN_PRIMARY,
      XmCR_LOSE_PRIMARY,   XmCR_CREATE,
      XmCR_TEAR_OFF_ACTIVATE, XmCR_TEAR_OFF_DEACTIVATE,
      XmCR_OBSCURED_TRAVERSAL
  };
typedef struct{ int reason; XEvent *event; } XmAnyCallbackStruct;
typedef struct{ int reason; XEvent *event; int click_count; } XmArrowButtonCallbackStruct;
typedef struct{ int reason; XEvent *event; Window window; } XmDrawingAreaCallbackStruct;
typedef struct{ int reason; XEvent *event; Window window; int click_count; } XmDrawnButtonCallbackStruct;
typedef struct{ int reason; XEvent *event; int click_count; } XmPushButtonCallbackStruct;
typedef struct{ int reason; XEvent *event; Widget widget; char *data; char *callbackstruct; } XmRowColumnCallbackStruct;
typedef struct{ int reason; XEvent *event; int value; int pixel; } XmScrollBarCallbackStruct;
typedef struct{ int reason; XEvent *event; int set; } XmToggleButtonCallbackStruct;
typedef struct{
  int reason;
  XEvent *event;
  XmString item;
  int item_length;
  int item_position;
  XmString *selected_items;
  int selected_item_count;
  int *selected_item_positions;
  char selection_type;
} XmListCallbackStruct;
typedef struct{ int reason; XEvent *event; XmString value; int length; } XmSelectionBoxCallbackStruct;
typedef struct{ int reason; XEvent *event; XmString value; int length; } XmCommandCallbackStruct;
typedef struct{
  int reason;
  XEvent *event;
  XmString value;
  int length;
  XmString mask;
  int mask_length;
  XmString dir;
  int dir_length;
  XmString pattern;
  int pattern_length;
} XmFileSelectionBoxCallbackStruct;
typedef struct{ int reason; XEvent *event; int value; } XmScaleCallbackStruct;
enum{ XmMULTICLICK_DISCARD, XmMULTICLICK_KEEP };
enum{ XmSHADOW_IN = 7, XmSHADOW_OUT };
enum{ XmARROW_UP, XmARROW_DOWN, XmARROW_LEFT, XmARROW_RIGHT };
enum{ XmNO_LINE, XmSINGLE_LINE,
      XmDOUBLE_LINE, XmSINGLE_DASHED_LINE,

```

```

XmDOUBLE_DASHED_LINE,    XmSHADOW_ETCHED_IN,
XmSHADOW_ETCHED_OUT,    XmSHADOW_ETCHED_IN_DASH,
XmSHADOW_ETCHED_OUT_DASH, XmINVALID_SEPARATOR_TYPE
};
enum{ XmPIXMAP = 1,        XmSTRING    };
enum{ XmWINDOW,          /* XmPIXMAP,*/  XmCURSOR = 2    };
enum{ XmMAX_ON_TOP, XmMAX_ON_BOTTOM, XmMAX_ON_LEFT, XmMAX_ON_RIGHT };
enum{ XmSINGLE_SELECT, XmMULTIPLE_SELECT,
XmEXTENDED_SELECT, XmBROWSE_SELECT };
enum{ XmSTATIC,          XmDYNAMIC    };
enum{ XmVARIABLE,        XmCONSTANT,   XmRESIZE_IF_POSSIBLE };
enum{ XmAUTOMATIC,      XmAPPLICATION_DEFINED };
enum{ /* XmSTATIC */     XmAS_NEEDED = 1 };
#define SW_TOP           1
#define SW_BOTTOM        0
#define SW_LEFT          2
#define SW_RIGHT         0
#define XmTOP_LEFT      (SW_TOP | SW_LEFT)
#define XmBOTTOM_LEFT   (SW_BOTTOM | SW_LEFT)
#define XmTOP_RIGHT     (SW_TOP | SW_RIGHT)
#define XmBOTTOM_RIGHT  (SW_BOTTOM | SW_RIGHT)
enum{ XmCOMMAND_ABOVE_WORKSPACE, XmCOMMAND_BELOW_WORKSPACE };
enum{ XmMULTI_LINE_EDIT, XmSINGLE_LINE_EDIT };
typedef enum{ XmTEXT_FORWARD, XmTEXT_BACKWARD } XmTextDirection;
typedef long XmTextPosition;
typedef Atom XmTextFormat;
#define XmFMT_8_BIT      ((XmTextFormat) XA_STRING) /* 8-bit text. */
#define XmFMT_16_BIT     ((XmTextFormat) 2) /* 16-bit text. */
#define FMT8BIT          XmFMT_8_BIT /* For backwards compatibility only.*/
#define FMT16BIT         XmFMT_16_BIT /* For backwards compatibility only.*/
typedef enum{
    XmSELECT_POSITION, XmSELECT_WHITESPACE,
    XmSELECT_WORD,    XmSELECT_LINE,
    XmSELECT_ALL,     XmSELECT_PARAGRAPH
} XmTextScanType;
typedef enum{
    XmHIGHLIGHT_NORMAL, XmHIGHLIGHT_SELECTED,
    XmHIGHLIGHT_SECONDARY_SELECTED
} XmHighlightMode;
typedef struct { char *ptr; int length; XmTextFormat format; } XmTextBlockRec, *XmTextBlock;
typedef struct{
    int reason;
    XEvent *event;
    Boolean doit;
    long currInsert, newInsert;
    long startPos, endPos;
    XmTextBlock text;
} XmTextVerifyCallbackStruct, *XmTextVerifyPtr;
typedef struct { wchar_t *wcsptr; int length; } XmTextBlockRecWcs, *XmTextBlockWcs;
typedef struct{
    int reason;
    XEvent *event;
    Boolean doit;
    long currInsert, newInsert;
    long startPos, endPos;
    XmTextBlockWcs text;
} XmTextVerifyCallbackStructWcs, *XmTextVerifyPtrWcs;
#define XmTextGetTopPosition XmTextGetTopCharacter
#define XmTextSetTopPosition XmTextSetTopCharacter

#define XmCOPY_FAILED 0
#define XmCOPY_SUCCEEDED 1
#define XmCOPY_TRUNCATED 2
enum{ XmDIALOG_NONE, XmDIALOG_APPLY_BUTTON,
    XmDIALOG_CANCEL_BUTTON, XmDIALOG_DEFAULT_BUTTON,
    XmDIALOG_OK_BUTTON, XmDIALOG_FILTER_LABEL,
    XmDIALOG_FILTER_TEXT, XmDIALOG_HELP_BUTTON,
    XmDIALOG_LIST, XmDIALOG_LIST_LABEL,
    XmDIALOG_MESSAGE_LABEL, XmDIALOG_SELECTION_LABEL,
    XmDIALOG_SYMBOL_LABEL, XmDIALOG_TEXT,
    XmDIALOG_SEPARATOR, XmDIALOG_DIR_LIST,
    XmDIALOG_DIR_LIST_LABEL
};
#define XmDIALOG_HISTORY_LIST XmDIALOG_LIST
#define XmDIALOG_PROMPT_LABEL XmDIALOG_SELECTION_LABEL
#define XmDIALOG_VALUE_TEXT XmDIALOG_TEXT
#define XmDIALOG_COMMAND_TEXT XmDIALOG_TEXT
#define XmDIALOG_FILE_LIST XmDIALOG_LIST
#define XmDIALOG_FILE_LIST_LABEL XmDIALOG_LIST_LABEL
enum{ XmDIALOG_MODELESS, XmDIALOG_PRIMARY_APPLICATION_MODAL,
    XmDIALOG_FULL_APPLICATION_MODAL, XmDIALOG_SYSTEM_MODAL
};

```

```

};
#define XmDIALOG_APPLICATION_MODAL XmDIALOG_PRIMARY_APPLICATION_MODAL
enum{ XmPLACE_TOP, XmPLACE_ABOVE_SELECTION, XmPLACE_BELOW_SELECTION } ;
#define XmFILE_DIRECTORY (1 << 0)
#define XmFILE_REGULAR (1 << 1)
#define XmFILE_ANY_TYPE (XmFILE_DIRECTORY | XmFILE_REGULAR)
enum{ XmDIALOG_WORK_AREA, XmDIALOG_PROMPT,
      XmDIALOG_SELECTION, XmDIALOG_COMMAND,
      XmDIALOG_FILE_SELECTION
};
enum{ XmDIALOG_TEMPLATE, XmDIALOG_ERROR,
      XmDIALOG_INFORMATION, XmDIALOG_MESSAGE,
      XmDIALOG_QUESTION, XmDIALOG_WARNING,
      XmDIALOG_WORKING
};
typedef enum{
  XmVISIBILITY_UNOBSURED, XmVISIBILITY_PARTIALLY_OBSCURED,
  XmVISIBILITY_FULLY_OBSCURED
} XmVisibility ;
typedef enum{
  XmTRAVERSE_CURRENT, XmTRAVERSE_NEXT,
  XmTRAVERSE_PREV, XmTRAVERSE_HOME,
  XmTRAVERSE_NEXT_TAB_GROUP, XmTRAVERSE_PREV_TAB_GROUP,
  XmTRAVERSE_UP, XmTRAVERSE_DOWN,
  XmTRAVERSE_LEFT, XmTRAVERSE_RIGHT
} XmTraversalDirection ;
typedef struct _XmTraverseObscuredCallbackStruct{
  int reason ;
  XEvent * event ;
  Widget traversal_destination ;
  XmTraversalDirection direction ;
} XmTraverseObscuredCallbackStruct ;
typedef unsigned char XmNavigationType;
typedef unsigned char XmButtonType;
typedef XmButtonType * XmButtonTypeTable;
typedef KeySym * XmKeySymTable;
typedef XmStringCharSet * XmStringCharSetTable;
enum{ XmPUSHBUTTON = 1,
      XmRADIOBUTTON,
      XmSEPARATOR,
      XmTITLE,
      XmTOGGLEBUTTON,
      XmCASCADEBUTTON,
      XmDOUBLE_SEPARATOR,
};
#define XmCHECKBUTTON XmTOGGLEBUTTON
typedef struct _XmSecondaryResourceDataRec{
  XmResourceBaseProc base_proc;
  XtPointer client_data;
  String name;
  String res_class;
  XtResourceList resources;
  Cardinal num_resources;
}XmSecondaryResourceDataRec, *XmSecondaryResourceData;
typedef long XmOffset;
typedef XmOffset *XmOffsetPtr;

```

Figure 10-74. Motif 1.2 Manifest Constants from XmStrDefs.h

```

#define XmS ""
#define XmCAccelerator "Accelerator"
#define XmCAcceleratorText "AcceleratorText"
#define XmCAdjustLast "AdjustLast"
#define XmCAdjustMargin "AdjustMargin"
#define XmCAlignment "Alignment"
#define XmCAllowOverlap "AllowOverlap"
#define XmCAnimationMask "AnimationMask"
#define XmCAnimationPixmap "AnimationPixmap"
#define XmCAnimationPixmapDepth "AnimationPixmapDepth"
#define XmCAnimationStyle "AnimationStyle"
#define XmCApplyLabelString "ApplyLabelString"
#define XmCArmCallback "ArmCallback"
#define XmCArmColor "ArmColor"
#define XmCArmPixmap "ArmPixmap"
#define XmCArrowDirection "ArrowDirection"
#define XmCAttachment "Attachment"
#define XmCAudibleWarning "AudibleWarning"
#define XmCAutoShowCursorPosition "AutoShowCursorPosition"
#define XmCAutoUnmanage "AutoUnmanage"
#define XmCAutomaticSelection "AutomaticSelection"
#define XmCAvailability "Availability"
#define XmCBackgroundPixmap "BackgroundPixmap"

```



```

#define XmCBlendModel "BlendModel"
#define XmCBlinkRate "BlinkRate"
#define XmCBottomShadowColor "BottomShadowColor"
#define XmCBottomShadowPixmap "BottomShadowPixmap"
#define XmCButtonAcceleratorText "ButtonAcceleratorText"
#define XmCButtonAccelerators "ButtonAccelerators"
#define XmCButtonCount "ButtonCount"
#define XmCButtonFontList "ButtonFontList"
#define XmCButtonMnemonicCharSets "ButtonMnemonicCharSets"
#define XmCButtonMnemonics "ButtonMnemonics"
#define XmCButtonSet "ButtonSet"
#define XmCButtonType "ButtonType"
#define XmCButtons "Buttons"
#define XmCCancelLabelString "CancelLabelString"
#define XmCChildHorizontalAlignment "ChildHorizontalAlignment"
#define XmCChildHorizontalSpacing "ChildHorizontalSpacing"
#define XmCChildPlacement "ChildPlacement"
#define XmCChildType "ChildType"
#define XmCChildVerticalAlignment "ChildVerticalAlignment"
#define XmCChildren "Children"
#define XmCClientData "ClientData"
#define XmCClipWindow "ClipWindow"
#define XmCColumns "Columns"
#define XmCCommandWindow "CommandWindow"
#define XmCCommandWindowLocation "CommandWindowLocation"
#define XmCConvertProc "ConvertProc"
#define XmCCursorBackground "CursorBackground"
#define XmCCursorForeground "CursorForeground"
#define XmCCursorPosition "CursorPosition"
#define XmCCursorPositionVisible "CursorPositionVisible"
#define XmCDarkThreshold "DarkThreshold"
#define XmCDecimalPoints "DecimalPoints"
#define XmCDefaultButtonShadowThickness "DefaultButtonShadowThickness"
#define XmCDefaultButtonType "DefaultButtonType"
#define XmCDefaultCopyCursorIcon "DefaultCopyCursorIcon"
#define XmCDefaultFontList "DefaultFontList"
#define XmCDefaultInvalidCursorIcon "DefaultInvalidCursorIcon"
#define XmCDefaultLinkCursorIcon "DefaultLinkCursorIcon"
#define XmCDefaultMoveCursorIcon "DefaultMoveCursorIcon"
#define XmCDefaultNoneCursorIcon "DefaultNoneCursorIcon"
#define XmCDefaultPosition "DefaultPosition"
#define XmCDefaultSourceCursorIcon "DefaultSourceCursorIcon"
#define XmCDefaultValidCursorIcon "DefaultValidCursorIcon"
#define XmCDeleteResponse "DeleteResponse"
#define XmCDesktopParent "DesktopParent"
#define XmCDialogStyle "DialogStyle"
#define XmCDialogTitle "DialogTitle"
#define XmCDialogType "DialogType"
#define XmCDirListItemCount "DirListItemCount"
#define XmCDirListItems "DirListItems"
#define XmCDirListLabelString "DirListLabelString"
#define XmCDirMask "DirMask"
#define XmCDirSearchProc "DirSearchProc"
#define XmCDirSpec "DirSpec"
#define XmCDirectory "Directory"
#define XmCDirectoryValid "DirectoryValid"
#define XmCDisarmCallback "DisarmCallback"
#define XmCDoubleClickInterval "DoubleClickInterval"
#define XmCDragContextClass "DragContextClass"
#define XmCDragDropFinishCallback "DragDropFinishCallback"
#define XmCDragIconClass "DragIconClass"
#define XmCDragInitiatorProtocolStyle "DragInitiatorProtocolStyle"
#define XmCDragMotionCallback "DragMotionCallback"
#define XmCDragOperations "DragOperations"
#define XmCDragOverMode "DragOverMode"
#define XmCDragProc "DragProc"
#define XmCDragReceiverProtocolStyle "DragReceiverProtocolStyle"
#define XmCDropProc "DropProc"
#define XmCDropRectangles "DropRectangles"
#define XmCDropSiteActivity "DropSiteActivity"
#define XmCDropSiteEnterCallback "DropSiteEnterCallback"
#define XmCDropSiteLeaveCallback "DropSiteLeaveCallback"
#define XmCDropSiteManagerClass "DropSiteManagerClass"
#define XmCDropSiteOperations "DropSiteOperations"
#define XmCDropSiteType "DropSiteType"
#define XmCDropStartCallback "DropStartCallback"
#define XmCDropTransferClass "DropTransferClass"
#define XmCDropTransfers "DropTransfers"
#define XmCEditable "Editable"
#define XmCEntryBorder "EntryBorder"
#define XmCEntryClass "EntryClass"

```

```
#define XmCExportTargets "ExportTargets"
#define XmCExposeCallback "ExposeCallback"
#define XmCExtensionType "ExtensionType"
#define XmCFileListItemCount "FileListItemCount"
#define XmCFileListItems "FileListItems"
#define XmCFileListLabelString "FileListLabelString"
#define XmCFileSearchProc "FileSearchProc"
#define XmCFileTypeMask "FileTypeMask"
#define XmCFillOnArm "FillOnArm"
#define XmCFillOnSelect "FillOnSelect"
#define XmCFilterLabelString "FilterLabelString"
#define XmCFontList "FontList"
#define XmCForegroundThreshold "ForegroundThreshold"
#define XmCHelpLabelString "HelpLabelString"
#define XmCHighlightColor "HighlightColor"
#define XmCHighlightOnEnter "HighlightOnEnter"
#define XmCHighlightPixmap "HighlightPixmap"
#define XmCHighlightThickness "HighlightThickness"
#define XmCHorizontalFontUnit "HorizontalFontUnit"
#define XmCHorizontalScrollBar "HorizontalScrollBar"
#define XmCHot "Hot"
#define XmCICCHandle "ICCHandle"
#define XmCImportTargets "ImportTargets"
#define XmCIncrement "Increment"
#define XmCIncremental "Incremental"
#define XmCIndicatorOn "IndicatorOn"
#define XmCIndicatorSize "IndicatorSize"
#define XmCIndicatorType "IndicatorType"
#define XmCInitialDelay "InitialDelay"
#define XmCInitialFocus "InitialFocus"
#define XmCInputCreate "InputCreate"
#define XmCInputMethod "InputMethod"
#define XmCInvalidCursorForeground "InvalidCursorForeground"
#define XmCIsAligned "IsAligned"
#define XmCIsHomogeneous "IsHomogeneous"
#define XmCItemCount "ItemCount"
#define XmCItems "Items"
#define XmCKeyboardFocusPolicy "KeyboardFocusPolicy"
#define XmCLabelFontList "LabelFontList"
#define XmCLabelInsensitivePixmap "LabelInsensitivePixmap"
#define XmCLabelPixmap "LabelPixmap"
#define XmCLabelString "LabelString"
#define XmCLabelType "LabelType"
#define XmCLightThreshold "LightThreshold"
#define XmCListLabelString "ListLabelString"
#define XmCListMarginHeight "ListMarginHeight"
#define XmCListMarginWidth "ListMarginWidth"
#define XmCListSizePolicy "ListSizePolicy"
#define XmCListSpacing "ListSpacing"
#define XmCListUpdated "ListUpdated"
#define XmCLogicalParent "LogicalParent"
#define XmCMainWindowMarginHeight "MainWindowMarginHeight"
#define XmCMainWindowMarginWidth "MainWindowMarginWidth"
#define XmCMappingDelay "MappingDelay"
#define XmCMarginBottom "MarginBottom"
#define XmCMarginHeight "MarginHeight"
#define XmCMarginLeft "MarginLeft"
#define XmCMarginRight "MarginRight"
#define XmCMarginTop "MarginTop"
#define XmCMarginWidth "MarginWidth"
#define XmCMask "Mask"
#define XmCMaxItems "MaxItems"
#define XmCMaxLength "MaxLength"
#define XmCMaxValue "MaxValue"
#define XmCMaximum "Maximum"
#define XmCMenuBar "MenuBar"
#define XmCMenuPost "MenuPost"
#define XmCMenuWidget "MenuWidget"
#define XmCMessageProc "MessageProc"
#define XmCMessageWindow "MessageWindow"
#define XmCMinimizeButtons "MinimizeButtons"
#define XmCMinimum "Minimum"
#define XmCMnemonic "Mnemonic"
#define XmCMnemonicCharSet "MnemonicCharSet"
#define XmCMoveOpaque "MoveOpaque"
#define XmCMultiClick "MultiClick"
#define XmCMustMatch "MustMatch"
#define XmCMwmDecorations "MwmDecorations"
#define XmCMwmFunctions "MwmFunctions"
#define XmCMwmInputMode "MwmInputMode"
#define XmCMwmMenu "MwmMenu"
```

```

#define XmCMwmMessages "MwmMessages"
#define XmCNavigationType "NavigationType"
#define XmCNeedsMotion "NeedsMotion"
#define XmCNoMatchString "NoMatchString"
#define XmCNoResize "NoResize"
#define XmCNoneCursorForeground "NoneCursorForeground"
#define XmCNotifyProc "NotifyProc"
#define XmCNumChildren "NumChildren"
#define XmCNumColumns "NumColumns"
#define XmCNumDropRectangles "NumDropRectangles"
#define XmCNumDropTransfers "NumDropTransfers"
#define XmCNumExportTargets "NumExportTargets"
#define XmCNumImportTargets "NumImportTargets"
#define XmCOffset "Offset"
#define XmCOKLabelString "OkLabelString"
#define XmCOperationChangedCallback "OperationChangedCallback"
#define XmCOperationCursorIcon "OperationCursorIcon"
#define XmCOptionLabel "OptionLabel"
#define XmCOptionMnemonic "OptionMnemonic"
#define XmCOutputCreate "OutputCreate"
#define XmCPacking "Packing"
#define XmCPageIncrement "PageIncrement"
#define XmCPaneMaximum "PaneMaximum"
#define XmCPaneMinimum "PaneMinimum"
#define XmCPattern "Pattern"
#define XmCPendingDelete "PendingDelete"
#define XmCPopupEnabled "PopupEnabled"
#define XmCPositionIndex "PositionIndex"
#define XmCPostFromButton "PostFromButton"
#define XmCPostFromCount "PostFromCount"
#define XmCPostFromList "PostFromList"
#define XmCPreditType "PreditType"
#define XmCProcessingDirection "ProcessingDirection"
#define XmCPromptString "PromptString"
#define XmCProtocolCallback "ProtocolCallback"
#define XmCPushButtonEnabled "PushButtonEnabled"
#define XmCQualifySearchDataProc "QualifySearchDataProc"
#define XmCRadioAlwaysOne "RadioAlwaysOne"
#define XmCRadioBehavior "RadioBehavior"
#define XmCRecomputeSize "RecomputeSize"
#define XmCRectangles "Rectangles"
#define XmCRepeatDelay "RepeatDelay"
#define XmCResizeCallback "ResizeCallback"
#define XmCResizeHeight "ResizeHeight"
#define XmCResizePolicy "ResizePolicy"
#define XmCResizeWidth "ResizeWidth"
#define XmCRowColumnType "RowColumnType"
#define XmCRows "Rows"
#define XmCRubberPositioning "RubberPositioning"
#define XmCSashHeight "SashHeight"
#define XmCSashIndent "SashIndent"
#define XmCSashWidth "SashWidth"
#define XmCScaleHeight "ScaleHeight"
#define XmCScaleMultiple "ScaleMultiple"
#define XmCScaleWidth "ScaleWidth"
#define XmCScroll "Scroll"
#define XmCScrollBarDisplayPolicy "ScrollBarDisplayPolicy"
#define XmCScrollBarPlacement "ScrollBarPlacement"
#define XmCScrollSide "ScrollSide"
#define XmCScrolledWindowMarginHeight "ScrolledWindowMarginHeight"
#define XmCScrolledWindowMarginWidth "ScrolledWindowMarginWidth"
#define XmCScrollingPolicy "ScrollingPolicy"
#define XmCSelectColor "SelectColor"
#define XmCSelectInsensitivePixmap "SelectInsensitivePixmap"
#define XmCSelectPixmap "SelectPixmap"
#define XmCSelectThreshold "SelectThreshold"
#define XmCSelectedItemCount "SelectedItemCount"
#define XmCSelectedItems "SelectedItems"
#define XmCSelectionArrayCount "SelectionArrayCount"
#define XmCSelectionLabelString "SelectionLabelString"
#define XmCSelectionPolicy "SelectionPolicy"
#define XmCSeparatorOn "SeparatorOn"
#define XmCSeparatorType "SeparatorType"
#define XmCSet "Set"
#define XmCShadowThickness "ShadowThickness"
#define XmCShadowType "ShadowType"
#define XmCShellUnitType "ShellUnitType"
#define XmCShowArrows "ShowArrows"
#define XmCShowAsDefault "ShowAsDefault"
#define XmCShowSeparator "ShowSeparator"
#define XmCShowValue "ShowValue"

```

```
#define XmCSimpleCheckBox "SimpleCheckBox"
#define XmCSimpleMenuBar "SimpleMenuBar"
#define XmCSimpleOptionMenu "SimpleOptionMenu"
#define XmCSimplePopupMenu "SimplePopupMenu"
#define XmCSimplePullDownMenu "SimplePullDownMenu"
#define XmCSimpleRadioBox "SimpleRadioBox"
#define XmCSizePolicy "SizePolicy"
#define XmCSliderSize "SliderSize"
#define XmCSource "Source"
#define XmCSourceCursorIcon "SourceCursorIcon"
#define XmCSourceIsExternal "SourceIsExternal"
#define XmCSourcePixmapIcon "SourcePixmapIcon"
#define XmCSourceWidget "SourceWidget"
#define XmCSourceWindow "SourceWindow"
#define XmCSpacing "Spacing"
#define XmCStartTime "StartTime"
#define XmCStateCursorIcon "StateCursorIcon"
#define XmCStringDirection "StringDirection"
#define XmCTearOffModel "TearOffModel"
#define XmCTextFontList "TextFontList"
#define XmCTextString "TextString"
#define XmCTextValue "TextValue"
#define XmCTitleString "TitleString"
#define XmCTopCharacter "TopCharacter"
#define XmCTopItemPosition "TopItemPosition"
#define XmCTopLevelEnterCallback "TopLevelEnterCallback"
#define XmCTopLevelLeaveCallback "TopLevelLeaveCallback"
#define XmCTopShadowColor "TopShadowColor"
#define XmCTopShadowPixmap "TopShadowPixmap"
#define XmCTransferProc "TransferProc"
#define XmCTransferStatus "TransferStatus"
#define XmCTraversalOn "TraversalOn"
#define XmCTraversalType "TraversalType"
#define XmCTreeUpdateProc "TreeUpdateProc"
#define XmCTroughColor "TroughColor"
#define XmCUnitType "UnitType"
#define XmCUnpostBehavior "UnpostBehavior"
#define XmCUnselectPixmap "UnselectPixmap"
#define XmCUpdateSliderSize "UpdateSliderSize"
#define XmCUseAsyncGeometry "UseAsyncGeometry"
#define XmCUserData "UserData"
#define XmCValidCursorForeground "ValidCursorForeground"
#define XmCValueChangedCallback "ValueChangedCallback"
#define XmCValueWcs "ValueWcs"
#define XmCVerifyBell "VerifyBell"
#define XmCVerticalAlignment "VerticalAlignment"
#define XmCVerticalFontUnit "VerticalFontUnit"
#define XmCVerticalScrollBar "VerticalScrollBar"
#define XmCVisibleItemCount "VisibleItemCount"
#define XmCVisibleWhenOff "VisibleWhenOff"
#define XmCVisualPolicy "VisualPolicy"
#define XmCWhichButton "WhichButton"
#define XmCWordWrap "WordWrap"
#define XmCWorkWindow "WorkWindow"
#define XmCXmString "XmString"
#define XmNaccelerator "accelerator"
#define XmNacceleratorText "acceleratorText"
#define XmNactivateCallback "activateCallback"
#define XmNadjustLast "adjustLast"
#define XmNadjustMargin "adjustMargin"
#define XmNalignment "alignment"
#define XmNallowOverlap "allowOverlap"
#define XmNallowResize "allowResize"
#define XmNanimationMask "animationMask"
#define XmNanimationPixmap "animationPixmap"
#define XmNanimationPixmapDepth "animationPixmapDepth"
#define XmNanimationStyle "animationStyle"
#define XmNapplyCallback "applyCallback"
#define XmNapplyLabelString "applyLabelString"
#define XmNarmCallback "armCallback"
#define XmNarmColor "armColor"
#define XmNarmPixmap "armPixmap"
#define XmNarrowDirection "arrowDirection"
#define XmNattachment "attachment"
#define XmNaudibleWarning "audibleWarning"
#define XmNautoShowCursorPosition "autoShowCursorPosition"
#define XmNautoUnmanage "autoUnmanage"
#define XmNautomaticSelection "automaticSelection"
#define XmNavailability "availability"
#define XmNblendModel "blendModel"
#define XmNblinkRate "blinkRate"
```

```

#define XmNbottomAttachment "bottomAttachment"
#define XmNbottomOffset "bottomOffset"
#define XmNbottomPosition "bottomPosition"
#define XmNbottomShadowColor "bottomShadowColor"
#define XmNbottomShadowPixmap "bottomShadowPixmap"
#define XmNbottomWidget "bottomWidget"
#define XmNbrowseSelectionCallback "browseSelectionCallback"
#define XmNbuttonAcceleratorText "buttonAcceleratorText"
#define XmNbuttonAccelerators "buttonAccelerators"
#define XmNbuttonCount "buttonCount"
#define XmNbuttonFontList "buttonFontList"
#define XmNbuttonMnemonicCharSets "buttonMnemonicCharSets"
#define XmNbuttonMnemonics "buttonMnemonics"
#define XmNbuttonSet "buttonSet"
#define XmNbuttonType "buttonType"
#define XmNbuttons "buttons"
#define XmNcancelButton "cancelButton"
#define XmNcancelCallback "cancelCallback"
#define XmNcancelLabelString "cancelLabelString"
#define XmNcascadePixmap "cascadePixmap"
#define XmNcascadingCallback "cascadingCallback"
#define XmNchildHorizontalAlignment "childHorizontalAlignment"
#define XmNchildHorizontalSpacing "childHorizontalSpacing"
#define XmNchildPlacement "childPlacement"
#define XmNchildPosition "childPosition"
#define XmNchildType "childType"
#define XmNchildVerticalAlignment "childVerticalAlignment"
#define XmNclientData "clientData"
#define XmNclipWindow "clipWindow"
#define XmNcolumns "columns"
#define XmNcommand "command"
#define XmNcommandChangedCallback "commandChangedCallback"
#define XmNcommandEnteredCallback "commandEnteredCallback"
#define XmNcommandWindow "commandWindow"
#define XmNcommandWindowLocation "commandWindowLocation"
#define XmNconvertProc "convertProc"
#define XmNcursorBackground "cursorBackground"
#define XmNcursorForeground "cursorForeground"
#define XmNcursorPosition "cursorPosition"
#define XmNcursorPositionVisible "cursorPositionVisible"
#define XmNdarkThreshold "darkThreshold"
#define XmNdecimalPoints "decimalPoints"
#define XmNdecrementCallback "decrementCallback"
#define XmNdefaultActionCallback "defaultActionCallback"
#define XmNdefaultButton "defaultButton"
#define XmNdefaultButtonShadowThickness "defaultButtonShadowThickness"
#define XmNdefaultButtonType "defaultButtonType"
#define XmNdefaultCopyCursorIcon "defaultCopyCursorIcon"
#define XmNdefaultFontList "defaultFontList"
#define XmNdefaultInvalidCursorIcon "defaultInvalidCursorIcon"
#define XmNdefaultLinkCursorIcon "defaultLinkCursorIcon"
#define XmNdefaultMoveCursorIcon "defaultMoveCursorIcon"
#define XmNdefaultNoneCursorIcon "defaultNoneCursorIcon"
#define XmNdefaultPosition "defaultPosition"
#define XmNdefaultSourceCursorIcon "defaultSourceCursorIcon"
#define XmNdefaultValidCursorIcon "defaultValidCursorIcon"
#define XmNdeleteResponse "deleteResponse"
#define XmNdesktopParent "desktopParent"
#define XmNdialogStyle "dialogStyle"
#define XmNdialogTitle "dialogTitle"
#define XmNdialogType "dialogType"
#define XmNdirListItemCount "dirListItemCount"
#define XmNdirListItems "dirListItems"
#define XmNdirListLabelString "dirListLabelString"
#define XmNdirMask "dirMask"
#define XmNdirSearchProc "dirSearchProc"
#define XmNdirSpec "dirSpec"
#define XmNdirectory "directory"
#define XmNdirectoryValid "directoryValid"
#define XmNdisarmCallback "disarmCallback"
#define XmNdoubleClickInterval "doubleClickInterval"
#define XmNdragCallback "dragCallback"
#define XmNdragContextClass "dragContextClass"
#define XmNdragDropFinishCallback "dragDropFinishCallback"
#define XmNdragIconClass "dragIconClass"
#define XmNdragInitiatorProtocolStyle "dragInitiatorProtocolStyle"
#define XmNdragMotionCallback "dragMotionCallback"
#define XmNdragOperations "dragOperations"
#define XmNdragOverMode "dragOverMode"
#define XmNdragProc "dragProc"
#define XmNdragReceiverProtocolStyle "dragReceiverProtocolStyle"

```

