

SPARC COMPLIANCE DEFINITION 2.1a

**SCD
2.1a**

SPARC International

FINAL RELEASE

2/26/99

© 1990-1999 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.

Introduction

This document is an addendum to SCD 2.1. The functions in this document will be added to the *SPARC Compliance Definition, Version 2.2*. However, application developers need early access to these functions and all SCD 2.1 conformant systems already provide compatible versions of these functions.

Included in this document are manual pages for some of the new functions. These manual pages are needed because

1. either the SVID lacks a manual page for the new function, or
2. the new function differs from the specification found in the SVID. (In such cases the SCD limits itself to a compatible subset of the SVID definition)

For the complete SCD 2 specification or information regarding application conformance contact SPARC International at (415)321-8692.

List of New Interfaces

Below is a list of new interfaces being added to the *SPARC Compliance Definition 2.1*. All of the interfaces described by this document are REQUIRED interfaces.

System Library Additions

The function `sbrk` has been added to the library `/usr/lib/ld.so.1`. Also, the SVID is missing a manual page for the function `ftok`, so the `ftok` manual page is included in this document. The function `sbrk` is deprecated immediately. Application programmers using `sbrk` should begin migrating to other SCD functions immediately.

C Library Additions

The functions `crypt`, `encrypt`, `getitimer`, `gettimeofday`, `modf`, `setitimer`, `setkey`, and `sysinfo` have been added to `/usr/lib/libc.so.1`.

Network Services Library Additions

The function `rpc_broadcast_exp` has been added to the library `/usr/lib/libnsl.so.1`.

Socket Library Added

A new library, `/usr/lib/libsocket.so.1`, and several new functions have been added to `/usr/lib/libnsl.so.1` to support applications which use sockets. All of these new interfaces are deprecated immediately. Application programmers using these interfaces should begin migrating to other SCD functions immediately.

Dynamic Linking Library Added

A new library, `/usr/lib/libdl.so.1`, is added to support those applications which need to access the systems dynamic linking capabilities at run-time.

X 11, Release 5 Support Added

Two new libraries, `/usr/lib/libX11.so.5` and `/usr/lib/libXt.so.5`, have been added to support applications which need X11, Release 5 support.

Libraries

Overview

This section defines the additions to the System Library, the C Library, the Network Services Library, the Socket Library, and the Dynamic Object File Loading Library. The System Library, the C Library, the Network Services Library, the Socket Library, and the Dynamic Object File Loading Library are REQUIRED interface sets.

The entries in the tables which define the function interfaces provided by various libraries have a superscript and some are also marked with a “strike-through” line. All entries with a superscript have an entry in the changes table describing differences in the SCD definition of the function and its base document (gABI, psABI, or SVID) definition. Entries marked with a “strike-through” line (~~strike through~~) are called out in the base documents but are excluded in the SCD.

The System Library

The libsys Interfaces

Every function defined in this section is a REQUIRED interface member.

Effective November 1st, 1993, the `sbrk` function interface is DEPRECATED. Reasons for this can be found in the `sbrk` description on page 6-9. The interface may be removed from the SCD as early as November 1st, 1996.

Table 1. `libsys` Contents

`ftok`¹
`sbrk`²

System Library Changes

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

#	Facility	Location	Description
1	ftok	SVID, Vol.I	Add description of function <code>ftok</code> . See the following man page for this function.
2	sbrk	SVID, Vol.I	Add description of the function <code>sbrk</code> . See the following man page for this function.
2	sbrk	gABI	Add the function <code>sbrk</code> to Table 6-3 on page 6-5.

ABI Extensions

The SCD includes several extra functions. These functions are not described in any of the base documents. A man page for each the functions follows.

NAME

`ftok` - standard interprocess communication package

SYNOPSIS

```
#include <sys/types.h>
#include <sys/ipc.h>

key_t ftok(char *path, int id);
```

DESCRIPTION

All interprocess communication facilities require the user to supply a key to be used by the `msgget`, `semget`, `shmget` system calls to obtain interprocess communication identifiers. One suggested method for forming a key is to use the `ftok` subroutine described below. Another way to compose keys is to include the project ID in the most significant byte and to use the remaining portion as a sequence number. There are many other ways to form keys, but it is necessary for each system to define standards for forming them. If some standard is not adhered to, it will be possible for unrelated processes to unintentionally interfere with each other's operation. It is still possible to interface intentionally. Therefore, it is strongly suggested that the most significant byte of a key in some sense refer to a project so that keys do not conflict across a given system.

`ftok` returns a key based on *path* and *id* that is usable in subsequent `msgget`, `semget`, and `shmget` system calls. *path* must be the path name of an existing file that is accessible to the process. *id* is a character that uniquely identifies a project. Note that `ftok` will return the same key for linked files when called with the same *id* and that it will return different keys when called with the same file name but different *ids*.

DIAGNOSTICS

`ftok` returns (`key_t`) -1 if *path* does not exist or if it is not accessible to the process.

NOTES

If the file whose *path* is passed to `ftok` is removed when keys still refer to the file, future calls to `ftok` with the same *path* and *id* will return an error. If the same file is recreated, then `ftok` is likely to return a different key than it did the original time it was called.

NAME

`sbrk` - query the current break value

SYNOPSIS

```
#include <unistd.h>

void *sbrk (const int 0);
```

DESCRIPTION

The function `sbrk` is used to query the amount of space allocated for the calling process's data segment [see `exec(BA_OS)`].

STATUS

This function is DEPRECATED effective November 1st, 1993. It may be removed from the SCD as early as November 1st, 1996.

DIAGNOSTICS

Upon successful completion, `sbrk` returns the current break value. Otherwise, a value of -1 is returned and `errno` is set to indicate the error. If `sbrk` is called with a non-zero value, the application is not portable.

Rationale

Calling `sbrk(0)` yields a value that, at one time, had a predictable, well defined interpretation. It has not had this property for many years, since the wide-spread usage of sparse, demand-paged address spaces. Its use is deprecated because the interpretation of the value returned is so highly variable as to be non-portable. It is more appropriately regarded as a function yielding a value relevant to one of many attributes of memory occupancy.

`sbrk(non-zero)` is not specified in any relevant standard, as its interactions with and dependencies upon other memory allocation mechanisms (e.g., `malloc`) are undefined. The use of `sbrk(non-zero)` is non-conforming since the implementation of system supplied functions may freely use such memory allocation mechanisms.

Applications desiring memory allocation functionality should use `malloc` for this purpose. Alternatively, applications may construct their own memory allocation arenas by building upon `mmap` and mappings to `/dev/zero`.

The C Library

The libc Interfaces

All of the functions defined in this section are REQUIRED interface members and must be supplied by `/usr/lib/libc.so.1`.

Table 2. libc Contents

crypt¹
encrypt²
getitimer³
gettimeofday⁴
modf⁵
setitimer⁶
setkey⁷
~~setlabel~~⁸
sysinfo⁹

C Library Changes

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

#	Facility	Location	Description
1	crypt(BA_LIB)	gABI	Add the function <code>crypt</code> to Figure 6-7 on page 6-11.
2	crypt(BA_LIB)	gABI	Add the function <code>encrypt</code> to Figure 6-7 on page 6-11.
3	getitimer(RT_OS)	gABI	Add the function <code>getitimer</code> to Figure 6-7 on page 6-11. 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.
4	gettimeofday(RT_OS)	gABI	Add the function <code>gettimeofday</code> to Figure 6-7 on page 6-11.
5	frexp(BA_LIB)	gABI	Add the function <code>modf</code> to Figure 6-6 on page 6-10.
6	getitimer(RT_OS)	gABI	Add the function <code>setitimer</code> to Figure 6-7 on page 6-11.
7	crypt(BA_LIB)	gABI	Add the function <code>setkey</code> to Figure 6-7 on page 6-11.
8	C Library	gABI	Remove the function <code>setLabel</code> from Figure 6-7 on page 6-11.
9	sysinfo	gABI	Add the function <code>sysinfo</code> to Figure 6-7 on page 6-11.

ABI Extensions

The SCD requires `/usr/lib/libc.so.1` to have functions which are not specified by the gABI. These extra functions are either not defined in the SVID, or, are defined differently in the SCD than in the SVID.

These functions are `crypt`, `setkey`, `encrypt`, and `sysinfo`. Manual pages for these additional/modified function definitions follow.

Rationale

The following man pages are for interfaces which do not have specifications in any of the base documents.

The library version number has remained 1, as these functions are correctly included in existing SCD conformant systems.

NAME

`crypt`, `setkey`, `encrypt` - generate string encoding

SYNOPSIS

```
char *crypt(char *key, char *salt);
void setkey(char *key);
void encrypt(char *block, int edflag);
```

DESCRIPTION

The function `crypt` is a string-encoding function.

The argument *key* is a string to be encoded. The argument *salt* is a two-character string chosen from the set [a-zA-Z0-9./]; this string is used to perturb the encoding algorithm, after which the string that *key* points to is used as the key to repeatedly encode a constant string. The returned value points to the encoded string. The first two characters are the *salt* itself, the remaining characters shall not be identical to the original value of *key*.

The functions `setkey` and `encrypt` provide (rather primitive) access to the encoding algorithm. The argument to `setkey` is a 64-bit string represented by a character array of length 64 containing only the characters with numerical value 0 and 1. The string is divided into groups of 8 and the low-order bit in each group is ignored; this gives a 56-bit key. This is the key that may be used with the above mentioned algorithm to encode the string *block* with the function `encrypt`; the encryption algorithm provided by the system may not actually use *key*.

The argument *block* to `encrypt` is a character array of length 64 containing only the characters with numerical value 0 and 1. The argument array is modified in place to a similar array representing the bits of the argument after having been subjected to the encoding algorithm using the key set by `setkey`.

If the argument *edflag* is zero, the string *block* is encoded. If the *edflag* is non-zero and the implementation supports decryption then the string *block* is decoded. If the *edflag* is non-zero and the implementation does not support decryption then `errno` is set to `ENOSYS`.

DIAGNOSTICS

Under the following conditions, these functions fail, and set `errno` to:

<code>ENOSYS</code>	<code>encrypt</code> was called with a non-zero value for <i>edflag</i> on a system that does not support decryption.
---------------------	---

USAGE

The return value of the function `crypt` points to static data that are overwritten by each call. A portable application shall not depend on portability of encrypted data, nor assume that decryption is supported on all SCD conforming platforms. Also, portable applications must set `errno` to zero before calling any of the functions since there are no function return values for `setkey` or `encrypt`.

Rationale

Encryption capability is often needed by an application that wants to provide some of its own license protection. The application needs to be able to depend on the system to provide an encryption service to do this even if the system does not provide a mechanism for decryption.

This standard does not require any particular underlying encryption algorithm, but only requires that the `crypt` function return a value that is not identical to the original. This leaves it to the system vendors to choose whatever algorithm they find to be appropriate, and alleviates any requirement for a system vendor to choose one that has export restrictions.

NAME

`sysinfo` - get system information strings

SYNOPSIS

```
#include <sys/systeminfo.h>

long sysinfo (int command, char *buf, long count);
```

DESCRIPTION

`sysinfo` copies information relating to the UNIX system on which the process is executing into the buffer pointed to by *buf*. *count* is the size of the buffer.

The *commands* available are:

<code>SI_SYSNAME</code>	Copy into the array pointed to by <i>buf</i> the string that would be returned by <code>uname</code> [see <code>uname(BA_OS)</code>] in the <i>sysname</i> field. This is the name of the implementation of the operating system, for example, <code>UNIX_SV</code> .
<code>SI_HOSTNAME</code>	<p>Copy into the array pointed to by <i>buf</i> a string that names the present host machine. This is the string that would be returned by <code>uname</code> in the <i>nodename</i> field. This hostname or nodename is often the name the machine is known by locally.</p> <p>The <i>hostname</i> is the name of this machine as a node in some network; different networks may have different names for the node, but presenting the nodename to the appropriate network Directory or name-to-address mapping service should produce a transport end point address. The name may not be fully qualified.</p> <p>Internet host names may be up to 256 bytes in length (plus the terminating null).</p>
<code>SI_RELEASE</code>	Copy into the array pointed to by <i>buf</i> the string that would be returned by <code>uname</code> in the <i>release</i> field. Typical values might be <code>4.2</code> , <code>4.0</code> , <code>3.2</code> .
<code>SI_VERSION</code>	Copy into the array pointed to by <i>buf</i> the string that would be returned by <code>uname</code> in the <i>version</i> field. The syntax and semantics of this string are defined by the system provider.
<code>SI_MACHINE</code>	Copy into the array pointed to by <i>buf</i> the string that would be returned by <code>uname</code> in the <i>machine</i> field.
<code>SI_ARCHITECTURE</code>	Copy into the array pointed to by <i>buf</i> a string describing the instruction set architecture of the current system, for example, <code>sparc</code> . These names may not match predefined names in the C language compilation system.
<code>SI_HW_PROVIDER</code>	Copies the name of the hardware manufacturer into the array pointed to by <i>buf</i> .
<code>SI_HW_SERIAL</code>	Copy into the array pointed to by <i>buf</i> a string which is the ASCII representation of the hardware-specific serial number of the physical machine on which the system call is executed. Note that this may be implemented in Read-Only Memory, via software constants set when building the operating system, or by other means, and may contain non-numeric characters. It is anticipated that manufacturers will not issue the same “serial number” to more than one physical machine. The pair of strings returned by <code>SI_HW_PROVIDER</code> and <code>SI_HW_SERIAL</code> is likely to be unique across all vendors’ System V implementations.
<code>SI_SRPC_DOMAIN</code>	Copies the Secure Remote Procedure Call domain name into the array pointed to by <i>buf</i> .

DIAGNOSTICS

Upon successful completion, the value returned indicates the buffer size in bytes required to hold the complete value and the terminating null character. If this value is no greater than the value passed in *count*, the entire string was copied; if this value is greater than *count*, the string copied into *buf* has been truncated to *count*-1 bytes plus a terminating null character.

Otherwise, a value of -1 is returned and `errno` is set to indicate the error.

Rationale

The commands included for `sysinfo` in SCD 2.2 are values that have been determined to be uniformly implemented on systems that have been presented for testing at the SCD 2.1 level. Also, the commands that require that the effective user-id be superuser are omitted.