```
#define XmNdropFinishCallback "dropFinishCallback"
#define XmNdropProc "dropProc"
#define XmNdropRectangles "dropRectangles"
#define XmNdropSiteActivity "dropSiteActivity"
#define XmNdropSiteEnterCallback "dropSiteEnterCallback"
#define XmNdropSiteLeaveCallback "dropSiteLeaveCallback"
#define XmNdropSiteManagerClass "dropSiteManagerClass"
#define XmNdropSiteOperations "dropSiteOperations"
#define XmNdropSiteType "dropSiteType"
#define XmNdropStartCallback "dropStartCallback"
#define XmNdropTransferClass "dropTransferClass"
#define XmNdropTransfers "dropTransfers"
#define XmNeditMode "editMode"
#define XmNeditable "editable"
#define XmNentryAlignment "entryAlignment"
#define XmNentryBorder "entryBorder"
#define XmNentryCallback "entryCallback"
#define XmNentryClass "entryClass"
#define XmNentryVerticalAlignment "entryVerticalAlignment"
#define XmNexportTargets "exportTargets"
#define XmNexposeCallback "exposeCallback"
#define XmNextendedSelectionCallback "extendedSelectionCallback"
#define XmNextensionType "extensionType"
#define XmNfileListItemCount "fileListItemCount"
#define XmNfileListItems "fileListItems"
#define XmNfileListLabelString "fileListLabelString"
#define XmNfileSearchProc "fileSearchProc"
#define XmNfileTypeMask "fileTypeMask"
#define XmNfillOnArm "fillOnArm"
#define XmNfillOnSelect "fillOnSelect"
#define XmNfilterLabelString "filterLabelString"
#define XmNfocusCallback "focusCallback"
#define XmNfocusMovedCallback "focusMovedCallback"
#define XmNfocusPolicyChanged "focusPolicyChanged"
#define XmNfontList "fontList"
#define XmNforegroundThreshold "foregroundThreshold"
#define XmNfractionBase "fractionBase"
#define XmNgainPrimaryCallback "gainPrimaryCallback"
#define XmNhelpCallback "helpCallback"
#define XmNhelpLabelString "helpLabelString"
#define XmNhighlightColor "highlightColor"
#define XmNhighlightOnEnter "highlightOnEnter"
#define XmNhighlightPixmap "highlightPixmap"
#define XmNhighlightThickness "highlightThickness"
#define XmNhistoryItemCount "historyItemCount"
#define XmNhistoryItems "historyItems"
#define XmNhistoryMaxItems "historyMaxItems"
#define XmNhistoryVisibleItemCount "historyVisibleItemCount"
#define XmNhorizontalFontUnit "horizontalFontUnit"
#define XmNhorizontalScrollBar "horizontalScrollBar"
#define XmNhorizontalSpacing "horizontalSpacing"
#define XmNhotX "hotX"
#define XmNhotY "hotY"
#define XmNiccHandle "iccHandle"
#define XmNimportTargets "importTargets"
#define XmNincrement "increment"
#define XmNincrementCallback "incrementCallback"
#define XmNincremental "incremental"
#define XmNindicatorOn "indicatorOn"
#define XmNindicatorSize "indicatorSize"
#define XmNindicatorType "indicatorType"
#define XmNinitialDelay "initialDelay"
#define XmNinitialFocus "initialFocus"
#define XmNinputCallback "inputCallback"
#define XmNinputCreate "inputCreate"
#define XmNinputMethod "inputMethod"
#define XmNinvalidCursorForeground "invalidCursorForeground"
#define XmNisAligned "isAligned"
#define XmNisHomogeneous "isHomogeneous"
#define XmNitemCount "itemCount"
#define XmNitems "items"
#define XmNkeyboardFocusPolicy "keyboardFocusPolicy"
#define XmNlabelFontList "labelFontList"
#define XmNlabelInsensitivePixmap "labelInsensitivePixmap"
#define XmNlabelPixmap "labelPixmap"
#define XmNlabelString "labelString"
#define XmNlabelType "labelType"
#define XmNleftAttachment "leftAttachment"
#define XmNleftOffset "leftOffset"
#define XmNleftPosition "leftPosition"
#define XmNleftWidget "leftWidget"
```

```

#define XmNlightThreshold "lightThreshold"
#define XmNlineSpace "lineSpace"
#define XmNlistItemCount "listItemCount"
#define XmNlistItems "listItems"
#define XmNlistLabelString "listLabelString"
#define XmNlistMarginHeight "listMarginHeight"
#define XmNlistMarginWidth "listMarginWidth"
#define XmNlistSizePolicy "listSizePolicy"
#define XmNlistSpacing "listSpacing"
#define XmNlistUpdated "listUpdated"
#define XmNlistVisibleItemCount "listVisibleItemCount"
#define XmNlogicalParent "logicalParent"
#define XmNlosePrimaryCallback "losePrimaryCallback"
#define XmNlosingFocusCallback "losingFocusCallback"
#define XmNmainWindowMarginHeight "mainWindowMarginHeight"
#define XmNmainWindowMarginWidth "mainWindowMarginWidth"
#define XmNmapCallback "mapCallback"
#define XmNmappingDelay "mappingDelay"
#define XmNmargin "margin"
#define XmNmarginBottom "marginBottom"
#define XmNmarginHeight "marginHeight"
#define XmNmarginLeft "marginLeft"
#define XmNmarginRight "marginRight"
#define XmNmarginTop "marginTop"
#define XmNmarginWidth "marginWidth"
#define XmNmask "mask"
#define XmNmaxLength "maxLength"
#define XmNmaximum "maximum"
#define XmNmenuAccelerator "menuAccelerator"
#define XmNmenuBar "menuBar"
#define XmNmenuCursor "menuCursor"
#define XmNmenuHelpWidget "menuHelpWidget"
#define XmNmenuHistory "menuHistory"
#define XmNmenuPost "menuPost"
#define XmNmessageAlignment "messageAlignment"
#define XmNmessageProc "messageProc"
#define XmNmessageString "messageString"
#define XmNmessageWindow "messageWindow"
#define XmNminimizeButtons "minimizeButtons"
#define XmNminimum "minimum"
#define XmNmnemonic "mnemonic"
#define XmNmnemonicCharSet "mnemonicCharSet"
#define XmNmodifyVerifyCallback "modifyVerifyCallback"
#define XmNmodifyVerifyCallbackWcs "modifyVerifyCallbackWcs"
#define XmNmotionVerifyCallback "motionVerifyCallback"
#define XmNmoveOpaque "moveOpaque"
#define XmNmultiClick "multiClick"
#define XmNmultipleSelectionCallback "multipleSelectionCallback"
#define XmNmustMatch "mustMatch"
#define XmNmwmDecorations "mwmDecorations"
#define XmNmwmFunctions "mwmFunctions"
#define XmNmwmInputMode "mwmInputMode"
#define XmNmwmMenu "mwmMenu"
#define XmNmwmMessages "mwmMessages"
#define XmNnavigationType "navigationType"
#define XmNneedsMotion "needsMotion"
#define XmNnoMatchCallback "noMatchCallback"
#define XmNnoMatchString "noMatchString"
#define XmNnoResize "noResize"
#define XmNnoneCursorForeground "noneCursorForeground"
#define XmNnotifyProc "notifyProc"
#define XmNnumColumns "numColumns"
#define XmNnumDropRectangles "numDropRectangles"
#define XmNnumDropTransfers "numDropTransfers"
#define XmNnumExportTargets "numExportTargets"
#define XmNnumImportTargets "numImportTargets"
#define XmNnumRectangles "numRectangles"
#define XmNoffsetX "offsetX"
#define XmNoffsetY "offsetY"
#define XmNokCallback "okCallback"
#define XmNokLabelString "okLabelString"
#define XmNoperationChangedCallback "operationChangedCallback"
#define XmNoperationCursorIcon "operationCursorIcon"
#define XmNoptionLabel "optionLabel"
#define XmNoptionMnemonic "optionMnemonic"
#define XmNoutputCreate "outputCreate"
#define XmNpacking "packing"
#define XmNpageDecrementCallback "pageDecrementCallback"
#define XmNpageIncrement "pageIncrement"
#define XmNpageIncrementCallback "pageIncrementCallback"
#define XmNpaneMaximum "paneMaximum"

```

```
#define XmNpaneMinimum "paneMinimum"
#define XmNpattern "pattern"
#define XmNpendingDelete "pendingDelete"
#define XmNpopupEnabled "popupEnabled"
#define XmNpositionIndex "positionIndex"
#define XmNpostFromButton "postFromButton"
#define XmNpostFromCount "postFromCount"
#define XmNpostFromList "postFromList"
#define XmNpreeditType "preeditType"
#define XmNprocessingDirection "processingDirection"
#define XmNpromptString "promptString"
#define XmNprotocolCallback "protocolCallback"
#define XmNpushButtonEnabled "pushButtonEnabled"
#define XmNqualifySearchDataProc "qualifySearchDataProc"
#define XmNradioAlwaysOne "radioAlwaysOne"
#define XmNradioBehavior "radioBehavior"
#define XmNrealizeCallback "realizeCallback"
#define XmNrecomputeSize "recomputeSize"
#define XmNrectangles "rectangles"
#define XmNrefigureMode "refigureMode"
#define XmNrepeatDelay "repeatDelay"
#define XmNresizable "resizable"
#define XmNresizeCallback "resizeCallback"
#define XmNresizeHeight "resizeHeight"
#define XmNresizePolicy "resizePolicy"
#define XmNresizeWidth "resizeWidth"
#define XmNrightAttachment "rightAttachment"
#define XmNrightOffset "rightOffset"
#define XmNrightPosition "rightPosition"
#define XmNrightWidget "rightWidget"
#define XmNrowColumnType "rowColumnType"
#define XmNrows "rows"
#define XmNrubberPositioning "rubberPositioning"
#define XmNsashHeight "sashHeight"
#define XmNsashIndent "sashIndent"
#define XmNsashShadowThickness "sashShadowThickness"
#define XmNsashWidth "sashWidth"
#define XmNscaledHeight "scaledHeight"
#define XmNscaledMultiple "scaledMultiple"
#define XmNscaledWidth "scaledWidth"
#define XmNscrollBarDisplayPolicy "scrollBarDisplayPolicy"
#define XmNscrollBarPlacement "scrollBarPlacement"
#define XmNscrollHorizontal "scrollHorizontal"
#define XmNscrollLeftSide "scrollLeftSide"
#define XmNscrollTopSide "scrollTopSide"
#define XmNscrollVertical "scrollVertical"
#define XmNscrolledWindowMarginHeight "scrolledWindowMarginHeight"
#define XmNscrolledWindowMarginWidth "scrolledWindowMarginWidth"
#define XmNscrollingPolicy "scrollingPolicy"
#define XmNselectColor "selectColor"
#define XmNselectInsensitivePixmap "selectInsensitivePixmap"
#define XmNselectPixmap "selectPixmap"
#define XmNselectThreshold "selectThreshold"
#define XmNselectedItemCount "selectedItemCount"
#define XmNselectedItems "selectedItems"
#define XmNselectionArrayCount "selectionArrayCount"
#define XmNselectionLabelString "selectionLabelString"
#define XmNselectionPolicy "selectionPolicy"
#define XmNseparatorOn "separatorOn"
#define XmNseparatorType "separatorType"
#define XmNset "set"
#define XmNshadow "shadow"
#define XmNshadowThickness "shadowThickness"
#define XmNshadowType "shadowType"
#define XmNshellUnitType "shellUnitType"
#define XmNshowArrows "showArrows"
#define XmNshowAsDefault "showAsDefault"
#define XmNshowSeparator "showSeparator"
#define XmNshowValue "showValue"
#define XmNsimpleCallback "simpleCallback"
#define XmNsingleSelectionCallback "singleSelectionCallback"
#define XmNsizePolicy "sizePolicy"
#define XmNskipAdjust "skipAdjust"
#define XmNsliderSize "sliderSize"
#define XmNsource "source"
#define XmNsourceCursorIcon "sourceCursorIcon"
#define XmNsourceIsExternal "sourceIsExternal"
#define XmNsourcePixmapIcon "sourcePixmapIcon"
#define XmNsourceWidget "sourceWidget"
#define XmNsourceWindow "sourceWindow"
#define XmNspacing "spacing"
```



```

#define XmNspotLocation "spotLocation"
#define XmNstartTime "startTime"
#define XmNstateCursorIcon "stateCursorIcon"
#define XmNstringDirection "stringDirection"
#define XmNsubMenuId "subMenuId"
#define XmNsymbolPixmap "symbolPixmap"
#define XmNtearOffMenuActivateCallback "tearOffMenuActivateCallback"
#define XmNtearOffMenuDeactivateCallback "tearOffMenuDeactivateCallback"
#define XmNtearOffModel "tearOffModel"
#define XmNtextAccelerators "textAccelerators"
#define XmNtextColumns "textColumns"
#define XmNtextFontList "textFontList"
#define XmNtextString "textString"
#define XmNtextTranslations "textTranslations"
#define XmNtextValue "textValue"
#define XmNtitleLabel "titleLabel"
#define XmNtoBottomCallback "toBottomCallback"
#define XmNtoPositionCallback "toPositionCallback"
#define XmNtoTopCallback "toTopCallback"
#define XmNtopAttachment "topAttachment"
#define XmNtopCharacter "topCharacter"
#define XmNtopItemPosition "topItemPosition"
#define XmNtopLevelEnterCallback "topLevelEnterCallback"
#define XmNtopLevelLeaveCallback "topLevelLeaveCallback"
#define XmNtopOffset "topOffset"
#define XmNtopPosition "topPosition"
#define XmNtopShadowColor "topShadowColor"
#define XmNtopShadowPixmap "topShadowPixmap"
#define XmNtopWidget "topWidget"
#define XmNtransferProc "transferProc"
#define XmNtransferStatus "transferStatus"
#define XmNtraversalCallback "traversalCallback"
#define XmNtraversalOn "traversalOn"
#define XmNtraversalType "traversalType"
#define XmNtraverseObscuredCallback "traverseObscuredCallback"
#define XmNtreeUpdateProc "treeUpdateProc"
#define XmNtroughColor "troughColor"
#define XmNunitType "unitType"
#define XmNunmapCallback "unmapCallback"
#define XmNunpostBehavior "unpostBehavior"
#define XmNunselectPixmap "unselectPixmap"
#define XmNupdateSliderSize "updateSliderSize"
#define XmNuseAsyncGeometry "useAsyncGeometry"
#define XmNuserData "userData"
#define XmNvalidCursorForeground "validCursorForeground"
#define XmNvalueChangedCallback "valueChangedCallback"
#define XmNvalueWcs "valueWcs"
#define XmNverifyBell "verifyBell"
#define XmNverticalFontUnit "verticalFontUnit"
#define XmNverticalScrollBar "verticalScrollBar"
#define XmNverticalSpacing "verticalSpacing"
#define XmNvisibleItemCount "visibleItemCount"
#define XmNvisibleWhenOff "visibleWhenOff"
#define XmNvisualPolicy "visualPolicy"
#define XmNwhichButton "whichButton"
#define XmNwordWrap "wordWrap"
#define XmNworkWindow "workWindow"
#define XmRAlignment "Alignment"
#define XmRAnimationMask "AnimationMask"
#define XmRAnimationPixmap "AnimationPixmap"
#define XmRAnimationStyle "AnimationStyle"
#define XmRArrowDirection "ArrowDirection"
#define XmRAtomList "AtomList"
#define XmRAttachment "Attachment"
#define XmRAudibleWarning "AudibleWarning"
#define XmRAvailability "Availability"
#define XmRBackgroundPixmap "BackgroundPixmap"
#define XmRBlendModel "BlendModel"
#define XmRBooleanDimension "BooleanDimension"
#define XmRBottomShadowPixmap "BottomShadowPixmap"
#define XmRButtonType "ButtonType"
#define XmRCallbackProc "CallbackProc"
#define XmRChar "Char"
#define XmRCharSetTable "CharSetTable"
#define XmRChildHorizontalAlignment "ChildHorizontalAlignment"
#define XmRChildPlacement "ChildPlacement"
#define XmRChildType "ChildType"
#define XmRChildVerticalAlignment "ChildVerticalAlignment"
#define XmRCommandWindowLocation "CommandWindowLocation"
#define XmRCompoundText "CompoundText"
#define XmRDefaultButtonType "DefaultButtonType"

```

```
#define XmRDeleteResponse "DeleteResponse"
#define XmRDialogStyle "DialogStyle"
#define XmRDialogType "DialogType"
#define XmRDoubleClickInterval "DoubleClickInterval"
#define XmRDragInitiatorProtocolStyle "DragInitiatorProtocolStyle"
#define XmRDragReceiverProtocolStyle "DragReceiverProtocolStyle"
#define XmRDropSiteActivity "DropSiteActivity"
#define XmRDropSiteOperations "DropSiteOperations"
#define XmRDropSiteType "DropSiteType"
#define XmRDropTransfers "DropTransfers"
#define XmRExtensionType "ExtensionType"
#define XmRFileTypeMask "FileTypeMask"
#define XmRFontList "FontList"
#define XmRGadgetPixmap "GadgetPixmap"
#define XmRHighlightPixmap "HighlightPixmap"
#define XmRHorizontalDimension "HorizontalDimension"
#define XmRHorizontalInt "HorizontalInt"
#define XmRHorizontalPosition "HorizontalPosition"
#define XmRIconAttachment "IconAttachment"
#define XmRImportTargets "ImportTargets"
#define XmRIndicatorType "IndicatorType"
#define XmRItemCount "ItemCount"
#define XmRItems "Items"
#define XmRKeySym "KeySym"
#define XmRKeySymTable "KeySymTable"
#define XmRKeyboardFocusPolicy "KeyboardFocusPolicy"
#define XmRLabelType "LabelType"
#define XmRListMarginHeight "ListMarginHeight"
#define XmRListMarginWidth "ListMarginWidth"
#define XmRListSizePolicy "ListSizePolicy"
#define XmRListSpacing "ListSpacing"
#define XmRManBottomShadowPixmap "ManBottomShadowPixmap"
#define XmRManForegroundPixmap "ManForegroundPixmap"
#define XmRManHighlightPixmap "ManHighlightPixmap"
#define XmRManTopShadowPixmap "ManTopShadowPixmap"
#define XmRMenuWidget "MenuWidget"
#define XmRMnemonic "Mnemonic"
#define XmRMultiClick "MultiClick"
#define XmRNavigationType "NavigationType"
#define XmRPacking "Packing"
#define XmRPrimForegroundPixmap "PrimForegroundPixmap"
#define XmRProc "Proc"
#define XmRProcessingDirection "ProcessingDirection"
#define XmRRectangleList "RectangleList"
#define XmRResizePolicy "ResizePolicy"
#define XmRRowColumnType "RowColumnType"
#define XmRScrollBarDisplayPolicy "ScrollBarDisplayPolicy"
#define XmRScrollBarPlacement "ScrollBarPlacement"
#define XmRScrollingPolicy "ScrollingPolicy"
#define XmRSelectedItemCount "SelectedItemCount"
#define XmRSelectedItems "SelectedItems"
#define XmRSelectionPolicy "SelectionPolicy"
#define XmRSelectionType "SelectionType"
#define XmRSeparatorType "SeparatorType"
#define XmRShadowType "ShadowType"
#define XmRShellHorizDim "ShellHorizDim"
#define XmRShellHorizPos "ShellHorizPos"
#define XmRShellUnitType "ShellUnitType"
#define XmRShellVertDim "ShellVertDim"
#define XmRShellVertPos "ShellVertPos"
#define XmRSizePolicy "SizePolicy"
#define XmRStringDirection "StringDirection"
#define XmRTearOffModel "TearOffModel"
#define XmRTopShadowPixmap "TopShadowPixmap"
#define XmRTransferStatus "TransferStatus"
#define XmRTraversalType "TraversalType"
#define XmRUnitType "UnitType"
#define XmRUNpostBehavior "UnpostBehavior"
#define XmRValueWcs "ValueWcs"
#define XmRVerticalAlignment "VerticalAlignment"
#define XmRVerticalDimension "VerticalDimension"
#define XmRVerticalInt "VerticalInt"
#define XmRVerticalPosition "VerticalPosition"
#define XmRVirtualBinding "VirtualBinding"
#define XmRVisibleItemCount "VisibleItemCount"
#define XmRVisualPolicy "VisualPolicy"
#define XmRWhichButton "WhichButton"
#define XmRXmBackgroundPixmap "XmBackgroundPixmap"
#define XmRXmString "XmString"
#define XmRXmStringCharSet "XmStringCharSet"
#define XmRXmStringTable "XmStringTable"
```

```

#define XmVosfActivate "osfActivate"
#define XmVosfAddMode "osfAddMode"
#define XmVosfBackSpace "osfBackSpace"
#define XmVosfBeginLine "osfBeginLine"
#define XmVosfCancel "osfCancel"
#define XmVosfClear "osfClear"
#define XmVosfCopy "osfCopy"
#define XmVosfCut "osfCut"
#define XmVosfDelete "osfDelete"
#define XmVosfDown "osfDown"
#define XmVosfEndLine "osfEndLine"
#define XmVosfHelp "osfHelp"
#define XmVosfInsert "osfInsert"
#define XmVosfLeft "osfLeft"
#define XmVosfMenu "osfMenu"
#define XmVosfMenuBar "osfMenuBar"
#define XmVosfPageDown "osfPageDown"
#define XmVosfPageLeft "osfPageLeft"
#define XmVosfPageRight "osfPageRight"
#define XmVosfPageUp "osfPageUp"
#define XmVosfPaste "osfPaste"
#define XmVosfPrimaryPaste "osfPrimaryPaste"
#define XmVosfQuickPaste "osfQuickPaste"
#define XmVosfRight "osfRight"
#define XmVosfSelect "osfSelect"
#define XmVosfUndo "osfUndo"
#define XmVosfUp "osfUp"
#define XmSFONTLIST_DEFAULT_TAG_STRING "FONTLIST_DEFAULT_TAG_STRING"
#define XmSxmFONTLIST_DEFAULT_TAG_STRING "XmFONTLIST_DEFAULT_TAG_STRING"
#define _XmConst /**/
#define XmSTRING_DEFAULT_CHARSET XmS
#define XmSTRING_ISO8859_1 "ISO8859-1"
#define XmFONTLIST_DEFAULT_TAG XmSFONTLIST_DEFAULT_TAG_STRING
#define XmFONTLIST_DEFAULT_TAG_STRING XmSxmFONTLIST_DEFAULT_TAG_STRING
#define XmVaCASCADEBUTTON "cascadeButton"
#define XmVaCHECKBUTTON "checkButton"
#define XmVaDOUBLE_SEPARATOR "doubleSeparator"
#define XmVaPUSHBUTTON "pushButton"
#define XmVaRADIOBUTTON "radioButton"
#define XmVaSEPARATOR "separator"
#define XmVaSINGLE_SEPARATOR "singleSeparator"
#define XmVaTOGGLEBUTTON "checkButton"
#define XmVaTITLE XtNtitle
#define XtCKeyboardFocusPolicy XmCKeyboardFocusPolicy
#define XtCShellUnitType XmCShellUnitType
#define XtNkeyboardFocusPolicy XmNkeyboardFocusPolicy
#define XtNshellUnitType XmNshellUnitType
#define XtRKeyboardFocusPolicy XmRKeyboardFocusPolicy
#define XmRPrimBottomShadowPixmap XmRBottomShadowPixmap
#define XmRPrimHighlightPixmap XmRHighlightPixmap
#define XmRPrimTopShadowPixmap XmRTopShadowPixmap
#define XmCAccelerators XtCAccelerators
#define XmCAllowShellResize XtCAllowShellResize
#define XmCArgc XtCArgc
#define XmCArgv XtCArgv
#define XmCBackground XtCBackground
#define XmCBaseHeight XtCBaseHeight
#define XmCBaseHeight XtCBaseHeight
#define XmCBaseWidth XtCBaseWidth
#define XmCBaseWidth XtCBaseWidth
#define XmCBitmap XtCBitmap
#define XmCBoolean XtCBoolean
#define XmCBorderColor XtCBorderColor
#define XmCBorderWidth XtCBorderWidth
#define XmCCallback XtCCallback
#define XmCColor XtCColor
#define XmCColormap XtCColormap
#define XmCCreatePopupChildProc XtCCreatePopupChildProc
#define XmCCursor XtCCursor
#define XmCDepth XtCDepth
#define XmCDimension XtRDimension
#define XmCEditMode XtREditMode
#define XmCEditType XtCEditType
#define XmCEventBindings XtCEventBindings
#define XmCFile XtCFile
#define XmCFont XtCFont
#define XmCFontSet XtCFontSet
#define XmCForeground XtCForeground
#define XmCFraction XtCFraction
#define XmCFunction XtCFunction
#define XmCGeometry XtCGeometry

```

```

#define XmCHSpace      XtCHSpace
#define XmCHeight     XtCHeight
#define XmCHeightInc  XtCHeightInc
#define XmCIconMask   XtCIconMask
#define XmCIconName   XtCIconName
#define XmCIconNameEncoding XtCIconNameEncoding
#define XmCIconPixmap XtCIconPixmap
#define XmCIconWindow XtCIconWindow
#define XmCIconX      XtCIconX
#define XmCIconY      XtCIconY
#define XmCIconic     XtCIconic
#define XmCIndex      XtCIndex
#define XmCInitialResourcesPersistent XtCInitialResourcesPersistent
#define XmCInitialState XtCInitialState
#define XmCInput      XtCInput
#define XmCInsertPosition XtCInsertPosition
#define XmCInterval   XtCInterval
#define XmCJustify     XtCJustify
#define XmCLabel      XtCLabel
#define XmCLength     XtCLength
#define XmCMappedWhenManaged XtCMappedWhenManaged
#define XmCMargin     XtCMargin
#define XmCMaxAspectX XtCMaxAspectX
#define XmCMaxAspectY XtCMaxAspectY
#define XmCMaxHeight  XtCMaxHeight
#define XmCMaxWidth   XtCMaxWidth
#define XmCMenuEntry  XtCMenuEntry
#define XmCMinAspectX XtCMinAspectX
#define XmCMinAspectY XtCMinAspectY
#define XmCMinHeight  XtCMinHeight
#define XmCMinWidth   XtCMinWidth
#define XmCNotify     XtCNotify
#define XmCOrientation XtCOrientation
#define XmCOVERRIDERedirect XtCOVERRIDERedirect
#define XmCParameter  XtCParameter
#define XmCPixmap     XtCPixmap
#define XmCPosition   XtCPosition
#define XmCReadOnly   XtCReadOnly
#define XmCResize     XtCResize
#define XmCReverseVideo XtCReverseVideo
#define XmCSaveUnder XtCSaveUnder
#define XmCScreen     XtCScreen
#define XmCScrollDCursor XtCScrollDCursor
#define XmCScrollHCursor XtCScrollHCursor
#define XmCScrollLCursor XtCScrollLCursor
#define XmCScrollProc XtCScrollProc
#define XmCScrollRCursor XtCScrollRCursor
#define XmCScrollUCursor XtCScrollUCursor
#define XmCScrollVCursor XtCScrollVCursor
#define XmCSelection  XtCSelection
#define XmCSelectionArray XtCSelectionArray
#define XmCSensitive  XtCSensitive
#define XmCSpace      XtCSpace
#define XmCString     XtCString
#define XmCTextOptions XtCTextOptions
#define XmCTextPosition XtCTextPosition
#define XmCTextSink   XtCTextSink
#define XmCTextSource XtCTextSource
#define XmCThickness  XtCThickness
#define XmCThumb      XtCThumb
#define XmCTitle      XtCTitle
#define XmCTitleEncoding XtCTitleEncoding
#define XmCTransient  XtCTransient
#define XmCTransientFor XtCTransientFor
#define XmCTranslations XtCTranslations
#define XmCVSpace     XtCVSpace
#define XmCValue      XtCValue
#define XmCVisual     XtCVisual
#define XmCWaitForWm  XtCWaitForWm
#define XmCWidget     XtCWidget
#define XmCWidth      XtCWidth
#define XmCWidthInc   XtCWidthInc
#define XmCWinGravity XtCWinGravity
#define XmCWindow     XtCWindow
#define XmCWindowGroup XtCWindowGroup
#define XmCWmTimeout  XtCWmTimeout
#define XmCX          XtCX
#define XmCY          XtCY
#define XmNaccelerators XtNaccelerators
#define XmNallowShellResize XtNallowShellResize
#define XmNancestorSensitive XtNancestorSensitive

```

```

#define XmNargc           XtNargc
#define XmNargv           XtNargv
#define XmNbackground    XtNbackground
#define XmNbackgroundPixmap XtNbackgroundPixmap
#define XmNbaseHeight     XtNbaseHeight
#define XmNbaseHeight     XtNbaseHeight
#define XmNbaseWidth      XtNbaseWidth
#define XmNbaseWidth      XtNbaseWidth
#define XmNbitmap         XtNbitmap
#define XmNborder         XtNborder
#define XmNborderColor    XtNborderColor
#define XmNborderPixmap   XtNborderPixmap
#define XmNborderWidth    XtNborderWidth
#define XmNcallback       XtNcallback
#define XmNchildren       XtNchildren
#define XmNcolormap       XtNcolormap
#define XmNcreatePopupChildProc XtNcreatePopupChildProc
#define XmNdepth          XtNdepth
#define XmNdestroyCallback XtNdestroyCallback
#define XmNeditType       XtNeditType
#define XmNfile           XtNfile
#define XmNfont           XtNfont
#define XmNfontSet        XtNfontSet
#define XmNforceBars      XtNforceBars
#define XmNforeground     XtNforeground
#define XmNfunction       XtNfunction
#define XmNgeometry       XtNgeometry
#define XmNheight         XtNheight
#define XmNheightInc      XtNheightInc
#define XmNhighlight      XtNhighlight
#define XmNiconMask       XtNiconMask
#define XmNiconName       XtNiconName
#define XmNiconNameEncoding XtNiconNameEncoding
#define XmNiconPixmap     XtNiconPixmap
#define XmNiconWindow     XtNiconWindow
#define XmNiconX          XtNiconX
#define XmNiconY          XtNiconY
#define XmNiconic         XtNiconic
#define XmNindex          XtNindex
#define XmNinitialResourcesPersistent XtNinitialResourcesPersistent
#define XmNinitialState   XtNinitialState
#define XmNinnerHeight    XtNinnerHeight
#define XmNinnerWidth     XtNinnerWidth
#define XmNinnerWindow    XtNinnerWindow
#define XmNinput          XtNinput
#define XmNinsertPosition XtNinsertPosition
#define XmNinternalHeight XtNinternalHeight
#define XmNinternalWidth  XtNinternalWidth
#define XmNjumpProc       XtNjumpProc
#define XmNjustify        XtNjustify
#define XmNlength         XtNlength
#define XmNlowerRight     XtNlowerRight
#define XmNmappedWhenManaged XtNmappedWhenManaged
#define XmNmaxAspectX     XtNmaxAspectX
#define XmNmaxAspectY     XtNmaxAspectY
#define XmNmaxHeight      XtNmaxHeight
#define XmNmaxWidth       XtNmaxWidth
#define XmNmenuEntry      XtNmenuEntry
#define XmNminAspectX     XtNminAspectX
#define XmNminAspectY     XtNminAspectY
#define XmNminHeight      XtNminHeight
#define XmNminWidth       XtNminWidth
#define XmNname           XtNname
#define XmNnotify         XtNnotify
#define XmNnumChildren     XtNnumChildren
#define XmNorientation    XtNorientation
#define XmNoverrideRedirect XtNoverrideRedirect
#define XmNparameter      XtNparameter
#define XmNpixmap         XtNpixmap
#define XmNpopdownCallback XtNpopdownCallback
#define XmNpopupCallback  XtNpopupCallback
#define XmNresize         XtNresize
#define XmNreverseVideo    XtNreverseVideo
#define XmNsaveUnder      XtNsaveUnder
#define XmNscreen         XtNscreen
#define XmNscrollDCursor  XtNscrollDCursor
#define XmNscrollHCursor  XtNscrollHCursor
#define XmNscrollLCursor  XtNscrollLCursor
#define XmNscrollProc     XtNscrollProc
#define XmNscrollRCursor  XtNscrollRCursor
#define XmNscrollUCursor  XtNscrollUCursor

```