Figure 1. Manifest Constants 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 */
```

The Network Services Library

The libnsl Interfaces

The interfaces found in Table 3 are also 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`.

Additionally, DEPRECATED functions needed for socket support can be found in Table 6-10.

Table 3. libns1 Contents,

rpc_broadcast_exp¹

Network Services Changes

The following are changes to the network services interfaces (detailed in the *System V Interface Definition, Third Edition*), as reported to SPARC International.

#	Facility	Location	Description
1	rpc_broadcast_exp	gABI	Add the function <code>rpc_broadcast_exp</code> to Figure 6-10 on page 6-14.

ABI Extension

The SCD includes an extra function `rpc_broadcast_exp`. This function is not described in any of the base documents. A man page for the function follows.

NAME

`rpc_broadcast_exp` - broadcast a call message specifying timeout

SYNOPSIS

```
#include <rpc/rpc.h>

enum clnt_stat rpc_broadcast_exp(const u_long prognum,
    const u_long versnum, const u_long procnum, const xdrproc_t xargs,
    caddr_t argsp, const xdrproc_t xresults, caddr_t resultsp,
    const resultproc_t eachresult, const int inittime, const int waittime,
    char *nettype);
```

DESCRIPTION

This function is like `rpc_broadcast()`, except that the initial timeout, *inittime*, and the maximum timeout, *waittime*, are specified in milliseconds.

inittime is the initial time that `rpc_broadcast_exp()` waits before resending the request. After the first resend, the re-transmission interval increases exponentially until it exceeds *waittime*.

The Socket Library

This section contains the socket internetworking interface, primarily used with the TCP/IP protocol suite. This is a REQUIRED interface set. It is also a DEPRECATED interface set effective November 1st, 1993. This interface set will not be removed from the SCD before November 1st, 1996.

This specification is based on *Programmer's Guide: Networking Interfaces* (Prentice-Hall, ISBN 0-13-020645-8).

All functions must be provided by systems from one or more of the libraries `/usr/lib/libnsl.so.1`, and `/usr/lib/libsocket.so.1`. Tables 4 and 5 document which system libraries are REQUIRED to provide each of the various socket functions.

Table 4. Socket Functions in libnsl

gethostbyaddr	inet_addr	inet_ntoa
gethostbyname	inet_netof	

Table 5. Socket Functions in libsocket

accept	getservbyport	recvfrom
bind	getsockname	recvmsg
connect	getsockopt	send
getpeername	inet_lnaof	sendmsg
getprotobyname	inet_makeaddr	sendto
getprotobynumber	inet_network	setsockopt
getprotoent	listen	shutdown
getservbyname	recv	socket

Structures and Manifest Constants

The Figures 2 through 5 contain the values of manifest constants and type declarations of the data types needed for the socket functions.

Figure 2. Manifest Constants and Data Types from <sys/socket.h>

```
/* Types */
#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_KEEPALIVE    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;          /* option on/off */
    int    l_linger;        /* linger time */
};

/* 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        /* internetwork: 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;          /* address family */
    char     sa_data[14];       /* up to 14 bytes of direct address */
};

/* Structure used by kernel to pass protocol * information in raw sockets. */

struct sockproto {
    u_short sp_family;          /* address family */
    u_short sp_protocol;        /* 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 ophdr {
    long      level;            /* protocol level affected */
    long      name;             /* option to modify */
    long      len;              /* length of option value */
};

#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;           /* the option */
    char     *val;              /* ptr to default value */
    int      len;               /* length of value */
};

/*
 * the opproc structure is used to build tables of options processing
 * functions for dooptions().
 */
struct opproc {
    int      level;             /* options level this function handles */
    int      (*func)();         /* the function */
};

/*
```

```

    * 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 */
};
```

Figure 3. 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; /* IP multicast address of group */
    struct in_addr imr_interface; /* local IP address of interface */
};
```

Figure 4. Manifest Constants and Data Types from <netdb.h>

```

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;           /* official service name */
    char    **s_aliases;       /* alias list */
    int     s_port;            /* port # */
    char    *s_proto;          /* protocol to use */
};

struct protoent {
    char    *p_name;           /* official protocol name */
    char    **p_aliases;       /* alias list */
    int     p_proto;           /* protocol # */
};

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

```

Figure 5. Manifest Constants and Data Types from <errno.h>

```
#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
```

Rationale

Two criteria must be satisfied for a function to be included in this specification.

- 1) Identical or similar implementation and documentation in all reference implementations.
- 2) Widespread use in applications for TCP/IP networking.

Basic Socket Interface Function Set

All the reference implementations inherit a basic set of socket functions from 4.2BSD relatively unchanged. All sixteen of these functions are included in this specification.

Another function,

```
int socketpair(int d, int type, int protocol, int sv[2]);
```

is excluded from the specification because it is only usable in the `AF_UNIX` address family, and is thus inapplicable to TCP/IP socket programming. In addition, there are differences in implementations; for example, one implementation adds `ENOBUFF` as an error code and another implementation returns a file descriptor number while other implementations return a success/failure code.

I/O Multiplexing

The `poll()` system call should be used for I/O multiplexing. The `select()` system call is not part of this specification because there is no implementation of the function which is binary compatible across all SPARC systems.

System Identifier

One implementation specification included two pairs of functions for getting and setting a symbolic hostname and a numeric hostid for the local system:

```
gethostname()
sethostname()
gethostid()
sethostid()
```

The functions `sethostname` and `sethostid` are not needed by normal applications and are excluded from this specification.

Applications which use `gethostname` should be changed to use `sysinfo(SI_SYSNAME, ...)`.

Applications which use `gethostid` should be changed to use `sysinfo(SI_HW_SERIAL, ...)`.

Ethernet Address Manipulation Routines

The specification excludes the functions from `ethers(3N)` for formatting Ethernet addresses, since display of Ethernet addresses is not generally needed in an application environment. These excluded functions are:

```
ether_aton()
ether_hostton()
ether_line()
ether_ntoa()
ether_ntohost()
```

BSD R* Support Functions

The support functions that implement the BSD R* protocols are excluded from the specification. These functions implement the specific R* client and server commands and are not generally used in common TCP/IP applications. These excluded functions are:

From `rcmd(3N)`:

```
rresvport()
```

```
rcmd()  
ruserok()  
From rexec(3N):  
rexec()  
From bindresvport(3N):  
bindresvport()
```

ioctl Calls

All socket ioctl calls are excluded from this specification. The two ioctl calls are

```
SIOCADDRT  
SIODELRT
```

These two calls relate to maintenance of the kernel routing tables and are implementation specific. These calls are only used by a user-level routing daemon and as such are not found in normal network applications.

Function Definitions

Basic Socket Interface Function Set

accept(3N)

```
int accept(int s, struct sockaddr *addr, int *addrlen);
```

bind(3N)

```
int bind(int s, struct sockaddr *name, int namelen);
```

connect(3N)

```
int connect(int s, struct sockaddr *name, int namelen);
```

getpeername(3N)

```
int getpeername(int s, struct sockaddr *name, int namelen);
```

getsockname(3N)

```
int getsockname(int s, struct sockaddr *name, int namelen);
```

One implementation spells out `struct linger` instead of referring to `<socket.h>`: delete this structure specification from man page and put in `<socket.h>`. One implementation gives numeric limits of 64Kbytes for `SO_SNDBUF` and `SO_RCVBUF`: delete these specific numeric limits.

```
int getsockopt(int s, int level, int optname, char *optval, int *optlen)
```

```
int setsockopt(int s, int level, int optname, char *optval, int *optlen)
```

One implementation makes the fourth argument `const`: delete this modifier.

listen(3N)

```
int listen(int s, int backlog)
```

recv(3N)

One implementation deletes `EFAULT`: delete `EFAULT` from the specification. Another implementation includes `ESTALE` and `EIO`: delete these error codes.

```
int recv(int s, char *buf, int len, int flags);
```

```
int recvfrom(int s, char *buf, int len, int flags, struct sockaddr *from,  
int *fromlen);
```

```
int recvmsg(int s, struct msghdr *msg, int flags);
```

send(3N)

```

    int send(int s, char *msg, int len, int flags);
    int sendto(int s, char *msg, int len, int flags, struct sockaddr *to, int tolen);
    int sendmsg(int s, msghdr *msg, int flags);
socket(3N)
    int socket(int domain, int type, int protocol);
shutdown(3N)
    int shutdown(int s, int how);
IP Address Manipulation Routines
inet(3N)
    unsigned long inet_addr(char *cp);
    int inet_network(char *cp);
    struct in_addr inet_makeaddr(int net, int lna);
    int inet_lnaof(struct in_addr in);
    int inet_netof(struct in_addr in);
    char *inet_ntoa(struct in_addr in);
Naming and Resource Lookup
gethostbyname(3N)
    struct hostent *gethostbyname(char *name)
    struct hostent *gethostbyaddr(char *addr, int length, int type)
getnetbyname(3N)
    struct netent *getnetbyname(char *name)
    struct netent *getnetbyaddr(long addr, int type)
getservbyname(3N)
    struct servent *getservbyname(char *name, char *proto)
    struct servent *getservbyport(int port, char *proto)
getprotobyname(3N)
    struct protoent *getprotobyname(char *name)
    struct protoent *getprotobyname(int proto)

```

The `gethostent()`, `getnetent()`, `getservent()`, `sethostent()`, `setnetent()`, `setservent()`, `endhostent()`, `endnetent()`, and `endservent()` routines found in all reference implementations are not included in this standard, since some specific implementations may not support the sequential search of a file implied by these routines.

DNS Resolver Interface

The following functions are not part of the specification. Their functionality is provided via `gethostbyname` and `gethostbyaddr`

```

resolver(3N)
    struct state _res;
    int res_mkquery(int op, char *dname, int class, int type, char *data, int datalen,
        struct rrec *newrr, char *buf, int buflen)

```

```
int res_send(char *msg, int msglen, char *answer, int anslen)
int res_init(void)
int dn_comp(char *exp_dn, char *comp_dn, int length, char **dnptrs,
            char **lastdnptr)
int dn_expand(char *msg, char *comp_dn, char exp_dn, int msglen, int length)
```

NAME

`accept` - accept a connection on a socket

SYNOPSIS

```
#include <sys/types.h>
#include <sys/socket.h>

int accept(int s, struct sockaddr *addr, int *addrlen);
```

DESCRIPTION

The argument *s* is a socket that has been created with `socket(3N)` and bound to an address with `bind(3N)`, and that is listening for connections after a call to `listen(3N)`. `accept` extracts the first connection on the queue of pending connections, creates a new socket with the properties of *s*, and allocates a new file descriptor, *ns*, for the socket. If no pending connections are present on the queue and the socket is not marked as non-blocking, `accept` blocks the caller until a connection is present. If the socket is marked as non-blocking and no pending connections are present on the queue, `accept` returns an error as described below. `accept` uses the `netconfig(4)` file to determine the STREAMS device file name associated with *s*. This is the device on which the connect indication will be accepted. The accepted socket, *ns*, is used to read and write data to and from the socket that connected to *ns*; it is not used to accept more connections. The original socket (*s*) remains open for accepting further connections.

The argument *addr* is a result parameter that is filled in with the address of the connecting entity as it is known to the communications layer. The exact format of the *addr* parameter is determined by the domain in which the communication occurs.

addrlen is a value-result parameter. Initially, it contains the amount of space pointed to by *addr*; on return it contains the length in bytes of the address returned.

`accept` is used with connection-based socket types, currently with `SOCK_STREAM`.

It is possible to `poll(2)` a socket for the purpose of an `accept` by polling it for a read. However, this will only indicate when a connect indication is pending; it is still necessary to call `accept`.

RETURN VALUES

`accept` returns -1 on error. If it succeeds, it returns a non-negative integer that is a descriptor for the accepted socket.

ERRORS

`accept` will fail if:

<code>EBADF</code>	The descriptor is invalid.
<code>ENODEV</code>	The protocol family and type corresponding to <i>s</i> could not be found in the <code>netconfig</code> file.
<code>ENOMEM</code>	There was insufficient user memory available to complete the operation.
<code>ENOSR</code>	There were insufficient STREAMS resources available to complete the operation.
<code>ENOTSOCK</code>	The descriptor does not reference a socket.
<code>EOPNOTSUPP</code>	The referenced socket is not of type <code>SOCK_STREAM</code> .
<code>EPROTO</code>	A protocol error has occurred; for example, the STREAMS protocol stack has not been initialized.
<code>EWouldBlock</code>	The socket is marked as non-blocking and no connections are present to be accepted.

NAME

bind - bind a name to a socket

SYNOPSIS

```
#include <sys/types.h>
#include <sys/socket.h>

int bind(int s, struct sockaddr *name, int namelen);
```

DESCRIPTION

bind assigns a name to an unnamed socket, *s*. When a socket is created with `socket(3N)`, it exists in a name space (address family) but has no *name* assigned. bind requests that the name pointed to by *name* be assigned to the socket. *namelen* specifies the size of *name*.

RETURN VALUES

If the bind is successful, a 0 value is returned. A return value of -1 indicates an error, which is further specified in the global `errno`.

ERRORS

The bind call will fail if:

EADDRINUSE	The specified address is already in use.
EADDRNOTAVAIL	The specified address is not available on the local machine.
EBADF	<i>s</i> is not a valid descriptor.
EINVAL	<i>namelen</i> is not the size of a valid address for the specified address family.
EINVAL	The socket is already bound to an address.
ENOSR	There were insufficient STREAMS resources for the operation to complete.
ENOTSOCK	<i>s</i> is a descriptor for a file, not a socket.

NAME

connect - initiate a connection on a socket

SYNOPSIS

```
#include <sys/types.h>
#include <sys/socket.h>

int connect(int s, struct sockaddr *name, int namelen);
```

DESCRIPTION

The parameter *s* is a socket. If it is of type `SOCK_DGRAM`, `connect` specifies the peer with which the socket is to be associated; this address is the address to which datagrams are to be sent if a receiver is not explicitly designated; it is the only address from which datagrams are to be received. If the socket *s* is of type `SOCK_STREAM`, `connect` attempts to make a connection to another socket. The other socket is specified by *name*. *name* is an address in the communication space of the socket. *namelen* specifies the size of data structure pointed to by *name*. Each communication space interprets the *name* parameter in its own way. If *s* is not bound, then it will be bound to an address selected by the underlying transport provider. Generally, stream sockets may successfully connect only once; datagram sockets may use `connect` multiple times to change their association. Datagram sockets may dissolve the association by connecting to a null address.

RETURN VALUES

If the connection or binding succeeds, then 0 is returned. Otherwise a -1 is returned and sets `errno` to indicate the error.

ERRORS

The call fails if:

<code>EADDRINUSE</code>	The address is already in use.
<code>EADDRNOTAVAIL</code>	The specified address is not available on the remote machine.
<code>EAFNOSUPPORT</code>	Addresses in the specified address family cannot be used with this socket.
<code>EALREADY</code>	The socket is non-blocking and a previous connection attempt has not yet been completed.
<code>EBADF</code>	<i>s</i> is not a valid descriptor.
<code>ECONNREFUSED</code>	The attempt to connect was forcefully rejected. The calling program should <code>close(2)</code> the socket descriptor, and issue another <code>socket(3N)</code> call to obtain a new descriptor before attempting another <code>connect</code> call.
<code>EINPROGRESS</code>	The socket is non-blocking and the connection cannot be completed immediately. It is possible to <code>poll(3C)</code> for completion by polling the socket for writing. However, this is only possible if the socket STREAMS module is the topmost module on the protocol stack with a write service procedure. This will be the normal case.
<code>EINTR</code>	The connection attempt was interrupted before any data arrived by the delivery of a signal.
<code>EINVAL</code>	<i>namelen</i> is not the size of a valid address for the specified address family.
<code>EISCONN</code>	The socket is already connected.
<code>ENETUNREACH</code>	The network is not reachable from this host.
<code>ENOSR</code>	There were insufficient STREAMS resources available to complete the operation.

NAME

gethostbyname, gethostbyaddr - get network host entry

SYNOPSIS

```
#include <sys/types.h>
#include <sys/socket.h>
#include <netdb.h>

struct hostent *gethostbyname(char *name);

struct hostent *gethostbyaddr(struct in_addr *addr,
    const sizeof(struct in_addr), const int AF_INET);
```

DESCRIPTION

gethostbyaddr, and gethostbyname each return a host entry.

The entry may come from the hosts file (see hosts(4)) or an implementation specific “hosts” table. The sources and their lookup order are unspecified.

gethostbyname searches for a host entry with a given hostname.

gethostbyaddr searches for a host entry with a given hostaddress.

The internal representation of a host entry is a structure defined in <netdb.h> with the following members:

```
char *h_name;
char **h_aliases;
int h_addrtype;
int h_length;
char **h_addr_list;
```

Host addresses are supplied in network byte order.

RETURN VALUES

gethostbyname and gethostbyaddr return a pointer to a struct hostent if they successfully locate the requested entry; otherwise they return NULL, and set an integer h_errno to indicate one of these errors: HOST_NOT_FOUND, TRY_AGAIN, NO_RECOVERY, NO_DATA and NO_ADDRESS (see /usr/include/netdb.h).

FILES

/etc/hosts

NOTES

All information is contained in a static area so it must be copied if it is to be saved.

NAME

getpeername - get name of connected peer

SYNOPSIS

```
int getpeername(int s, struct sockaddr *name, int *namelen);
```

DESCRIPTION

getpeername returns the name of the peer connected to socket *s*. The `int` pointed to by the *namelen* parameter should be initialized to indicate the amount of space pointed to by *name*. On return it contains the actual size of the *name* returned (in bytes). The *name* is truncated if the buffer provided is too small.

RETURN VALUES

If successful, getpeername returns 0; otherwise it returns -1 and sets `errno` to indicate the error.

ERRORS

The call succeeds unless:

EBADF	The argument <i>s</i> is not a valid descriptor.
ENOMEM	There was insufficient user memory for the operation to complete.
ENOSR	There were insufficient STREAMS resources available for the operation to complete.
ENOTCONN	The socket is not connected.
ENOTSOCK	The argument <i>s</i> is not a socket.

NAME

getprotobyname, getprotobynumber, getprotoent - get protocol entry

SYNOPSIS

```
#include <netdb.h>

struct protoent *getprotobyname(char *name);
struct protoent *getprotobynumber(int proto);
struct protoent *getprotoent(void);
```

DESCRIPTION

getprotoent, getprotobyname, and getprotobynumber each return a protocol entry. The entry may come from /etc/protocols or an implementation defined place. *name* is a pointer to one of the strings "tcp", "udp", or "icmp". *proto* is one of the values 6 (tcp), 17 (udp), 0 (ip), or 1 (icmp).

getprotoent enumerates protocol entries: successive calls to getprotoent will return either successive protocol entries or NULL. Enumeration may not be supported by some sources.

The internal representation of a protocol entry is a protoent structure defined in <netdb.h> with the following members:

```
char *p_name;
char **p_aliases;
int p_proto;
```

RETURN VALUES

getprotobyname and getprotobynumber return a pointer to a struct protoent if they successfully locate the requested entry; otherwise they return NULL.

getprotoent returns a pointer to a struct protoent if it successfully enumerates an entry; otherwise it returns NULL, indicating the end of the enumeration.

FILES

/etc/protocols

NOTES

All information is contained in a static area so it must be copied if it is to be saved.

Use of getprotoent is deprecated.

NAME

getservbyname, getservbyport - get service entry

SYNOPSIS

```
#include <netdb.h>

struct servent *getservbyname(char *name, char *proto);
struct servent *getservbyport(int port, char *proto);
```

DESCRIPTION

getservbyname, and getservbyport each return a service entry.

The entry may come from the services file (see services(4)) or an implementation defined source.

getservbyname searches for a service entry with a given service name.

getservbyport searches for a service entry with a given port number and, if the protocol name is non-NULL, the protocol.

name is a pointer to one of the strings "tcp" or "udp". *port* is the number of a well-known port.

The internal representation of a service entry is a struct `servent` defined in `<netdb.h>` with the following members:

```
char *s_name;
char **s_aliases;
int s_port;
char *s_proto;
```

RETURN VALUES

getservbyname and getservbyport return a pointer to a struct `servent` if they successfully locate the requested entry; otherwise they return NULL.

FILES

/etc/services

NOTES

All information is contained in a static area, so it must be copied if it is to be saved.

NAME

getsockname - get socket name

SYNOPSIS

```
#include <sys/types.h>
#include <sys/sockets.h>

int getsockname(int s, struct sockaddr *name, int *namelen);
```

DESCRIPTION

getsockname returns the current name for socket *s*. The *namelen* parameter should be initialized to indicate the amount of space pointed to by *name*. On return it contains the actual size in bytes of the *name* returned.

RETURN VALUES

If successful, getsockname returns 0; otherwise it returns -1 and sets *errno* to indicate the error.

ERRORS

The call succeeds unless:

EBADF	The argument <i>s</i> is not a valid file descriptor.
ENOMEM	There was insufficient memory available for the operation to complete.
ENOSR	There were insufficient STREAMS resources available for the operation to complete.
ENOTSOCK	The argument <i>s</i> is not a socket.

NAME

`inet_addr`, `inet_network`, `inet_makeaddr`, `inet_lnaof`, `inet_netof`, `inet_ntoa` - Internet address manipulation

SYNOPSIS

```
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>

unsigned long inet_addr(char *cp);
int inet_network(char *cp);
struct in_addr inet_makeaddr(int net, int lna);
int inet_lnaof(struct in_addr in);
int inet_netof(struct in_addr in);
char *inet_ntoa(struct in_addr in);
```

DESCRIPTION

The `inet_addr` and `inet_network` routines interpret a character string, *cp*, representing numbers expressed in the Internet standard '.' notation, returning numbers suitable for use as Internet addresses and Internet network numbers, respectively. The routine `inet_makeaddr` takes an Internet network number, *net*, and a local network address, *lna*, and constructs an Internet address from it. The routines `inet_netof` and `inet_lnaof` break apart an Internet host address, *in*, returning the network number and local network address part, respectively.

The routine `inet_ntoa` returns a pointer to a string in the base 256 notation "d.d.d.d" described below.

All Internet addresses are returned in network order (bytes ordered from left to right). All network numbers and local address parts are returned as machine format integer values.

INTERNET ADDRESSES

Values specified using the '.' notation take one of the following forms:

```
a.b.c.d
a.b.c
a.b
a
```

When four parts are specified, each is interpreted as a byte of data and assigned, from left to right, to the four bytes of an Internet address.

When a three part address is specified, the last part is interpreted as a 16-bit quantity and placed in the right most two bytes of the network address. This makes the three part address format convenient for specifying Class B network addresses as "128.net.host".

When a two part address is supplied, the last part is interpreted as a 24-bit quantity and placed in the right most three bytes of the network address. This makes the two part address format convenient for specifying Class A network addresses as "net.host".

When only one part is given, the value is stored directly in the network address without any byte rearrangement.

All numbers supplied as "parts" in a '.' notation may be decimal, octal, or hexadecimal, as specified

in the C language (that is, a leading 0x or 0X implies hexadecimal; otherwise, a leading 0 implies octal; otherwise, the number is interpreted as decimal).

RETURN VALUES

The value -1 is returned by `inet_addr` and `inet_network` for malformed requests.

The routines `inet_netof` and `inet_lnaof` break apart Internet host addresses, returning the network number and local network address part, respectively.

The routine `inet_ntoa` returns a pointer to a string in the base 256 notation “d.d.d.d” described below.

NAME

`listen` - listen for connections on a socket

SYNOPSIS

```
#include <sys/types.h>
#include <sys/sockets.h>
```

```
int listen(int s, int backlog);
```

DESCRIPTION

To accept connections, a socket, *s*, is first created with `socket(3N)`, a *backlog* for incoming connections is specified with `listen` and then the connections are accepted with `accept(3N)`. The `listen` call applies only to sockets of type `SOCK_STREAM` or `SOCK_SEQPACKET`.

The *backlog* parameter defines the maximum length the queue of pending connections may grow to. If a connection request arrives with the queue full, the client will receive an error with an indication of `ECONNREFUSED`.

RETURN VALUES

A 0 return value indicates success; -1 indicates an error.

ERRORS

The call fails if:

<code>EBADF</code>	The argument <i>s</i> is not a valid file descriptor.
<code>ENOTSOCK</code>	The argument <i>s</i> is not a socket.
<code>EOPNOTSUPP</code>	The socket is not of a type that supports the operation <code>listen</code> .

NOTES

There is currently no backlog limit.

NAME

`recv`, `recvfrom`, `recvmsg` - receive a message from a socket

SYNOPSIS

```
#include <sys/types.h>
#include <sys/socket.h>
#include <sys/uio.h>

int recv(int s, char *buf, int len, int flags);
int recvfrom(int s, char *buf, int len, int flags, struct sockaddr *from,
             int *fromlen);
int recvmsg(int s, struct msghdr *msg, int flags);
```

DESCRIPTION

`recv`, `recvfrom`, and `recvmsg` are used to receive messages from another socket. `recv` may be used only on a connected socket (see `connect(3N)`), while `recvfrom` and `recvmsg` may be used to receive data on a socket whether it is in a connected state or not. *s* is a socket created with `socket(3N)`. *buf* is a pointer to the buffer to receive the data and *len* is its size in bytes.

If *from* is not a NULL pointer, the source address of the message is filled in. *fromlen* is a value-result parameter, initialized to the size of the buffer associated with *from*, and modified on return to indicate the actual size of the address stored there. The length of the message is returned. If a message is too long to fit in the supplied buffer, excess bytes may be discarded depending on the type of socket the message is received from (see `socket(3N)`).

If no messages are available at the socket, the receive call waits for a message to arrive, unless the socket is nonblocking (see `fcntl(2)`) in which case -1 is returned with the external variable `errno` set to `EWOULDBLOCK`.

The `poll` call may be used to determine when more data arrives.

The flags parameter is formed by ORing one or more of the following:

<code>MSG_OOB</code>	Read any out-of-band data present on the socket rather than the regular in-band data.
<code>MSG_PEEK</code>	Peek at the data present on the socket; the data is returned, but not consumed, so that a subsequent receive operation will see the same data.

The `recvmsg` call uses a `struct msghdr, msg`, to minimize the number of directly supplied parameters. This structure is defined in `<sys/socket.h>` and includes the following members:

```
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;
```

Here `msg_name` and `msg_namelen` specify the destination address if the socket is unconnected; `msg_name` may be given as a NULL pointer if no names are desired or required. The `msg_iov` and `msg_iovlen` describe the scatter-gather locations, as described in `read(2)`. A buffer to receive any access rights sent along with the message is specified in `msg_accrights`, which has length `msg_accrightslen`.

RETURN VALUES

These calls return the number of bytes received, or -1 if an error occurred.

ERRORS

The calls fail if:

EBADF	<i>s</i> is an invalid file descriptor.
EINTR	The operation was interrupted by delivery of a signal before any data was available to be received.
ENOMEM	There was insufficient user memory available for the operation to complete.
ENOSR	There were insufficient STREAMS resources available for the operation to complete.
ENOTSOCK	<i>s</i> is not a socket.
EWOULDBLOCK	The socket is marked non-blocking and the requested operation would block.

NAME

`send`, `sendto`, `sendmsg` - send a message from a socket

SYNOPSIS

```
#include <sys/types.h>
#include <sys/socket.h>

int send(int s, char *buf, int len, int flags);
int sendto(int s, char *buf, int len, int flags, struct sockaddr *to, int tolen);
int sendmsg(int s, struct msghdr *msg, int flags);
```

DESCRIPTION

`send`, `sendto`, and `sendmsg` are used to transmit a message to another transport end-point. `send` may be used only when the socket is in a connected state, while `sendto` and `sendmsg` may be used at any time. *s* is a socket created with `socket(3N)`. *buf* points to a buffer containing the data to be sent. *len* is number of bytes to be sent.

The address of the target is given by *to* with *tolen* specifying its size. The length of the message is given by *len*. If the message is too long to pass atomically through the underlying protocol, then the error `EMSGSIZE` is returned, and the message is not transmitted.