```

#define XmNscrollVCursor      XtNscrollVCursor
#define XmNselection         XtNselection
#define XmNselectionArray    XtNselectionArray
#define XmNsensitive         XtNsensitive
#define XmNshown             XtNshown
#define XmNspace             XtNspace
#define XmNstring            XtNstring
#define XmNtextOptions       XtNtextOptions
#define XmNtextSink          XtNtextSink
#define XmNtextSource        XtNtextSource
#define XmNthickness         XtNthickness
#define XmNthumb             XtNthumb
#define XmNthumbProc         XtNthumbProc
#define XmNtitle              XtNtitle
#define XmNtitleEncoding     XtNtitleEncoding
#define XmNtop               XtNtop
#define XmNtransient         XtNtransient
#define XmNtransientFor      XtNtransientFor
#define XmNtransientFor      XtNtransientFor
#define XmNtranslations      XtNtranslations
#define XmNupdate            XtNupdate
#define XmNuseBottom         XtNuseBottom
#define XmNuseRight          XtNuseRight
#define XmNvalue             XtNvalue
#define XmNvisual            XtNvisual
#define XmNwaitForWm         XtNwaitForWm
#define XmNwidth             XtNwidth
#define XmNwidthInc          XtNwidthInc
#define XmNwinGravity        XtNwinGravity
#define XmNwindow            XtNwindow
#define XmNwindowGroup       XtNwindowGroup
#define XmNwmTimeout         XtNwmTimeout
#define XmNx                  XtNx
#define XmNy                  XtNy
#define XmRAcceleratorTable  XtRAcceleratorTable
#define XmRAtom              XtRAtom
#define XmRBitmap            XtRBitmap
#define XmRBool              XtRBool
#define XmRBoolean           XtRBoolean
#define XmRCallProc          XtRCallProc
#define XmRCallback          XtRCallback
#define XmRCardinal          XtRCardinal
#define XmRColor             XtRColor
#define XmRColormap          XtRColormap
#define XmRCursor           XtRCursor
#define XmRDimension         XtRDimension
#define XmRDisplay           XtRDisplay
#define XmREditMode          XtREditMode
#define XmREnum              XtREnum
#define XmRFile              XtRFile
#define XmRFloat             XtRFloat
#define XmRFont              XtRFont
#define XmRFontSet           XtRFontSet
#define XmRFontStruct        XtRFontStruct
#define XmRFunction          XtRFunction
#define XmRGeometry          XtRGeometry
#define XmRImmediate         XtRImmediate
#define XmRInitialState      XtRInitialState
#define XmRInt               XtRInt
#define XmRJustify           XtRJustify
#define XmRLongBoolean       XtRLongBoolean
#define XmROrientation       XtROrientation
#define XmRObject            XtRObject
#define XmRPixel             XtRPixel
#define XmRPixmap            XtRPixmap
#define XmRPointer           XtRPointer
#define XmRPosition          XtRPosition
#define XmRScreen            XtRScreen
#define XmRShort             XtRShort
#define XmRString            XtRString
#define XmRStringArray       XtRStringArray
#define XmRStringTable       XtRStringTable
#define XmRTextPosition      XtCTextPosition
#define XmRTranslationTable  XtRTranslationTable
#define XmRUnsignedChar      XtRUnsignedChar
#define XmRVisual            XtRVisual
#define XmRWidget            XtRWidget
#define XmRWidgetClass       XtRWidgetClass
#define XmRWidgetList        XtRWidgetList
#define XmRWindow            XtRWindow

```

Windowing and Terminal Interface (64-bit psABI) - EXPERIMENTAL

Overview

The following table identifies the actual version numbers and reference names for 64-bit ABI windowing and terminal shared interfaces:

Library	Reference Name
libMrm	<code>/usr/dt/lib/sparcv9/libMrm.so.3</code>
libX11	<code>/usr/lib/sparcv9/libX11.so.5</code>
libXext	<code>/usr/lib/sparcv9/libXext.so.0</code>
libXol	<code>/usr/lib/sparcv9/libXol.so.3</code>
libXm	<code>/usr/dt/lib/sparcv9/libXm.so.3</code>
libXt	<code>/usr/lib/sparcv9/libXol.so.5</code>

CHAPTER 11: Development Environments

SCD
2.4

Development Environments

Overview

This chapter is common to both the 32-bit ABI and 64-bit ABI, except that the 64-bit ABI is EXPERIMENTAL.

It contains the commands for application programs as listed in the *System V Application Binary Interface (Third Edition)*, and described in the *System V Interface Definition, (Third Edition)*.

Table 11-1. Software Packaging tools

pkgproto pkgtrans pkgmk

Development Environments Changes

#	Facility	Location	Description
1	Commands	gABI	Remove command section page 11-1, 11-2, 11-3: The SCD places no requirements on the nature of the development tools, if any, that are provided on a system. In particular, the six commands as, cc, ld, m4, lex, and yacc are not part of the SCD.

SCD

2.4

Networking

Overview

All information regarding File System Structure and Contents may be found in Chapter 12 of the *System V Application Binary Interface (Third Edition)*.

Networking Changes

The following are changes to the *System V Application Binary Interface (Third Edition)*, the *System V Application Binary Interface - SPARC Processor Supplement (Third Edition)*, and the *System V Interface Definition (Third Edition)* as reported to SPARC International.

#	Facility	Location	Description
1	Required STREAMS Devices and Modules	gABI	Addition - To Figure 12-1 on page 12-2, add <i>/dev/icmp</i> and <i>/dev/arp</i> .



SCD

2.4

Symbols

6P-4	%f28 3P-11
% 3P-11, 3P-22	%f29 3P-10, 3P-11
%29 3P-11	%f3 3P-10, 3P-11, 3P-14, 3P-15
%4 3P-11	%f30 3P-11
%asi 3P-7, 3P-10, 3P-21, 3P-22, 3P-24	%f31 3P-10, 3P-11
%ccr 3P-7, 3P-10, 3P-21, 3P-22	%f5 3P-11
%d 3P-12	%f6 3P-11
%d0 3P-10, 3P-11, 3P-13, 3P-15, 3P-22	%f7 3P-10, 3P-11, 3P-14
%d0..%d10 3P-12	%f8 3P-11
%d10 3P-11	%f9 3P-11
%d12 3P-11	%fp 3P-6, 3P-10, 3P-25
%d12..%d30 3P-12	%fp+BIAS 3P-33
%d14 3P-11	%fp+BIAS+176 3P-14
%d16 3P-11, 3P-15	%fp+BIAS+184 3P-14
%d18 3P-11	%fprs 3P-10, 3P-22
%d2 3P-10, 3P-11	%fpsr 3P-7
%d20 3P-11, 3P-14	%fsr 3P-7, 3P-10, 3P-22
%d22 3P-11	%g0 3P-7, 3P-10, 3P-15, 3P-21, 3P-30, 3P-32
%d24 3P-11	%g1 3P-7, 3P-10, 3P-20, 3P-25, 5P-5
%d26 3P-11	%g2 3P-7, 3P-10
%d28 3P-11	%g3 3P-7, 3P-10
%d30 3P-10, 3P-11	%g4 3P-7, 3P-10
%d32 3P-10	%g5 3P-7, 3P-10, 3P-32
%d4 3P-11, 3P-14	%g6 3P-7, 3P-10
%d6 3P-10, 3P-11, 3P-15	%g7 3P-7, 3P-10
%d62 3P-10	%hh 3P-30, 4-2, 4P-8
%d8 3P-11, 3P-14	%hi 5P-5
%f 3P-12	%hm 4-2, 4P-8
%f0 3P-10, 3P-11, 3P-13, 3P-15	%i 3P-12
%f1 3P-10, 3P-11, 3P-14	%i0 3P-6, 3P-8, 3P-10, 3P-13, 3P-14, 3P-15
%f10 3P-11	%i0..%i5 3P-12
%f11 3P-11, 3P-14, 3P-15	%i1 3P-6, 3P-8, 3P-10, 3P-13, 3P-14
%f12 3P-11	%i2 3P-6, 3P-8, 3P-10, 3P-14, 3P-15
%f13 3P-11, 3P-14	%i3 3P-6, 3P-8, 3P-10, 3P-14
%f14 3P-11	%i4 3P-6, 3P-8, 3P-14, 3P-15
%f15 3P-11	%i5 3P-6, 3P-8, 3P-10, 3P-14
%f16 3P-11	%i6 3P-6, 3P-8, 3P-10
%f17 3P-11	%i7 3P-6, 3P-8, 3P-10
%f18 3P-11	%i7+8 3P-10, 3P-15
%f19 3P-11	%i0 3-2, 3P-6, 3P-8, 3P-10, 3P-22, 3P-32
%f2 3P-10, 3P-11	%i1 3-2, 3P-6, 3P-8, 3P-21, 3P-32
%f20 3P-11	%i2 3P-6, 3P-8
%f21 3P-11	%i3 3P-6, 3P-8
%f22 3P-11	%i4 3P-6, 3P-8, 3P-15
%f23 3P-11	%i44 4P-9
%f24 3P-11	%i5 3P-6, 3P-8
%f25 3P-11	%i6 3P-6, 3P-8, 3P-21
%f26 3P-11	%i7 3P-6, 3P-8, 3P-10, 3P-21, 3P-30, 3P-31
%f27 3P-11	%lm 3P-30, 4-2, 4P-8
	%npc 3P-21

%o 3P-12	%r29 3P-6
%o0 3-1, 3P-6, 3P-10, 3P-11, 3P-12, 3P-13, 3P-14, 3P-15, 3P-21, 3P-22, 3P-30	%r3 3P-7
%o1 3-1, 3P-6, 3P-10, 3P-11, 3P-14, 3P-21, 3P-30	%r30 3P-6
%o2 3P-6, 3P-10, 3P-11, 3P-14, 3P-15, 3P-22	%r31 3P-6
%o3 3P-6, 3P-10, 3P-11, 3P-14, 3P-22	%r4 3P-7
%o4 3P-6, 3P-11, 3P-14, 3P-15, 3P-22	%r5 3P-7
%o5 3P-6, 3P-10, 3P-11, 3P-12, 3P-14	%r6 3P-7
%o6 3P-6, 3P-10	%r7 3P-7
%o7 3P-6, 3P-10, 3P-15	%r8 3P-6
%o7+8 3P-15	%r9 3P-6
%pc 3P-21, 3P-24, 3P-32	%sp 3P-6, 3P-9, 3P-10, 3P-21, 3P-25
%q0 3P-7, 3P-10, 3P-11, 3P-13, 3P-15	%sp+BIAS 3P-33
%q0.%q8 3P-12	%sp+BIAS+128 3P-11, 3P-12
%q12 3P-7, 3P-11	%sp+BIAS+136 3P-11
%q16 3P-7, 3P-11, 3P-14	%sp+BIAS+144 3P-11
%q20 3P-7, 3P-11	%sp+BIAS+152 3P-11
%q24 3P-7, 3P-11, 3P-14	%sp+BIAS+160 3P-11
%q28 3P-7, 3P-10, 3P-11	%sp+BIAS+168 3P-11, 3P-12
%q32 3P-7, 3P-10	%sp+BIAS+176 3P-11, 3P-12
%q36 3P-7	%sp+BIAS+184 3P-11
%q4 3P-7, 3P-10, 3P-11	%sp+BIAS+192 3P-11
%q40 3P-7	%sp+BIAS+200 3P-11, 3P-34
%q44 3P-7	%sp+BIAS+208 3P-11
%q48 3P-7	%sp+BIAS+216 3P-11
%q52 3P-7	%sp+BIAS+224 3P-11
%q56 3P-7	%sp+BIAS+232 3P-11
%q60 3P-7, 3P-10	%sp+BIAS+240 3P-11
%q8 3P-7, 3P-11	%sp+BIAS+248 3P-11, 3P-12
%r0 3P-7	%tick 3P-24
%r1 3P-7	%xcc 3P-32
%r10 3P-6	%y 3P-7, 3P-10, 3P-21, 3P-22, 3P-24
%r11 3P-6	.div 6-27, 6-32
%r12 3P-6	.got 4P-3
%r13 3P-6	.Lcasei 3P-32
%r14 3P-6	.Ldef 3P-32
%r15 3P-6	.PLT1 5P-6
%r16 3-2, 3P-6	.pltN 5P-7
%r17 3-2, 3P-6	.rela 4P-3
%r18 3P-6	/dev 9-1
%r19 3P-6	/dev/ 9-1
%r2 3P-7	/dev/arp 7-1, 12-1
%r20 3P-6	/dev/console 9-1
%r21 3P-6	/dev/icmp 7-1, 12-1
%r22 3P-6	/dev/lpX 9-1
%r23 3P-6	/dev/mt 9-1
%r24 3P-6	/dev/null 2-1, 9-1
%r25 3P-6	/dev/rmt 9-1
%r26 3P-6	/dev/tcp 7-1
%r27 3P-6	/dev/tty 9-1
%r28 3P-6	/dev/udp 7-1

/dev/zero 9-1
 /etc/group 6-4
 /etc/passwd 6-4
 /home/dir 3P-26
 /home/dir/bin
 /usr/bin 3P-26
 /usr/bin 7-1
 /usr/dt/lib/libMrm.so.1.2 10-1, 10-55
 /usr/dt/lib/libMrm.so.3 10-1, 10-55
 /usr/dt/lib/libXm.so.1.2 10-1
 /usr/dt/lib/libXm.so.1.2, 10-55
 /usr/dt/lib/libXm.so.3 10-1, 10-55
 /usr/dt/lib/sparcv9/libMrm.so.3 10P-1
 /usr/dt/lib/sparcv9/libXm.so.3 10P-1
 /usr/lib 5-1, 5P-3
 /usr/lib/ld.so.1 6-9
 /usr/lib/lib/libXrm.so.1.2 10-1
 /usr/lib/libaio.so.1 6-24
 /usr/lib/libc.so.1 6-9, 6-26
 /usr/lib/libdl.so.1 6-9, 6-45
 /usr/lib/libelf.so.1 6-9, 6-46
 /usr/lib/libintl.so.1 6-9, 6-50
 /usr/lib/libm.so.1 6-9, 6-51
 /usr/lib/libMrm.so.1.2 10-55
 /usr/lib/libMrm.so.3 10-1, 10-55
 /usr/lib/libnisdb.so.1 6-9, 6-52
 /usr/lib/libnsl.so.1 6-9, 6-54, 6P-1
 /usr/lib/libposix4.so.1 6-59
 /usr/lib/libpthread.so.1 6-60
 /usr/lib/libresolv.so.1 6-9, 6-64, 6P-1
 /usr/lib/librpcsvc.so.1 6-9, 6-60, 6-66
 /usr/lib/libsocket.so.1 6-9, 6-67, 6P-1
 /usr/lib/libthread.so.1 6-9, 6-77, 6P-1
 /usr/lib/libucb.so.1 6-9, 6-80
 /usr/lib/libw.so.1 6-9, 6-81, 6P-1
 /usr/lib/libX11.5 10-2
 /usr/lib/libX11.so.4 10-1, 10-2
 /usr/lib/libX11.so.5 10-1, 10-2
 /usr/lib/libXext.so.0 10-1, 10-26
 /usr/lib/libXm.so.1.2 10-1, 10-55
 /usr/lib/libXm.so.3 10-1, 10-55
 /usr/lib/libXol.so.3 10-1, 10-48
 /usr/lib/libXt.so.4 10-1, 10-27
 /usr/lib/libXt.so.5 10-1, 10-27
 /usr/lib/sparcv9 5P-3
 /usr/lib/sparcv9/ld.so.1 5P-8
 /usr/lib/sparcv9/libX11.so.5 10P-1
 /usr/lib/sparcv9/libXext.so.0 10P-1
 /usr/lib/sparcv9/libXol.so.3 10P-1
 /usr/lib/sparcv9/libXol.so.5 10P-1
 /usr/sbin 7-1
 /var/tmp/ 6P-11
 _ 6-22
 __errno 6-23, 6-77, 6-78
 __align_cpy_n 3P-14
 __builtin_alloca 6-32
 __data 6P-9
 __dtou 6-3
 __ftoll 6-28
 __ftou 6-3
 __huge_val 6-27
 __job 6-27
 __pdata 6P-9
 __posix_asctime_r 6-22, 6-23
 __posix_ctime_r 6-22, 6-23
 __posix_getgrgid_r 6-22, 6-23
 __posix_getgrnam_r 6-22, 6-23
 __posix_getlogin_r 6-22, 6-23
 __posix_getpwnam_r 6-22, 6-23
 __posix_getpwuid_r 6-22, 6-23
 __posix_readdir_r 6-22, 6-23
 __posix_sigwait 6-22, 6-23
 __posix_ttyname_r 6-22, 6-23
 __priocntl 6-32
 __proc 6P-9
 __prof 6P-9
 __pthread_rwlock_magic 6-62
 __pthread_rwlock_pad1 6-62
 __pthread_rwlock_pad2 6-62
 __pthread_rwlock_pad3 6-62
 __pthread_rwlock_readers 6-62
 __pthread_rwlock_type 6-62
 __signal 6P-9
 __svcxprt 6P-7
 _altzone 6-4, 6-27
 _CS_LFS_CFLAGS 6-16
 _CS_LFS_LDFLAGS 6-16
 _CS_LFS_LIBS 6-16
 _CS_LFS_LINTFLAGS 6-16
 _CS_LFS64_CFLAGS 6-16
 _CS_LFS64_LDFLAGS 6-16
 _CS_LFS64_LIBS 6-16
 _CS_LFS64_LINTFLAGS 6-16
 _CS_PATH 6-43
 _ctype 6-27
 _daylight 6-27
 _dummy 10-11
 _DYNAMIC 5G-1, 5P-3, 5P-6
 _environ 6-3, 6-27
 _euw1 6-81

- _eucw2 6-81
- _eucw3 6-81
- _FILE_OFFSET_BITS 6-10, 6-11, 6-14, 6-15
- _FSTYPSZ 6P-10
- _getdate_err 6-27
- _GLOBAL_OFFSET_TABLE_ 3P-28, 5P-4
- _job 6-27
- _JBLLEN 6P-8
- _longimp 6-80
- _multibyte 6-81
- _nderror 6-54
- _NGREG 6P-14
- _null_auth 6-54
- _numeric 6-27
- _once 6-62
- _PC_FILESIZEBITS 6-16
- _pcw 6-81
- _POSIX_NO_TRUNC 6-4
- _prioset 6-32
- _pthread_attr 6-62
- _pthread_cond 6-62
- _pthread_condattr 6-62
- _pthread_mutex 6-62
- _pthread_mutexattr 6-62
- _Q_add 6-32
- _Q_qtou 6-3
- _Q_sub 6-32
- _REENTRANT 6-78
- _res 6-65
- _S_un 6P-16
- _Screen 10-11
- _scrw1 6-81
- _scrw2 6-81
- _scrw3 6-81
- _setjmp 6-80
- _SIGJBLLEN 6P-8
- _ST_FSTYPSZ 6-11, 6-15, 6P-10
- _timezone 6-27
- _tzname 6-27
- _XAllocScratch 10-6
- _XcmsCCC 10-10
- _Xdebug 10-6
- _XDisplay 10-11
- _XExtData 10-11
- _XFlush 10-6
- _XFlushGCCache 10-6
- _XFontSet 10-11
- _XFTWVER 6-33
- _XGC 10-11
- _XIC 10-12
- _XIM 10-12
- _XRead 10-6
- _XReadPad 10-6
- _XRegion 10-12
- _XReply 10-6
- _XrmHashBucketRec 10-11
- _XSend 10-6
- _XSetLastRequestRead 10-6
- _XtInherit 10-42
- _XtInheritTranslations 10-32
- “Application Binary Objects Record Library Names They Depend On” on page 18 6-17
- Numerics**
- 64 bit sparc 6-5
- 64-bit ABI 1-2
- 64-bit integer 6-26
- A**
- a_fcn 3P-26
- A_PROB 6-57
- a_ptr 3P-26
- A_REBOOT 6-37
- A_REMOUNT 6-37
- A_SHUTDOWN 6-37
- a_type 3P-25, 3P-26
- a_un 3P-26
- a_val 3P-26
- abbrevMenuBarClassRec 10-50
- abbrevMenuBarWidgetClass 10-50
- ABI 6-10
 - generic 1-7
 - processor specific 1-7
- ABI Errata 2-1
- abort 6-6
- accepted_reply 6P-7
- acos 6-51
- acosh 6-51
- Active Users 7-3
- AD_BOOT 6-37
- AD_HALT 6-37
- AD_IBOOT 6-37
- add 3P-28, 3P-32
- Address Resolution Protocol 7-2
- AEXC 3P-10
- aexc 3P-23
- AF_802 6-74, 6-75
- AF_APPLETALK 6-74, 6-75
- AF_CCITT 6-74
- AF_CHAOS 6-74
- AF_DATAKIT 6-74
- AF_DECnet 6-74
- AF_DLI 6-74, 6-75

AF_ECMA 6-74
 AF_GOSIP 6-74, 6-75
 AF_HYLINK 6-74, 6-75
 AF_IMPLINK 6-74
 AF_INET 6-74, 6-75
 AF_LAT 6-74, 6-75
 AF_MAX 6-74, 6-75
 AF_NBS 6-74
 AF_NIT 6-74, 6-75
 AF_NS 6-74
 AF_OSI 6-74, 6-75
 AF_OSINET 6-74, 6-75
 AF_PUP 6-74
 AF_SNA 6-74
 AF_UNIX 6-74
 AF_UNSPEC 6-74
 AF_X25 6-74, 6-75
 AFI 6-74
 aio.h 6-13
 aio_cancel 6-59
 aio_cancel64 6-12, 6-59
 aio_erno 6-25
 aio_error 6-59
 aio_error64 6-12
 aio_fsync 6-59
 aio_fsync64 6-12
 AIO_INPROGRESS 6-25
 aio_offset 6-13
 aio_read 6-59
 aio_read64 6-12
 aio_result_t 6-25
 aio_return 6-25, 6-59
 aio_return64 6-12
 aio_suspend 6-59
 aio_suspend64 6-12
 aio_write 6-59
 aio_write64 6-12
 aiocb 6-13
 aiocb64 6-13
 aioread64 6-12
 aiowrite64 6-12
 alloca 6-32
 alloca.h 6-32
 AllocateTextBuffer 10-49
 alphasort 6-80
 alphasort64 6-12, 6-80
 altzone 6-27
 AM 3P-24
 anedW.h 10-65
 ANSI 1-6
 ANSI C 3P-2, 3P-12, 3P-27
 Any private RJE service 7-3
 Apple Talk 6-74
 Application Binary Interface 1-1
 Application Program Commands 8-1
 ApplicationShellClassPart 10-46
 ApplicationShellClassRec 10-46
 applicationShellClassRec 10-32
 ApplicationShellPart 10-46
 ApplicationShellRec 10-47
 ApplicationShellWidget 10-47
 ApplicationShellWidgetClass 10-36
 applicationShellWidgetClass 10-32
 ar 8-1
 ar_date 6-49
 ar_gid 6-49
 ar_mode 6-49
 ar_name 6-49
 ar_rawname 6-49
 ar_size 6-49
 ar_uid 6-49
 AREGTYPE 8-2
 Arg 10-34
 argc 3P-22
 ArgList 10-34
 argv 3P-22
 ARP 7-1
 arp 7-2
 arpa/nameser.h 6-64
 as 11-1
 AS_CMD 2-1
 as_hash 6-49
 as_name 6-49
 as_off 6-49
 ascount 6-65
 asctime_r 6-22, 6-23
 ASI 3P-1, 3P-24
 ASI_PRIMARY 3P-16
 ASI_PRIMARY_LITTLE 3P-16
 ASI_PRIMARY_NOFAULT 3P-11, 3P-16, 3P-21, 3P-24
 ASI_PRIMARY_NOFAULT_LITTLE 3P-16
 asin 6-51
 asinh 6-51
 asrset_t 6P-14
 async_data_error 3P-18, 3P-20
 Asynchronous File I/O 6-24
 atan 6-51
 atan2 6-51
 atanh 6-51
 atexit 3P-25

- attr 6-57
- AU_CMD 2-1, 8-3
- audience i
 - application developers i
 - system implementors i
- auth_destroy 6-57, 6-58
- authdes_cred 6P-7
- authdes_fullname 6P-7
- authsys_create_default 6-55
- auxv_t 3P-26
- B**
- BA_ENV 6-26
- BA_LIB 3P-34, 6-4, 6-7, 6-26, 7-1
- BA_OS 3P-11, 3P-16, 3P-17, 3P-18, 3P-22, 3P-25, 6-4, 6-5, 6-26
- BackwardScanTextBuffer 10-49
- BAD_SYS 6-57
- basename 8-1
- baseWindowShellClassRec 10-50
- baseWindowShellWidgetClass 10-50
- baud 6-57
- bcmp 6-80
- bcopy 6-80
- BIAS 3P-8, 3P-10, 3P-12, 3P-25, 3P-33
- binary 1-1
- bind 6-67
- bindtextdomain 6-50
- Bit-Fields 3-2
- BitmapPad
 - unsafe, use XBitmapPad 10-7
- BitmapUnit
 - unsafe, use XBitmapUnit 10-7
- BlackPixel
 - unsafe, use XBlackPixel 10-7
- BlackPixelOfScreen
 - unsafe, use XBlackPixelOfScreen 10-7
- blkcnt_t 6-11, 6-15
- blkcnt64_t 6-15
- BLKTYPE 8-2
- BlockTable 10-54
- BOGUS_OBJ 6-70
- bool_t 6-58, 6P-8
- Boolean 10-33
- BOOT_TIME 6-44
- BSD 6-2, 7-2
- BSDNET 7-2
- BSP 6-74
- BU_CMD 2-1, 8-2, 8-3
- Buffer 10-54
- BufferElement 10-54
- BulletinB.h 10-59
- bulletinBoardClassRec 10-50
- bulletinBoardWidgetClass 10-50
- BUS_ADDRERR 3P-21
- BUS_OBJERR 3P-21
- buttonClassRec 10-50
- buttonGadgetClass 10-50
- buttonGadgetClassRec 10-50
- buttonWidgetClass 10-50
- byte8 4P-6
- bzero 6-80
- C**
- caddr_t 6-8, 6-58, 6P-8, 6P-11, 6P-13
- CALL 3P-28, 6-57
- call 3P-28
- call_body 6P-7
- captionClassRec 10-50
- CaptionWidget 10-52
- CaptionWidgetClass 10-52
- captionWidgetClass 10-50
- Cardinal 10-33
- CAS 3P-21
- CascadeB.h 10-59
- CascadeBG.h 10-59
- cat 8-1
- categoryClassRec 10-50
- CategoryWidget 10-52
- CategoryWidgetClass 10-52
- categoryWidgetClass 10-50
- cbirt 6-51
- cc 11-1
- cc_t 6P-11
- CCITT 6-74
- CCITT X.25 6-74
- CCR 3P-11, 3P-20, 3P-22, 3P-24
- cd 8-1
- CD-ROM 2-1, 2-2
- CDS 6-70
- ceil 6-51
- CellsOfScreen
 - unsafe, use XCellsOfScreen 10-7
- cexc 3P-23
- char 6-11
- Character Generator 7-3
- chargen 7-3
- chars 10-12
- checkBoxClassRec 10-50
- CheckBoxWidget 10-52
- CheckBoxWidgetClass 10-52
- checkBoxWidgetClass 10-50
- chgrp 8-1

chmod 8-1
 chown 8-1
 CHRTYPE 8-2
 CLE 3P-24
 ClientWhitePointOfCCC
 unsafe, use XClientWhitePointOfCCC 10-7
 clientWhitePt 10-10
 clnt_create_vers 6-55
 clnt_perror 6-55
 clnt_sperror 6-55
 clnt_tli_create 6-55
 clnt_vc_create 6-55
 clnttcp_create 6-55
 clntudp_bufcreate 6-55
 clntudp_create 6-55
 clock_getres 6-59
 clock_gettime 6-59
 clock_settime 6-59
 clock_t 6P-12
 cmp 8-1
 colorConvertArgs 10-32
 Command.h 10-59
 commands 11-1
 COMMON 3P-2
 CompositeClassExtension 10-40
 CompositeClassExtensionRec 10-40
 CompositeClassPart 10-40
 CompositeClassRec 10-40
 compositeClassRec 10-32
 CompositePart 10-40
 CompositePartPtr 10-40
 CompositePtr 10-40
 CompositeRec 10-40
 CompositeWidget 10-33
 CompositeWidgetClass 10-33
 compositeWidgetClass 10-32
 compress 8-1
 cond_broadcast 6-23, 6-78
 cond_destroy 6-23, 6-78
 cond_init 6-23, 6-78
 cond_signal 6-23, 6-78
 cond_t 6-79
 cond_timedwait 6-23, 6-78
 cond_wait 6-23, 6-78
 connect 6-67
 connect 6-67
 ConnectionNumber
 unsafe, use XConnectionNumber 10-7
 ConstraintClassExtension 10-41
 ConstraintClassExtensionRec 10-41
 ConstraintClassPart 10-41
 ConstraintClassRec 10-41
 constraintClassRec 10-32
 ConstraintPart 10-41
 ConstraintRec 10-41
 ConstraintWidget 10-41
 ConstraintWidgetClass 10-33
 constraintWidgetClass 10-32
 contents of SCD 2.2 i
 ControlAreaWidget 10-52
 ControlAreaWidgetClass 10-52
 controlAreaWidgetClass 10-50
 controlClassRec 10-50
 ControlLayout 10-52
 ControlMapIndex 10-10
 CONTTYPE 8-2
 CopyBuffer 10-49
 copysign 6-51
 CopyTextBufferBlock 10-49
 CoreClassPart 10-42
 CoreClassRec 10-42
 CorePart 10-41
 CoreRec 10-41
 CoreWidget 10-33
 CoreWidgetClass 10-33
 coreWidgetClass 10-32
 cos 6-51
 cosh 6-51
 cp 8-1
 cpio 2-1, 8-1
 creat64 6-12, 6-23, 6-28
 crypt 6-4
 ctime 6-4
 ctime_r 6-22, 6-23
 CutPaste.h 10-59
D
 d K&R 3P-12
 d_align 6-48
 d_buf 6-48
 D_HUNG 6-57
 d_ino 6-14, 6-36
 d_name 6-14, 6-36
 d_off 6-14, 6-36, 6-48, 6-49
 d_ptr 5G-1, 5P-2, 6-49
 d_reclen 6-14, 6-36
 d_size 6-48
 d_tag 5G-1, 6-49
 d_type 6-48
 d_un 5G-1, 6-49
 d_val 5G-1, 6-49

- d_version 6-48
- Data Structures
 - Motif 10-59, 10-60, 10-61, 10-62, 10-63, 10-64, 10-65, 10-66, 10-67
 - Xol 10-52