A return value of -1 indicates locally detected errors only. It does not implicitly mean the message was not delivered.

If the socket does not have enough buffer space available to hold the message being sent, `send` blocks, unless the socket has been placed in non-blocking I/O mode (see `fcntl(2)`). The `poll` call may be used to determine when it is possible to send more data.

The flags parameter is formed from the bitwise OR of zero or more of the following:

<code>MSG_OOB</code>	Send out-of-band data on sockets that support this notion. The underlying protocol must also support out-of-band data. Only <code>SOCK_STREAM</code> sockets created in the <code>AF_INET</code> address family support out-of-band data.
<code>MSG_DONTROUTE</code>	The <code>SO_DONTROUTE</code> option is turned on for the duration of the operation. It is used only by diagnostic or routing programs.

See `recv(3N)` for a description of the `msghdr` structure.

RETURN VALUES

These calls return the number of bytes sent, or -1 if an error occurred.

ERRORS

The calls fail if:

<code>EBADF</code>	<i>s</i> is an invalid file descriptor.
<code>EINTR</code>	The operation was interrupted by delivery of a signal before any data could be buffered to be sent.
<code>EINVAL</code>	<i>tolen</i> is not the size of a valid address for the specified address family.
<code>EMSGSIZE</code>	The socket requires that message be sent atomically, and the message was too long.
<code>ENOMEM</code>	There was insufficient memory available to complete the operation.
<code>ENOSR</code>	There were insufficient STREAMS resources available for the operation to complete.
<code>ENOTSOCK</code>	<i>s</i> is not a socket.

EWouldBlock

The socket is marked non-blocking and the requested operation would block.

NAME

getsockopt, setsockopt - get and set options on sockets

SYNOPSIS

```
#include <sys/types.h>
#include <sys/socket.h>

int getsockopt(int s, int level, int optname, void *optval, int *optlen);
int setsockopt(int s, int level, int optname, void *optval, int optlen);
```

DESCRIPTION

getsockopt and setsockopt manipulate options associated with a socket, *s*. Options may exist at multiple protocol levels; they are always present at the uppermost socket level.

When manipulating socket options, the level at which the option resides and the name of the option must be specified. To manipulate options at the socket level, *level* is specified as `SOL_SOCKET`. To manipulate options at any other level, *level* is the protocol number of the protocol that controls the option. For example, to indicate that an option is to be interpreted by the TCP protocol, *level* is set to the TCP protocol number (see `getprotobyname(3N)`).

The parameters *optval* and *optlen* are used to access option values for `setsockopt`. For `getsockopt`, they identify a buffer in which the value(s) for the requested option(s) are to be returned. For `getsockopt`, *optlen* is a value-result parameter, initially containing the size of the buffer pointed to by *optval*, and modified on return to indicate the actual size of the value returned. Use a 0 *optval* if no option value is to be supplied or returned.

optname and any specified options are passed uninterpreted to the appropriate protocol module for interpretation. The include file `<sys/socket.h>` contains definitions for the socket-level options described below. Options at other protocol levels vary in format and name.

Most socket-level options take an `int` for *optval*. For `setsockopt`, the *optval* parameter should be non-zero to enable a boolean option, or zero if the option is to be disabled. `SO_LINGER` uses a `struct linger` parameter that specifies the desired state of the option and the linger interval (see below). `struct linger` is defined in `<sys/socket.h>`. `struct linger` contains the following members:

<code>l_onoff</code>	option on/off
<code>l_linger</code>	linger time

The following options are recognized at the socket level. Except as noted, each may be examined with `getsockopt` and set with `setsockopt`.

<code>SO_DEBUG</code>	toggle recording of debugging information
<code>SO_REUSEADDR</code>	toggle local address reuse
<code>SO_KEEPAIVE</code>	toggle keep connections alive
<code>SO_DONTROUTE</code>	toggle routing bypass for outgoing messages
<code>SO_LINGER</code>	linger on close if data is present
<code>SO_BROADCAST</code>	toggle permission to transmit broadcast messages
<code>SO_OOBINLINE</code>	toggle reception of out-of-band data in band
<code>SO_SNDBUF</code>	set buffer size for output
<code>SO_RCVBUF</code>	set buffer size for input
<code>SO_TYPE</code>	get the type of the socket (get only)
<code>SO_ERROR</code>	get and clear error on the socket (get only)

`SO_DEBUG` enables debugging in the underlying protocol modules. `SO_REUSEADDR` indicates that the rules used in validating addresses supplied in a `bind(3N)` call should allow reuse of local addresses. `SO_KEEPAIVE` enables the periodic transmission of messages on a connected socket. If the connected party fails to respond to these messages, the connection is considered broken and processes using the socket are notified using a `SIGPIPE` signal. `SO_DONTROUTE` indicates that outgoing messages should bypass the standard routing facilities. Instead, messages are directed to the appropriate network interface according to the network portion of the destination address.

`SO_LINGER` controls the action taken when unsent messages are queued on a socket and a `close(2)` is performed. If the socket promises reliable delivery of data and `SO_LINGER` is set, the system will block the process on the close attempt until it is able to transmit the data or until it decides it is unable to deliver the information (a timeout period, termed the linger interval, is specified in the `setsockopt` call when `SO_LINGER` is requested). If `SO_LINGER` is disabled and a close is issued, the system will process the close in a manner that allows the process to continue as quickly as possible.

The option `SO_BROADCAST` requests permission to send broadcast datagrams on the socket. With protocols that support out-of-band data, the `SO_OOBINLINE` option requests that out-of-band data be placed in the normal data input queue as received; it will then be accessible with `recv` or `read` calls without the `MSG_OOB` flag.

`SO_SNDBUF` and `SO_RCVBUF` are options that adjust the normal buffer sizes allocated for output and input buffers, respectively. The buffer size may be increased for high-volume connections or may be decreased to limit the possible backlog of incoming data.

Finally, `SO_TYPE` and `SO_ERROR` are options used only with `getsockopt`. `SO_TYPE` returns the type of the socket (for example, `SOCK_STREAM`). It is useful for servers that inherit sockets on startup. `SO_ERROR` returns any pending error on the socket and clears the error status. It may be used to check for asynchronous errors on connected datagram sockets or for other asynchronous errors.

RETURN VALUES

If successful, `getsockopt` returns 0; otherwise it returns -1 and sets `errno` to indicate the error.

ERRORS

The call succeeds unless:

<code>EBADF</code>	The argument <code>s</code> is not a valid file descriptor.
<code>ENOMEM</code>	There was insufficient memory available for the operation to complete.
<code>ENOPROTOPT</code>	The option is unknown at the level indicated.
<code>ENOSR</code>	There were insufficient STREAMS resources available for the operation to complete.
<code>ENOTSOCK</code>	The argument <code>s</code> is not a socket.

NAME

shutdown - shut down part of a full-duplex connection

SYNOPSIS

```
int shutdown(int s, int how);
```

DESCRIPTION

The `shutdown` call shuts down all or part of a full-duplex connection on the socket associated with *s*. If *how* is 0, then further receives will be disallowed. If *how* is 1, then further sends will be disallowed. If *how* is 2, then further sends and receives will be disallowed.

RETURN VALUES

A 0 is returned if the call succeeds, -1 if it fails.

ERRORS

The call succeeds unless:

EBADF	<i>s</i> is not a valid file descriptor.
ENOMEM	There was insufficient user memory available for the operation to complete.
ENOSR	There were insufficient STREAMS resources available for the operation to complete.
ENOTCONN	The specified socket is not connected.
ENOTSOCK	<i>s</i> is not a socket.

NOTES

The *how* values should be defined constants.

NAME

`socket` - create an endpoint for communication

SYNOPSIS

```
#include <sys/types.h>
#include <sys/socket.h>

int socket(int domain, int type, int protocol);
```

DESCRIPTION

`socket` creates an endpoint for communication and returns a descriptor.

The *domain* parameter specifies a communications domain within which communication will take place; this selects the protocol family which should be used. The protocol family generally is the same as the address family for the addresses supplied in later operations on the socket. These families are defined in the include file `<sys/socket.h>`.

The only supported protocol family is `PF_INET`.

The socket has the indicated *type*, which specifies the communication semantics. Currently defined *types* are:

```
SOCK_STREAM
SOCK_DGRAM
SOCK_SEQPACKET
```

A `SOCK_STREAM` type provides sequenced, reliable, two-way connection-based byte streams. An out-of-band data transmission mechanism may be supported. A `SOCK_DGRAM` socket supports datagrams (connectionless, unreliable messages of a fixed (typically small) maximum length). A `SOCK_SEQPACKET` socket may provide a sequenced, reliable, two-way connection-based data transmission path for datagrams of fixed maximum length; a consumer may be required to read an entire packet with each `read` system call. This facility is protocol specific, and presently not implemented for any protocol family.

protocol specifies a particular protocol to be used with the socket. Normally only a single protocol exists to support a particular socket type within a given protocol family. However, multiple protocols may exist, in which case a particular protocol must be specified in this manner. The protocol number to use is particular to the “communication domain” in which communication is to take place. If a protocol is specified by the caller, then it will be packaged into a socket level option request and sent to the underlying protocol layers.

Sockets of type `SOCK_STREAM` are full-duplex byte streams, similar to pipes. A stream socket must be in a connected state before any data may be sent or received on it. A connection to another socket is created with a `connect(3N)` call. Once connected, data may be transferred using `read(2)` and `write(2)` calls or some variant of the `send(3N)` and `recv(3N)` calls. When a session has been completed, a `close(2)` may be performed. Out-of-band data may also be transmitted as described on the `send(3N)` manual page and received as described on the `recv(3N)` manual page.

The communications protocols used to implement a `SOCK_STREAM` insure that data is not lost or duplicated. If a piece of data for which the peer protocol has buffer space cannot be successfully transmitted within a reasonable length of time, then the connection is considered broken and calls will indicate an error with -1 returns and with `ETIMEDOUT` as the specific code in the global variable `errno`. The protocols optionally keep sockets “warm” by forcing transmissions roughly every minute in the absence of other activity. An error is then indicated if no response can be elicited on an otherwise idle connection for an extended period (for instance 5 minutes). A `SIGPIPE` signal is raised if a process sends on a broken stream; this causes naive processes, which do not handle the signal, to exit.

`SOCK_SEQPACKET` sockets employ the same system calls as `SOCK_STREAM` sockets. The only difference is that `read` calls will return only the amount of data requested, and any remaining in the arriving packet will be discarded.

`SOCK_DGRAM` sockets allow datagrams to be sent to correspondents named in `sendto` calls. Datagrams are generally received with `recvfrom`, which returns the next datagram with its return address.

An `ioctl(2)` call can be used to specify a process group to receive a `SIGURG` signal when the out-of-band data arrives. It may also enable non-blocking I/O and asynchronous notification of I/O events with `SIGPOLL` signals.

The operation of sockets is controlled by socket level options. These options are defined in the file `<sys/socket.h>`. `setsockopt` and `getsockopt(3N)` are used to set and get options, respectively.

RETURN VALUES

A -1 is returned if an error occurs. Otherwise the return value is a descriptor referencing the socket.

ERRORS

The `socket` call fails if:

<code>EACCES</code>	Permission to create a socket of the specified type and/or protocol is denied.
<code>EMFILE</code>	The per-process descriptor table is full.
<code>ENOMEM</code>	Insufficient user memory is available.
<code>ENOSR</code>	There were insufficient STREAMS resources available to complete the operation.
<code>EPROTONOSUPPORT</code>	The protocol type or the specified protocol is not supported within this domain.

Dynamic Object File Loading

All interfaces described in this section are REQUIRED. Systems must supply `/usr/lib/libdl.so.1`. The functions supplied by this library listed in Table 6. Manifest constants for these functions are defined in Figure 6.

Manual pages for the required functions are missing from the SVID so they have been included after Figure 6.

Introduction

The run-time dynamic linking facilities of the system are made available to the executing application program through the functions in `libdl`:

`dlopen`, `dlclose`, `dlsym`, and `dlerror`

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 following terms are used in this specification:

For a *program* to *reference* a symbol means for the program to use the storage value associated with that symbol. To reference a data symbol means (a) to retrieve the value stored in the location associated with that symbol, or (b) to store a value into the location associated with that symbol. To reference a function symbol means to (a) use the value directly by calling that function, or (b) to obtain its value via a call to `dlsym`, presumably in order to call the function later.

For a *program* to *contain a reference* to a symbol means that the program has been constructed in such a way that it will reference a symbol that is not defined within it. In the C language, this is done by declaring a data or function to have the `extern` attribute. The *reference* that the program contains is an indication to the linker and loader of what the name of the symbol is, and the fact that it will be found in some other program. For details on how this is implemented in a SPARC executable file, see the *System V Application Binary Interface* and the *System V Application Binary Interface, SPARC Processor Supplement*.

Two kinds of objects are mentioned in these specifications. A *data object* is the storage location associated with a symbol in an application program. A *shared object* is (a) a file on disk that was created by linking a program as a shared object, or (b) such a file that has been loaded into memory and prepared for execution. When the word “object” is used without qualification in this specification, it means shared object, and usually the shared object in memory.

For an *object* to *reference another object* means that the first object has been linked with the second object in such a way as to create `DT_NEEDED` entries that cause the second object to be loaded automatically with the first object. (See Chapter 5 of this document.)