- data_access_error 3P-18, 3P-20
- data_access_exception 3P-18, 3P-20
- data_access_MMU_miss 3P-18, 3P-20
- data_access_protection 3P-18, 3P-20
- date 8-1
- DATLOCK 6-38
- datum 6-34
- daylight 3P-29, 6-27
- Daytime 7-3
- daytime 7-3
- db_action 6-53
- DB_ADD 6-53
- db_add_entry 6-52
- DB_ALL 6-53
- DB_BADOBJECT 6-53
- DB_BADQUERY 6-53
- DB_BADTABLE 6-53
- db_checkpoint 6-52
- db_create_table 6-52
- db_destroy_table 6-52
- DB_FIRST 6-53
- db_first_entry 6-52
- db_free_result 6-52
- db_initialize 6-52
- DB_INTERNAL_ERROR 6-53
- db_list_entries 6-52
- DB_LOOKUP 6-53
- DB_MEMORY_LIMIT 6-53
- DB_NEXT 6-53
- db_next_desc 6-53
- db_next_desc_len 6-53
- db_next_desc_val 6-53
- db_next_entry 6-52
- DB_NOTFOUND 6-53
- DB_NOTUNIQUE 6-53
- DB_REMOVE 6-53
- db_remove_entry 6-52
- DB_RESET_NEXT 6-53
- db_reset_next_entry 6-52
- db_result 6-53
- db_standby 6-52
- db_status 6-53
- DB_STORAGE_LIMIT 6-53
- DB_SUCCESS 6-53
- db_table_exists 6-52
- DBL_MANT_DIG 6-51
- DBL_MAX_EXP 6-51
- DBM 6-34
- dbm_clearerr 6-34
- dbm_error 6-34
- dcgettext 6-50
- dd 2-1, 8-1
- DEAD_PROCESS 6-44
- decimal_form 6-32
- decimal_mode 6-32
- decimal_record 6-32
- decimal_string_form 6-32
- DECIMAL_STRING_LENGTH 6-32
- DECnet 6-74
- DefaultColormap
 - unsafe, use XDefaultColormap 10-7
- DefaultColormapOfScreen
 - unsafe, use XDefaultColormapOfScreen 10-7
- DefaultDepth
 - unsafe, use XDefaultDepth 10-7
- DefaultDepthOfScreen
 - unsafe, use XDefaultDepthOfScreen 10-7
- DefaultGC
 - unsafe, use XDefaultGC 10-7
- DefaultGCOfScreen
 - unsafe, use XDefaultGCOfScreen 10-7
- DefaultRootWindow
 - unsafe, use XDefaultRootWindow 10-7
- DefaultScreen
 - unsafe, use XDefaultScreen 10-7
- DefaultScreenOfDisplay
 - unsafe, use XDefaultScreenOfDisplay 10-7
- DefaultVisual
 - unsafe, use XDefaultVisual 10-7
- DefaultVisualOfScreen
 - unsafe, use XDefaultVisualOfScreen 10-7
- defname 6-65
- definition of audience i
- definition of purpose i
- delta 10-12
- Dependencies 6-9
- DEPRECATED 1-3, 1-7
- Depth 10-11
- dev_len 6-57
- dev_t 6-10, 6-15, 6-42, 6-76, 6P-12
- device 6-57
- df 8-1
- dgettext 6-50
- dial.h 6-57
- DialogS.h 10-60
- Digital Press 10-1

- Dimension 10-33
- DIR 6-8
- DIRECTORY_OBJ 6-70
- directory_obj 6-71
- dirent 6-36
- dirent.h 6-8, 6-14, 6-22
- dirent64 6-14
- dirname 8-1
- DIRTY 8-2
- DisableScreenInterval 10-10
- DisableScreenSaver 10-10
- Discard 7-3
- discard 7-3
- Display 10-10
- Display 10-11
- Display.h 10-60
- DisplayCells
 - unsafe, use XDisplayCells 10-7
- DisplayHeight
 - unsafe, use XDisplayHeight 10-7
- DisplayHeightMM
 - unsafe, use XDisplayHeightMM 10-7
- DisplayOfCCC
 - unsafe, use XDisplayOfCCC 10-7
- DisplayOfScreen
 - unsafe, use XDisplayOfScreen 10-7
- DisplayPlanes
 - unsafe, use XDisplayPlanes 10-7
- DisplayString
 - unsafe, use XDisplayString 10-7
- DisplayWidth
 - unsafe, use XDisplayWidth 10-8
- DisplayWidthMM
 - unsafe, use XDisplayWidthMM 10-8
- division_by_zero 3P-18, 3P-20
- DL 3P-23
- DL_info 6-45
- dl_info 6-45
- dladdr 6-45
- dlclose 6-45
- dlerror 6-45
- dlfcn.h 6-45
- dli_fbase 6-45
- dli_fname 6-45
- dli_saddr 6-45
- dli_sname 6-45
- dlopen 6-45
- dlsym 6-45
- dn_comp 6-64
- dn_expand 6-64
- DNANS 6-70
- DNS 6-70
- dnsrch 6-65
- do_armask 6-71
- do_servers 6-71
- doconfig 6-55
- DoesBackingStore
 - unsafe, use XDoesBackingStore 10-8
- DoesSaveUnders
 - unsafe, use XDoesSaveUnders 10-8
- domain 7-3
- Domain Name Server 7-3
- dpy 10-10
- DragC.h 10-60
- DragIcon.h 10-61
- DragOverS.h 10-61
- DropSMgr.h 10-62
- DT_JMP_REL 5P-2, 5P-6
- DT_NEEDED 5-1, 6-9, 6-77
- DT_PLTGOT 5P-2
- DT_REGISTER 5P-3
- DT_RPATH 5-1
- DT_SONAME 6-17
- DT_SPARC_REGISTER 5P-3
- DU 3P-23
- DV_NT_A 6-57
- DV_NT_E 6-57
- DV_NT_K 6-57
- Dynamic Linking 5-1
- dynamic_linker_0 5P-6
- dynamic_linker_1 5P-6
- Dynamically linking, something 5-1
- E**
- e_ehsize 4G-1, 6-48
- e_entry 4G-1, 6-48
- e_exit 6-44
- e_flags 4G-1, 4P-1, 6-48
- e_ident 4P-1, 6-48
- e_machine 4G-1, 4P-1, 6-48
- e_phentsize 4G-1, 6-48
- e_phnum 4G-1, 6-48
- e_phoff 4G-1, 6-48
- e_shentsize 4G-1, 6-48
- e_shnum 4G-1, 6-48
- e_shoff 4G-1, 6-48
- e_shstrndx 4G-1, 6-48
- e_termination 6-44
- e_type 4G-1, 6-48
- e_version 4G-1, 6-48
- EADDRINUSE 6-68

EADDRNOTAVAIL 6-68
EAFNOSUPPORT 6-68
EAGAIN 6-3, 6-5, 6-68
EALREADY 6-68
ec_flags 6-71
ec_value 6-71
Echo 7-3
echo 7-3, 8-1
ECONNREFUSED 6-68
ed 8-1
edcompatible 8-2
EditResult 10-51
EF_SPARC_EXT_MASK 4P-1
EF_SPARC_HAL_R1 4P-1
EF_SPARC_SUN_US1 4P-1
EF_SPARC_SUN_US3 4P-1
EF_SPARC64_MM 4P-1
EF_SPARC64_PSO 4P-1
EF_SPARC64_RMO 4P-1
EF_SPARC64_TSO 4P-1
EF_SPARC64bit_MM 4P-1
EL_CLASS 4P-1, 6-47
EL_DATA 4P-1, 6-47
EL_MAG0 6-47
EL_MAG1 6-47
EL_MAG2 6-47
EL_MAG3 6-47
e_ident 4G-1
EL_NIDENT 4G-1, 6-47, 6-48
EL_VERSION 6-47
EINPROGRESS 6-68
EINVAL 3P-20
EISCONN 6-68
ELF 3P-24, 4P-1, 4P-4, 6-1, 6-2, 6-17
Elf 6-47
Elf_Arhdr 6-49
Elf_Arsym 6-49
elf_begin 6-46
ELF_C_CLR 6-48
ELF_C_FDDONE 6-48
ELF_C_FDREAD 6-48
ELF_C_NULL 6-48
ELF_C_NUM 6-48
ELF_C_RDWR 6-48
ELF_C_READ 6-48
ELF_C_SET 6-48
ELF_C_WRITE 6-48
Elf_Cmd 6-48
elf_cntl 6-46
Elf_Data 6-48
elf_end 6-46
elf_errmsg 6-46
elf_errno 6-46
ELF_F_DIRTY 6-47
ELF_F_LAYOUT 6-47
elf_fill 6-46
elf_flagdata 6-46
elf_flagehdr 6-46
elf_flagelf 6-46
elf_flagphdr 6-46
elf_flagscn 6-46
elf_flagsshr 6-46
elf_getarhdr 6-46
elf_getarsym 6-46
elf_getbase 6-46
elf_getdata 6-46
elf_getident 6-46
elf_getscn 6-46
elf_hash 6-5, 6-46
ELF_K_AR 6-49
ELF_K_COFF 6-49
ELF_K_ELF 6-49
ELF_K_NONE 6-49
ELF_K_NUM 6-49
Elf_Kind 6-49
elf_kind 6-46
elf_ndxscn 6-46
elf_newdata 6-46
elf_newscn 6-46
elf_next 6-46
elf_nextscn 6-46
elf_rand 6-46
elf_rawdata 6-46
elf_rawfile 6-46
Elf_Scn 6-47
elf_strptr 6-46
ELF_T_ADDR 6-48
ELF_T_BYTE 6-48
ELF_T_DYN 6-48
ELF_T_EHDR 6-48
ELF_T_HALF 6-48
ELF_T_NUM 6-48
ELF_T_OFF 6-48
ELF_T_PHDR 6-48
ELF_T_REL 6-48
ELF_T_RELA 6-48
ELF_T_SHDR 6-48
ELF_T_SWORD 6-48
ELF_T_SYM 6-48
ELF_T_WORD 6-48

Elf_Type 6-48
elf_update 6-46
elf_version 6-46
Elf_Void 6-47, 6-48
Elf32_Addr 6-47, 6-48, 6-49
Elf32_Dyn 6-49
Elf32_Ehdr 6-48
elf32_fsize 6-46
ELF32_FSZ_ADDR 6-47
ELF32_FSZ_HALF 6-47
ELF32_FSZ_OFF 6-47
ELF32_FSZ_SWORD 6-47
ELF32_FSZ_WORD 6-47
elf32_getehdr 6-46
elf32_getphdr 6-46
elf32_getshdr 6-46
Elf32_Half 6-47, 6-48, 6-49
elf32_newehdr 6-46
elf32_newphdr 6-46
Elf32_Off 6-47, 6-48, 6-49
Elf32_Phdr 6-48
ELF32_R_SYM 4G-3
ELF32_R_TYPE 4G-3
ELF32_Rel 4G-3
Elf32_Rel 4G-3, 6-49
Elf32_Rela 6-49
Elf32_Shdr 4G-2, 6-48
Elf32_Sword 6-47, 6-49
Elf32_Sym 6-49
Elf32_Word 6-47, 6-48, 6-49
elf32_xlatetof 6-46
elf32_xlatetom 6-46
Elf64_Addr 4G-1, 4G-2, 4G-3, 5G-1, 5P-4
Elf64_Dyn 5G-1
Elf64_Dyn_DYNAMIC 5G-1
Elf64_Ehdr 4G-1
Elf64_Half 4G-1, 4G-2
Elf64_Off 4G-1, 4G-2, 5G-1
Elf64_Phdr 5G-1
ELF64_R_INFO 4G-3
ELF64_R_SYM 4G-3
ELF64_R_TYPE 4G-3
ELF64_R_TYPE_DATA 4P-4, 4P-6
ELF64_R_TYPE_ID 4P-4
ELF64_R_TYPE_INFO 4P-4
ELF64_Rel 4G-3
Elf64_Rel 4G-3
Elf64_Rela 4G-3, 4P-6
Elf64_Shdr 4G-2
ELF64_ST_INFO 4P-3
Elf64_Sword 4G-1
Elf64_Sxword 4G-1, 4G-3
Elf64_Sym 4G-2
Elf64_Word 4G-1, 4G-2, 4G-3, 5G-1
ELFCLASS32 5P-3, 6-47
ELFCLASS64 4G-1, 4G-2, 4G-3, 4P-1, 5P-3, 6-47
ELFCLASSNONE 6-47
ELFDATA2LSB 4G-2, 6-47
ELFDATA2MSB 4G-2, 4P-1, 6-47
ELFDATANONE 6-47
ELFMAG0 6-47
ELFMAG1 6-47
ELFMAG2 6-47
ELFMAG3 6-47
EM_SPARC64 4P-1
EMPTY 6-44
EMSGSIZE 6-68
EN_ASN1 6-70
EN_BINARY 6-70
EN_CRYPT 6-70
EN_MODIFIED 6-70
EN_XDR 6-70
ENAMETOOLONG 6-4
encrypt 6-4
EndCurrentTextBufferWord 10-49
endgrent 6-4
endhostent 6-55
endnetconfig 6-55
endnetent 6-67
endnetpath 6-55
endpoint 6-70, 6-71
endservent 6-67
ENETUNREACH 6-68
ENOPROTOOPT 6-68
ENOTCONN 6-68
ENOTSOCK 6-68
entry_col 6-71
ENTRY_OBJ 6-70
entry_obj 6-53, 6-71
entry_object_p 6-53
environ 6-27
EOPNOTSUPP 6-68
ep_val 6-71
EPROTONOSUPPORT 6-68
EPROTOTYPE 6-68
EQUIVALENCE 3P-2
erf 6-51
erfc 6-51
errno 6-27, 6-78

- errno.h 6-68, 6-78
- esolv 6-64
- ether_aton 6-67
- ether_hostton 6-67
- ether_line 6-67
- ether_ntoa 6-67
- ether_ntohost 6-67
- ETIMEDOUT 6-68
- eucwidth_t 6-81
- EV_CURRENT 6-47
- EV_NONE 6-47
- EventMask 10-34
- EventMaskOfScreen
 - unsafe, use XEventMaskOfScreen 10-8
- EventObj 10-52
- EventObjClass 10-52
- eventObjClass 10-50
- eventObjClassRec 10-50
- EWOULDBLOCK 6-68
- ex 8-1, 8-2, 8-3
- exception 6-51
- exclusivesClassRec 10-50
- ExclusivesWidget 10-52
- ExclusivesWidgetClass 10-52
- exclusivesWidgetClass 10-50
- exec 3P-22, 3P-25
- Execution Environment i, 9-1
- exit 6-6
- exit_status 6-44
- exp 6-51
- EXPERIMENTAL 1-2, 1-3, 1-7
- Experimental 1-3
- expm1 6-51
- Exported Data
 - libX11 10-6
 - libXol 10-50
 - libXt 10-32
- expr 8-1
- extended 6-32
- extension 10-11
- F**
- f_basetype 6-11, 6-15, 6P-10
- f_bavail 6-11, 6-15
- f_bfree 6-11, 6-15
- f_blocks 6-11, 6-15
- f_bsize 6-11, 6-15
- f_favail 6-11, 6-15
- f_ffree 6-11, 6-15
- f_files 6-11, 6-15
- f_filler 6-11, 6-15
- f_flag 6-11, 6-15
- F_FREESP 6-3, 6-8, 6-14
- F_FREESP64 6-14
- f_frsiz 6-11, 6-15
- f_fsid 6-11, 6-15
- f_fstr 6-11, 6-15
- F_GETLK 6-14
- F_GETLK64 6-14
- f_namemax 6-11, 6-15
- F_SETLK 6-14
- F_SETLK64 6-14
- F_SETLKW 6-14
- F_SETLKW64 6-14
- fabs 6-51
- false 8-1
- fcc0 3P-23
- fcc1 3P-23
- fcc2 3P-23
- fcc3 3P-23
- fcntl 6-3
- fcntl.h 6-8, 6-14
- FD_CLR 6-41
- FD_ISSET 6-41
- FD_SET 6-41
- fd_set 6-41
- FD_SETSIZE 6-41, 6P-7
- FD_ZERO 6-41
- fdatasync 6-59
- fdopen 6-4
- FEF 3P-10, 3P-21, 3P-23
- fgetgrent 6-4
- fgetgrent_r 6-23
- fgetpos64 6-12
- fgetpwent_r 6-23
- fgetwc 6-81, 6-82
- fgetws 6-81
- FIFOTYPE 8-2
- fildev 6-4
- FILE 6P-11
- File Transfer Protocol 7-3
- File transfer protocol 7-2
- find 8-1, 8-3
- Finger 7-3
- finger 7-2, 7-3
- first_error 10-11
- first_event 10-11
- FirstExtensionError 10-10
- fixed_dotfrac_form 6-32
- fixed_int_form 6-32
- fixed_intdot_form 6-32
- fixed_intdotfrac_form 6-32

floating_dotfrac_form 6-32
floating_int_form 6-32
floating_intdot_form 6-32
floating_intdotfrac_form 6-32
floating-point 7G-1
floatingpoint.h 6-32
flock64 6-14
flockfile 6-23
floor 6-51
FLUSH 3P-1
FLUSHW 3P-21
fmod 6-51
fmtmsg 8-1
font_set 10-12
FontChange 10-10
footerPanelClassRec 10-50
FooterPanelWidget 10-52
FooterPanelWidgetClass 10-52
footerPanelWidgetClass 10-50
fopen 6-80
fopen64 6-12
fork 6-22, 6-23
fork1 6-23, 6-78
Form.h 10-63
Formats and Protocols i
formats and protocols 7-1
formClassRec 10-50
FormConstraints 10-52
FormWidget 10-52
FormWidgetClass 10-52
formWidgetClass 10-50
FORTRAN 3P-2
ForwardScanTextBuffer 10-49
fp_class_type 6-32
fp_direction_type 6-32
fp_disabled 3P-18, 3P-20
fp_exception_field_type 6-32
fp_exception_ieee_754 3P-18, 3P-20
fp_exception_other 3P-18, 3P-20
fp_rnd 6-33
fpclass_t 6-33
fpos_t 6-10
fpos64_t 6-14
fpq 6P-14
fpregset_t 6P-14
FPRS 3P-22, 3P-23, 3P-24
fpu 6P-14
fpu_fr 6P-14
fputwc 6-81
fputws 6-81
Frame.h 10-63
free_private 10-11
FreeBuffer 10-49
freenetconfignt 6-55
FreeTextBuffer 10-49
freopen 6-80
freopen64 6-12
fsblkcnt_t 6-11, 6-15
fsblkcnt64_t 6-15
fseeko64 6-12
fsetpos64 6-12
fsfilent_t 6-11, 6-15
fsfilent64_t 6-15
FSR 3-2, 3P-11, 3P-23
fstat 6-3
fstat64 6-12
fstatvfs64 6-12
FSTYPSZ 6-11, 6-15
ftello64 6-12
ftime 6-80
ftp 7-2, 7-3
ftp-data 7-3
ftruncate 6-8
ftruncate64 6-12
ftt 3P-23
ftw.h 6-33
ftw64 6-12
funlockfile 6-23
G
gABI 2-1, 5-1, 6-7, 6-9, 6-10, 6-80
gamma 6-51
gamutCompClientData 10-10
gamutCompProc 10-10
gaugeClassRec 10-50
GaugeWidget 10-52
GaugeWidgetClass 10-52
gaugeWidgetClass 10-50
GC 10-11
GCLastBit 10-10
gencat 8-1
generic ABI 1-7
getc_unlocked 6-23
getcwd 6-4
getdate_err 6-27
getdents64 6-12
getdtablesize 6-80
getgrent 6-4
getgrent(BA_LIB) 6-4
getgrent_r 6-23
getgrgid_r 6-22, 6-23

- getgrnam_r 6-22, 6-23
- gethostid 6-80
- gethostname 6-80
- getitimer 6-4, 6-5, 6-8
- getitimer() 6-8
- getlogin_r 6-22, 6-23
- getnetbyaddr 6-67
- getnetbyname 6-67
- getnetent 6-67
- GetOIBusyCursor 10-49
- GetOIDuplicateCursor 10-49
- GetOIMoveCursor 10-49
- GetOIPanCursor 10-49
- GetOIQuestionCursor 10-49
- GetOIStandardCursor 10-49
- GetOISWGeometries 10-49
- GetOITargetCursor 10-49
- getpagesize 6-80
- getpeername 6-67
- getpriority 6-80
- getprotobyname 6-67
- getprotobynumber 6-67
- getprotoent 6-67
- getpublickey 6-55
- getpwent(BA_LIB) 6-4
- getpwent_r 6-23
- getpwnam_r 6-22
- getpwuid_r 6-22, 6-23
- getrlimit64 6-12
- getrusage 6-80
- getservbyname 6-67
- getservbyport 6-67
- getservent 6-67
- getsockname 6-67
- getsockopt 6-67
- gettext 6-50
- GetTextBufferBlock 10-49
- GetTextBufferBuffer 10-49
- GetTextBufferChar 10-49
- GetTextBufferLine 10-49
- GetTextBufferLocation 10-49
- gettimeofday 6-5, 6-80
- gettxt 8-1
- getwd 6-80
- getwidth 6-81
- getws 6-81
- gid_t 6-10, 6-15
- gmtime_r 6-23
- goto 3P-34
- gr_flags 6-71
- gr_members_len 6-71
- gr_members_val 6-71
- greg_t 6P-14
- gregset_t 6P-14
- grep 8-1
- GROUP_OBJ 6-70
- group_obj 6-71
- GrowBuffer 10-49
- grp.h 6-22
- gwindows_t 6P-14
- H**
- h_addr 6-68
- h_addr_lis 6-68
- h_addr_list 6-68
- h_addrtype 6-68
- h_aliases 6-68
- h_length 6-68
- h_name 6-68
- h44 4P-9
- HAL 4P-1
- half16 4P-6
- halfword 4P-9
- Hashing 6-5
- he X extensions 10-1
- HeightMMOfScreen
 - unsafe, use XHeightMMOfScreen 10-8
- HeightOfScreen
 - unsafe, use XHeightOfScreen 10-8
- helpClassRec 10-50
- HelpWidget 10-52
- HelpWidgetClass 10-52
- helpWidgetClass 10-50
- Host Name Server 7-3
- HOST_NOT_FOUND 6-68
- HostDelet 10-10
- hostent 6-68
- HostInsert 10-10
- hostname 7-3
- howmany 6-41
- htonl 6-67
- htons 6-67
- hypot 6-51
- HZ 6-8
- I**
- IBM SNA 6-74
- icc 3P-20, 3P-23
- ICMP 7-1
- iconv.h 6-33
- iconv_t 6-33
- id 6-65, 8-1

IDI 6-74
 idtype 6-40
 IE 3P-24
 IEEE 1-6, 2-2, 3-2, 7G-1
 IEEE 754 7G-1
 IEEE 802.2 6-74
 ieefp.h 6-33
 ILL_BD 6-57
 illegal_instruction 3P-18, 3P-20
 ILLTRAP 3P-21
 ilogb 6-51
 ImageByteOrder
 unsafe, use XImageByteOrder 10-8
 imm22 4-2, 4P-6, 4P-8
 imm44 4P-9
 IMPDEP1 3P-1
 IMPDEP2 3P-1
 imr_interface 6-69
 imr_multiaddr 6-69
 in_addr 6-65, 6-69, 6P-16
 IncrementTextBufferLocation 10-49
 Index i
 index 6-80
 inet_addr 6-55
 inet_lnaof 6-67
 inet_makeaddr 6-67
 inet_netof 6-55
 inet_network 6-67
 inet_ntoa 6-55
 inf_form 6-32
 infinity_form 6-32
 Information server for logged on users 7-2
 INIT_PROCESS 6-44
 initstate 6-80
 ino_t 6-10, 6-15
 ino64_t 6-14, 6-15
 inproc 6-7
 InsertIntoBuffer 10-49
 installf 2-1
 instruction_access_error 3P-18, 3P-20
 instruction_access_exception 3P-18, 3P-20
 instruction_access_MMU_miss 3P-18, 3P-20
 INT_MAX 6-51
 INT_MIN 6-51
 Interactive terminal services 7-2
 Interface 1-3
 Internet Protocol Suite 7-1
 Introduction i
 INTRPT 6-57
 inttypes.h 6P-13
 invalid_form 6-32
 iov 6-4
 IOV_MAX 6-4
 iovcnt 6-4
 iovec 6P-13
 iovec_t 6P-13
 IP 7-1
 IP protocols 7-1
 IP_ADD_MEMBERSHIP 6-69
 IP_DEFAULT_MULTICAST_LOOP 6-69
 IP_DEFAULT_MULTICAST_TTL 6-69
 IP_DROP_MEMBERSHIP 6-69
 IP_HDRINCL 6-69
 ip_mreq 6-69
 IP_MULTICAST_IF 6-69
 IP_MULTICAST_LOOP 6-69
 IP_MULTICAST_TTL 6-69
 IP_OPTIONS 6-69
 IP_RECVDSTADDR 6-69
 IP_RECVOPTS 6-69
 IP_RECVRETOPTS 6-69
 IP_RETOPTS 6-69
 IP_TOS 6-69
 IP_TTL 6-69
 ipc_perm 6P-8
 IPPROTO_TCP 6-58
 IPPROTO_UDP 6-58
 IPS 7-1
 IQUERY 6-64
 isblkcnt_t 6-11
 isenglish 6-81
 isideogram 6-81
 isnumber 6-81
 ISO 1-6
 ISO 8802 6-74
 ISO 966 2-1
 ISO 9660 2-1, 2-2
 ISO 9660-1988 1-5, 1-6, 1-7
 ISO/IEC 10149 1-5, 1-6, 2-1
 ISO-TSAP 7-3
 iso-tsap 7-3
 isphonogram 6-81
 isspecial 6-81
 iswalnum 6-81
 iswalph 6-81
 iswcntrl 6-81
 iswdigit 6-81
 iswgraph 6-81
 iswlower 6-81
 iswprint 6-81

- iswpunct 6-81
- iswspace 6-81
- iswupper 6-81
- iswxdigit 6-81
- IVY 6-70
- J**
- j0 6-51
- j1 6-51
- James Gettys 10-1
- jmp_buf 6P-8
- jmp1 3P-21, 3P-32
- jn 6-51
- K**
- K&R C 3P-12
- KE_OS 3P-17, 6-4, 6-26
- kill 8-1
- killpg 6-80
- L**
- l 11-1
- l_len 6-14
- l_linger 6-73
- l_onoff 6-73
- l_pad 6-14
- l_pid 6-14
- L_PROB 6-57
- l_start 6-14
- l_sysid 6-14
- l_type 6-14
- l_whence 6-14
- Label.h 10-63
- LabelG.h 10-63
- Large Files Support 6-12
- LastExtensionError 10-10
- LastKnownRequestProcessed
 - unsafe, use XLastKnownRequestProcessed 10-8
- LastTextBufferLocation 10-49
- LastTextBufferPosition 10-49
- LAT 6-74
- LD_BIND_NOW 5P-7
- LD_LIBRARY_PATH 5-1
- LD_LIBRARY_PATH 5-1, 5G-1, 5P-3
- LD_LIBRARY_PATH_64 5P-3
- LDDF 3P-2
- LDQF 3P-2
- LDX 3P-29
- ldx 3P-28, 3P-31, 3P-32
- len 6-75
- level 6-75
- lex 11-1
- lgamma 6-51
- li_attr_len 6-71
- li_attr_val 6-71
- li_rtype 6-71
- libaio 6-1, 6-12, 6-24
- libc 1-9, 1-10, 6-1, 6-9, 6-12, 6-23, 6-26, 6P-1
- libdl 6-1, 6-9, 6-45, 6P-1
- libelf 1-10, 6-1, 6-9, 6-46, 6P-1
- libintl 1-10, 6-1, 6-9, 6-50
- liblfl 6-12, 6-21, 6-23
- libm 1-10, 6-1, 6-9, 6P-1
- libMrm
 - entry points 10-59
- libnisdb 1-10, 6-1, 6-9, 6-52
- libnsl 1-10, 6-1, 6-9, 6-54, 6-55, 6-56, 6P-1
- libposix4 6-1, 6-12, 6-59, 6-60
- libpthread 6-1, 6-23, 6-60
- Libraries i
- libresolv 1-10, 6-1, 6-9, 6-64, 6P-1
- librpcsv 1-10
- librpcsvc 6-2, 6-9, 6-59, 6-60, 6-61, 6-62, 6-66
- libsocket 1-10, 6-2, 6-9, 6-67, 6P-1
- libsys 6-1
 - support routines 6-27
- libsys SPARC support routines 6-27, 6-28, 6-29, 6-30, 6-31, 6-32, 6-33, 6-34, 6-35, 6-36, 6-37, 6-38, 6-39, 6-40, 6-41, 6-42, 6-43, 6-44
- libthread 6-2, 6-9, 6-12, 6-77, 6P-1
- libthread, 6-23
- libucb 1-10, 6-2, 6-9, 6-12, 6-80
- libw 1-10, 6-2, 6-9, 6-81, 6P-1
- libX 10-1, 10P-1
 - contents 10-3, 10-4, 10-5, 10-6
- libXext 10-1, 10P-1
 - contents 10-26
- libXm 10-1, 10P-1
 - contents 10-56, 10-57
- libXm.so.1.2 6-17
- libXm.so.3 6-17
- libXmu 1-10
- libXol 10-1, 10P-1
 - contents 10-49
 - version number 10-48
- libXrm 10-1, 10P-1
- libXt 10-1, 10P-1
 - contents 10-28, 10-29
- limits.h 6-33
- Line 10-54
- line 6-57, 8-1
- LineOfPosition 10-49
- LineTable 10-54
- linger 6-73
- LINK_OBJ 6-70

-
- link_obj 6-71
 - lio_listio 6-59
 - lio_listio64 6-12
 - List.h 10-63
 - listClassRec 10-50
 - listen 6-67, 6-73
 - listPaneClassRec 10-50
 - ListPaneWidget 10-52
 - ListPaneWidgetClass 10-52
 - listPaneWidgetClass 10-50
 - ln 8-1
 - localtime 3P-29
 - localtime_r 6-23
 - LocationOfPosition 10-49
 - lockf 6-5
 - lockf64 6-12
 - LockMapIndex 10-10
 - log 6-51
 - LOG_ALERT 6-43
 - LOG_AUTH 6-43
 - LOG_CONS 6-43
 - LOG_CRIT 6-43
 - LOG_CRON 6-43
 - LOG_DAEMON 6-43
 - LOG_DEBUG 6-43
 - LOG_EMERG 6-43
 - LOG_ERR 6-43
 - LOG_INFO 6-43
 - LOG_KERN 6-43
 - LOG_LFMT 6-43
 - LOG_LOCAL0 6-43
 - LOG_LOCAL1 6-43
 - LOG_LOCAL2 6-43
 - LOG_LOCAL3 6-43
 - LOG_LOCAL4 6-43
 - LOG_LOCAL5 6-43
 - LOG_LOCAL6 6-43
 - LOG_LOCAL7 6-43
 - LOG_LPR 6-43
 - LOG_MAIL 6-43
 - LOG_MASK 6-43
 - LOG_NDELAY 6-43
 - LOG_NEWS 6-43
 - LOG_NOTICE 6-43
 - LOG_NOWAIT 6-43
 - LOG_ODELAY 6-43
 - LOG_PID 6-43
 - LOG_SYSLOG 6-43
 - LOG_UPTO 6-43
 - LOG_USER 6-43
 - LOG_UUCP 6-43
 - LOG_WARNING 6-43
 - log10 6-51
 - log1p 6-51
 - logb 6-5
 - LOGIN_PROCESS 6-44
 - logname 8-1
 - long 6-10, 6-11, 6-15, 6-41
 - long long 6-11, 6-26
 - longjmp 3P-34, 6-5, 6-6, 6-80
 - longlong_t 6-41, 6-42
 - LookupOfInputEvent 10-49
 - Low-Level System Information i
 - lp 8-1
 - ls 8-1
 - lseek64 6-12
 - lstat 6-3
 - lstat64 6-12
 - M**
 - m4 11-1
 - m44 4P-9
 - MADV_NORMAL 6-38
 - MADV_RANDOM 6-38
 - MADV_SEQUENTIAL 6-38
 - MADV_WILLNEED 6-38
 - magClassRec 10-50
 - MagWidget 10-52
 - MagWidgetClass 10-52
 - magWidgetClass 10-50
 - main 3P-22
 - MainW.h 10-63
 - major_opcode 10-11
 - major_t 6-42, 6P-12
 - make 8-1
 - malloc 3P-17
 - managerClassRec 10-50
 - ManagerWidget 10-52
 - ManagerWidgetClass 10-52
 - managerWidgetClass 10-50
 - Manifest Constants
 - Motif 10-59, 10-60, 10-61, 10-62, 10-63, 10-64, 10-65, 10-67, 10-70
 - OLIT 10-51
 - X Extensions 10-26
 - Xlib, 10-13
 - MAP_FAILED 6-38
 - MAP_NORESERVE 6-38
 - Mask 10-10
 - MAXADDR 6-65
 - MaxCmapsOfScreen
 - unsafe, use XMaxCmapsOfScreen 10-8
 - MAXDNAME 6-64, 6-65
-

- MAXDNSRCH 6-65
- MAXNS 6-65
- mcontext_t 6P-14
- mctl 6-80
- mem_address_not_aligned 3P-18, 3P-20
- MEMBAR 3P-24
- memory_address_not_aligned 3P-19
- menuButtonClassRec 10-50
- MenuButtonGadget 10-52
- MenuButtonGadgetClass 10-52
- menuButtonGadgetClass 10-50
- menuButtonGadgetClassRec 10-50
- MenuButtonWidget 10-52
- MenuButtonWidgetClass 10-52
- menuButtonWidgetClass 10-50
- MenuShell.h 10-63
- menuShellClassRec 10-50
- MenuShellWidget 10-52
- MenuShellWidgetClass 10-52
- menuShellWidgetClass 10-50
- MessageB.h 10-63
- MinCmapsOfScreen
 - unsafe, use XMinCmapsOfScreen 10-8
- minor_t 6-42, 6P-12
- mkdir 8-1
- mknod 6-3
- mkstemp64 6-12
- MM 3P-24
- mmap 3P-16, 3P-17, 6-4
- mmap64 6-12
- mnttab 6-36
- Mod1MapIndex 10-10
- Mod2MapIndex 10-10
- Mod3MapIndex 10-10
- Mod4MapIndex 10-10
- Mod5MapIndex 10-10
- mode_t 6-10, 6-15
- modem 6-57
- Modifiers 10-33
- MotifWmHints 10-64
- MotifWmInfo 10-65
- movgu 3P-32
- mq_close 6-59
- mq_getattr 6-59
- mq_notify 6-59
- mq_open 6-59
- mq_receive 6-59
- mq_send 6-59
- mq_unlink 6-59
- MrmCloseHierarchy 10-59
- MrmCode 10-64
- MrmCount 10-64
- MrmFetchBitmapLiteral 10-59
- MrmFetchColorLiteral 10-59
- MrmFetchIconLiteral 10-59
- MrmFetchLiteral 10-59
- MrmFetchSetValues 10-59
- MrmFetchWidget 10-59
- MrmFetchWidgetOverride 10-59
- MrmFlag 10-64
- MrmGroup 10-64
- MrmHierarchy 10-64
- MrmInitialize 10-59
- MrmOffset 10-64
- MrmOpenHierarchy 10-59
- MrmOpenHierarchyPerDisplay 10-59
- MrmOsOpenParam 10-64
- MrmOsOpenParamPtr 10-64
- MrmPublic.h 10-63
- MRMRegisterArg 10-64
- MrmRegisterArg 10-64
- MrmRegisterArglist 10-64
- MrmRegisterClass 10-59
- MrmRegisterNames 10-59
- MrmRegisterNamesInHierarchy 10-59
- MrmResource_id 10-64
- MrmSCode 10-64
- MrmSize 10-64
- MrmType 10-64
- MS-DOS 2-1
- MSG_DONTROUTE 6-75
- MSG_MAXIOVLEN 6-75
- MSG_OOB 6-75
- MSG_PEEK 6-75
- msghdr 6-75
- mutex_destroy 6-23, 6-78
- mutex_init 6-23, 6-78
- mutex_lock 6-23, 6-78
- mutex_trylock 6-23, 6-78
- mutex_unlock 6-23, 6-78
- mutext_t 6-79
- mv 8-1
- MwmHints 10-65
- MwmInfo 10-65
- N**
 - n_addrtype 6-68
 - n_aliases 6-68
 - n_bytes 6-57
 - n_len 6-57
 - n_name 6-49, 6-68

n_net 6-68
n_numaux 6-49
n_sclass 6-49
n_senum 6-49
n_type 6-49
n_value 6-49
name 6-75, 7-3
NAME_MAX 6-4
nan_form 6-32
nanosleep 6-59
nanstring_form 6-32
NBBY 6-41
nc_flag 6-7
nchar 10-12
nchars 10-12
ndbm.h 6-34
netbuf 6P-12
netconfig 6-7
netdb.h 6-68
netdir_getbyaddr 6-55
netdir_options 6-55
netdir_perror 6-55
netdir_sperror 6-55
netent 6-68
netinet/in.h 6-69, 6P-16
netname2host 6-55
netname2user 6-55
netobj 6-57
NETSTAT 7-3
netstat 7-3
nettype 6-7
Network Time Protocol 7-3
NEW_TIME 6-44
nextinfo 6-53
NextLocation 10-49
NextRequest
 unsafe, use XNextRequest 10-8
NextTextBufferWord 10-49
NFBITS 6-41
nftw64 6-12
NIC Host Name Server 7-3
nice 6-80
nicname 7-3
NIS 6-1, 6-70
nis_add 6-55
nis_add_entry 6-55
nis_addmember 6-55
nis_attr 6-70
nis_checkpoint 6-55
nis_clone_object 6-55
nis_creategroup 6-55
nis_destroy_object 6-55
nis_destroygroup 6-55
nis_dir_cmp 6-55
nis_domain_of 6-55
nis_error 6-72
nis_first_entry 6-55
nis_freenames 6-55
nis_freeresult 6-55
nis_freeservlist 6-55
nis_freetags 6-55
nis_getnames 6-55
nis_getservlist 6-55
nis_lerror 6-55
nis_list 6-55
nis_local_directory 6-55
nis_local_group 6-55
nis_local_host 6-55
NIS_MAXATTR 6-70
NIS_MAXATTRNAME 6-70
NIS_MAXATTRVAL 6-70
NIS_MAXCOLUMNS 6-70
NIS_MAXLINKS 6-70
NIS_MAXNAMELEN 6-70
NIS_MAXPATH 6-70
NIS_MAXREPLICAS 6-70
NIS_MAXSTRINGLEN 6-70
nis_mkdir 6-55
nis_name 6-70, 6-71
nis_object 6-72
nis_oid 6-72
NIS_PK_DH 6-70
NIS_PK_KERB 6-70
NIS_PK_NONE 6-70
NIS_PK_RSA 6-70
nis_removemember 6-55
nis_result 6-72, 6-73
nis_server 6-71
nis_servstate 6-55
nis_sperno 6-55
nis_sperror 6-55
nis_stats 6-55
nis_verifygroup 6-55
NIST 2-2
nlink_t 6-10, 6-15
nlist 6-46, 6-49
NO_ADDRESS 6-68
NO_ANS 6-57
NO_BD_A 6-57
NO_BD_K 6-57

NO_DATA 6-68
NO_Ldv 6-57
NO_OBJ 6-70
NO_RECOVERY 6-68
NOASSIGN 6-57
NOBUFS 6-68
NODEV 6-42
non-ANSI 6-5
nonexclusivesClassRec 10-50
NonexclusivesWidget 10-52
NonexclusivesWidgetClass 10-52
nonexclusivesWidgetClass 10-50
NORUN 6-57
noticeShellClassRec 10-50
NoticeShellWidget 10-52
NoticeShellWidgetClass 10-52
noticeShellWidgetClass 10-50
Novell 1-5
NS 3-2, 3P-10, 3P-23
nsaddr 6-65
nsaddr_list 6-65
NSC Hyperchannel 6-74
nscout 6-65
nstype 6-70
ntohl 6-67
ntohs 6-67
ntp 7-3
NULL 6-7
NULLSTRING 10-11

O

O_LARGEFILE 6-14
oa_otype 6-70
oa_rights 6-70
oar_mask 6-70
objdata 6-72
objdata_u 6-72
Object 10-35
object file format 4-1
Object Files i
ObjectClass 10-35
objectClass 10-32
ObjectClassPart 10-43
ObjectClassRec 10-43
objectClassRec 10-32
ObjectPart 10-43
ObjectRec 10-43
objects 6-73
objects_len 6-53
objects_val 6-53
oblongButtonClassRec 10-50
OblongButtonGadget 10-52
OblongButtonGadgetClass 10-52
oblongButtonGadgetClass 10-50
oblongButtonGadgetClassRec 10-50
OblongButtonWidget 10-52
OblongButtonWidgetClass 10-52
oblongButtonWidgetClass 10-50
off_t 6-10, 6-11, 6-13, 6-47, 6-48, 6P-12
off64_t 6-13, 6-14, 6-15
offsetof 6-35
OIAddCallback 10-49
OIBitMask 10-52
OICallAcceptFocus 10-49
OICallCallbacks 10-49
OICanAcceptFocus 10-49
OICategorySetPage 10-49
OLD_TIME 6-44
OIDefine 10-52
OIDragAndDrop 10-49
OIError 10-49
OIGet50PercentGrey 10-49
OIGet75PercentGrey 10-49
OIGetApplicationResources 10-49
OIGetApplicationValues 10-49
OIGetBeepVolume 10-49
OIGetCurrentFocusWidget 10-49
OIGrabDragPointer 10-49
OIHasCallbacks 10-49
OIHasFocus 10-49
OIInitialize 10-49
OLIT 1-5
OILayoutScrolledWindow 10-49
OIListDelete 10-53
OIListItem 10-53
OIListItemPointer 10-49
OIListToken 10-53
OIMenuPopdown 10-49
OIMenuPopup 10-49
OIMenuPost 10-49
OIMenuUnpost 10-49
OIMoveFocus 10-49
OIQueryAcceleratorDisplay 10-49
OIQueryMnemonicDisplay 10-49
OIRChar 10-50
OIRegisterColorTupleListConverter 10-49
OIRegisterHelp 10-49
OIRemoveCallback 10-49
OISameSize 10-52
OIScanDirection 10-51
OIScanType 10-51

- OlScrollbarVerify 10-53
- OlSetErrorHandler 10-49
- OlSetGaugeValue 10-49
- OlSetInputFocus 10-49
- OlSetVaDisplayErrorMsgHandler 10-49
- OlSetVaDisplayWarningMsgHandler 10-49
- OlSetWarningHandler 10-49
- OlSliderVerify 10-53
- OlSWGeometries 10-53
- OlTextEditClearBuffer 10-49
- OlTextEditCopyBuffer 10-49
- OlTextEditCopySelection 10-49
- OlTextEditGetCursorPosition 10-49
- OlTextEditGetLastPosition 10-49
- OlTextEditInsert 10-49
- OlTextEditPaste 10-49
- OlTextEditReadSubString 10-49
- OlTextEditRedraw 10-49
- OlTextEditResize 10-49
- OlTextEditSetCursorPosition 10-49
- OlTextEditTextBuffer 10-49
- OlTextEditUpdate 10-49
- OlTextFieldCopyString 10-49
- OlTextFieldGetString 10-49
- OlTextFieldVerify 10-54
- OlTextFieldVerifyPointer 10-54
- OlTextMarginCallData 10-53
- OlTextMarginCallDataPointer 10-53
- OlTextModifyCallData 10-53
- OlTextModifyCallDataPointer 10-53
- OlTextMotionCallData 10-53
- OlTextMotionCallDataPointer 10-53
- OlTextPostModifyCallData 10-54
- OlTextPostModifyCallDataPointer 10-54
- OlToolkitInitialize 10-49
- OlUngrabDragPointer 10-49
- OlUpdateDisplay 10-49
- OlVaDisplayErrorMsg 10-49
- OlVaDisplayWarningMsg 10-49
- OlVerifyOpType 10-51
- OlWarning 10-49
- OlWMPProtocolAction 10-49
- ong long 6-10
- OOB 6-73
- Opaque 10-33
- Open 10-1
- Open Look Widget 10-1
- open64 6-12, 6-23, 6-78
- opproc 6-75
- optarg 6-27
- optdefault 6-75
- opterr 6-27
- opthdr 6-75
- optind 6-27
- OPTIONAL 1-3, 1-4, 2-1
- options 6-65
- OPTLEN 6-75
- optname 6-75
- optopt 6-27
- OPTVAL 6-75
- OR 3-2
- or 3P-28, 3P-32
- organization i
- OSF/Motif 1-5
- OSI 6-74
- outproc 6-7
- OverrideShellClassPart 10-45
- OverrideShellClassRec 10-45
- overrideShellClassRec 10-32
- OverrideShellPart 10-45
- OverrideShellRec 10-45
- OverrideShellWidget 10-45
- OverrideShellWidgetClass 10-36
- overrideShellWidgetClass 10-32
- P**
- p_aliases 6-68
- p_align 5G-1, 6-48
- P_ALL 6-40
- P_CID 6-40
- p_filesz 5G-1, 6-48
- p_flags 5G-1, 6-48
- P_GID 6-40
- p_lid 6-40
- p_lidtype 6-40
- P_LWPID 6-40
- p_memsz 5G-1, 6-48
- p_name 6-68
- P_OFFLINE 6-39
- p_offset 5G-1, 6-48
- P_ONLINE 6-39
- p_op 6-40
- p_paddr 5G-1, 6-48
- P_PPID 6-40
- p_proto 6-68
- p_rid 6-40
- p_ridtype 6-40
- P_SID 6-40
- P_STATUS 6-39
- p_type 5G-1, 6-48
- p_vaddr 5G-1, 6-48

P1003.2 8-3
pad 6-44
Page 10-54
PageQueue 10-54
PageTable 10-54
passwd 8-1
PATH_MAX 6-4
PBIND_QUERY 6-39
PC 3P-24
pc_cid 6-39
pc_clinfo 6-39
PC_CLINFOSZ 6-39
pc_clname 6-39
PC_CLNMSZ 6-39
pc_clparms 6-39
PC_CLPARMSZ 6-39
PC_VERSION 6-39
pcinfo 6-39
pcinfo_t 6-39
pcparms 6-39
pcparms_t 6-39
PF_802 6-75
PF_APPLETALK 6-75
PF_CCITT 6-74
PF_CHAOS 6-74
PF_DATAKIT 6-74
PF_DECnet 6-74
PF_DLI 6-75
PF_ECMA 6-74
PF_GOSIP 6-75
PF_HYLINK 6-75
PF_IMPLINK 6-74
PF_INET 6-74
PF_LAT 6-75
PF_MAX 6-75
PF_NBS 6-74
PF_NIT 6-75
PF_NS 6-74
PF_OSI 6-75
PF_OSINET 6-75
PF_PUP 6-74
PF_SNA 6-74
PF_UNIX 6-74
PF_UNSPEC 6-74
PF_X25 6-75
pfmt.h 6-34
pg 8-1
Physical Page 5P-1
pi_clock 6-39
PI_FPUTYPE 6-39
pi_fputypes 6-39
pi_processor_type 6-39
pi_state 6-39
PI_TYPELEN 6-39
PIC 3P-28, 3P-29, 3P-31
pid_t 6-14
PIPE_MAX 6P-5
pipes 6-74
Pixel 10-33
pkgadd 2-1
pkgask 2-1
pkgchk 2-1
pkginfo 2-1
pkgmk 11-1
pkgparam 2-1
pkgproto 11-1
pkgrm 2-1
pkgtrans 11-1
PlanesOfScreen
 unsafe, use XPlanesOfScreen 10-8
PLT 3P-10
PLT1 5P-6
PLT2 5P-6
pm_port 6-58
pm_prog 6-58
pm_prot 6-58
pm_vers 6-58
PMAP 6-58
pmap 6-58
pmap_getmaps 6-55
pmap_getport 6-55
pmap_rmtcall 6-55
pmap_set 6-55
pmap_unset 6-55
pmaplist 6-58
pml_map 6-58
pml_next 6-58
popupWindowShellClassRec 10-50
PopupWindowShellWidget 10-52
PopupWindowShellWidgetClass 10-52
popupWindowShellWidgetClass 10-50
port 111 7-1
portals 6-74
Position 10-33
PositionOfLine 10-49
PositionOfLocation 10-49
POSIX 1-6, 1-7, 2-2, 6-1, 6-2, 6-4, 6-5
POSIX 1003.1-1990 1-6
POSIX4 6-59, 6-60
pow 6-51

-
- pPerScrnInfo 10-10
 - pr 8-1, 8-3
 - pread64 6-12
 - Preface i
 - Prentice 1-5
 - Prentice Hall 1-5
 - Prentice-Hall 1-6
 - PreviousLocation 10-49
 - PreviousTextBufferWord 10-49
 - pri 6-43
 - primitiveClassRec 10-50
 - PrimitiveWidget 10-52
 - PrimitiveWidgetClass 10-52
 - primitiveWidgetClass 10-50
 - printf(BA_LIB) 6-26
 - prioctl 6-39, 8-1
 - prioctlset 6-39
 - PRIV 3P-24
 - PRIVATE 6-38
 - private_data 10-11
 - PRIVATE_OBJ 6-70
 - privileged_action 3P-18, 3P-20
 - privileged_opcode 3P-18, 3P-20
 - PROC_DATA 6-38
 - PROC_TEXT 6-38
 - processor_info_t 6-39
 - processorid_t 6-42
 - PROCLOCK 6-38
 - Program Loading and Dynamic Linking i
 - program loading and linking 5-1
 - PropMotifWmHints 10-65
 - PropMotifWmInfo 10-65
 - PropMwmHints 10-65
 - PropMwmInfo 10-65
 - PROT_EXEC 6-38
 - PROT_NONE 6-4, 6-38
 - PROT_READ 6-38
 - PROT_WRITE 6-4, 6-38
 - ProtocolRevision
 - unsafe, use XProtocolRevision 10-8
 - ProtocolVersion
 - unsafe, use XProtocolVersion 10-8
 - protosw 6-76
 - psABI i, 1-7, 2-1, 4-1, 5-1, 6-3, 6-8, 6-9
 - psignal 6-80
 - PSO 4P-1
 - PSR 3-2, 3P-20
 - PSTATE 3P-23
 - PT_INTERP 5-1
 - pthread.h 6-62
 - pthread_attr_setdetachstate 6-61
 - pthread_attr_t 6-62
 - pthread_attrp 6-62
 - PTHREAD_CANCEL_ASYNCHRONOUS 6-63
 - PTHREAD_CANCEL_DEFERRED 6-63
 - PTHREAD_CANCEL_DISABLE 6-63
 - PTHREAD_CANCEL_ENABLE 6-63
 - PTHREAD_CANCELED 6-63
 - pthread_cond_data 6-62
 - pthread_cond_flag 6-62
 - PTHREAD_COND_INITIALIZER 6-63
 - pthread_cond_t 6-62
 - pthread_cond_timedwait 6-61
 - pthread_cond_type 6-62
 - pthread_condattr_t 6-62
 - pthread_condattrp 6-62
 - PTHREAD_CREATE_DETACHED 6-63
 - PTHREAD_CREATE_JOINABLE 6-63
 - PTHREAD_EXPLICIT_SCHED 6-63
 - PTHREAD_INHERIT_SCHED 6-63
 - pthread_key_t 6-62
 - pthread_kill 6-61
 - pthread_mutex_data 6-62
 - PTHREAD_MUTEX_DEFAULT 6-63
 - PTHREAD_MUTEX_ERRORCHECK 6-63
 - pthread_mutex_flag 6-62
 - pthread_mutex_flags 6-62
 - PTHREAD_MUTEX_INITIALIZER 6-63
 - pthread_mutex_lock 6-62
 - pthread_mutex_lock64 6-62
 - PTHREAD_MUTEX_NORMAL 6-63
 - pthread_mutex_owner64 6-62
 - pthread_mutex_pad 6-62
 - PTHREAD_MUTEX_RECURSIVE 6-63
 - pthread_mutex_t 6-62
 - pthread_mutex_type 6-62
 - pthread_mutexattr_destroy 6-61
 - pthread_mutexattr_getshared 6-61
 - pthread_mutexattr_t 6-62
 - pthread_mutexattrp 6-62
 - PTHREAD_ONCE_DONE 6-63
 - PTHREAD_ONCE_INIT 6-63
 - PTHREAD_ONCE_NOTDONE 6-63
 - pthread_once_pad 6-62
 - pthread_once_t 6-62
 - PTHREAD_PROCESS_PRIVATE 6-63
 - PTHREAD_PROCESS_SHARED 6-63
 - PTHREAD_RWLOCK_INITIALIZER 6-63
 - pthread_rwlock_t 6-62
 - pthread_rwlock_unlock 6-61
-

- PTHREAD_SCOPE_PROCESS 6-63
- PTHREAD_SCOPE_SYSTEM 6-63
- pthread_t 6-62
- ptrdiff_t 6P-10
- publication conventions i
 - page format i
 - typography i
- purpose i
- PushB.h 10-65
- PushBG.h 10-65
- pushpinClassRec 10-50
- PushpinWidget 10-52
- PushpinWidgetClass 10-52
- pushpinWidgetClass 10-50
- putc_unlocked 6-23
- putchar_unlocked 6-23
- putmntent 6-36
- putws 6-81
- pwd 8-1
- pwd.h 6-22
- pwrite64 6-12
- Q**
- qelem 6-35
- QLength
 - unsafe, use XQLength 10-8
- qne 3P-23
- Quad-Precision 7G-1
- quad-precision 7G-1
- QUERY 6-64
- R**
- r_addend 4G-3, 4P-6, 6-49
- r_aliases 6-58
- r_class 6-64
- r_data 6-64
- r_info 4G-3, 4P-4, 4P-6, 6-49
- r_name 6-58
- r_number 6-58
- r_offset 4G-3, 4P-6, 6-49
- r_size 6-64
- R_SPARC_10 4-1, 4P-7
- R_SPARC_11 4-1, 4P-7
- R_SPARC_13 4P-7, 4P-8
- R_SPARC_16 4P-7, 4P-9
- R_SPARC_22 4P-7, 4P-8
- R_SPARC_32 4P-7, 4P-8
- R_SPARC_5 4-1, 4P-7
- R_SPARC_6 4-1, 4P-7
- R_SPARC_64 4-1, 4P-7, 4P-8, 4P-9
- R_SPARC_7 4-1, 4-2, 4P-7, 4P-8
- R_SPARC_8 4P-7
- R_SPARC_COPY 4P-7, 4P-8
- R_SPARC_DISP16 4P-7
- R_SPARC_DISP32 4P-7
- R_SPARC_DISP64 4P-7
- R_SPARC_DISP8 4P-7
- R_SPARC_GLOB_DAT 4-2, 4P-7, 4P-8, 5P-3
- R_SPARC_GLOB_JMP 4-1, 4-2
- R_SPARC_GOT10 4P-7
- R_SPARC_GOT13 4P-7, 4P-8
- R_SPARC_GOT22 4P-7, 4P-8
- R_SPARC_H44 4P-7, 4P-9
- R_SPARC_HH 3P-27
- R_SPARC_HH22 4-1, 4-2, 4P-7, 4P-8
- R_SPARC_HI22 4-2, 4P-7, 4P-8, 4P-9
- R_SPARC_HIPLT22 4-1, 4P-7
- R_SPARC_HIX22 4P-7, 4P-9
- R_SPARC_HM10 4-1, 4-2, 4P-7, 4P-8
- R_SPARC_JMP_SLOT 4P-7, 4P-8, 5P-7
- R_SPARC_L44 4P-7, 4P-9
- R_SPARC_LM 3P-27
- R_SPARC_LM22 4-1, 4-2, 4P-7, 4P-8
- R_SPARC_LO10 4-2, 4P-7, 4P-8, 4P-9
- R_SPARC_LOPLT10 4-1, 4P-7
- R_SPARC_LOX10 4P-7, 4P-9
- R_SPARC_M44 4P-7, 4P-9
- R_SPARC_NONE 4P-7
- R_SPARC_OLO10 4-1, 4-2, 4P-7, 4P-8
- R_SPARC_PC_HH22 4-1, 4-2, 4P-7, 4P-8
- R_SPARC_PC_HM10 4-1, 4-2, 4P-7, 4P-8
- R_SPARC_PC_LM22 4-1, 4-2, 4P-7, 4P-8
- R_SPARC_PC10 4P-7
- R_SPARC_PC22 4-2, 4P-7, 4P-8
- R_SPARC_PCPLT10 4-1, 4P-7
- R_SPARC_PCPLT22 4-1, 4P-7
- R_SPARC_PCPLT32 4-1, 4P-7
- R_SPARC_PLT32 4-1, 4P-7
- R_SPARC_PLT64 4P-7
- R_SPARC_REGISTER 4P-3, 4P-7, 4P-9
- R_SPARC_RELATIVE 4P-7, 4P-8
- R_SPARC_UA16 4P-7, 4P-9
- R_SPARC_UA32 4P-7, 4P-8
- R_SPARC_UA64 4P-7, 4P-9
- R_SPARC_WDISP16 4-1, 4P-7
- R_SPARC_WDISP19 4-1, 4P-7
- R_SPARC_WDISP22 4P-7
- R_SPARC_WDISP30 4P-7, 4P-8
- R_SPARC_WPLT30 4P-7, 4P-8
- r_ttl 6-64
- r_type 6-64
- r_zone 6-64

rand 6-80
rand_r 6-23
random 6-80
rcmd 6-67
rcp 7-2
RD 3-2, 3P-10, 3P-23
RDASI 3P-24
RDASR 3P-24
rdate 7-2
RDCCR 3P-24
RDFPRS 3P-24
RDPC 3P-24
RDTICK 3P-24
RDY 3P-24
re_comp 6-80
re_exec 6-80
read 3-1, 6-4
readdir_r 6-22, 6-23
readdir64 6-12
readdir64_r 6-12
ReadFileIntoBuffer 10-49
ReadFileIntoTextBuffer 10-49
ReadStringIntoBuffer 10-49
ReadStringIntoTextBuffer 10-49
readv 6-4
readv() 6-4
reboot 6-80
rectButtonClassRec 10-50
RectButtonWidget 10-52
RectButtonWidgetClass 10-52
rectButtonWidgetClass 10-50
RectObj 10-35
RectObjClass 10-35
rectObjClass 10-32
RectObjClassPart 10-44
RectObjClassRec 10-44
rectObjClassRec 10-32
RectObjPart 10-43
RectObjRec 10-43
recv 6-67
recvfrom 6-67
recvmsg 6-67
RED 3P-24
Region 10-12
Region 10-12
registerrpc 6-55
RegisterTextBufferScanFunctions 10-49
RegisterTextBufferUpdate 10-49
RegisterTextBufferWordDefinition 10-49
REGTYPE 8-2
rejected_reply 6P-7
related publications 1-4
remainder 6-51
Remote chat utility 7-2
Remote file copy 7-2
Remote terminal services 7-2
Remote uptime statistics 7-2
Remote user information service 7-2
Remote user shell 7-2
removef 2-1
rename 6-4
ReplaceBlockInTextBuffer 10-49
ReplaceCharInTextBuffer 10-49
REQUIRED 1-2, 1-3, 1-4
RES_AAONLY 6-65
RES_DEBUG 6-65
RES_DEFNAMES 6-65
RES_DNSRCH 6-65
RES_IGNTC 6-65
RES_INIT 6-65
res_init 6-64
res_mkquery 6-64
RES_RECURSE 6-65
res_send 6-64
RES_STAYOPEN 6-65
RES_USEVC 6-65
resource.h 6-10
RESTORE 3P-12
resultproc_t 6-58
retrans 6-65
retry 6-65
rexc 6-67
RFC 7-1, 7-2
RFC 1340 7-1
rfc1078 7-3
rfc1288 7-2
rfc826 7-2
rfc959 7-2
rindex 6-80
rint 6-51
rlim_cur 6-15
RLIM_INFINITY 6-10, 6P-5
rlim_max 6-15
RLIM_SAVED_CUR 6-15, 6P-5
RLIM_SAVED_MAX 6-15, 6P-5
rlim_t 6-10, 6-15, 6P-5
RLIM64_INFINITY 6-15
RLIM64_SAVED_CUR 6-15
RLIM64_SAVED_MAX 6-15
rlim64_t 6-15

rlimit 6-13
rlimit64 6-13, 6-15
rlogin 7-2
rm 8-1
rmdir 8-1
RMO 4P-1
rusers 6-66
Robert W. Scheifler 10-1
Rock Ridge 2-2
RootWindow
 unsafe, use XRootWindow 10-8
RootWindowOfScreen
 unsafe, use XRootWindowOfScreen 10-8
RowColumn.h 10-66
RPC 6-2, 6-37
RPC SVC 6-60
rpc.h 6-8
rpc/rpc.h 6-8, 6-58, 6P-6, 6P-7, 6P-8
rpc/rpcent.h 6-58
rpc/xdr.h 6-57
RPC_ANYFD 6-8
RPC_ANYSOCK 6-8
rpc_broadcast 6-55
rpc_broadcast_exp 6-7, 6-55
rpc_call 6-7, 6-55
rpc_clnt_calls 6-7
rpc_createerr 6-54
rpc_msg 6P-7
rpc_reg 6-55
rpc_svc_err 6-7
rpcb_getaddr 6-55
rpcb_gettime 6-55
rpcb_rmtcall 6-55
rpcb_set 6-55
rpcb_unset 6-55
rpcbind 7-1
rpcent 6-58
rpcsvc/nis.h 6-70
rrec 6-64
RS_ENV 6-7
RS_LIB 6-7
rs1 3P-21, 3P-22
rs2 3P-21, 3P-22
rsh 7-2
rt_maxpri 6-40
RT_NOCHANGE 6-40
RT_OS 6-5
rt_pri 6-40
RT_TQDEF 6-40
RT_TQINF 6-40
rt_tqnsecs 6-40
rt_tqsecs 6-40
rtinfo 6-40
RTLD_GLOBAL 6-45
RTLD_LAZY 6-45
RTLD_NEXT 6-45
RTLD_NOW 6-45
rtparms 6-40
rtparms_t 6-40
rubberTileClassRec 10-50
RubberTileWidget 10-52
RubberTileWidgetClass 10-52
rubberTileWidgetClass 10-50
RUN_LVL 6-44
ruserok 6-67
rusers 6-66
rw_rdlock 6-23, 6-78
rw_tryrdlock 6-23, 6-78
rw_trywrlock 6-23, 6-78
rw_unlock 6-23, 6-78
rw_wrlock 6-23, 6-78
rwall 6-66
rwho 7-2
rwindow 6P-14
rwalk_destroy 6-23, 6-78
rwalk_init 6-23, 6-78
rwalk_t 6-79
S
S_addr 6-69
s_aliases 6-68
s_b1 6-69
s_b2 6-69
s_b3 6-69
s_b4 6-69
s_name 6-68
s_port 6-68
s_proto 6-68
S_un 6-69
S_un_b 6-69
S_un_w 6-69
s_w1 6-69
s_w2 6-69
sa_data 6-74
sa_family 6-74
sa_flags 6P-9
sa_handler 6-8
sac.h 6-57
SAV 1-9
SAVE 3P-12, 3P-19
save 3P-9

-
- SaveResult 10-51
 - SaveTextBuffer 10-49
 - sbrk 6-4
 - Scalb 6-5
 - scalbn 6-51
 - Scale.h 10-66
 - scandir 6-80
 - scandir64 6-12, 6-80
 - scanf 6-26
 - scanf(BA_LIB) 6-26
 - ScanResult 10-51
 - SCD 1-1, 1-9, 2-1, 3-1, 6-80, 7-1, 11-1
 - SCD 2.2
 - terminology 1-1
 - SCD 2.3 10-1
 - SCD 2.4 2-2, 6-4, 6-26, 9-1, 10-2
 - SCD 2.x 8-3
 - SCD2.4 2-1
 - sched.h 6-62
 - SCHED_FIFO 6-62
 - sched_get_priority_max 6-59
 - sched_get_priority_min 6-59
 - sched_getparam 6-59
 - sched_getscheduler 6-59
 - SCHED_OTHER 6-62
 - SCHED_RR 6-62
 - sched_rr_get_interval 6-59
 - sched_setparam 6-59
 - sched_setscheduler 6-59
 - sched_yield 6-59
 - Screen 10-11
 - Screen.h 10-66
 - screenConvertArg 10-32
 - ScreenCount
 - unsafe, use XScreenCount 10-8
 - screenNumber 10-10
 - ScreenNumberOfCCC
 - unsafe, use XScreenNumberOfCCC 10-8
 - ScreenOfDisplay
 - unsafe, use XScreenOfDisplay 10-8
 - ScreenWhiteOfCCC
 - unsafe, use XScreenWhiteOfCCC 10-8
 - Scrollbar.h 10-66
 - scrollbarClassRec 10-50
 - ScrollbarWidget 10-52
 - ScrollbarWidgetClass 10-52
 - scrollbarWidgetClass 10-50
 - ScrolledW.h 10-66
 - scrolledWindowClassRec 10-50
 - ScrolledWindowWidget 10-53
 - ScrolledWindowWidgetClass 10-53
 - scrolledWindowWidgetClass 10-50
 - ScrollingListWidget 10-53
 - ScrollingListWidgetClass 10-53
 - scrollingListWidgetClass 10-50
 - SCT 1-9
 - search.h 6-35
 - sed 8-1
 - SEEK_CUR 6-3