Table 6. libdl Contents

dlclose

dLError

Figure 6. 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 */
```

ABI Extensions

The dynamic linking routines required by the SCD are not described in any of the the base documents. Descriptions for each of the required functions follows.

NAME

`dlclose` - close a shared object

SYNOPSIS

```
#include <dlfcn.h>

int dlclosel(void *handle);
```

DESCRIPTION

The function `dlclose` disassociates from the current process a shared object previously opened by `dlopen`.

handle is a value that was returned from a previous call to `dlopen`. It designates the shared object whose *pathname* was specified in that previous call to `dlopen`.

Once an object has been dissasociated from the process using `dlclose`, its symbols and those of any objects that were loaded automatically as a result of opening the object designated by *handle* are no longer available to `dlsym` via *handle*.

In order for `dlclose` to dissasociate an object from a process, there must have been exactly one `dlclose` executed for each `dlopen` that was executed. Thus if a `dlopen` was executed once for a *pathname*, `dlclose` would have to be executed once with the handle that was returned for *pathname*. If a `dlopen` were executed twice for the same *pathname*, the disassociation would occur only after the second `dlclose`.

A successful invocation of `dlclose` does not guarantee that the objects associated with *handle* will actually be removed from the address space of the process, even if the object has been disassociated from the process and its symbols are no longer available through *handle*. Objects loaded by one invocation of `dlopen` may also be loaded by another invocation of `dlopen`. The same object may also be opened multiple times. An object may be removed from the address space by the system only after all references to that object through an explicit `dlopen` invocation have been closed and all other objects that reference that object have also been closed. Even then, however, it is unspecified in this standard whether the object will actually be removed from the address space.

When the system removes an object from the process address space, the object's termination function is executed. The termination function for each object is specified by the `DT_FINI` entry in that object's `.dynamic` section. The exact timing of the execution of termination function relative to the timing of the `dlclose` that release the object is unspecified in this standard.

An SCD-conforming application will not have any processing dependencies upon the system's removal or non-removal of an object from the process address space following `dlclose`.

DIAGNOSTICS

If the referenced object was successfully closed, `dlclose` returns 0. If the object could not be closed, or if *handle* does not refer to an open object, `dlclose` returns a non-0 value. More detailed diagnostic information will be available through `dlerror`.

NOTES

The following notes are a consequence of that fact that this standard does not specify whether an object ever is actually removed from a process address space:

Once a program has executed a sequence of `dlclose` operations that would permit the system to remove an object from the process address space, the result of the program's executing any reference to symbols defined in that object are unspecified in this standard.

Once a program has executed a sequence of `dlclose` operations that would permit the system to remove an object from the process address space, if the program executes another `dlopen` for that object, it is unspecified in this standard whether the object is actually loaded again and whether the object's data will be in its initial state.

NAME

`dlerror` - get diagnostic information

SYNOPSIS

```
#include <dlfcn.h>

char *dlerror(void);
```

DESCRIPTION

The function `dlerror` returns a null-terminated character string (with no trailing newline) that describes the last error that occurred during dynamic linking processing. If no dynamic linking errors have occurred since the last invocation of `dlerror`, `dlerror` returns `NULL`. Thus, invoking `dlerror` a second time, immediately following a prior invocation, will result in `NULL` being returned.

NOTES

The messages returned by `dlerror` may reside in a static buffer that is overwritten on each call to `dlerror`. Application code should not write to this buffer. Programs wishing to preserve an error message should make their own copies of that message.

NAME

`dlopen` - open a shared object

SYNOPSIS

```
#include <dlfcn.h>

void *dlopen(char *pathname, int mode);
```

DESCRIPTION

The function `dlopen` is one of a family of routines that give the user direct access to the dynamic linking facilities.

The function `dlopen` makes a shared object available to a running process. `dlopen` returns to the process a *handle* the process must use to identify the object on subsequent calls to `dlsym` and `dlclose`. This value must not be interpreted in any way by the process. (See Rationale)

pathname is the path name of the object to be opened; it may be an absolute path or relative to the current directory. If the value of *pathname* is 0, `dlopen` will make the symbols contained in the original `a.out`, and all of the objects that were loaded at program startup with the `a.out`, available through `dlsym`.

If the value of *pathname* is not zero, and no file specified by *pathname* has already been loaded into the address space, the file specified by *pathname* will be loaded. If the file specified by *pathname* contains `DT_NEEDED` entries for other shared objects, those objects will automatically be loaded by `dlopen`. The directory search path to be used to find both *pathname* and the other objects to be automatically loaded are given in Chapter 5 of this document and in the normative references specified there.

Objects whose names resolve to the same absolute or relative path name may be opened any number of times either using `dlopen` or automatically as a result of executing `dlopen` for an object that uses them. However, the object referenced is loaded only once into the address space of the current process. This means that the object only takes up space once; there is only one copy of its static data; and the static data are initialized only once, when the initial load takes place.

When a shared object is brought into the address space of a process, it may contain references to symbols whose addresses are not known until the object is loaded. These references must be relocated before the symbols can be accessed. The *mode* parameter governs when these relocations take place and may have the following values:

<code>RTLD_LAZY</code>	Under this <i>mode</i> , only references to data symbols are relocated when the object is loaded. References to functions are not relocated until a given function is referenced for the first time by the executing program. This <i>mode</i> should result in better performance, since a process may not reference all of the functions in any given shared object.
<code>RTLD_NOW</code>	Under this <i>mode</i> , all necessary relocations are performed when the object is first loaded. This may result in some wasted effort, if relocations are performed for functions that are never referenced, but is useful for applications that need to know as soon as an object is loaded that all symbols referenced during execution will be available.

The *mode* parameter only takes effect when an object is initially loaded. If `RTLD_LAZY` is specified in the first `dlopen` for an object, and `RTLD_NOW` is specified for the second `dlopen` of the same object, the second `dlopen` will not cause any relocations to be performed.

The *mode* parameter is required, and always overrides the value of the `LD_BIND_NOW` environment variable.

When the system loads an object for the first time, the object's initialization function is executed. The initialization function for each object is specified by the `DT_INIT` entry in that object's

.dynamic section. If multiple objects are loaded as a result of `dlopen`, the order initialization functions are called is unspecified.

Objects loaded by a single invocation of `dlopen` may import symbols from one another or from any object loaded automatically with `a.out` during program startup, but objects loaded by one `dlopen` invocation may not directly reference symbols from objects loaded by a different `dlopen` invocation. Those symbols may, however, be referenced indirectly using `dlsym`.

Rationale

The functions `dlopen` and `dlclose` may not work in a manner consistent with the way the functions `open` and `close` work. For example, if the same file is opened twice, the `open` function will return unique file descriptors for each `open` operation. Using `dlopen` to open the same file multiple times may return the same file handle every time. The result is that if the first file handle for a `dlopen` call is used more than once as a parameter to `dlclose`, there may be unexpected side effects.

DIAGNOSTICS

If the file specified by *pathname* cannot be found, cannot be opened for reading, is not a shared object, or if an error occurs during the process of loading the file specified by *pathname* or relocating its symbolic references, `dlopen` will return `NULL`. More detailed diagnostic information will be available through `dlerror`.

NOTES

The same object referenced by different path names may be loaded multiple times. For example, given the object `/usr/home/me/mylibs/mylib.so`, and assuming the current working directory is `/usr/home/me/workdir`,

```
...
void *handle1;
void *handle2;

handle1 = dlopen("../mylibs/mylib.so", RTLD_LAZY);
handle2 = dlopen("/usr/home/me/mylibs/mylib.so", RTLD_LAZY);
...
```

results in `mylibs.so` being loaded twice for the current process. On the other hand, given the same object and current working directory, if `LD_LIBRARY_PATH=/usr/home/me/mylibs`, then

```
...
void *handle1;
void *handle2;

handle1 = dlopen("mylib.so", RTLD_LAZY);
handle2 = dlopen("/usr/home/me/mylibs/mylib.so", RTLD_LAZY);
...
```

results in `mylibs.so` being loaded only once.

Users who wish to gain access to the symbol table of the `a.out` itself using `dlopen(0, mode)` should be aware that some symbols defined in the `a.out` may not be available to the dynamic linker. The symbol table created by `ld` for use by the dynamic linker might contain only a subset of the symbols originally defined in the `a.out`: specifically, those referenced by the shared objects with which the `a.out` is linked.

NAME

`dlsym` - get the address of a symbol in a shared object

SYNOPSIS

```
#include <dlfcn.h>

void *dlsym(void *handle, char *name);
```

DESCRIPTION

The function `dlsym` allows a process to obtain the address of a symbol defined within a shared object previously opened by `dlopen`.

handle is a value returned by a call to `dlopen`; the corresponding shared object must not have been disassociated from the executing process using `dlclose`. *name* is the symbol's name as a character string.

`dlsym` searches for the named symbol in the shared object designated by *handle* and in all shared objects loaded automatically as a result of loading the object referenced by *handle* [see `dlopen(3X)`].

EXAMPLES

The following example shows how one can use `dlopen` and `dlsym` to access either function or data objects. For simplicity, error checking has been omitted.