 - SEEK_END 6-3
 - SEEK_SET 6-3
 - SelectioB.h 10-66
 - Self-Modifying 3-1
 - sem_close 6-59
 - sem_destroy 6-59
 - sem_init 6-59
 - sem_open 6-59
 - sem_post 6-59
 - sem_trywait 6-59
 - sem_unlink 6-59
 - sem_wait 6-59
 - sema_destroy 6-23, 6-78
 - sema_init 6-23, 6-78
 - sema_post 6-23, 6-78
 - sema_t 6-79
 - sema_trywait 6-23, 6-78
 - sema_wait 6-23, 6-78
 - semid_ds 6P-8
 - send 6-67
 - sendmsg 6-67
 - sendto 6-67
 - SeparatoG.h 10-66
 - Separator.h 10-66
 - ServerVendor
 - unsafe, use XServerVendor 10-8
 - setbuffer 6-80
 - setcontext 6-23, 6-78
 - setgrent 6-4
 - sethi 3P-28, 3P-30, 3P-31, 3P-32, 5P-7
 - sethostent 6-56
 - sethostname 6-80
 - setitimer 6-5, 6-8
 - setjmp 3P-11, 3P-34, 6-80
 - setjmp.h 6P-8
 - setkey 6-4
 - setlinebuf 6-80
 - setnetconfig 6-56
 - setnetent 6-67
 - setnetpath 6-56
 - setpriority 6-80
-

setprocset 6-40
setprotoent 6-67
setregid 6-80
setreuid 6-80
setrlimit 3P-17
setrlimit64 6-12
setrpcent 6-56
setservent 6-67
setsockopt 6-67
setstate 6-80
settimeofday 6-80
sh 8-1, 8-3
sh(BU_CMD) 8-3
sh_addr 4G-2, 6-48
sh_addralign 4G-2, 6-48
sh_entsize 4G-2, 6-48
sh_flags 4G-2, 6-48
sh_info 4G-2, 4P-2, 6-48
sh_link 4G-2, 4P-2, 6-48
sh_name 4G-2, 6-48
sh_offset 4G-2, 6-48
sh_size 4G-2, 6-48
sh_type 4G-2, 6-48
shadow.h 6-35
Shape Extension 1-5
Share Library Names 6-9, 6-10
SHARED 6-38
Shared Library Names 6-9
ShellClassExtension 10-44
ShellClassExtensionRec 10-44
ShellClassPart 10-44
ShellClassRec 10-44
shellClassRec 10-32
ShellPart 10-45
ShellRec 10-45
ShellWidget 10-45
ShellWidgetClass 10-36
shellWidgetClass 10-32
SHF_EXCLUDE 4P-2
SHF_ORDERED 4P-2
ShiftMapIndex 10-10
shm_open 6-59
shm_unlink 6-59
shmid_ds 6P-8
SHN_ABS 4P-3, 4P-9
SHN_AFTER 4P-2
SHN_BEFORE 4P-2
SHN_UNDEF 4P-3, 4P-4, 5P-4
short 6-44
SHT_DYNAMIC 6-47
SHT_DYNSYM 4-1, 6-47
SHT_HASH 6-47
SHT_NOBITS 6-47
SHT_NOTE 6-47
SHT_NULL 6-47
SHT_PROGBITS 6-47
SHT_REL 6-47
SHT_RELA 6-47
SHT_STRTAB 6-47
SHT_SYMTAB 6-47
shutdown 6-67
SL_ARCHITECTURE 6-37
SL_HOSTNAME 6-37
SL_HW_PROVIDER 6-37
SL_HW_SERIAL 6-37
SL_MACHINE 6-37
SL_RELEASE 6-37
SL_SRPC_DOMAIN 6-37
SL_SYSNAME 6-37
SL_VERSION 6-37
sigaction 6-23, 6-78, 6P-9
sigaltstack 6-5, 6P-9
sigbits 6P-9
sigblock 6-80
SIGBUS 3P-18
sigdisp_t sa_disp 6-8
SIGEMT 3P-18
sigev_notify 6-35
sigev_signo 6-35
sigev_value 6-35
sigevent 6-35
SIGFPE 3P-18, 3P-19
SIGILL 3P-18, 3P-19
siginfo.h 6-35
siginfo_t 6P-9
siginterrupt 6-80
sigjmp_buf 6P-8
signal 3P-11, 3P-18, 6-80
signal.h 6-5, 6-8, 6-22, 6P-9
significand 6-51
sigpause 6-80
sigprocmask 6-23, 6-78
SIGPROF 6-8
sigqueue 6-59
SIGSEGV 3P-18
sigset_t 6P-9
sigsetjmp 3P-11
sigsetmask 6-80
sigstack 6-80, 6P-9
SIGSYS 3P-19

-
- sigtimedwait 6-59
 - SIGTRAP 3P-18
 - sigval 6-35
 - sigvec 6-80
 - SIGVTALRM 6-8
 - sigwait 6-23, 6-78
 - sigwaitinfo 6-59
 - simm13 3P-21, 3P-22, 4-2, 4P-8
 - Simple Mail Transport Protocol 7-3
 - sin 6-51
 - sin_addr 6-69
 - sin_family 6-69
 - sin_port 6-69
 - sin_zero 6-69
 - sinh 6-51
 - SIR 3P-24
 - sival_int 6-35
 - sival_ptr 6-35
 - size_t 6-4, 6-5, 6-47, 6-48, 6P-10
 - sleep 6-23, 6-78, 6-80, 8-1
 - sliderClassRec 10-50
 - SliderWidget 10-53
 - SliderWidgetClass 10-53
 - sliderWidgetClass 10-50
 - sllx 3P-31, 3P-32
 - smtp 7-3
 - SO_ACCEPTCONN 6-73
 - SO_BROADCAST 6-73
 - SO_DEBUG 6-73
 - SO_DONTROUTE 6-73
 - SO_ERROR 6-73
 - SO_KEEPALIVE 6-73
 - SO_LINGER 6-73
 - SO_OOBINLINE 6-73
 - SO_PROTOTYPE 6-73
 - SO_RCVBUF 6-73
 - SO_RCVLOWAT 6-73
 - SO_RCVTIMEO 6-73
 - SO_REUSEADDR 6-73
 - SO_SNDBUF 6-73
 - SO_SNDLOWAT 6-73
 - SO_SNDTIMEO 6-73
 - SO_TYPE 6-73
 - SO_USELOOPBACK 6-73
 - SOCK_DGRAM 6-73
 - SOCK_RAW 6-73
 - SOCK_RDM 6-73
 - SOCK_SEQPACKET 6-73
 - SOCK_STREAM 6-73, 6-76
 - sockaddr 6-74
 - sockaddr_in 6-65, 6-69
 - Socket 6-67, 6-68
 - socket 6-67, 6-74
 - socketpair 6-67
 - socknewproto 6-75
 - sockopt 6-74
 - sockproto 6-74
 - socksysreq 6-75
 - Software Installation i
 - SOL_SOCKET 6-74
 - SOMAXCONN 6-75
 - sort 8-1
 - sort_list 6-65
 - sp_family 6-74
 - sp_protocol 6-74
 - SPARC 1-1, 1-4, 1-9, 3P-1, 4P-6, 5P-2, 5P-8, 6-3
 - SPARC ABI 1-1
 - SPARC Application Verifier 1-9
 - SPARC International 1-9, 3P-19, 3P-20, 4-1, 4P-1, 5-1, 6-2, 8-2, 9-1, 12-1
 - SPARC Processor Supplement i
 - SPARC V8 1-4
 - SPARC V9 1-4, 3P-1, 3P-21, 4P-1
 - SPARC_MAXREGWINDOW 6P-14
 - sparcv9/libaio 6-2
 - sparcv9/libc 6-2
 - sparcv9/libdl 6-2
 - sparcv9/libelf 6-2
 - sparcv9/libm 6-2
 - sparcv9/libnsl 6-2
 - sparcv9/libposix4 6-2
 - sparcv9/libpthread 6-2
 - sparcv9/libresolv 6-2
 - sparcv9/libsocket 6-2
 - sparcv9/libthread 6-2
 - sparcv9/libw 6-2
 - speed 6-57
 - speed_t 6P-11
 - sprintf 6-80
 - spwd 6-35
 - sqrt 6-51
 - srand 6-80
 - srandom 6-80
 - SS_DISABLE 6P-9
 - ss_flags 6-5
 - ss_size 6-5
 - ss_sp 6-5
 - st_atim 6-10, 6-15
 - st_atime 6P-10
 - st_blksize 6-10, 6-15
 - st_blocks 6-11, 6-15, 6P-10
-

st_ctim 6-10, 6-15
st_ctime 6P-10
st_dev 6-10, 6-15
st_fstype 6-11, 6-15, 6P-10
st_gid 6-10, 6-15
st_info 4G-2, 4P-3, 6-49
st_ino 6-10, 6-15
st_mode 6-10, 6-15
st_mtim 6-10, 6-15
st_mtime 6P-10
st_name 4G-2, 4P-3, 6-49
st_nlink 6-10, 6-15
st_other 4G-2, 4P-3, 6-49
st_pad1 6-10, 6-15
st_pad2 6-10, 6-15
st_pad3 6-10
st_pad4 6-11, 6-15
st_rdev 6-10, 6-15
ST_RETURN_FROM_DEFERRED_TRAP 3P-22
st_shndx 4G-2, 4P-3, 4P-4, 5P-4, 6-49
st_size 4G-2, 4P-3, 6-10, 6-15, 6-49
st_uid 6-10, 6-15
st_value 4G-2, 4P-3, 4P-4, 5P-4, 6-49
stack_t 6P-9
StartCurrentTextBufferWord 10-49
stat 6-3, 6-10, 6-13, 6P-10
stat.h 6-10
stat64 6-12, 6-13, 6-15
state 6-65
statictextClassRec 10-50
StaticTextWidget 10-53
StaticTextWidgetClass 10-53
staticTextWidgetClass 10-50
statictextWidgetClass 10-50
STATUS 6-64
Status 10-10
statvfs 6-11, 6P-10
statvfs.h 6-11
statvfs_t 6-11, 6P-10
statvfs64 6-12, 6-15
statvfs64_t 6-15
STB_GLOBAL 4-1, 4P-3
STB_LOCAL 4P-3
STB_WEAK 4P-3
STBAR 3P-24
stdarg.h 3P-33
stddef.h 6-35, 6P-10
STDF 3P-2
stdio.h 6-10, 6-14, 6P-11
stdlib.h 6-6
STO_SPARC_REGISTER_G2 4P-3
STO_SPARC_REGISTER_G3 4P-3
STQF 3P-2
strbuf 6P-11
STREAMS 12-1
strfdinsert 6P-11
String 10-33
stropts.h 6P-11
strpeek 6P-11
strrecvfd 6P-11
strtok_r 6-23
struct aio_result_t 6-25
struct iov 6-4
struct protoent 6-68
struct servent 6-68
STT_FUNC 5P-4
STT_REGISTER 5P-3
STT_SPARC_REGISTER 4P-3
stty 8-1
stx 3P-28, 3P-31
su 8-1
subcc 3P-32
Substitution 10-34
SubstitutionRec 10-34
sum 8-1
SUNYP 6-70
supdup 7-3
SUPDUP Protocol 7-3
svc_auth_reg 6-56
svc_create 6-56
svc_destroy 6-56
svc_dg_create 6-56
svc_dg_enablecache 6-56
svc_exit 6-56
svc_fd_create 6-56
svc_fdset 6-7, 6-54
svc_freeargs 6-56
svc_getargs 6-56
svc_getcaller 6-56
svc_getreq_common 6-56
svc_getreq_poll 6-56
svc_getreqset 6-56
svc_getrppcaller 6-56
svc_pollset 6-56
svc_raw_create 6-56
svc_reg 6-56
svc_register 6-56
svc_req 6P-7
svc_sendreply 6-56
svc_tli_create 6-56

svc_tp_create 6-56
 svc_unregister 6-56
 svc_vc_create 6-56
 svcerr_auth 6-56
 svcerr_decode 6-56
 svcerr_noproc 6-56
 svcerr_noprogram 6-56
 svcerr_progvers 6-7, 6-56
 svcerr_systemerr 6-56
 svcerr_weakauth 6-56
 svdfd_create 6-56
 svcraw_create 6-56
 svctcp_create 6-56
 svcudp_bufcreate 6-56
 svcudp_create 6-56
 SVCXPRT 6-7, 6P-7
 SVID 6-2, 6-4, 6-5, 6-7, 6-46, 6-50, 6-80, 8-2, 8-3
 SWAP 3P-21
 swapent 6-41
 swapent_t 6-41
 swapes 6-41
 swapes_t 6-41
 swaptable 6-41
 swaptbl_t 6-41
 symlink 6-4
 SYMTYPE 8-2
 sync_instruction_memory 3P-1
 synch.h 6-79
 sys/asynch.h 6-25, 6-47, 6-49, 6-51, 6-53, 6-65
 sys/dirent.h 6-13, 6-36
 sys/euc.h 6-81
 sys/fcntl.h 6-13
 sys/fstyp.h 6-36
 sys/lock.h 6-38
 sys/mman.h 6-38
 sys/mnttab.h 6-36
 sys/netconfig.h 6P-4
 sys/param.h 6-8
 sys/priocntl.h 6-39
 sys/processor.h 6-39
 sys/procset.h 6-40
 sys/regset.h 6P-13, 6P-14
 sys/resource.h 6-13, 6P-5
 sys/ressource.h 6-15
 sys/rtpriocntl.h 6-40
 sys/sem.h 6P-8
 sys/shm.h 6P-8
 sys/siginfo.h 6P-9
 sys/socket.h 6-73
 sys/stat.h 6-15, 6P-10
 sys/statvfs.h 6-15, 6P-10
 sys/stdio.h 6-13
 sys/swap.h 6-41
 sys/systeminfo.h 6-37
 sys/time.h 6-41, 6P-12
 sys/tiuser.h 6-8
 sys/tspriocntl.h 6-42
 sys/types.h 6-8, 6-11, 6-15, 6-42, 6P-12
 sys/uadmin.h 6-37
 sys/utrap.h 3P-20, 6P-15
 sys/vfstab.h 6-37
 sys/wait.h 6-8
 sysconf 3P-16
 sysinfo 6-5, 6-37
 syslog.h 6-43
 systat 7-3
 System Commands i, 8-1
 System Compliance Test 1-9
 System V i, 8-1, 11-1
 System V Application Binary Interface 1-4, 1-5
 System V Application Binary Interface SPARC Processor Supplement 1-5
 System V Interface Definition 1-5
 system(BA_OS) 6-4
T
 t_accept 6-56
 t_alloc 6-7
 t_bind 6P-12
 t_erno 6-54
 t_free 6-56
 t_getinfo 6-56
 t_getstate 6-7, 6-56
 t_info 6P-12
 t_listen 6-56
 t_look 6-56
 t_open 6-56
 t_optmgmt 6-56, 6P-12
 t_rcv 6-56
 t_rcvconnect 6-56
 t_rcvdis 6-56
 t_rcvrel 6-56
 t_rcvudata 6-56
 t_rcvuderr 6-56
 t_snd 6-56
 t_snddis 6-56
 t_sndrel 6-56
 t_sndudata 6-56
 t_sterror 6-56
 t_sync 6-56
 t_uderr 6P-12
 t_unbind 6-56

TA_ASN1 6-70
TA_BINARY 6-70
TA_CASE 6-70
TA_CRYPT 6-70
TA_MODIFIED 6-70
TA_SEARCHABLE 6-70
TA_XDR 6-70
table_col 6-71
TABLE_OBJ 6-70
table_obj 6-72
TabTable 10-53
taddr2uaddr 6-56
tag_overflow 3P-18, 3P-20
tail 8-1
talk 7-2
tan 6-51
tanh 6-51
tar.h 8-2
tc_flags 6-71
tc_name 6-71
tc_rights 6-71
tcfalg_t 6P-11
TCP 6-74, 7-1
TCP/IP 6-67
tcpmux 7-3
tee 8-1
telnet 7-2, 7-3
telno 6-57
TEM 3-2, 3P-10, 3P-23
Terminal Connection 7-3
terminfo 7-1
termio 6-57
termios 6-5
termios.h 6P-11
test 8-1
Text.h 10-66
TextBlock 10-54
TextBuffer 10-54
textdomain 6-50
TextEditWidget 10-54
TextEditWidgetClass 10-54
textEditWidgetClass 10-50
TextF.h 10-66
textFieldClassRec 10-50
TextFieldWidget 10-54
TextFieldWidgetClass 10-54
textFieldWidgetClass 10-50
TextFileStatus 10-51
TextLine 10-54
TextLocation 10-54
TextPage 10-54
TextPosition 10-54
TextUndoHint 10-54
TextUndoItem 10-54
TextUpdateCallback 10-54
TextUpdateFunction 10-54
tftp 7-3
TGEXEC 8-2
TGREAD 8-2
TGWRITE 8-2
The SPARC Architecture Manual 1-5
The X Window System 1-5
THR_BOUND 6-79
thr_continue 6-23, 6-78
thr_create 6-23, 6-78
THR_DAEMON 6-79
THR_DETACHED 6-79
thr_exit 6-23, 6-78
thr_getconcurrency 6-23, 6-78
thr_getprio 6-23, 6-78
thr_getspecific 6-23, 6-78
thr_join 6-23, 6-78
thr_keycreate 6-23, 6-78
thr_kill 6-23, 6-78
thr_main 6-23, 6-78
thr_min_stack 6-23, 6-78
THR_NEW_LWP 6-79
thr_self 6-23, 6-78
thr_setconcurrency 6-23, 6-78
thr_setprio 6-23, 6-78
thr_setspecific 6-23, 6-78
thr_sigsetmask 6-23, 6-78
thr_suspend 6-23, 6-78
THR_SUSPENDED 6-79
thr_yield 6-23, 6-78
thread.h 6-79
thread_key_t 6-79
thread_t 6-79
TICK 3P-24
Time 7-3
time 7-3
time.h 6-22
time_t 6-47, 6P-8
timer_create 6-59
timer_delete 6-59
timer_getoverrun 6-59
timer_gettime 6-59
timer_settime 6-59
times 6-80
timespec 6P-12

timestruc_t 6-10, 6-15, 6P-12
 timeval 6-44, 6P-12
 timezone 6-27, 6P-12
 T-imm22 4-1
 tiuser.h 6-8, 6P-12
 TLI 7-1
 TMAGIC 8-2
 TMAGLEN 8-2
 tmpfile64 6-12
 TOEXEC 8-2
 Toolkit 10-1
 TopLevelShellClassPart 10-46
 TopLevelShellClassRec 10-46
 topLevelShellClassRec 10-32
 TopLevelShellPart 10-46
 TopLevelShellRec 10-46
 TopLevelShellWidget 10-46
 TopLevelShellWidgetClass 10-36
 topLevelShellWidgetClass 10-32
 TOREAD 8-2
 touch 8-1
 tolower 6-81
 TOWRITE 8-2
 toupper 6-81
 tpicommon.h 6P-15
 tr 8-1
 TransientShellClassPart 10-45
 TransientShellClassRec 10-46
 transientShellClassRec 10-32
 TransientShellPart 10-46
 TransientShellRec 10-46
 TransientShellWidget 10-46
 TransientShellWidgetClass 10-36
 transientShellWidgetClass 10-32
 trap_instruction 3P-18, 3P-20
 Trivial File Transfer 7-3
 true 8-1
 truncate 6-8
 truncate() 6-8
 truncate64 6-12
 TRY_AGAIN 6-68
 ts_maxupri 6-42
 TS_NOCHANGE 6-42
 ts_upri 6-42
 ts_uprilm 6-42
 TSGID 8-2
 T-simm13 4-1
 tsinfo 6-42
 tsinfo_t 6-42
 TSO 4P-1
 tsparms 6-42
 tsparms_t 6-42
 TSUID 8-2
 TSVTX 8-2
 tty 8-1
 ttyname_r 6-22, 6-23
 TUEXEC 8-2
 TUREAD 8-2
 TUWRITE 8-2
 tv_sec 6P-10
 TVERSION 8-2
 TVERSLEN 8-2
 TXTLOCK 6-38
 tzname 6-27
 tzset 6-4
 U
 u_char 6-69
 u_int 6-53, 6-70
 u_long 6-7, 6-8, 6-58, 6-70
 u_longlong_t 6-42
 u_short 6-8, 6-74
 uaddr 6-70
 uaddr2taddr 6-56
 ualarm 6-80
 UC_ASR 6P-13
 uc_filler 6P-13
 uc_flags 6P-13
 uc_link 6P-13
 uc_mcontext 6P-13
 uc_sigmask 6P-13
 uc_stack 6P-13
 UCB 6-2
 ucontext 6P-13
 ucontext.h 6P-13
 ucontext_t 6P-13
 UDP 6-74, 7-1
 uid 2-1
 uid_t 6-10, 6-15, 6P-12
 uio.h 6P-13
 ulong_t 6-7
 UltraSPARC III 4P-1
 UltraSPARC1 4P-1
 umask 8-1
 uname 6-3, 8-1
 uncompress 8-1
 undial 6-56
 ungetwc 6-81
 UNI 2-2
 unistd.h 6-16, 6-22, 6-43, 6P-13
 UNIX 8-1, 8-3

Unix 1-5
UNKNOWN 6-70
UNLOCK 6-38
UnregisterTextBufferUpdate 10-49
Unsafe Macros
 X Library 10-7
unsigned char 4G-2
unsigned long 6-10
UPDATEA 6-64
UPDATED 6-64
UPDATEM 6-64
UPDATEMA 6-64
USER_PROCESS 6-44
user2netname 6-56
ushort_t 6P-8
USL 1-5
usleep 6-80
ustar 8-2
USYNC_PROCESS 6-79
USYNC_THREAD 6-79
UT_ASYNC_DATA_ERROR 3P-20
UT_DATA_ERROR 3P-20, 3P-21
UT_DATA_EXCEPTION 3P-20, 3P-21
UT_DATA_PROTECTION 3P-20, 3P-21
UT_DIVISION_BY_ZERO 3P-20
ut_exit 6-44
UT_FP_DISABLED 3P-20, 3P-21, 3P-22, 3P-23
UT_FP_EXCEPTION_IEEE_754 3P-20
UT_FP_EXCEPTION_OTHER 3P-20
ut_host 6-44
ut_id 6-44
UT_ILLEGAL_INSTRUCTION 3P-20, 3P-21
UT_ILLTRAP_INSTRUCTION 3P-20, 3P-21
UT_INSTRUCTION_ERROR 3P-20, 3P-21
UT_INSTRUCTION_EXCEPTION 3P-20, 3P-21
UT_INSTRUCTION_PROTECTION 3P-20, 3P-21
ut_line 6-44
UT_MEM_ADDRESS_NOT_ALIGNED 3P-19, 3P-20
ut_pid 6-44
UT_PRIVILEGED_ACTION 3P-20
UT_PRIVILEGED_OPCODE 3P-20, 3P-21
ut_session 6-44
ut_syslen 6-44
UT_TAG_OVERFLOW 3P-20
ut_time 6-44
UT_TRAP_INSTRUCTION_16 3P-20
UT_TRAP_INSTRUCTION_31 3P-20
ut_tv 6-44
ut_type 6-44
ut_user 6-44
utmp 6-44
utmpx 6-44
utmpx.h 6-44
utrap_handler_t 6P-15
uucp 8-1
UUCP Path Service 7-3
uucp-path 7-3
uulog 8-1
uustat 8-1
uux 8-1
UXP/DS 6-5
V
varargs.h 3P-33
V-d2/disp14 4-1
V-disp19 4-1
V-disp22 4-1
vendor 6G-1
VendorRelease
 unsafe, use XVendorRelease 10-8
VendorS.h 10-67
VendorShellClassPart 10-47
VendorShellClassRec 10-47
vendorShellClassRec 10-32, 10-50
VendorShellPart 10-47
VendorShellRec 10-47
VendorShellWidget 10-47
VendorShellWidgetClass 10-39
vendorShellWidgetClass 10-32, 10-50, 10-58
ver 3P-23
vfork 3P-11
vfs_automnt 6-37
vfs_fsckdev 6-37
vfs_fsckpass 6-37
vfs_fstype 6-37
VFS_LINE_MAX 6-37
vfs_mntopts 6-37
vfs_mountp 6-37
vfs_special 6-37
VFS_TOOFEW 6-37
VFS_TOOLONG 6-37
VFS_TOOMANY 6-37
vfstab 6-37
vi 8-1
V-imm22 4-1, 4P-7
V-imm5 4-1
V-imm6 4-1
V-imm7 4-1
VirtKeys.h 10-67
Visual 10-10
Visual 10-11

-
- visual 10-10
 - VisualOfCCC
 - unsafe, use XVisualOfCCC 10-8
 - volatile 3P-10, 3P-11
 - V-simm10 4-1
 - V-simm11 4-1
 - V-simm12 4-1
 - V-simm13 4-1
 - vsprintf 6-80
 - V-word32 4-1
 - V-xword64 4-1
 - W**
 - wait 8-1
 - wait.h 6-8
 - wait3 6-80
 - waitid 6-4
 - wc 8-1
 - wchar.h 6-82
 - wchar_t 6-6, 6-82, 6P-10
 - wctype.h 6-82
 - WhitePixel
 - unsafe, use XWhitePixel 10-8
 - WhitePixelOfScreen
 - unsafe, use XWhitePixelOfScreen 10-8
 - whitePtAdjClientData 10-10
 - whitePtAdjProc 10-10
 - whitespace_form 6-32
 - who 8-1, 8-3
 - Who Is 7-3
 - Who is up 7-3
 - Widget 10-33
 - WidgetClass 10-33
 - widgetClass 10-32
 - WidgetClassRec 10-42
 - widgetClassRec 10-32
 - WidgetList 10-33
 - WidgetRec 10-41
 - WidthMMOfScreen
 - unsafe, use XWidthMMOfScreen 10-8
 - WidthOfScreen
 - unsafe, use XWidthOfScreen 10-8
 - Window Save 3-2
 - Windowing and Terminal Interfaces i
 - wint_t 6-82
 - WShellClassPart 10-45
 - WShellClassRec 10-45
 - wmShellClassRec 10-32
 - WShellPart 10-45
 - WShellRec 10-45
 - WShellWidget 10-45
 - WShellWidgetClass 10-36
 - wmShellWidgetClass 10-32
 - WRASI 3P-24
 - WRASR 3P-24
 - WRCCR 3P-24
 - WRFPRS 3P-24
 - writev 6-4
 - WRY 3P-24
 - wscasecmp 6-81
 - wscat 6-81
 - wchr 6-81
 - wscmp 6-81
 - wscol 6-81
 - wscoll 6-81
 - wscopy 6-81
 - wscspn 6-81
 - wsdup 6-81
 - wslen 6-81
 - wsncasecmp 6-81
 - wsncat 6-81
 - wsncmp 6-81
 - wsncpy 6-81
 - wspbrk 6-81
 - wsprintf 6-81
 - wsrchr 6-81
 - wsscanf 6-81
 - wsspn 6-81
 - wstod 6-81
 - wstok 6-81
 - wstol 6-81
 - wsxfrm 6-81
 - WTRACED 6-4
 - WTRAPPED 6-4
 - X**
 - X 7-3
 - X Toolkit Intrinsic 1-5
 - X Window Service 7-3
 - X.25 6-74
 - X/Open 1-7
 - x_base 6P-8
 - x_control 6P-8
 - x_destroy 6P-8
 - x_getbytes 6P-8
 - x_getint32 6P-8
 - x_getlong 6P-8
 - x_getpostn 6P-8
 - x_handy 6P-8
 - x_inline 6P-8
 - x_op 6P-8
 - x_private 6P-8
-

X_PROTOCOL 10-10
X_PROTOCOL_REVISION 10-10
x_public 6P-8
x_putbytes 6P-8
x_putint32 6P-8
x_putlong 6P-8
x_setpostn 6P-8
X11 1-4
X11/Composite.h 10-33
X11/CompositeP.h 10-40
X11/ConstrainP.h 10-41
X11/Constraint.h 10-33
X11/Core.h 10-33
X11/CoreP.h 10-41
X11/extensions/shape.h
 Manifest Constants 10-26
X11/Intrinsic.h> 10-33
X11/IntrinsicP.h 10-42
X11/keysymdef.h 10-13
 Manifest Constants 10-13
X11/keysymdef.h 10-9
X11/Object.h 10-35
X11/RectObj.h 10-35
X11/Shell.h 10-35
X11/ShellP.h 10-44
X11/StringDefs.h 10-36
X11/Vendor.h 10-39
X11/VendorP.h 10-47
X11/X.h 10-10
X11/X.h 10-9
X11/Xcms.h 10-10
X11/Xcms.h 10-9
X11/Xlib. 6-68, 6-70
X11/Xlib.h 10-11, 10-12
X11/Xlib.h 10-9
X11/Xresource.h 10-11
X11/Xresource.h 10-9
X11/Xutil.h 10-12
X11/Xutil.h 10-9
X11R4 10-2
X500 6-70
XActivateScreenSaver 10-3
XAddExtension 10-3
XAddHost 10-3
XAddHosts 10-3
XAddPixel 10-3
XAddToExtensionList 10-3
XAddToSaveSet 10-3
XAllocClassHint 10-3
XAllocColor 10-3
XAllocColorCells 10-3
XAllocColorPlanes 10-3
XAllocIconSize 10-3
XAllocNamedColor 10-3
XAllocSizeHints 10-3
XAllocStandardColormap 10-3
XAllocWMHints 10-3
XAllowEvents 10-3
XAllPlanes 10-3
XAutoRepeatOff 10-3
XAutoRepeatOn 10-3
XBaseFontNameListOffFontSet 10-3
XBell 10-3
XBitmapBitOrder 10-3
XBitmapPad 10-3, 10-7
XBitmapUnit 10-3, 10-7
XBlackPixel 10-3, 10-7
XBlackPixelOfScreen 10-3, 10-7
xcc 3P-23
XCellsOfScreen 10-3, 10-7
XChangeActivePointerGrab 10-3
XChangeGC 10-3
XChangeKeyboardControl 10-3
XChangeKeyboardMapping 10-3
XChangePointerControl 10-3
XChangeProperty 10-3
XChangeSaveSet 10-3
XChangeWindowAttributes 10-3
XCheckIfEvent 10-3
XCheckMaskEvent 10-3
XCheckTypedEvent 10-3
XCheckTypedWindowEvent 10-3
XCheckWindowEvent 10-3
XCHS 6-70
XCirculateSubwindows 10-3
XCirculateSubwindowsDown 10-3
XCirculateSubwindowsUp 10-3
XClearArea 10-3
XClearWindow 10-3
XClientWhitePointOfCCC 10-7
XClipBox 10-3
XCloseDisplay 10-3
XCloseIM 10-3
XcmsAddColorSpace 10-3
XcmsAddFunctionSet 10-3
XcmsAllocColor 10-3
XcmsAllocNamedColor 10-3
XcmsCCC 10-10
XcmsCCC 10-10
XcmsCCCOfColormap 10-3

XcmsCCCRec 10-10
XcmsCIELabClipab 10-3
XcmsCIELabClipL 10-3
XcmsCIELabClipLab 10-3
XcmsCIELabColorSpace 10-6
XcmsCIELabQueryMaxC 10-3
XcmsCIELabQueryMaxL 10-3
XcmsCIELabQueryMaxLC 10-3
XcmsCIELabQueryMinL 10-3
XcmsCIELabToCIEXYZ 10-3
XcmsCIELabWhiteShiftColors 10-3
XcmsCIELuvClipL 10-3
XcmsCIELuvClipLuv 10-3
XcmsCIELuvClipuv 10-3
XcmsCIELuvColorSpace 10-6
XcmsCIELuvQueryMaxC 10-3
XcmsCIELuvQueryMaxL 10-3
XcmsCIELuvQueryMaxLC 10-3
XcmsCIELuvQueryMinL 10-3
XcmsCIELuvToCIEuvY 10-3
XcmsCIELuvWhiteShiftColors 10-3
XcmsCIEuvYColorSpace 10-6
XcmsCIEuvYToCIELuv 10-3
XcmsCIEuvYToCIEXYZ 10-3
XcmsCIEuvYToTekHVC 10-3
XcmsCIExyYColorSpace 10-6
XcmsCIExyYToCIEXYZ 10-3
XcmsCIEXYZColorSpace 10-6
XcmsCIEXYZToCIELab 10-3
XcmsCIEXYZToCIEuvY 10-3
XcmsCIEXYZToCIExyY 10-3
XcmsCIEXYZToRGBi 10-3
XcmsClientWhitePointOfCCC 10-3
XcmsColor 10-10
XcmsCompressionPro 10-10
XcmsConvertColors 10-3
XcmsCreateCCC 10-3
XcmsDefaultCCC 10-3
XcmsDisplayOfCCC 10-3
XcmsFormatOfPrefix 10-3
XcmsFreeCCC 10-3
XcmsLinearRGBFunctionSet 10-6
XcmsLookupColor 10-3
XcmsPerScrnInfo 10-10
XcmsPrefixOfFormat 10-3
XcmsQueryBlack 10-3
XcmsQueryBlue 10-3
XcmsQueryColor 10-3
XcmsQueryColors 10-3
XcmsQueryGreen 10-3
XcmsQueryRed 10-3
XcmsQueryWhite 10-3
XcmsRGBColorSpace 10-6
XcmsRGBiColorSpace 10-6
XcmsRGBiToCIEXYZ 10-3
XcmsRGBiToRGB 10-3
XcmsRGBToRGBi 10-3
XcmsScreenFreeProc 10-10
XcmsScreenInitProc 10-10
XcmsScreenNumberOfCCC 10-3
XcmsScreenWhitePointOfCCC 10-3
XcmsSetCompressionProc 10-3
XcmsSetWhiteAdjustProc 10-3
XcmsSetWhitePoint 10-3
XcmsStoreColor 10-3
XcmsStoreColors 10-3
XcmsTekHVCClipC 10-3
XcmsTekHVCClipV 10-3
XcmsTekHVCClipVC 10-3
XcmsTekHVCColorSpace 10-6
XcmsTekHVCQueryMaxC 10-3
XcmsTekHVCQueryMaxV 10-3
XcmsTekHVCQueryMaxVC 10-3
XcmsTekHVCQueryMaxVSamples 10-3
XcmsTekHVCQueryMinV 10-3
XcmsTekHVCToCIEuvY 10-3
XcmsTekHVCWhiteShiftColors 10-3
XcmsUNDEFINEDColorSpace 10-6
XcmsVisualOfCCC 10-3
XcmsWhiteAdjustProc 10-10
XcmsWhiteAdjustProc 10-10
XConfigureWindow 10-3
XConnectionNumber 10-3, 10-7
XContextDependentDrawing 10-3
XConvertSelection 10-3
XCopyArea 10-3
XCopyColormapAndFree 10-3
XCopyGC 10-3
XCopyPlane 10-3
XCreateBitmapFromData 10-3
XCreateColormap 10-3
XCreateFontCursor 10-3
XCreateFontSet 10-3
XCreateGC 10-3
XCreateGlyphCursor 10-3
XCreateIC 10-3
XCreateImage 10-3
XCreatePixmap 10-3
XCreatePixmapCursor 10-3
XCreatePixmapFromBitmapData 10-3

XCreateRegion 10-3
XCreateSimpleWindow 10-3
XCreateWindow 10-3
XDefaultColormap 10-3, 10-7
XDefaultColormapOfScreen 10-3, 10-7
XDefaultDepth 10-3, 10-7
XDefaultDepthOfScreen 10-3, 10-7
XDefaultGC 10-4, 10-7
XDefaultGCOfScreen 10-3, 10-7
XDefaultRootWindow 10-3, 10-7
XDefaultScreen 10-3, 10-7
XDefaultScreenOfDisplay 10-3, 10-7
XDefaultString 10-3
XDefaultVisual 10-4, 10-7
XDefaultVisualOfScreen 10-4, 10-7
XDefineCursor 10-4
XDeleteContext 10-4
XDeleteModifiermapEntry 10-4
XDeleteProperty 10-4
XDestroyIC 10-4
XDestroyImage 10-4
XDestroyRegion 10-4
XDestroySubwindows 10-4
XDestroyWindow 10-4
XDisableAccessControl 10-4
XDisplayCells 10-4, 10-7
XDisplayHeight 10-4, 10-7
XDisplayHeightMM 10-4, 10-7
XDisplayKeycodes 10-4
XDisplayMotionBufferSize 10-4
XDisplayName 10-4
XDisplayOfCCC 10-7
XDisplayOfIM 10-4
XDisplayOfScreen 10-4, 10-7
XDisplayPlanes 10-4, 10-7
XDisplayString 10-4, 10-7
XDisplayWidth 10-4, 10-8
XDisplayWidthMM 10-4, 10-8
XDoesBackingStore 10-4, 10-8
XDoesSaveUnders 10-4, 10-8
XDR 6P-8, 7G-1
xdr_accepted_reply 6-56
xdr_array 6-56
xdr_authsys_parms 6-56
xdr_hyper 6G-1
xdr_int32 6G-1
xdr_long_double 6G-1
xdr_op 6P-8
xdr_ops 6P-8
xdr_reference 6-56
xdr_rejected_reply 6-56
xdr_replymsg 6-56
xdr_setpos 6-56
xdr_short 6-56
xdr_sizeof 6-56
xdr_sprayarr 6-66
xdr_spraycumul 6-66
xdr_string 6-56
xdr_u_hyper 6G-1
xdr_u_char 6-56
xdr_u_int 6-56
xdr_u_int32 6G-1
xdr_u_long 6-56
xdr_u_short 6-56
xdr_union 6-56
xdr_vector 6-56
xdr_void 6-56
xdr_wrapstring 6-56
XDrawArc 10-4
XDrawArcs 10-4
XDrawImageString 10-4
XDrawImageString16 10-4
XDrawLine 10-4
XDrawLines 10-4
XDrawPoint 10-4
XDrawPoints 10-4
XDrawRectangle 10-4
XDrawRectangles 10-4
XDrawSegments 10-4
XDrawString 10-4
XDrawString16 10-4
XDrawText 10-4
XDrawText16 10-4
xdrmem_create 6-56
xdrproc_t 6-7, 6P-8
xdrrec_create 6-56
xdrrec_endofrecord 6-56
xdrrec_eof 6-56
xdrrec_readbytes 6-56
xdrrec_skiprecord 6-56
xdrstdio_create 6-56
XEHeadOfExtensionList 10-4
XEmptyRegion 10-4
XEnableAccessControl 10-4
XEqualRegion 10-4
XEROX NS 6-74
XESetCloseDisplay 10-4
XESetCopyGC 10-4
XESetCreateFont 10-4
XESetCreateGC 10-4

XESetError 10-4
XESetErrorString 10-4
XESetEventToWire 10-4
XESetFlushGC 10-4
XESetFreeFont 10-4
XESetFreeGC 10-4
XESetPrintErrorValues 10-4
XESetWireToError 10-4
XESetWireToEvent 10-4
XEventMaskOfScreen 10-4, 10-8
XEventsQueued 10-4
XExtCodes 10-11
XExtCodes 10-11
XExtData 10-11
XExtData 10-11
XExtentsOfFontSet 10-4
XFetchBuffer 10-4
XFetchBytes 10-4
XFetchName 10-4
XFillArc 10-4
XFillArcs 10-4
XFillPolygon 10-4
XFillRectangle 10-4
XFillRectangles 10-4
XFilterEvent 10-4
XFindContext 10-4
XFindOnExtensionList 10-4
XFlush 10-4
XFlushGC 10-4
XFontSet 10-12
XFontSet 10-11
XFontsOfFontSet 10-4
XForceScreenSaver 10-4
XFree 10-4
XFreeColormap 10-4
XFreeColors 10-4
XFreeCursor 10-4
XFreeExtensionList 10-4
XFreeFont 10-4
XFreeFontInfo 10-4
XFreeFontNames 10-4
XFreeFontPath 10-4
XFreeFontSet 10-4
XFreeGC 10-4
XFreeModifiermap 10-4
XFreePixmap 10-4
XFreeStringList 10-4
XGCContextFromGC 10-4
XGeometry 10-4
XGetAtomName 10-4
XGetClassHint 10-4
XGetCommand 10-4
XGetDefault 10-4
XGetErrorDatabaseText 10-4
XGetErrorText 10-4
XGetFontPath 10-4
XGetFontProperty 10-4
XGetGCValues 10-4
XGetGeometry 10-4
XGetIconName 10-4
XGetIconSizes 10-4
XGetICValues 10-4
XGetImage 10-4
XGetIMValues 10-4
XGetInputFocus 10-4
XGetKeyboardControl 10-4
XGetKeyboardMapping 10-4
XGetModifierMapping 10-4
XGetMotionEvents 10-4
XGetNormalHints 10-4
XGetPixel 10-4
XGetPointerControl 10-4
XGetPointerMapping 10-4
XGetRGBColormaps 10-4
XGetScreenSaver 10-4
XGetSelectionOwner 10-4
XGetSizeHints 10-4
XGetStandardColormap 10-4
XGetSubImage 10-4
XGetTextProperty 10-4
XGetTransientForHint 10-4
XGetVisualInfo 10-4
XGetWindowAttributes 10-4
XGetWindowProperty 10-4
XGetWMCClientMachine 10-4
XGetWMC colormapWindows 10-4
XGetWMHints 10-4
XGetWMIconName 10-4
XGetWMName 10-4
XGetWMNormalHints 10-4
XGetWMProtocols 10-4
XGetWMSizeHints 10-4
XGrabButton 10-4
XGrabKey 10-4
XGrabKeyboard 10-4
XGrabPointer 10-4
XGrabServer 10-4
XHeightMMOfScreen 10-4, 10-8
XHeightOfScreen 10-4, 10-8
XIC 10-12

XIconifyWindow 10-4
 XIfEvent 10-4
 XIM 10-12
 XImageByteOrder 10-4, 10-8
 XIMOfIC 10-4
 XInitExtension 10-4
 XInsertModifiermapEntry 10-4
 XInstallColormap 10-4
 XInternAtom 10-4
 XIntersectRegion 10-4
 XK_abreve 10-17
 XK_Arabic_ha 10-20
 XK_Arabic_lam 10-20
 XK_Arabic_tah 10-20
 XK_C 10-15
 XK_Cyrillic_je 10-20
 XK_Down 10-13
 XK_ediaeresis 10-17
 XK_F8 10-14
 XK_guillemotleft 10-16
 XK_kana_TA 10-19
 XK_KP_1 10-13
 XK_KP_8 10-13
 XK_ncedilla 10-18
 XK_Pause 10-13
 XK_R11 10-14
 XKeycodeToKeysym 10-5
 XKeysymToKeycode 10-5
 XKeysymToString 10-5
 XKillClient 10-5
 XLastKnownRequestProcessed 10-5, 10-8
 Xlib 10-1
 XListDepths 10-5
 XListExtensions 10-5, 10-6
 XListFonts 10-5
 XListFontsWithInfo 10-5
 XListHosts 10-5
 XListInstalledColormaps 10-5
 XListPixmapFormats 10-5
 XListProperties 10-5
 XLoadFont 10-5
 XLoadQueryFont 10-5
 XLocaleOfFontSet 10-5
 XLocaleOfIM 10-5
 XLookupColor 10-5
 XLookupKeysym 10-5
 XLookupString 10-5
 XLowerWindow 10-5
 XmActivateProtocol 10-56
 XmAddProtocolCallback 10-56
 XmAddProtocols 10-56
 XmAddTabGroup 10-56
 XmAnyCallbackStruct 10-68
 XmAnyICCCallback 10-60
 XmAnyICCCallbackStruct 10-60
 XMapRaised 10-5
 XMapSubwindows 10-5
 XMapWindow 10-5
 XmArrowButtonCallbackStruct 10-68
 xmArrowButtonClassRec 10-58
 XmArrowButtonGadget 10-59
 XmArrowButtonGadgetClass 10-59
 xmArrowButtonGadgetClass 10-58
 xmArrowButtonGadgetClassRec 10-58
 XmArrowButtonWidget 10-59
 XmArrowButtonWidgetClass 10-59
 xmArrowButtonWidgetClass 10-58
 XMaskEvent 10-5
 XMatchVisualInfo 10-5
 XMaxCmapsOfScreen 10-5, 10-8
 XMaxRequestSize 10-5
 XmbDrawImageString 10-5
 XmbDrawString 10-5
 XmbDrawText 10-5
 XmbLookupString 10-5
 XmbResetIC 10-5
 XmbSetWMProperties 10-5
 XmbTextEscapement 10-5
 XmbTextExtents 10-5
 XmbTextItem 10-12
 XmbTextListToTextProperty 10-5
 XmbTextPerCharExtents 10-5
 XmbTextPropertyToTextList 10-5
 xmBulletinBoardClassRec 10-58
 XmBulletinBoardWidget 10-59
 XmBulletinBoardWidgetClass 10-59
 xmBulletinBoardWidgetClass 10-58
 XmButtonType 10-70
 XmButtonTypeTable 10-70
 xmCascadeButtonClassRec 10-58
 XmCascadeButtonGadget 10-59
 XmCascadeButtonGadgetClass 10-59
 xmCascadeButtonGadgetClass 10-58
 xmCascadeButtonGadgetClassRec 10-58
 XmCascadeButtonGadgetHighlight 10-56
 xmCascadeButtonGCacheObjClassRec 10-58
 XmCascadeButtonGCacheObject 10-59
 XmCascadeButtonHighlight 10-56
 XmCascadeButtonWidget 10-59
 XmCascadeButtonWidgetClass 10-59

xmCascadeButtonWidgetClass 10-58
XmChangeColor 10-56
XmClipboardCancelCopy 10-56
XmClipboardCopy 10-56
XmClipboardCopyByName 10-56
XmClipboardEndCopy 10-56
XmClipboardEndRetrieve 10-56
XmClipboardInquireCount 10-56
XmClipboardInquireFormat 10-56
XmClipboardInquireLength 10-56
XmClipboardInquirePendingItems 10-56
XmClipboardLock 10-56
XmClipboardPendingList 10-59
XmClipboardPendingRec 10-59
XmClipboardRegisterFormat 10-56
XmClipboardRetrieve 10-56
XmClipboardStartCopy 10-56
XmClipboardStartRetrieve 10-56
XmClipboardUndoCopy 10-56
XmClipboardUnlock 10-56
XmClipboardWithdrawFormat 10-56
XmCommandAppendValue 10-56
XmCommandCallbackStruct 10-68
xmCommandClassRec 10-58
XmCommandError 10-56
XmCommandGetChild 10-56
XmCommandSetValue 10-56
XmCommandWidget 10-59
XmCommandWidgetClass 10-59
xmCommandWidgetClass 10-58
XmConvertUnits 10-56
XmCreateArrowButton 10-56
XmCreateArrowButtonGadget 10-56
XmCreateBulletinBoard 10-56
XmCreateBulletinBoardDialog 10-56
XmCreateCascadeButton 10-56
XmCreateCascadeButtonGadget 10-56
XmCreateCommand 10-56
XmCreateDialogShell 10-56
XmCreateDragIcon 10-56
XmCreateDrawingArea 10-56
XmCreateDrawnButton 10-56
XmCreateErrorDialog 10-56
XmCreateFileSelectionBox 10-56
XmCreateFileSelectionDialog 10-56
XmCreateForm 10-56
XmCreateFormDialog 10-56
XmCreateFrame 10-56
XmCreateInformationDialog 10-56
XmCreateLabel 10-56
XmCreateLabelGadget 10-56
XmCreateList 10-56
XmCreateMainWindow 10-56
XmCreateMenuBar 10-56
XmCreateMenuShell 10-56
XmCreateMessageBox 10-56
XmCreateMessageDialog 10-56
XmCreateOptionMenu 10-56
XmCreatePanedWindow 10-56
XmCreatePopupMenu 10-56
XmCreatePromptDialog 10-56
XmCreatePulldownMenu 10-56
XmCreatePushButton 10-56
XmCreatePushButtonGadget 10-56
XmCreateQuestionDialog 10-56
XmCreateRadioBox 10-56
XmCreateRowColumn 10-56
XmCreateScale 10-56
XmCreateScrollBar 10-56
XmCreateScrolledList 10-56
XmCreateScrolledText 10-56
XmCreateScrolledWindow 10-56
XmCreateSelectionBox 10-56
XmCreateSelectionDialog 10-56
XmCreateSeparator 10-56
XmCreateSeparatorGadget 10-56
XmCreateSimpleCheckBox 10-56
XmCreateSimpleMenuBar 10-56
XmCreateSimpleOptionMenu 10-56
XmCreateSimplePopupMenu 10-56
XmCreateSimplePulldownMenu 10-56
XmCreateSimpleRadioBox 10-56
XmCreateTemplateDialog 10-56
XmCreateText 10-56
XmCreateTextField 10-56
XmCreateToggleButton 10-56
XmCreateToggleButtonGadget 10-56
XmCreateWarningDialog 10-56
XmCreateWorkArea 10-56
XmCreateWorkingDialog 10-56
XmCvtCTToXmString 10-56
XmCvtStringToUnitType 10-56
XmCvtXmStringToCT 10-56
XmDeactivateProtocol 10-56
xmDesktopClass 10-58
xmDesktopClassRec 10-58
xmDesktopObjectClass 10-58
XmDestroyPixmap 10-56
xmDialogShellClassRec 10-58
xmDialogShellExtClassRec 10-58

xmDialogShellExtObjectClass 10-58
XmDialogShellWidget 10-60
XmDialogShellWidgetClass 10-60
xmDialogShellWidgetClass 10-58
XmDisplay 10-60
XmDisplayClass 10-60
xmDisplayClass 10-58
xmDisplayClassRec 10-58
xmDisplayObjectClass 10-58
XmDragCancel 10-56
XmDragContext 10-60
XmDragContextClass 10-60
xmDragContextClass 10-58
xmDragContextClassRec 10-58
XmDragDropFinishCallback 10-61
XmDragDropFinishCallbackStruct 10-61
xmDragIconClassRec 10-58
XmDragIconObject 10-61
XmDragIconObjectClass 10-61
xmDragIconObjectClass 10-58
XmDragMotionCallback 10-61
XmDragMotionCallbackStruct 10-61
xmDragOverShellClassRec 10-58
XmDragOverShellWidget 10-61
XmDragOverShellWidgetClass 10-61
xmDragOverShellWidgetClass 10-58
XmDragProcCallback 10-62
XmDragProcCallbackStruct 10-62
XmDragStart 10-56
XmDrawingAreaCallbackStruct 10-68
xmDrawingAreaClassRec 10-58
XmDrawingAreaWidget 10-61
XmDrawingAreaWidgetClass 10-61
xmDrawingAreaWidgetClass 10-58
XmDrawnButtonCallbackStruct 10-68
xmDrawnButtonClassRec 10-58
XmDrawnButtonWidget 10-61
XmDrawnButtonWidgetClass 10-61
xmDrawnButtonWidgetClass 10-58
XmDropFinishCallback 10-61
XmDropFinishCallbackStruct 10-61
XmDropProcCallback 10-62
XmDropProcCallbackStruct 10-62
XmDropSiteConfigureStackingOrder 10-56
XmDropSiteEndUpdate 10-56
XmDropSiteEnterCallback 10-60
XmDropSiteEnterCallbackStruct 10-60
XmDropSiteLeaveCallback 10-61
XmDropSiteLeaveCallbackStruct 10-61
xmDropSiteManagerClassRec 10-58
XmDropSiteManagerObject 10-62
XmDropSiteManagerObjectClass 10-62
xmDropSiteManagerObjectClass 10-58
XmDropSiteQueryStackingOrder 10-56
XmDropSiteRegister 10-56
XmDropSiteRetrieve 10-56
XmDropSiteStartUpdate 10-56
XmDropSiteUnregister 10-56
XmDropSiteUpdate 10-56
XmDropSiteVisuals 10-62
XmDropSiteVisualsRec 10-62
XmDropStartCallback 10-61
XmDropStartCallbackStruct 10-61
XmDropTransferAdd 10-56
xmDropTransferClassRec 10-58
XmDropTransferEntry 10-62
XmDropTransferEntryRec 10-62
XmDropTransferObject 10-62
XmDropTransferObjectClass 10-62
xmDropTransferObjectClass 10-58
XmDropTransferStart 10-56
xmExtClassRec 10-58
xmExtObjectClass 10-58
XmFileSelectionBoxCallbackStruct 10-68
xmFileSelectionBoxClassRec 10-58
XmFileSelectionBoxGetChild 10-56
XmFileSelectionBoxWidget 10-63
XmFileSelectionBoxWidgetClass 10-63
xmFileSelectionBoxWidgetClass 10-58
XmFileSelectionDoSearch 10-56
XmFontContext 10-67
XmFontList 10-67
XmFontListAdd 10-56
XmFontListAppendEntry 10-56
XmFontListCopy 10-56
XmFontListCreate 10-56
XmFontListEntry 10-67
XmFontListEntryCreate 10-56
XmFontListEntryFree 10-56
XmFontListEntryGetFont 10-56
XmFontListEntryGetTag 10-56
XmFontListEntryLoad 10-56
XmFontListFree 10-56
XmFontListFreeFontContext 10-56
XmFontListGetNextFont 10-56
XmFontListInitFontContext 10-56
XmFontListNextEntry 10-56
XmFontListRemoveEntry 10-56
XmFontType 10-67
xmFormClassRec 10-58

XmFormWidget 10-63
XmFormWidgetClass 10-63
xmFormWidgetClass 10-58
xmFrameClassRec 10-58
XmFrameWidget 10-63
XmFrameWidgetClass 10-63
xmFrameWidgetClass 10-58
XmGadget 10-67
XmGadgetClass 10-67
xmGadgetClass 10-58
xmGadgetClassRec 10-58
XmGetAtomName 10-56
XmGetColorCalculation 10-56
XmGetColors 10-56
XmGetDestination 10-56
XmGetDragContext 10-56
XmGetFocusWidget 10-56
XmGetMenuCursor 10-56
XmGetPixmap 10-56
XmGetPixmapByDepth 10-56
XmGetPostedFromWidget 10-56
XmGetSecondaryResourceData 10-56
XmGetTabGroup 10-56
XmGetTearOffControl 10-56
XmGetVisibility 10-56
XmGetXmDisplay 10-56
XmGetXmScreen 10-56
XmHighlightMode 10-69
XmID 10-60
XMinCmapsOfScreen 10-5, 10-8
XmInstallImage 10-56
XmInternAtom 10-56
XmIsMotifWMRunning 10-56
XmIsTraversable 10-56
XmKeySymTable 10-70
xmLabelClassRec 10-58
XmLabelGadget 10-63
XmLabelGadgetClass 10-63
xmLabelGadgetClass 10-58
xmLabelGadgetClassRec 10-58
xmLabelGCacheObjClassRec 10-58
XmLabelGCacheObject 10-63
XmLabelWidget 10-63
XmLabelWidgetClass 10-63
xmLabelWidgetClass 10-58
XmListAddItem 10-56
XmListAddItems 10-56
XmListAddItemsUnselected 10-56
XmListAddItemUnselected 10-56
XmListCallbackStruct 10-68
xmListClassRec 10-58
XmListDeleteAllItems 10-56
XmListDeleteItem 10-56
XmListDeleteItems 10-56
XmListDeleteItemsPos 10-56
XmListDeletePos 10-56
XmListDeletePositions 10-56
XmListDeselectAllItems 10-56
XmListDeselectItem 10-56
XmListDeselectPos 10-56
XmListGetKbdItemPos 10-56
XmListGetMatchPos 10-56
XmListGetSelectedPos 10-56
XmListItemExists 10-56
XmListItemPos 10-56
XmListPosSelected 10-56
XmListPosToBounds 10-56
XmListReplaceItems 10-56
XmListReplaceItemsPos 10-56
XmListReplaceItemsPosUnselected 10-56
XmListReplaceItemsUnselected 10-56
XmListReplacePositions 10-56
XmListSelectItem 10-56
XmListSelectPos 10-56
XmListSetAddMode 10-56
XmListSetBottomItem 10-56
XmListSetBottomPos 10-56
XmListSetHorizPos 10-56
XmListSetItem 10-56
XmListSetKbdItemPos 10-56
XmListSetPos 10-56
XmListUpdateSelectedList 10-56
XmListWidget 10-63
XmListWidgetClass 10-63
xmListWidgetClass 10-58
XmListYToPos 10-56
xmMainWindowClassRec 10-58
XmMainWindowSep1 10-57
XmMainWindowSep2 10-57
XmMainWindowSep3 10-57
XmMainWindowSetAreas 10-57
XmMainWindowWidget 10-63
XmMainWindowWidgetClass 10-63
xmMainWindowWidgetClass 10-58
xmManagerClassRec 10-58
XmManagerWidget 10-67
XmManagerWidgetClass 10-67
xmManagerWidgetClass 10-58
XmMapSegmentEncoding 10-57
XmMenuPosition 10-57

xmMenuShellClassRec 10-58
 XmMenuShellWidget 10-63
 XmMenuShellWidgetClass 10-63
 xmMenuShellWidgetClass 10-58
 xmMessageBoxClassRec 10-58
 XmMessageBoxGetChild 10-57
 XmMessageBoxWidget 10-63
 XmMessageBoxWidgetClass 10-63
 xmMessageBoxWidgetClass 10-58
 XmNavigationType 10-70
 XmOffset 10-70
 XmOffsetPtr 10-70
 XmOperationChangedCallback 10-61
 XmOperationChangedCallbackStruct 10-61
 XmOptionButtonGadget 10-57
 XmOptionLabelGadget 10-57
 XMMoveResizeWindow 10-5
 XMMoveWindow 10-5
 xmPanedWindowClassRec 10-58
 XmPanedWindowWidget 10-65
 XmPanedWindowWidgetClass 10-65
 xmPanedWindowWidgetClass 10-58
 xmPrimitiveClassRec 10-58
 XmPrimitiveWidget 10-67
 XmPrimitiveWidgetClass 10-67
 xmPrimitiveWidgetClass 10-58
 XmProcessTraversal 10-57
 xmProtocolClassRec 10-58
 xmProtocolObjectClass 10-58
 XmPushButtonCallbackStruct 10-68
 xmPushButtonClassRec 10-58
 XmPushButtonGadget 10-65
 XmPushButtonGadgetClass 10-65
 xmPushButtonGadgetClass 10-58
 xmPushButtonGadgetClassRec 10-58
 xmPushButtonGCacheObjClassRec 10-58
 XmPushButtonGCacheObject 10-65
 XmPushButtonWidget 10-65
 XmPushButtonWidgetClass 10-65
 xmPushButtonWidgetClass 10-58
 XmQmotif 10-58
 XmRegisterSegmentEncoding 10-57
 XmRemoveProtocolCallback 10-57
 XmRemoveProtocols 10-57
 XmRemoveTabGroup 10-57
 XmRepTypeAddReverse 10-57
 XmRepTypeEntry 10-65
 XmRepTypeEntryRec 10-65
 XmRepTypeGetId 10-57
 XmRepTypeGetNameList 10-57
 XmRepTypeGetRecord 10-57
 XmRepTypeGetRegistered 10-57
 XmRepTypeId 10-65
 XmRepTypeInstallTearOffModelConverter 10-57
 XmRepTypeList 10-65
 XmRepTypeListRec 10-65
 XmRepTypeRegister 10-57
 XmRepTypeValidValue 10-57
 XmResolveAllPartOffsets 10-57
 XmResolvePartOffsets 10-57
 XmRowColumnCallbackStruct 10-68
 xmRowColumnClassRec 10-58
 XmRowColumnWidget 10-66
 XmRowColumnWidgetClass 10-66
 xmRowColumnWidgetClass 10-58
 xmSashClassRec 10-58
 xmSashWidgetClass 10-58
 XmScaleCallbackStruct 10-68
 xmScaleClassRec 10-58
 XmScaleGetValue 10-57
 XmScaleSetValue 10-57
 XmScaleWidget 10-66
 XmScaleWidgetClass 10-66
 xmScaleWidgetClass 10-58
 XmScreen 10-66
 XmScreenClass 10-66
 xmScreenClass 10-58
 xmScreenClassRec 10-58
 xmScreenObjectClass 10-58
 XmScrollBarCallbackStruct 10-68
 xmScrollBarClassRec 10-58
 XmScrollBarGetValues 10-57
 XmScrollBarSetValues 10-57
 XmScrollBarWidget 10-66
 XmScrollBarWidgetClass 10-66
 xmScrollBarWidgetClass 10-58
 xmScrolledWindowClassRec 10-58
 XmScrolledWindowSetAreas 10-57
 XmScrolledWindowWidget 10-66
 XmScrolledWindowWidgetClass 10-66
 xmScrolledWindowWidgetClass 10-58
 XmScrollVisible 10-57
 XmSecondaryResourceData 10-70
 XmSecondaryResourceDataRec 10-70
 XmSelectionBoxCallbackStruct 10-68
 xmSelectionBoxClassRec 10-58
 XmSelectionBoxGetChild 10-57
 XmSelectionBoxWidget 10-66
 XmSelectionBoxWidgetClass 10-66
 xmSelectionBoxWidgetClass 10-58

xmSeparatorClassRec 10-58
XmSeparatorGadget 10-66
XmSeparatorGadgetClass 10-66
xmSeparatorGadgetClass 10-58
xmSeparatorGadgetClassRec 10-58
xmSeparatorGCacheObjClassRec 10-58
XmSeparatorGCacheObject 10-66
XmSeparatorWidget 10-66
XmSeparatorWidgetClass 10-66
xmSeparatorWidgetClass 10-58
XmSetColorCalculation 10-57
XmSetFontUnit 10-57
XmSetFontUnits 10-57
XmSetMenuCursor 10-57
XmSetProtocolHooks 10-57
xmShellExtClassRec 10-58
xmShellExtObjectClass 10-58
XmStrDefs.h 10-70
XmString 10-67
XmStringBaseline 10-57
XmStringByteCompare 10-57
XmStringCharSet 10-67
XmStringCharSetTable 10-70
XmStringCompare 10-57
XmStringComponentType 10-67
XmStringConcat 10-57
XmStringContext 10-67
XmStringCopy 10-57
XmStringCreate 10-57
XmStringCreateLocalized 10-57
XmStringCreateLtoR 10-57
XmStringCreateSimple 10-57
XmStringDirection 10-67
XmStringDirectionCreate 10-57
XmStringDraw 10-57
XmStringDrawImage 10-57
XmStringDrawUnderline 10-57
XmStringEmpty 10-57
XmStringExtent 10-57
XmStringFree 10-57
XmStringFreeContext 10-57
XmStringGetLtoR 10-57
XmStringGetNextComponent 10-57
XmStringGetNextSegment 10-57
XmStringHasSubstring 10-57