```
void *handle;
int i, *iptr;
int (*fptr)(int);
/* open the needed object */
handle = dlopen("/usr/mydir/libx.so", RTLD_LAZY);
/* find address of function and data objects */
fptr = (int (*)(int))dlsym(handle, "some_function");
iptr = (int *)dlsym(handle, "int_object");
/* invoke function, passing value of integer as a parameter */
i = (*fptr)(*iptr);
```

DIAGNOSTICS

If *handle* does not refer to a valid object opened by `dlopen`, or if the named symbol cannot be found within any of the objects associated with *handle*, `dlsym` will return `NULL`. More detailed diagnostic information will be available through `dlerror`.

Miscellaneous ABI 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
	Dependencies Among Libraries	gABI	Change - On page 6-2, at the statement which begins "Application 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 <code>DT_NEEDED</code> entries. 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.
	<sys/types.h>	psABI	Addition - On page 6-65, in Figure 6-41, 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>

The X Library

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).

All SCD 2.2 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 X Library Interfaces

The interfaces listed below in Table 7 have been included in SCD 2.2 because they are REQUIRED to be present on all compliant systems, in the dynamic library `/usr/lib/libX11.so.5`.

Table 8 contains the exported data which are also REQUIRED to be present in `/usr/lib/libX11.so.5`. The format of these entries is: `data[size]`.

Conformant systems are also REQUIRED to have `/usr/lib/libX11.so.4` in order to support SPARC applications written to conform to versions 2.0 and 2.1 of the *SPARC Compliance Definition*. Since the X Version 11, Release 5 specification is a proper superset of the X Version 11, Release 4 specification, system vendors can provide this support by simply making a link to `/usr/lib/libX11.so.5` for the file `/usr/lib/libX11.so.4` as part of the system installation process. If an application executable references `/usr/lib/libX11.so.4` and does not reference `/usr/lib/libX11.so.5`, the application may not use any functionality defined by X11R5 but not defined by X11R4. See the *SPARC Compliance Definition 2.1* for a list of X11R4 components. This file name, `/usr/lib/libX11.so.4`, is deprecated effective November 1st, 1993; this file name may go away as early as November 1st, 1996 so new applications should not rely on its existence.

Figures 7 through 14 detail the manifest constants and visible data structures associated with the X library.

Table 7. Contents of libX

XActivateScreenSaver	XcmsAddFunctionSet
XAddExtension	XcmsAllocColor
XAddHost	XcmsAllocNamedColor
XAddHosts	XcmsCCCOfColormap
XAddPixel	XcmsCIELabClipab
XAddToExtensionList	XcmsCIELabClipL
XAddToSaveSet	XcmsCIELabClipLab
XAllocClassHint	XcmsCIELabQueryMaxC
XAllocColor	XcmsCIELabQueryMaxL
XAllocColorCells	XcmsCIELabQueryMaxLC
XAllocColorPlanes	XcmsCIELabQueryMinL
XAllocIconSize	XcmsCIELabToCIEXYZ
XAllocNamedColor	XcmsCIELabWhiteShiftColors
XAllocSizeHints	XcmsCIELuvClipL
XAllocStandardColormap	XcmsCIELuvClipLuv
XAllocWMHints	XcmsCIELuvClipuv
XAllowEvents	XcmsCIELuvQueryMaxC
XAllPlanes	XcmsCIELuvQueryMaxL
XAutoRepeatOff	XcmsCIELuvQueryMaxLC
XAutoRepeatOn	XcmsCIELuvQueryMinL
XBaseFontNameListOfFontSet	XcmsCIELuvToCIEuvY
XBell	XcmsCIELuvWhiteShiftColors
XBitmapBitOrder	XcmsCIEuvYToCIELuv
XBitmapPad	XcmsCIEuvYToCIEXYZ
XBitmapUnit	XcmsCIEuvYToTekHVC
XBlackPixel	XcmsCIExyYToCIEXYZ
XBlackPixelOfScreen	XcmsCIEXYZToCIELab
XCellsOfScreen	XcmsCIEXYZToCIEuvY
XChangeActivePointerGrab	XcmsCIEXYZToCIExyY
XChangeGC	XcmsCIEXYZToRGBi
XChangeKeyboardControl	XcmsClientWhitePointOfCCC
XChangeKeyboardMapping	XcmsConvertColors
XChangePointerControl	XcmsCreateCCC
XChangeProperty	XcmsDefaultCCC
XChangeSaveSet	XcmsDisplayOfCCC
XChangeWindowAttributes	XcmsFormatOfPrefix
XCheckIfEvent	XcmsFreeCCC
XCheckMaskEvent	XcmsLookupColor
XCheckTypedEvent	XcmsPrefixOfFormat
XCheckTypedWindowEvent	XcmsQueryBlack
XCheckWindowEvent	XcmsQueryBlue
XCirculateSubwindows	XcmsQueryColor
XCirculateSubwindowsDown	XcmsQueryColors
XCirculateSubwindowsUp	XcmsQueryGreen
XClearArea	XcmsQueryRed
XClearWindow	XcmsQueryWhite
XClipBox	XcmsRGBiToCIEXYZ
XCloseDisplay	XcmsRGBiToRGB
XCloseIM	XcmsRGBToRGBi
XcmsAddColorSpace	XcmsScreenNumberOfCCC

XcmsScreenWhitePointOfCCC	XDeleteProperty
XcmsSetCompressionProc	XDestroyIC
XcmsSetWhiteAdjustProc	XDestroyImage
XcmsSetWhitePoint	XDestroyRegion
XcmsStoreColor	XDestroySubwindows
XcmsStoreColors	XDestroyWindow
XcmsTekHVCClipC	XDisableAccessControl
XcmsTekHVCClipV	XDisplayCells
XcmsTekHVCClipVC	XDisplayHeight
XcmsTekHVCQueryMaxC	XDisplayHeightMM
XcmsTekHVCQueryMaxV	XDisplayKeycodes
XcmsTekHVCQueryMaxVC	XDisplayMotionBufferSize
XcmsTekHVCQueryMaxVSamples	XDisplayName
XcmsTekHVCQueryMinV	XDisplayOfIM
XcmsTekHVCToCIEuvY	XDisplayOfScreen
XcmsTekHVCWhiteShiftColors	XDisplayPlanes
XcmsVisualOfCCC	XDisplayString
XConfigureWindow	XDisplayWidth
XConnectionNumber	XDisplayWidthMM
XContextDependentDrawing	XDoesBackingStore
XConvertSelection	XDoesSaveUnders
XCopArea	XDrawArc
XCopColormapAndFree	XDrawArcs
XCopGC	XDrawImageString
XCopPlane	XDrawImageStringl6
XCreateBitmapFromData	XDrawLine
XCreateColormap	XDrawLines
XCreateFontCursor	XDrawPoint
XCreateFontSet	XDrawPoints
XCreateGC	XDrawRectangle
XCreateGlyphCursor	XDrawRectangles
XCreateIC	XDrawSegments
XCreateImage	XDrawString
XCreatePixmap	XDrawStringl6
XCreatePixmapCursor	XDrawText
XCreatePixmapFromBitmapData	XDrawTextl6
XCreateRegion	XEHeadOfExtensionList
XCreateSimpleWindow	XEmptyRegion
XCreateWindow	XEnableAccessControl
XDefaultColormap	XEqualRegion
XDefaultColormapOfScreen	XESetCloseDisplay
XDefaultDepth	XESetCopyGC
XDefaultDepthOfScreen	XESetCreateFont
XDefaultGC	XESetCreateGC
XDefaultGCOfScreen	XESetError
XDefaultRootWindow	XESetErrorString
XDefaultScreen	XESetEventToWire
XDefaultScreenOfDisplay	XESetFlushGC
XDefaultString	XESetFreeFont
XDefaultVisual	XESetFreeGC
XDefaultVisualOfScreen	XESetPrintErrorValues
XDefineCursor	XESetWireToError
XDeleteContext	XESetWireToEvent
XDeleteModifiermapEntry	XEventMaskOfScreen

XEventsQueued	XGetPixel
XExtentsOfFontSet	XGetPointerControl
XFetchBuffer	XGetPointerMapping
XFetchBytes	XGetRGBColormaps
XFetchName	XGetScreenSaver
XFillArc	XGetSelectionOwner
XFillArcs	XGetSizeHints
XFillPolygon	XGetStandardColormap
XFillRectangle	XGetSubImage
XFillRectangles	XGetTextProperty
XFilterEvent	XGetTransientForHint
XFindContext	XGetVisualInfo
XFindOnExtensionList	XGetWindowAttributes
XFlush	XGetWindowProperty
XFlushGC	XGetWMClientMachine
XFontsOfFontSet	XGetWMColormapWindows
XForceScreenSaver	XGetWMHints
XFree	XGetWMIconName
XFreeColormap	XGetWMName
XFreeColors	XGetWMNormalHints
XFreeCursor	XGetWMProtocols
XFreeExtensionList	XGetWMSizeHints
XFreeFont	XGrabButton
XFreeFontInfo	XGrabKey
XFreeFontNames	XGrabKeyboard
XFreeFontPath	XGrabPointer
XFreeFontSet	XGrabServer
XFreeGC	XHeightMMOfScreen
XFreeModifiermap	XHeightOfScreen
XFreePixmap	XIconifyWindow
XFreeStringList	XIfEvent
XGContextFromGC	XImageByteOrder
XGeometry	XIMOfIC
XGetAtomName	XInitExtension
XGetClassHint	XInsertModifiermapEntry
XGetCommand	XInstallColormap
XGetDefault	XInternAtom
XGetErrorDatabaseText	XIntersectRegion
XGetErrorText	XKeycodeToKeysym
XGetFontPath	XKeysymToKeycode
XGetFontProperty	XKeysymToString
XGetGCValues	XKillClient
XGetGeometry	XLastKnownRequestProcessed
XGetIconName	XListDepths
XGetIconSizes	XListExtensions
XGetICValues	XListFonts
XGetImage	XListFontsWithInfo
XGetIMValues	XListHosts
XGetInputFocus	XListInstalledColormaps
XGetKeyboardControl	XListPixmapFormats
XGetKeyboardMapping	XListProperties
XGetModifierMapping	XLoadFont
XGetMotionEvents	XLoadQueryFont
XGetNormalHints	XLocaleOfFontSet

XLocaleOfIM	XQueryExtension
XLookupColor	XQueryFont
XLookupKeysym	XQueryKeymap
XLookupString	XQueryPointer
XLowerWindow	XQueryTextExtents
XMapRaised	XQueryTextExtents16
XMapSubwindows	XQueryTree
XMapWindow	XRaiseWindow
XMaskEvent	XReadBitmapFile
XMatchVisualInfo	XRebindKeysym
XMaxCmapsOfScreen	XRecolorCursor
XMaxRequestSize	XReconfigureWMWindow
XmbDrawImageString	XRectInRegion
XmbDrawString	XRefreshKeyboardMapping
XmbDrawText	XRemoveFromSaveSet
XmbLookupString	XRemoveHost
XmbResetIC	XRemoveHosts
XmbSetWMPProperties	XReparentWindow
XmbTextEscapement	XResetScreenSaver
XmbTextExtents	XResizeWindow
XmbTextListToTextProperty	XResourceManagerString
XmbTextPerCharExtents	XRestackWindows
XmbTextPropertyToTextList	XrmCombineDatabase
XMinCmapsOfScreen	XrmCombineFileDatabase
XMoveResizeWindow	XrmDestroyDatabase
XMoveWindow	XrmEnumerateDatabase
XNewModifiermap	XrmGetDatabase
XNextEvent	XrmGetFileDatabase
XNextRequest	XrmGetResource
XNoOp	XrmGetStringDatabase
XOffsetRegion	XrmInitialize
XOpenDisplay	XrmLocaleOfDatabase
XOpenIM	XrmMergeDatabases
XParseColor	XrmParseCommand
XParseGeometry	XrmPermStringToQuark
XPeekEvent	XrmPutFileDatabase
XPeekIfEvent	XrmPutLineResource
XPending	XrmPutResource
Xpermalloc	XrmPutStringResource
XPlanesOfScreen	XrmQGetResource
XPointInRegion	XrmQGetSearchList
XPolygonRegion	XrmQGetSearchResource
XProtocolRevision	XrmQPutResource
XProtocolVersion	XrmQPutStringResource
XPutBackEvent	XrmQuarkToString
XPutImage	XrmSetDatabase
XPutPixel	XrmStringToBindingQuarkList
XQLength	XrmStringToQuark
XQueryBestCursor	XrmStringToQuarkList
XQueryBestSize	XrmUniqueQuark
XQueryBestStipple	XRootWindow
XQueryBestTile	XRootWindowOfScreen
XQueryColor	XRotateBuffers
XQueryColors	XRotateWindowProperties

XSaveContext	XSetWindowBackgroundPixmap
XScreenCount	XSetWindowBorder
XScreenNumberOfScreen	XSetWindowBorderPixmap
XScreenOfDisplay	XSetWindowBorderWidth
XScreenResourceString	XSetWindowColormap
XSelectInput	XSetWMClientMachine
XSendEvent	XSetWMColormapWindows
XServerVendor	XSetWMHints
XSetAccessControl	XSetWMIconName
XSetAfterFunction	XSetWMName
XSetArcMode	XSetWMNormalHints
XSetBackground	XSetWMProperties
XSetClassHint	XSetWMProtocols
XSetClipMask	XSetWMSizeHints
XSetClipOrigin	XShrinkRegion
XSetClipRectangles	XStoreBuffer
XSetCloseDownMode	XStoreBytes
XSetCommand	XStoreColor
XSetDashes	XStoreColors
XSetErrorHandler	XStoreName
XSetFillRule	XStoreNamedColor
XSetFillStyle	XStringListToTextProperty
XSetFont	XStringToKeysym
XSetFontPath	XSubImage
XSetForeground	XSubtractRegion
XSetFunction	XSupportsLocale
XSetGraphicsExposures	XSync
XSetICFocus	XSynchronize
XSetIconName	XTextExtents
XSetIconSizes	XTextExtentsl6
XSetICValues	XTextPropertyToStringList
XSetInputFocus	XTextWidth
XSetIOErrorHandler	XTextWidthl6
XSetLineAttributes	XTranslateCoordinates
XSetLocaleModifiers	XUndefineCursor
XSetModifierMapping	XUngrabButton
XSetNormalHints	XUngrabKey
XSetPlaneMask	XUngrabKeyboard
XSetPointerMapping	XUngrabPointer
XSetRegion	XUngrabServer
XSetRGBColormaps	XUninstallColormap
XSetScreenSaver	XUnionRectWithRegion
XSetSelectionOwner	XUnionRegion
XSetSizeHints	XUnloadFont
XSetStandardColormap	XUnmapSubwindows
XSetStandardProperties	XUnmapWindow
XSetState	XUnsetICFocus
XSetStipple	XVaCreateNestedList
XSetSubwindowMode	XVendorRelease
XSetTextProperty	XVisualIDFromVisual
XSetTile	XWarpPointer
XSetTransientForHint	XwcDrawImageString
XSetTSTOrigin	XwcDrawString
XSetWindowBackground	XwcDrawText

XwcFreeStringList
XwcLookupString
XwcResetIC
XwcTextEscapement
XwcTextExtents
XwcTextListToTextProperty
XwcTextPerCharExtents
XwcTextPropertyToTextList
XWhitePixel
XWhitePixelOfScreen
XWidthMMOfScreen
XWidthOfScreen
XWindowEvent
XWithdrawWindow
XWMGeometry
XWriteBitmapFile
XXorRegion

Table 8. Exported Data for libX.

```

XcmsCIELabColorSpace[ 0x18 ]
XcmsCIELuvColorSpace[ 0x18 ]
XcmsCIEuvYColorSpace[ 0x18 ]
XcmsCIExyYColorSpace[ 0x18 ]
XcmsCIEXYZColorSpace[ 0x18 ]
XcmsLinearRGBFunctionSet[ ? ]
XcmsRGBColorSpace[ 0x18 ]
XcmsRGBiColorSpace[ 0x18 ]
XcmsTekHVCColorSpace[ 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 9 below lists each of the unsafe macros and its safe function counterpart from the X library.

Table 9: ABI Unsafe Macros

Unsafe Macro	Equivalent X Function
<code>BitmapPad</code>	<code>XBitmapPad</code>
<code>BitmapUnit</code>	<code>XBitmapUnit</code>
<code>BlackPixel</code>	<code>XBlackPixel</code>
<code>BlackPixelOfScreen</code>	<code>XBlackPixelOfScreen</code>
<code>CellsOfScreen</code>	<code>XCellsOfScreen</code>
<code>ClientWhitePointOfCCC</code>	<code>XClientWhitePointOfCCC</code>
<code>ConnectionNumber</code>	<code>XConnectionNumber</code>
<code>DefaultColormap</code>	<code>XDefaultColormap</code>
<code>DefaultColormapOfScreen</code>	<code>XDefaultColormapOfScreen</code>
<code>DefaultDepth</code>	<code>XDefaultDepth</code>
<code>DefaultDepthOfScreen</code>	<code>XDefaultDepthOfScreen</code>
<code>DefaultGC</code>	<code>XDefaultGC</code>
<code>DefaultGCOfScreen</code>	<code>XDefaultGCOfScreen</code>
<code>DefaultRootWindow</code>	<code>XDefaultRootWindow</code>
<code>DefaultScreen</code>	<code>XDefaultScreen</code>
<code>DefaultScreenOfDisplay</code>	<code>XDefaultScreenOfDisplay</code>
<code>DefaultVisual</code>	<code>XDefaultVisual</code>
<code>DefaultVisualOfScreen</code>	<code>XDefaultVisualOfScreen</code>
<code>DisplayCells</code>	<code>XDisplayCells</code>

Table 9: ABI Unsafe Macros

Unsafe Macro	Equivalent X Function
DisplayHeight	XDisplayHeight
DisplayHeightMM	XDisplayHeightMM
DisplayOfCCC	XDisplayOfCCC
DisplayOfScreen	XDisplayOfScreen
DisplayPlanes	XDisplayPlanes
DisplayString	XDisplayString
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

Table 9: ABI Unsafe Macros

Unsafe Macro	Equivalent X Function
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 7.

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/Xatom.h>`,
- `<X11/Xresource.h>`,
- `<X11/Xutil.h>`,
- `<X11/Xcms.h>`,
- `<X11/keysymdef.h>`, and
- `<X11/cursorfont.h>`.

All header definitions are based on X, version 11, release 5 from the MIT X Consortium.

Figure 7. Manifest Constants and Data Types from <X11/Xlib.h>

```
typedef char *XPointer;

#define Bool int
#define Status int
#define True 1
#define False 0

#define QueuedAlready 0
#define QueuedAfterReading 1
#define QueuedAfterFlush 2

#define AllPlanes          ((unsigned long)~0L)

/*
 * Extensions need a way to hang private data on some structures.
 */
typedef struct _XExtData {
    int number; /* number returned by XRegisterExtension */
    struct _XExtData *next; /* next item on list of data for structure */
    int (*free_private)(); /* called to free private storage */
    XPointer private_data; /* data private to this extension. */
} XExtData;

/*
 * This file contains structures used by the extension mechanism.
 */
typedef struct {
    int extension; /* public to extension, cannot be changed */
    int extension_number; /* extension number */
    int major_opcode; /* major op-code assigned by server */
    int first_event; /* first event number for the extension */
    int first_error; /* first error number for the extension */
} XExtCodes;

/*
 * Data structure for retrieving info about pixmap formats.
 */
typedef struct {
    int depth;
    int bits_per_pixel;
    int scanline_pad;
} XPixmapFormatValues;

/*
 * Data structure for setting graphics context.
 */
typedef struct {
    int function; /* logical operation */
    unsigned long plane_mask; /* plane mask */
    unsigned long foreground; /* foreground pixel */
    unsigned long background; /* background pixel */
}
```

```

    int line_width;           /* line width */
    int line_style;           /* LineSolid, LineOnOffDash, LineDoubleDash */
    int cap_style;            /* CapNotLast, CapButt,
                               CapRound, CapProjecting */
    int join_style;           /* JoinMiter, JoinRound, JoinBevel */
    int fill_style;           /* FillSolid, FillTiled,
                               FillStippled, FillOpaeueStippled */
    int fill_rule;            /* EvenOddRule, WindingRule */
    int arc_mode;             /* ArcChord, ArcPieSlice */
    Pixmap tile;              /* tile pixmap for tiling operations */
    Pixmap stipple;           /* stipple 1 plane pixmap for stippling */
    int ts_x_origin;          /* offset for tile or stipple operations */
    int ts_y_origin;
    Font font;                /* default text font for text operations */
    int subwindow_mode;       /* ClipByChildren, IncludeInferiors */
    Bool graphics_exposures; /* boolean, should exposures be generated */
    int clip_x_origin;        /* origin for clipping */
    int clip_y_origin;
    Pixmap clip_mask;         /* bitmap clipping; other calls for rects */
    int dash_offset;          /* patterned/dashed line information */
    char dashes;
} XGCValues;

/*
 * Graphics context. The contents of this structure are implementation
 * dependent. A GC should be treated as opaque by application code.
 */

typedef struct _XGC *GC;

/*
 * Visual structure; contains information about colormapping possible.
 */
typedef struct Visual;

/*
 * Depth structure; contains information for each possible depth.
 */
typedef struct Depth;

/*
 * Information about the screen. The contents of this structure are
 * implementation dependent. A Screen should be treated as opaque
 * by application code.
 */
typedef struct _Screen Screen;

/*
 * Format structure; describes ZFormat data the screen will understand.
 */
typedef struct ScreenFormat;

/*
 * Data structure for setting window attributes.