XmStringHeight 10-57
XmStringInitContext 10-57
XmStringLength 10-57
XmStringLineCount 10-57
XmStringNConcat 10-57
XmStringNCopy 10-57
XmStringPeekNextComponent 10-57
XmStringSegmentCreate 10-57
XmStringSeparatorCreate 10-57
XmStringTable 10-67
XmStringWidth 10-57
XmTargetsAreCompatible 10-57
xmTearOffButtonClassRec 10-58
xmTearOffButtonWidgetClass 10-58
XmTextBlock 10-69
XmTextBlockRec 10-69
XmTextBlockRecWcs 10-69
XmTextBlockWcs 10-69
xmTextClassRec 10-58
XmTextClearSelection 10-57
XmTextCopy 10-57
XmTextCut 10-57
XmTextDirection 10-69
XmTextDisableRedisplay 10-57
XmTextEnableRedisplay 10-57
xmTextFieldClassRec 10-58
XmTextFieldClearSelection 10-57
XmTextFieldCopy 10-57
XmTextFieldCut 10-57
XmTextFieldGetBaseline 10-57
XmTextFieldGetEditable 10-57
XmTextFieldGetInsertionPosition 10-57
XmTextFieldGetLastPosition 10-57
XmTextFieldGetMaxLength 10-57
XmTextFieldGetSelection 10-57
XmTextFieldGetSelectionPosition 10-57
XmTextFieldGetSelectionWcs 10-57
XmTextFieldGetString 10-57
XmTextFieldGetStringWcs 10-57
XmTextFieldGetSubstring 10-57
XmTextFieldGetSubstringWcs 10-57
XmTextFieldInsert 10-57
XmTextFieldInsertWcs 10-57
XmTextFieldPaste 10-57
XmTextFieldPosToXY 10-57
XmTextFieldRemove 10-57
XmTextFieldReplace 10-57
XmTextFieldReplaceWcs 10-57
XmTextFieldSetAddMode 10-57
XmTextFieldSetEditable 10-57
XmTextFieldSetHighlight 10-57
XmTextFieldSetInsertionPosition 10-57
XmTextFieldSetMaxLength 10-57
XmTextFieldSetSelection 10-57
XmTextFieldSetString 10-57

XmTextFieldSetStringWcs 10-57
XmTextFieldShowPosition 10-57
XmTextFieldWidget 10-66
XmTextFieldWidgetClass 10-66
xmTextFieldWidgetClass 10-58
XmTextFieldXYToPos 10-57
XmTextFindString 10-57
XmTextFindStringWcs 10-57
XmTextFormat 10-69
XmTextGetBaseLine
 obsolete, use XmTextGetBaseline 10-57
XmTextGetBaseline 10-57
XmTextGetEditable 10-57
XmTextGetInsertionPosition 10-57
XmTextGetLastPosition 10-57
XmTextGetMaxLength 10-57
XmTextGetSelection 10-57
XmTextGetSelectionPosition 10-57
XmTextGetSelectionWcs 10-57
XmTextGetSource 10-57
XmTextGetString 10-57
XmTextGetStringWcs 10-57
XmTextGetSubstring 10-57
XmTextGetSubstringWcs 10-57
XmTextGetTopCharacter 10-57
XmTextInsert 10-57
XmTextInsertWcs 10-57
XmTextPaste 10-57
XmTextPosition 10-69
XmTextPosToXY 10-57
XmTextRemove 10-57
XmTextReplace 10-57
XmTextReplaceWcs 10-57
XmTextScanType 10-69
XmTextScroll 10-57
XmTextSetAddMode 10-57
XmTextSetEditable 10-57
XmTextSetHighlight 10-57
XmTextSetInsertionPosition 10-57
XmTextSetMaxLength 10-57
XmTextSetSelection 10-57
XmTextSetSource 10-57
XmTextSetString 10-57
XmTextSetStringWcs 10-57
XmTextSetTopCharacter 10-57
XmTextShowPosition 10-57
XmTextSource 10-66
XmTextVerifyCallbackStruct 10-69
XmTextVerifyCallbackStructWcs 10-69
XmTextVerifyPtr 10-69
XmTextVerifyPtrWcs 10-69
XmTextWidget 10-66
XmTextWidgetClass 10-66
xmTextWidgetClass 10-58
XmTextXYToPos 10-57
XmToggleButtonCallbackStruct 10-68
xmToggleButtonClassRec 10-58
XmToggleButtonGadget 10-66
XmToggleButtonGadgetClass 10-66
xmToggleButtonGadgetClass 10-58
xmToggleButtonGadgetClassRec 10-58
XmToggleButtonGadgetGetState 10-57
XmToggleButtonGadgetSetState 10-57
xmToggleButtonGCacheObjClassRec 10-58
XmToggleButtonGCacheObject 10-66
XmToggleButtonGetState 10-57
XmToggleButtonSetState 10-57
XmToggleButtonWidget 10-66
XmToggleButtonWidgetClass 10-66
xmToggleButtonWidgetClass 10-58
XmTopLevelEnterCallback 10-60
XmTopLevelEnterCallbackStruct 10-60
XmTopLevelLeaveCallback 10-60
XmTopLevelLeaveCallbackStruct 10-60
XmTrackingEvent 10-57
XmTrackingLocate 10-57
XmTranslateKey 10-57
XmTraversalDirection 10-70
XmTraverseObscuredCallbackStruct 10-70
XmUninstallImage 10-57
XmUpdateDisplay 10-57
XmVaCreateSimpleCheckBox 10-57
XmVaCreateSimpleMenuBar 10-57
XmVaCreateSimpleOptionMenu 10-57
XmVaCreateSimplePopupMenu 10-57
XmVaCreateSimplePulldownMenu 10-57
XmVaCreateSimpleRadioBox 10-57
xmVendorShellExtClassRec 10-58
xmVendorShellExtObjectClass 10-58
XmVendorShellWidget 10-67
XmVendorShellWidgetClass 10-67
XmVisibility 10-70
XmWidgetGetBaselines 10-57
XmWidgetGetDisplayRect 10-57
xmWorldClass 10-58
xmWorldClassRec 10-58
xmWorldObjectClass 10-58
XNewModifiermap 10-5
XNextEvent 10-5
XNextRequest 10-5, 10-8

XNoOp 10-5
XOffsetRegion 10-5
XOpenDisplay 10-5
XOpenIM 10-5
XParseColor 10-5
XParseGeometry 10-5
XPeekevent 10-5
XPeekevent 10-5
XPending 10-5
Xpermalloc 10-5
XPG3 1-7
XPG4 6-80
XPG5 6-6
XPlanesOfScreen 10-5, 10-8
XPointer 10-10, 10-11
XPointInRegion 10-5
XPolygonRegion 10-5
XProtocolRevision 10-5, 10-8
XProtocolVersion 10-5, 10-8
xprt_register 6-56
xprt_unregister 6-56
XPutBackEvent 10-5
XPutImage 10-5
XPutPixel 10-5
XQLength 10-5, 10-8
XQueryBestCursor 10-5
XQueryBestSize 10-5
XQueryBestStipple 10-5
XQueryBestTile 10-5
XQueryColor 10-5
XQueryColors 10-5
XQueryExtension 10-5
XQueryFont 10-5
XQueryKeymap 10-5
XQueryPointer 10-5
XQueryTextExtents 10-5
XQueryTextExtents16 10-5
XQueryTree 10-5
XRaiseWindow 10-5
XReadBitmapFile 10-5
XRebindKeysym 10-5
XRecolorCursor 10-5
XReconfigureWMWindow 10-5
XRectInRegion 10-5
XRefreshKeyboardMapping 10-5
XRemoveFromSaveSet 10-5
XRemoveHost 10-5
XRemoveHosts 10-5
XReparentWindow 10-5
XResetScreenSaver 10-5
XResizeWindow 10-5
XResourceManagerString 10-5
XRestackWindows 10-5
XrmCombineDatabase 10-5
XrmCombineFileDatabase 10-5
XrmDatabase 10-11
XrmDestroyDatabase 10-5
XrmEnumerateDatabase 10-5
XrmGetDatabase 10-5
XrmGetFileDatabase 10-5
XrmGetResource 10-5
XrmGetStringDatabase 10-5
XrmHashBucket 10-11
XrmInitialize 10-5
XrmLocaleOfDatabase 10-5
XrmMergeDatabases 10-5
XrmParseCommand 10-5
XrmPermStringToQuark 10-6
XrmPutFileDatabase 10-5
XrmPutLineResource 10-5
XrmPutResource 10-5
XrmPutStringResource 10-5
XrmQGetResource 10-5
XrmQGetSearchList 10-5
XrmQGetSearchResource 10-5
XrmQPutResource 10-5
XrmQPutStringResource 10-5
XrmQuarkToString 10-5
XrmResource 10-42
XrmResourceList 10-42
XrmSetDatabase 10-5
XrmString 10-11
XrmString 10-11
XrmStringToBindingQuarkList 10-5
XrmStringToQuark 10-5
XrmStringToQuarkList 10-5
XrmUniqueQuark 10-5
XRootWindow 10-5, 10-8
XRootWindowOfScreen 10-5, 10-8
XRotateBuffers 10-5
XRotateWindowProperties 10-5
xrs_t 6P-14
XSaveContext 10-5
XScreenCount 10-5, 10-8
XScreenNumberOfCCC 10-8
XScreenNumberOfScreen 10-5
XScreenOfDisplay 10-5, 10-8
XScreenResourceString 10-5
XScreenWhiteOfCCC 10-8
XSelectInput 10-5

XSendEvent 10-5
XServerVendor 10-5, 10-8
XSetAccessControl 10-5
XSetAfterFunction 10-5
XSetArcMode 10-5
XSetBackground 10-5
XSetClassHint 10-5
XSetClipMask 10-6
XSetClipOrigin 10-5
XSetClipRectangles 10-5
XSetCloseDownMode 10-5
XSetCommand 10-5
XSetDashes 10-5
XSetErrorHandler 10-5
XSetFillRule 10-5
XSetFillStyle 10-5
XSetFont 10-5
XSetFontPath 10-5
XSetForeground 10-5
XSetFunction 10-5
XSetGraphicsExposures 10-6
XSetICFocus 10-6
XSetIconName 10-6
XSetIconSizes 10-6
XSetICValues 10-6
XSetInputFocus 10-6
XSetIOErrorHandler 10-6
XSetLineAttributes 10-6
XSetLocaleModifiers 10-6
XSetModifierMapping 10-6
XSetNormalHints 10-6
XSetPlaneMask 10-6
XSetPointerMapping 10-6
XSetRegion 10-6
XSetRGBColormaps 10-6
XSetScreenSaver 10-6
XSetSelectionOwner 10-6
XSetSizeHints 10-6
XSetStandardColormap 10-6
XSetStandardProperties 10-6
XSetState 10-6
XSetStipple 10-6
XSetSubwindowMode 10-6
XSetTextProperty 10-6
XSetTile 10-6
XSetTransientForHint 10-6
XSetTSTOrigin 10-6
XSetWindowBackground 10-6
XSetWindowBackgroundPixmap 10-6
XSetWindowBorder 10-6
XSetWindowBorderPixmap 10-6
XSetWindowBorderWidth 10-6
XSetWindowColormap 10-6
XSetWMClientMachine 10-6
XSetWMC colormapWindows 10-6
XSetWMHints 10-6
XSetWMIconName 10-6
XSetWMName 10-6
XSetWMNormalHints 10-6
XSetWMProperties 10-6
XSetWMPprotocols 10-6
XSetWMSizeHints 10-6
XShapeCombineMask 10-26
XShapeCombineRectangles 10-26
XShapeCombineRegion 10-26
XShapeCombineShape 10-26
XShapeGetRectangles 10-26
XShapeInputSelected 10-26
XShapeOffsetShape 10-26
XShapeQueryExtension 10-26
XShapeQueryExtents 10-26
XShapeQueryVersion 10-26
XShapeSelectInput 10-26
XShrinkRegion 10-6
XSI 1-7
XStoreBuffer 10-6
XStoreBytes 10-6
XStoreColor 10-6
XStoreColors 10-6
XStoreName 10-6
XStoreNamedColor 10-6
XStringListToTextProperty 10-6
XStringToKeysym 10-6
XSubImage 10-6
XSubtractRegion 10-6
XSupportsLocale 10-6
XSync 10-6
XSynchronize 10-6
XtAccelerators 10-33
XtAcceptFocusProc 10-42
XtActionHookId 10-33
XtActionHookProc 10-33
XtActionList 10-33
XtActionProc 10-33
XtActionsRec 10-33
XtAddActions
 obsolete, use XtAppAddActions 10-30
XtAddCallback 10-28, 10-29
XtAddCallbacks 10-28
XtAddConverter

- obsolete, use XtSetTypeConverter 10-30
- XtAddEventHandler 10-28
- XtAddExposureToRegion 10-28
- XtAddGrab 10-28
- XtAddInput
 - obsolete, use XtAppAddInput 10-30
- XtAddRawEventHandler 10-28
- XtAddressMode 10-33
- XtAddTimeOut
 - obsolete, use XtAppAddTimeOut 10-30
- XtAddWorkProc
 - obsolete, use XtAppAddWorkProc 10-30
- XtAllocateGC 10-28
- XtAlmostProc 10-42
- XtAppAddActionHook 10-28
- XtAppAddActions 10-28
- XtAppAddConverter
 - obsolete, use XtAppSetTypeConverter 10-30
- XtAppAddInput 10-28
- XtAppAddTimeOut 10-28
- XtAppAddWorkProc 10-28
- XtAppContext 10-33
- XtAppCreateShell 10-28
- XtAppError 10-28
- XtAppErrorMsg 10-28
- XtAppGetErrorDatabase 10-28
- XtAppGetErrorDatabaseText 10-28
- XtAppGetSelectionTimeout 10-28
- XtAppInitialize 10-28
- XtAppMainLoop 10-28
- XtAppNextEvent 10-28
- XtAppPeekEvent 10-28
- XtAppPending 10-28
- XtAppProcessEvent 10-28
- XtAppReleaseCacheRefs 10-28
- XtAppSetErrorHandler 10-28
- XtAppSetErrorHandlerMsg 10-28
- XtAppSetFallbackResources 10-28
- XtAppSetSelectionTimeout 10-28
- XtAppSetTypeConverter 10-28
- XtAppSetWarningHandler 10-28
- XtAppSetWarningMsgHandler 10-28
- XtAppWarning 10-28
- XtAppWarningMsg 10-28
- XtArgsFunc 10-42
- XtArgsProc 10-42
- XtArgVal 10-33
- XtAugmentTranslations 10-28
- XtBoundAccActions 10-33
- XtBoundActions 10-33
- XtBuildEventMask 10-28
- XtCacheRef 10-33
- XtCacheType 10-33
- XtCallAcceptFocus 10-28
- XtCallActionProc 10-28
- XtCallbackExclusive 10-28
- XtCallbackList 10-34
- XtCallbackNone 10-28
- XtCallbackNonexclusive 10-28
- XtCallbackPopdown 10-28
- XtCallbackProc 10-34
- XtCallbackRec 10-34
- XtCallbackReleaseCacheRef 10-28
- XtCallbackReleaseCacheRefList 10-28
- XtCallbackStatus 10-34
- XtCallCallbackList 10-28
- XtCallCallbacks 10-28
- XtCallConverter 10-28
- XtCalloc 10-28
- XtCancelConvertSelectionProc 10-34
- XtCaseProc 10-33
- XtClass 10-28
- XtCloseDisplay 10-28
- XtConfigureWidget 10-28, 10-42
- XtConvert
 - obsolete, use XtConvertAndStore 10-30
- XtConvertAndStore 10-28
- XtConvertArgList 10-33
- XtConvertArgProc 10-33
- XtConvertArgRec 10-33
- XtConvertCase 10-28
- XtConverter 10-33
- XtConvertSelectionIncrProc 10-34
- XtConvertSelectionProc 10-34
- XtCreateApplicationContext 10-28
- XtCreateApplicationShell
 - obsolete, use XtAppCreateShell 10-30
- XtCreateManagedWidget 10-28
- XtCreatePopupChildProc 10-34
- XtCreatePopupShell 10-28
- XtCreateWidget 10-28
- XtCreateWindow 10-28, 10-42
- XtCvtColorToPixel 10-28
- XtCvtIntToBool 10-28
- XtCvtIntToBoolean 10-28
- XtCvtIntToColor 10-28
- XtCvtIntToFloat 10-28
- XtCvtIntToFont 10-28
- XtCvtIntToPixel 10-28
- XtCvtIntToPixmap 10-28

-
- XtCvtIntToShort 10-28
 - XtCvtIntToUnsignedChar 10-28
 - XtCvtStringToAcceleratorTable 10-28
 - XtCvtStringToAtom 10-28
 - XtCvtStringToBool 10-28
 - XtCvtStringToBoolean 10-28
 - XtCvtStringToCursor 10-28
 - XtCvtStringToDimension 10-28
 - XtCvtStringToDisplay 10-28
 - XtCvtStringToFile 10-28
 - XtCvtStringToFloat 10-28
 - XtCvtStringToFont 10-28
 - XtCvtStringToFontSet 10-28
 - XtCvtStringToFontStruct 10-28
 - XtCvtStringToInitialState 10-28
 - XtCvtStringToInt 10-28
 - XtCvtStringToPixel 10-28
 - XtCvtStringToShort 10-28
 - XtCvtStringToTranslationTable 10-28
 - XtCvtStringToUnsignedChar 10-28
 - XtCvtStringToVisual 10-28
 - XtCXtToolkitError 10-32
 - XtDatabase 10-28
 - XtDestroyApplicationContext 10-28
 - XtDestroyGC
 - obsolete, use XtReleaseGC 10-30
 - XtDestroyWidget 10-28
 - XtDestructor 10-33
 - XtDirectConvert
 - obsolete, use XtCallConverter 10-30
 - XtDisownSelection 10-28
 - XtDispatchEvent 10-28
 - XtDisplay 10-28
 - XtDisplayInitialize 10-28
 - XtDisplayOfObject 10-28
 - XtDisplayStringConversionWarning 10-28
 - XtDisplayToApplicationContext 10-28
 - XtEnum 10-33
 - XtError
 - obsolete, use XtAppErrorMsg or XtAppError 10-30
 - XtErrorHandler 10-34
 - XtErrorMsg
 - obsolete, use XtAppErrorMsg 10-30
 - XtErrorMsgHandler 10-34
 - XtEventHandler 10-33
 - XtEventTable 10-33
 - XtExposeProc 10-42
 - XTextExtents 10-6
 - XTextExtents16 10-6
 - XTextPropertyToStringList 10-6
 - XTextWidth 10-6
 - XTextWidth16 10-6
 - XtFilePredicate 10-34
 - XtFindFile 10-28
 - XtFree 10-28
 - XtGCMask 10-33
 - XtGeometryHandler 10-42
 - XtGeometryMask 10-33
 - XtGeometryResult 10-34
 - XtGetActionKeysym 10-28
 - XtGetActionList 10-28
 - XtGetApplicationNameAndClass 10-28
 - XtGetApplicationResources 10-28
 - XtGetConstraintResourceList 10-28
 - XtGetErrorDatabase
 - obsolete, use XtAppGetErrorDatabase 10-30
 - XtGetErrorDatabaseText
 - obsolete, use XtAppGetErrorDatabaseText 10-30
 - XtGetGC 10-28
 - XtGetKeysymTable 10-28
 - XtGetMultiClickTime 10-28
 - XtGetResourceList 10-28
 - XtGetSelectionRequest 10-28
 - XtGetSelectionTimeout
 - obsolete, use XtAppGetSelectionTimeout 10-30
 - XtGetSelectionValue 10-28
 - XtGetSelectionValueIncremental 10-28
 - XtGetSelectionValues 10-28
 - XtGetSelectionValuesIncremental 10-28
 - XtGetSubresources 10-28
 - XtGetSubvalues 10-28
 - XtGetValues 10-28
 - XtGrabButton 10-28
 - XtGrabKey 10-28
 - XtGrabKeyboard 10-28
 - XtGrabKind 10-34
 - XtGrabPointer 10-28
 - XtHasCallbacks 10-28
 - XtInitialize
 - obsolete, use XtAppInitialize 10-30
 - XtInitializeWidgetClass 10-28
 - XtInitProc 10-42
 - XtInputCallbackProc 10-34
 - XtInputId 10-33
 - XtInputMask 10-34
 - XtInsertEventHandler 10-28
 - XtInsertRawEventHandler 10-28
 - XtInstallAccelerators 10-28
 - XtInstallAllAccelerators 10-28
 - XtIntervalId 10-33
-

XtIsApplicationShell 10-28
 XtIsComposite 10-28
 XtIsConstraint 10-28
 XtIsManaged 10-28
 XtIsObject 10-28
 XtIsOverrideShell 10-28
 XtIsRealized 10-28
 XtIsRectObj 10-28
 XtIsSensitive 10-28
 XtIsShell 10-28
 XtIsSubclass 10-28
 XtIsTopLevelShell 10-28
 XtIsTransientShell 10-28
 XtIsVendorShell 10-28
 XtIsWidget 10-28
 XtIsWMShell 10-28
 XtKeyProc 10-33
 XtKeysymToKeycodeList 10-28
 XtLanguageProc 10-34
 XtLastTimestampProcessed 10-28
 XtListPosition 10-34
 XtLoseSelectionIncrProc 10-34
 XtLoseSelectionProc 10-34
 XtMainLoop
 obsolete, use XtAppMainLoop 10-30
 XtMakeGeometryRequest 10-28
 XtMakeResizeRequest 10-28
 XtMalloc 10-28
 XtManageChild 10-28
 XtManageChildren 10-28
 XtMapWidget 10-28
 XtMenuPopupAction 10-28
 XtMergeArgLists 10-28
 XtMoveWidget 10-28, 10-42
 XtName 10-28
 XtNameToWidget 10-28
 XtNewString 10-28
 XtNextEvent
 obsolete, use XtAppNextEvent 10-30
 XtOpenDisplay 10-28
 XtOrderProc 10-33
 XtOverrideTranslations 10-28
 XtOwnSelection 10-28
 XtOwnSelectionIncremental 10-28
 XtParent 10-28
 XtParseAcceleratorTable 10-28
 XtParseTranslationTable 10-28
 XtPeekEvent
 obsolete, use XtAppPeekEvent 10-30
 XtPending
 obsolete, use XtAppPending 10-30
 XtPointer 10-33
 XtPopdown 10-28
 XtPopdownID 10-34
 XtPopdownIDRec 10-34
 XtPopup 10-28
 XtPopupSpringLoaded 10-28
 XtProc 10-42
 XtProcessEvent
 obsolete, use XtAppProcessEvent 10-30
 XtQueryGeometry 10-28
 XtTranslateCoordinates 10-6
 XtRealizeProc 10-42
 XtRealizeWidget 10-28
 XtRealloc 10-28
 XtRegisterCaseConverter 10-28
 XtRegisterGrabAction 10-28
 XtReleaseGC 10-28
 XtRemoveActionHook 10-28
 XtRemoveAllCallbacks 10-28
 XtRemoveCallback 10-28
 XtRemoveCallbacks 10-28
 XtRemoveEventHandler 10-28
 XtRemoveGrab 10-28
 XtRemoveInput 10-28
 XtRemoveRawEventHandler 10-28
 XtRemoveTimeOut 10-28
 XtRemoveWorkProc 10-28
 XtRequestId 10-34
 XtResizeWidget 10-28, 10-42
 XtResizeWindow 10-28, 10-42
 XtResolvePathname 10-28
 XtResource 10-34
 XtResourceDefaultProc 10-34
 XtResourceList 10-34
 XtScreen 10-28
 XtScreenDatabase 10-28
 XtScreenOfObject 10-28
 XtSelectionCallbackProc 10-34
 XtSelectionDoneIncrProc 10-34
 XtSelectionDoneProc 10-34
 XtSetErrorHandler
 obsolete, use XtAppSetErrorHandler 10-30
 XtSetErrorMsgHandler
 obsolete, use XtAppSetErrorMsgHandler 10-30
 XtSetKeyboardFocus 10-28
 XtSetKeyTranslator 10-28
 XtSetLanguageProc 10-28
 XtSetMappedWhenManaged 10-28
 XtSetMultiClickTime 10-28

-
- XtSetSelectionTimeout
 - obsolete, use XtAppSetSelectionTimeout 10-30
 - XtSetSensitive 10-28
 - XtSetSubvalues 10-28
 - XtSetTypeConverter 10-28
 - XtSetValues 10-28
 - XtSetValuesFunc 10-42
 - XtSetWarningHandler
 - obsolete, use XtAppSetWarningHandler 10-30
 - XtSetWarningMsgHandler
 - obsolete, use XtAppSetWarningMsgHandler 10-30
 - XtSetWMColormapWindows 10-28
 - XtShellStrings
 - Not safe to use 10-32
 - XtStringConversionWarning
 - obsolete, use XtDisplayStringConversionWarning 10-30
 - XtStringProc 10-42
 - XtStrings
 - Not safe to use 10-32
 - XtSuperclass 10-28
 - XtTimerCallbackProc 10-34
 - XtTM 10-42
 - XtTMRec 10-42
 - XtToolkitInitialize 10-28
 - XtTranslateCoords 10-28
 - XtTranslateKey 10-29
 - XtTranslateKeycode 10-29
 - XtTranslations 10-33
 - XtTypeConverter 10-33
 - XtUngrabButton 10-29
 - XtUngrabKey 10-29
 - XtUngrabKeyboard 10-29
 - XtUngrabPointer 10-29
 - XtUninstallTranslations 10-29
 - XtUnmanageChild 10-29
 - XtUnmanageChildren 10-29
 - XtUnmapWidget 10-29
 - XtUnrealizeWidget 10-29
 - XtVaAppCreateShell 10-29
 - XtVaAppInitialize 10-29
 - XtVaCreateArgsList 10-29
 - XtVaCreateManagedWidget 10-29
 - XtVaCreatePopupShell 10-29
 - XtVaCreateWidget 10-29
 - XtVaGetApplicationResources 10-29
 - XtVaGetSubresources 10-29
 - XtVaGetSubvalues 10-29
 - XtVaGetValues 10-29
 - XtValueMask 10-33
 - XtVarArgsList 10-34
 - XtVaSetSubvalues 10-29
 - XtVaSetValues 10-29
 - XtVersionType 10-42
 - XtWarning
 - obsolete, use XtAppWarningMsg or XtAppWarning 10-30
 - XtWarningMsg
 - obsolete, use XtAppWarningMsg 10-30
 - XtWidgetClassProc 10-42
 - XtWidgetGeometry 10-33
 - XtWidgetProc 10-42
 - XtWidgetToApplicationContext 10-29
 - XtWindow 10-29
 - XtWindowOfObject 10-29
 - XtWindowToWidget 10-29
 - XtWorkProc 10-34
 - XtWorkProcId 10-33
 - XUndefineCursor 10-6
 - XUnggrabButton 10-6
 - XUnggrabKey 10-6
 - XUnggrabKeyboard 10-6
 - XUnggrabPointer 10-6
 - XUnggrabServer 10-6
 - XUinstallColormap 10-6
 - XUnionRectWithRegion 10-6
 - XUnionRegion 10-6
 - XUnloadFont 10-6
 - XUnmapSubwindows 10-6
 - XUnmapWindow 10-6
 - XUnsetICFocus 10-6
 - XVaCreateNestedList 10-6
 - XVendorRelease 10-6, 10-8
 - XVisualIDFromVisual 10-6
 - XVisualOfCCC 10-8
 - XWarpPointer 10-6
 - XwcDrawImageString 10-6
 - XwcDrawString 10-6
 - XwcDrawText 10-6
 - XwcFreeStringList 10-6
 - XwcLookupString 10-6
 - XwcResetIC 10-6
 - XwcTextEscapement 10-6
 - XwcTextExtents 10-6
 - XwcTextListToTextProperty 10-6
 - XwcTextPerCharExtents 10-6
 - XwcTextPropertyToTextList 10-6
 - XWhitePixel 10-6, 10-8
 - XWhitePixelOfScreen 10-6, 10-8
 - XWidthMMOfScreen 10-6, 10-8
 - XWidthOfScreen 10-6, 10-8
 - XWindowEvent 10-6
-

XWithdrawWindow 10-6
XWMGeometry 10-6
xword 4P-9
xword64 4P-6
XWriteBitmapFile 10-6
XXorRegion 10-6

Y

Y 3P-24
y0 6-51
y1 6-51
yacc 11-1
yn 6-51
yp_all 6-56
yp_bind 6-56
yp_first 6-56
yp_get_default_domain 6-56
yp_master 6-56
yp_match 6-56
yp_next 6-56
yp_order 6-56, 6-57, 6-58
yp_unbind 6-56
yperr_string 6-56
ypprot_err 6-56

Z

zattr_ndx 6-70
zattr_val 6-70
zattr_val_len 6-70
zattr_val_val 6-70
ZONEINIT 6-64
ZONEREF 6-64
zotypes 6-70