```

```

*/
typedef struct {
    Pixmap background_pixmap; /* background or None or ParentRelative */
    unsigned long background_pixel; /* background pixel */
    Pixmap border_pixmap; /* border of the window */
    unsigned long border_pixel; /* border pixel value */
    int bit_gravity; /* one of bit gravity values */
    int win_gravity; /* one of the window gravity values */
    int backing_store; /* NotUseful, WhenMapped, Always */
    unsigned long backing_planes; /* planes to be preseved if possible */
    unsigned long backing_pixel; /* value to use in restoring planes */
    Bool save_under; /* should bits under be saved? (popups) */
    long event_mask; /* set of events that should be saved */
    long do_not_propagate_mask; /* set of events that should not propagate */
    Bool override_redirect; /* boolean value for override-redirect */
    Colormap colormap; /* color map to be associated with window */
    Cursor cursor; /* cursor to be displayed (or None) */
} XSetWindowAttributes;

typedef struct {
    int x, y; /* location of window */
    int width, height; /* width and height of window */
    int border_width; /* border width of window */
    int depth; /* depth of window */
    Visual *visual; /* the associated visual structure */
    Window root; /* root of screen containing window */
    int class; /* InputOutput, InputOnly */
    int bit_gravity; /* one of bit gravity values */
    int win_gravity; /* one of the window gravity values */
    int backing_store; /* NotUseful, WhenMapped, Always */
    unsigned long backing_planes; /* planes to be preserved if possible */
    unsigned long backing_pixel; /* value to be used when restoring planes */
    Bool save_under; /* boolean, should bits under be saved? */
    Colormap colormap; /* color map to be associated with window */
    Bool map_installed; /* boolean, is color map currently installed */
    int map_state; /* IsUnmapped, IsUnviewable, IsViewable */
    long all_event_masks; /* set of events all people have interest in */
    long your_event_mask; /* my event mask */
    long do_not_propagate_mask; /* set of events that should not propagate */
    Bool override_redirect; /* boolean value for override-redirect */
    Screen *screen; /* back pointer to correct screen */
} XWindowAttributes;

/*
 * Data structure for host setting; getting routines.
 */

typedef struct XHostAddress;

/*
 * Data structure for "image" data, used by image manipulation routines.
 */
typedef struct _XImage {

```

```

    int width, height;          /* size of image */
    int xoffset;                /* number of pixels offset in X direction */
    int format;                 /* XYBitmap, XYPixmap, ZPixmap */
    char *data;                 /* pointer to image data */
    int byte_order;              /* data byte order, LSBFirst, MSBFirst */
    int bitmap_unit;             /* quant. of scanline 8, 16, 32 */
    int bitmap_bit_order;        /* LSBFirst, MSBFirst */
    int bitmap_pad;              /* 8, 16, 32 either XY or ZPixmap */
    int depth;                   /* depth of image */
    int bytes_per_line;          /* accelerator to next line */
    int bits_per_pixel;          /* bits per pixel (ZPixmap) */
    unsigned long red_mask;       /* bits in z arrangment */
    unsigned long green_mask;
    unsigned long blue_mask;
    XPointer obdata;             /* hook for the object routines to hang on */
    struct funcs {                /* image manipulation routines */
        struct _XImage *(*create_image)();
        int (*destroy_image)();
        unsigned long (*get_pixel)();
        int (*put_pixel)();
        struct _XImage *(*sub_image)();
        int (*add_pixel)();
    } f;
} XImage;

/*
 * Data structure for XReconfigureWindow
 */
typedef struct {
    int x, y;
    int width, height;
    int border_width;
    Window sibling;
    int stack_mode;
} XWindowChanges;

/*
 * Data structure used by color operations
 */
typedef struct {
    unsigned long pixel;
    unsigned short red, green, blue;
    char flags; /* do_red, do_green, do_blue */
    char pad;
} XColor;

/*
 * Data structures for graphics operations. On most machines, these are
 * congruent with the wire protocol structures, so reformatting the data
 * can be avoided on these architectures.
 */
typedef struct {
    short x1, y1, x2, y2;
} XSegment;

```

```

typedef struct {
    short x, y;
} XPoint;

typedef struct {
    short x, y;
    unsigned short width, height;
} XRectangle;

typedef struct {
    short x, y;
    unsigned short width, height;
    short angle1, angle2;
} XArc;

/* Data structure for XChangeKeyboardControl */

typedef struct {
    int key_click_percent;
    int bell_percent;
    int bell_pitch;
    int bell_duration;
    int led;
    int led_mode;
    int key;
    int auto_repeat_mode; /* On, Off, Default */
} XKeyboardControl;

/* Data structure for XGetKeyboardControl */

typedef struct {
    int key_click_percent;
    int bell_percent;
    unsigned int bell_pitch, bell_duration;
    unsigned long led_mask;
    int global_auto_repeat;
    char auto_repeats[32];
} XKeyboardState;

/* Data structure for XGetMotionEvents. */

typedef struct {
    Time time;
    short x, y;
} XTimeCoord;

/* Data structure for X{Set,Get}ModifierMapping */

typedef struct {
    int max_keypermod; /* The server's max # of keys per modifier */
    KeyCode *modifiermap; /* An 8 by max_keypermod array of modifiers */
} XModifierKeymap;

```

```

/*
 * Display datatype maintaining display specific data.
 * The contents of this structure are implementation dependent.
 * A Display should be treated as opaque by application code.
 */
typedef struct _XDisplay Display;

/*
 * A "XEvent" structure always has type as the first entry. This
 * uniquely identifies what kind of event it is. The second entry
 * is always a pointer to the display the event was read from.
 * The third entry is always a window of one type or another,
 * carefully selected to be useful to toolkit dispatchers. (Except
 * for keymap events, which have no window.) You
 * must not change the order of the three elements or toolkits will
 * break! The pointer to the generic event must be cast before use to
 * access any other information in the structure.
 */

/*
 * Definitions of specific events.
 */
typedef struct {
    int type;                /* of event */
    unsigned long serial;    /* # of last request processed by server */
    Bool send_event;        /* true if this came from a SendEvent request */
    Display *display;       /* Display the event was read from */
    Window window;         /* "event" window it is reported relative to */
    Window root;           /* root window that the event occurred on */
    Window subwindow;      /* child window */
    Time time;             /* milliseconds */
    int x, y;              /* pointer x, y coordinates in event window */
    int x_root, y_root;    /* coordinates relative to root */
    unsigned int state;     /* key or button mask */
    unsigned int keycode;   /* detail */
    Bool same_screen;      /* same screen flag */
} XKeyEvent;
typedef XKeyEvent XKeyPressedEvent;
typedef XKeyEvent XKeyReleasedEvent;

typedef struct {
    int type;                /* of event */
    unsigned long serial;    /* # of last request processed by server */
    Bool send_event;        /* true if this came from a SendEvent request */
    Display *display;       /* Display the event was read from */
    Window window;         /* "event" window it is reported relative to */
    Window root;           /* root window that the event occurred on */
    Window subwindow;      /* child window */
    Time time;             /* milliseconds */
    int x, y;              /* pointer x, y coordinates in event window */
    int x_root, y_root;    /* coordinates relative to root */
    unsigned int state;     /* key or button mask */
    unsigned int button;    /* detail */
    Bool same_screen;      /* same screen flag */
}

```

```

} XButtonEvent;
typedef XButtonEvent XButtonPressedEvent;
typedef XButtonEvent XButtonReleasedEvent;

typedef struct {
    int type;                /* of event */
    unsigned long serial;    /* # of last request processed by server */
    Bool send_event;        /* true if this came from a SendEvent request */
    Display *display;        /* Display the event was read from */
    Window window;          /* "event" window reported relative to */
    Window root;            /* root window that the event occurred on */
    Window subwindow;       /* child window */
    Time time;              /* milliseconds */
    int x, y;               /* pointer x, y coordinates in event window */
    int x_root, y_root;     /* coordinates relative to root */
    unsigned int state;     /* key or button mask */
    char is_hint;           /* detail */
    Bool same_screen;       /* same screen flag */
} XMotionEvent;
typedef XMotionEvent XPointerMovedEvent;

typedef struct {
    int type;                /* of event */
    unsigned long serial;    /* # of last request processed by server */
    Bool send_event;        /* true if this came from a SendEvent request */
    Display *display;        /* Display the event was read from */
    Window window;          /* "event" window reported relative to */
    Window root;            /* root window that the event occurred on */
    Window subwindow;       /* child window */
    Time time;              /* milliseconds */
    int x, y;               /* pointer x, y coordinates in event window */
    int x_root, y_root;     /* coordinates relative to root */
    int mode;               /* NotifyNormal, NotifyGrab, NotifyUngrab */
    int detail;
    /*
     * NotifyAncestor, NotifyVirtual, NotifyInferior,
     * NotifyNonLinear, NotifyNonLinearVirtual
     */
    Bool same_screen;       /* same screen flag */
    Bool focus;             /* boolean focus */
    unsigned int state;     /* key or button mask */
} XCrossingEvent;
typedef XCrossingEvent XEnterWindowEvent;
typedef XCrossingEvent XLeaveWindowEvent;

typedef struct {
    int type;                /* FocusIn or FocusOut */
    unsigned long serial;    /* # of last request processed by server */
    Bool send_event;        /* true if this came from a SendEvent request */
    Display *display;        /* Display the event was read from */
    Window window;          /* window of event */
    int mode;               /* NotifyNormal, NotifyGrab, NotifyUngrab */
    int detail;
    /*

```

```

        * NotifyAncestor, NotifyVirtual, NotifyInferior,
        * NotifyNonLinear, NotifyNonLinearVirtual, NotifyPointer,
        * NotifyPointerRoot, NotifyDetailNone
    */
} XFocusChangeEvent;
typedef XFocusChangeEvent XFocusInEvent;
typedef XFocusChangeEvent XFocusOutEvent;

/* generated on EnterWindow and FocusIn when KeyMapState selected */
typedef struct {
    int type;
    unsigned long serial; /* # of last request processed by server */
    Bool send_event; /* true if this came from a SendEvent request */
    Display *display; /* Display the event was read from */
    Window window;
    char key_vector[32];
} XKeymapEvent;

typedef struct {
    int type;
    unsigned long serial; /* # of last request processed by server */
    Bool send_event; /* true if this came from a SendEvent request */
    Display *display; /* Display the event was read from */
    Window window;
    int x, y;
    int width, height;
    int count; /* if non-zero, at least this many more */
} XExposeEvent;

typedef struct {
    int type;
    unsigned long serial; /* # of last request processed by server */
    Bool send_event; /* true if this came from a SendEvent request */
    Display *display; /* Display the event was read from */
    Drawable drawable;
    int x, y;
    int width, height;
    int count; /* if non-zero, at least this many more */
    int major_code; /* core is CopyArea or CopyPlane */
    int minor_code; /* not defined in the core */
} XGraphicsExposeEvent;

typedef struct {
    int type;
    unsigned long serial; /* # of last request processed by server */
    Bool send_event; /* true if this came from a SendEvent request */
    Display *display; /* Display the event was read from */
    Drawable drawable;
    int major_code; /* core is CopyArea or CopyPlane */
    int minor_code; /* not defined in the core */
} XNoExposeEvent;

typedef struct {
    int type;

```

```

        unsigned long serial;    /* # of last request processed by server */
        Bool send_event;        /* true if this came from a SendEvent request */
        Display *display;       /* Display the event was read from */
        Window window;
        int state;              /* Visibility state */
    } XVisibilityEvent;

typedef struct {
    int type;
    unsigned long serial;    /* # of last request processed by server */
    Bool send_event;        /* true if this came from a SendEvent request */
    Display *display;       /* Display the event was read from */
    Window parent;          /* parent of the window */
    Window window;          /* window id of window created */
    int x, y;               /* window location */
    int width, height;      /* size of window */
    int border_width;       /* border width */
    Bool override_redirect; /* creation should be overridden */
} XCreateWindowEvent;

typedef struct {
    int type;
    unsigned long serial;    /* # of last request processed by server */
    Bool send_event;        /* true if this came from a SendEvent request */
    Display *display;       /* Display the event was read from */
    Window event;
    Window window;
} XDestroyWindowEvent;

typedef struct {
    int type;
    unsigned long serial;    /* # of last request processed by server */
    Bool send_event;        /* true if this came from a SendEvent request */
    Display *display;       /* Display the event was read from */
    Window event;
    Window window;
    Bool from_configure;
} XUnmapEvent;

typedef struct {
    int type;
    unsigned long serial;    /* # of last request processed by server */
    Bool send_event;        /* true if this came from a SendEvent request */
    Display *display;       /* Display the event was read from */
    Window event;
    Window window;
    Bool override_redirect; /* boolean, is override set... */
} XMapEvent;

typedef struct {
    int type;
    unsigned long serial;    /* # of last request processed by server */
    Bool send_event;        /* true if this came from a SendEvent request */
    Display *display;       /* Display the event was read from */

```

```

        Window parent;
        Window window;
    } XMapRequestEvent;

typedef struct {
    int type;
    unsigned long serial; /* # of last request processed by server */
    Bool send_event;      /* true if this came from a SendEvent request */
    Display *display;      /* Display the event was read from */
    Window event;
    Window window;
    Window parent;
    int x, y;
    Bool override_redirect;
} XReparentEvent;

typedef struct {
    int type;
    unsigned long serial; /* # of last request processed by server */
    Bool send_event;      /* true if this came from a SendEvent request */
    Display *display;      /* Display the event was read from */
    Window event;
    Window window;
    int x, y;
    int width, height;
    int border_width;
    Window above;
    Bool override_redirect;
} XConfigureEvent;

typedef struct {
    int type;
    unsigned long serial; /* # of last request processed by server */
    Bool send_event;      /* true if this came from a SendEvent request */
    Display *display;      /* Display the event was read from */
    Window event;
    Window window;
    int x, y;
} XGravityEvent;

typedef struct {
    int type;
    unsigned long serial; /* # of last request processed by server */
    Bool send_event;      /* true if this came from a SendEvent request */
    Display *display;      /* Display the event was read from */
    Window window;
    int width, height;
} XResizeRequestEvent;

typedef struct {
    int type;
    unsigned long serial; /* # of last request processed by server */
    Bool send_event;      /* true if this came from a SendEvent request */
    Display *display;      /* Display the event was read from */

```

```

    Window parent;
    Window window;
    int x, y;
    int width, height;
    int border_width;
    Window above;
    int detail;          /* Above, Below, TopIf, BottomIf, Opposite */
    unsigned long value_mask;
} XConfigureRequestEvent;

typedef struct {
    int type;
    unsigned long serial; /* # of last request processed by server */
    Bool send_event;      /* true if this came from a SendEvent request */
    Display *display;      /* Display the event was read from */
    Window event;
    Window window;
    int place;             /* PlaceOnTop, PlaceOnBottom */
} XCirculateEvent;

typedef struct {
    int type;
    unsigned long serial; /* # of last request processed by server */
    Bool send_event;      /* true if this came from a SendEvent request */
    Display *display;      /* Display the event was read from */
    Window parent;
    Window window;
    int place;             /* PlaceOnTop, PlaceOnBottom */
} XCirculateRequestEvent;

typedef struct {
    int type;
    unsigned long serial; /* # of last request processed by server */
    Bool send_event;      /* true if this came from a SendEvent request */
    Display *display;      /* Display the event was read from */
    Window window;
    Atom atom;
    Time time;
    int state;             /* NewValue, Deleted */
} XPropertyEvent;

typedef struct {
    int type;
    unsigned long serial; /* # of last request processed by server */
    Bool send_event;      /* true if this came from a SendEvent request */
    Display *display;      /* Display the event was read from */
    Window window;
    Atom selection;
    Time time;
} XSelectionClearEvent;

typedef struct {
    int type;
    unsigned long serial; /* # of last request processed by server */

```

```

        Bool send_event;          /* true if this came from a SendEvent request */
        Display *display;        /* Display the event was read from */
        Window owner;
        Window requestor;
        Atom selection;
        Atom target;
        Atom property;
        Time time;
    } XSelectionRequestEvent;

typedef struct {
    int type;
    unsigned long serial;        /* # of last request processed by server */
    Bool send_event;            /* true if this came from a SendEvent request */
    Display *display;            /* Display the event was read from */
    Window requestor;
    Atom selection;
    Atom target;
    Atom property;              /* ATOM or None */
    Time time;
} XSelectionEvent;

typedef struct {
    int type;
    unsigned long serial;        /* # of last request processed by server */
    Bool send_event;            /* true if this came from a SendEvent request */
    Display *display;            /* Display the event was read from */
    Window window;
    Colormap colormap;          /* COLORMAP or None */
    Bool new;
    int state;                  /* ColormapInstalled, ColormapUninstalled */
} XColormapEvent;

typedef struct {
    int type;
    unsigned long serial;        /* # of last request processed by server */
    Bool send_event;            /* true if this came from a SendEvent request */
    Display *display;            /* Display the event was read from */
    Window window;
    Atom message_type;
    int format;
    union {
        char b[20];
        short s[10];
        long l[5];
    } data;
} XClientMessageEvent;

typedef struct {
    int type;
    unsigned long serial;        /* # of last request processed by server */
    Bool send_event;            /* true if this came from a SendEvent request */
    Display *display;            /* Display the event was read from */
    Window window;              /* unused */
}

```

```

        int request;           /* one of MappingModifier, MappingKeyboard,
                                MappingPointer */
        int first_keycode;     /* first keycode */
        int count;             /* defines range of change w. first_keycode*/
    } XMappingEvent;

typedef struct {
    int type;
    Display *display;          /* Display the event was read from */
    XID resourceid;            /* resource id */
    unsigned long serial;      /* serial number of failed request */
    unsigned char error_code;   /* error code of failed request */
    unsigned char request_code; /* Major op-code of failed request */
    unsigned char minor_code;   /* Minor op-code of failed request */
} XErrorEvent;

typedef struct {
    int type;
    unsigned long serial;      /* # of last request processed by server */
    Bool send_event;           /* true if this came from a SendEvent request */
    Display *display;          /* Display the event was read from */
    Window window;             /* window on which event was requested in event mask */
} XAnyEvent;

/*
 * this union is defined so Xlib can always use the same sized
 * event structure internally, to avoid memory fragmentation.
 */
typedef union _XEvent {
    int type;                  /* must not be changed; first element */
    XAnyEvent xany;
    XKeyEvent xkey;
    XButtonEvent xbutton;
    XMotionEvent xmotion;
    XCrossingEvent xcrossing;
    XFocusChangeEvent xfocus;
    XExposeEvent xexpose;
    XGraphicsExposeEvent xgraphicsexpose;
    XNoExposeEvent xnoexpose;
    XVisibilityEvent xvisibility;
    XCreateWindowEvent xcreatewindow;
    XDestroyWindowEvent xdestroywindow;
    XUnmapEvent xunmap;
    XMapEvent xmap;
    XMapRequestEvent xmaprequest;
    XReparentEvent xreparent;
    XConfigureEvent xconfigure;
    XGravityEvent xgravity;
    XResizeRequestEvent xresizerequest;
    XConfigureRequestEvent xconfigurerequest;
    XCirculateEvent xcirculate;
    XCirculateRequestEvent xcirculaterequest;
    XPropertyEvent xproperty;
    XSelectionClearEvent xselectionclear;

```

```

        XSelectionRequestEvent xselectionrequest;
        XSelectionEvent xselection;
        XColormapEvent xcolormap;
        XClientMessageEvent xclient;
        XMappingEvent xmapping;
        XErrorEvent xerror;
        XKeymapEvent xkeymap;
        long pad[24];
    } XEvent;

/*
 * per character font metric information.
 */
typedef struct {
    short        lbearing;        /* origin to left edge of raster */
    short        rbearing;        /* origin to right edge of raster */
    short        width;           /* advance to next char's origin */
    short        ascent;          /* baseline to top edge of raster */
    short        descent;         /* baseline to bottom edge of raster */
    unsigned short attributes;    /* per char flags (not predefined) */
} XCharStruct;

/*
 * To allow arbitrary information with fonts, there are additional properties
 * returned.
 */
typedef struct {
    Atom name;
    unsigned long card32;
} XFontProp;

typedef struct {
    XExtData      *ext_data;      /* hook for extension to hang data */
    Font          fid;            /* Font id for this font */
    unsigned       direction;     /* hint about direction the font is painted */
    unsigned       min_char_or_byte2; /* first character */
    unsigned       max_char_or_byte2; /* last character */
    unsigned       min_bytel;     /* first row that exists */
    unsigned       max_bytel;     /* last row that exists */
    Bool          all_chars_exist; /* flag if all characters have non-zero size */
    unsigned       default_char;  /* char to print for undefined character */
    int           n_properties;   /* how many properties there are */
    XFontProp     *properties;    /* pointer to array of additional properties */
    XCharStruct    min_bounds;    /* minimum bounds over all existing char */
    XCharStruct    max_bounds;    /* maximum bounds over all existing char */
    XCharStruct    *per_char;     /* first_char to last_char information */
    int           ascent;         /* log. extent above baseline for spacing */
    int           descent;        /* log. descent below baseline for spacing */
} XFontStruct;

/*
 * PolyText routines take these as arguments.
 */
typedef struct {

```

```

    char *chars;                /* pointer to string */
    int nchars;                 /* number of characters */
    int delta;                  /* delta between strings */
    Font font;                  /* font to print it in, None don't change */
} XTextItem;

typedef struct {                /* normal 16 bit characters are two bytes */
    unsigned char byte1;
    unsigned char byte2;
} XChar2b;

typedef struct {
    XChar2b *chars;             /* two byte characters */
    int nchars;                 /* number of characters */
    int delta;                  /* delta between strings */
    Font font;                  /* font to print it in, None don't change */
} XTextItem16;

typedef union { Display *display;
                GC gc;
                Visual *visual;
                Screen *screen;
                ScreenFormat *pixmap_format;
                XFontStruct *font; } XDataObject;

typedef struct {
    XRectangle      max_ink_extent;
    XRectangle      max_logical_extent;
} XFontSetExtents;

typedef struct _XFontSet *XFontSet;

typedef struct {
    char            *chars;
    int             nchars;
    int             delta;
    XFontSet        font_set;
} XmbTextItem;

typedef struct {
    wchar_t         *chars;
    int             nchars;
    int             delta;
    XFontSet        font_set;
} XwcTextItem;

typedef void (*XIMProc)();

typedef struct _XIM *XIM;
typedef struct _XIC *XIC;

typedef unsigned long XIMStyle;

typedef struct {

```

```

    unsigned short count_styles;
    XIMStyle *supported_styles;
} XIMStyles;

#define XIMPreeditArea          0x0001L
#define XIMPreeditCallbacks     0x0002L
#define XIMPreeditPosition     0x0004L
#define XIMPreeditNothing      0x0008L
#define XIMPreeditNone         0x0010L
#define XIMStatusArea          0x0100L
#define XIMStatusArea          0x0100L
#define XIMStatusCallbacks     0x0200L
#define XIMStatusNothing       0x0400L
#define XIMStatusNone          0x0800L

#define XNVaNestedList "XNVaNestedList"
#define XNQueryInputStyle "queryInputStyle"
#define XNClientWindow "clientWindow"
#define XNInputStyle "inputStyle"
#define XNFocusWindow "focusWindow"
#define XNResourceName "resourceName"
#define XNResourceClass "resourceClass"
#define XNGeometryCallback "geometryCallback"
#define XNFilterEvents "filterEvents"
#define XNPreeditStartCallback "preeditStartCallback"
#define XNPreeditDoneCallback "preeditDoneCallback"
#define XNPreeditDrawCallback "preeditDrawCallback"
#define XNPreeditCaretCallback "preeditCaretCallback"
#define XNPreeditAttributes "preeditAttributes"
#define XNStatusStartCallback "statusStartCallback"
#define XNStatusDoneCallback "statusDoneCallback"
#define XNStatusDrawCallback "statusDrawCallback"
#define XNStatusAttributes "statusAttributes"
#define XNArea "area"
#define XNAreaNeeded "areaNeeded"
#define XNSpotLocation "spotLocation"
#define XNColormap "colorMap"
#define XNStdColormap "stdColorMap"
#define XNForeground "foreground"
#define XNBackground "background"
#define XNBackgroundPixmap "backgroundPixmap"
#define XNFontSet "fontSet"
#define XNLineSpace "lineSpace"
#define XNCursor "cursor"
#define XBufferOverflow -1
#define XLookupNone 1
#define XLookupChars 2
#define XLookupKeySym 3
#define XLookupBoth 4

typedef XPointer XVaNestedList;

typedef struct {
    XPointer client_data;

```

```

    XIMProc callback;
} XIMCallback;

typedef unsigned long XIMFeedback;

#define XIMReverse      1
#define XIMUnderline    (1<<1)
#define XIMHighlight    (1<<2)
#define XIMPrimary      (1<<5)
#define XIMSecondary    (1<<6)
#define XIMTertiary     (1<<7)

typedef struct _XIMText {
    unsigned short length;
    XIMFeedback *feedback;
    Bool encoding_is_wchar;
    union {
        char *multi_byte;
        wchar_t *wide_char;
    } string;
} XIMText;

typedef struct _XIMPreeditDrawCallbackStruct {
    int caret;           /* Cursor offset within pre-edit string */
    int chg_first;       /* Starting change position */
    int chg_length;      /* Length of the change in character count */
    XIMText *text;
} XIMPreeditDrawCallbackStruct;

typedef enum {
    XIMForwardChar, XIMBackwardChar,
    XIMForwardWord, XIMBackwardWord,
    XIMCaretUp, XIMCaretDown,
    XIMNextLine, XIMPreviousLine,
    XIMLineStart, XIMLineEnd,
    XIMAbsolutePosition,
    XIMDontChange
} XIMCaretDirection;

typedef enum {
    XIMIsInvisible,      /* Disable caret feedback */
    XIMIsPrimary,        /* UI defined caret feedback */
    XIMIsSecondary       /* UI defined caret feedback */
} XIMCaretStyle;

typedef struct _XIMPreeditCaretCallbackStruct {
    int position;         /* Caret offset within pre-edit string */
    XIMCaretDirection direction; /* Caret moves direction */
    XIMCaretStyle style;  /* Feedback of the caret */
} XIMPreeditCaretCallbackStruct;

typedef enum {
    XIMTextType,
    XIMBitmapType

```

```
} XIMStatusDataType;

typedef struct _XIMStatusDrawCallbackStruct {
    XIMStatusDataType type;
    union {
        XIMText *text;
        Pixmap bitmap;
    } data;
} XIMStatusDrawCallbackStruct;
```

Figure 8. Manifest Constants and Data Types from <X11/X.h>

```

#define X_PROTOCOL      11                /* current protocol version */
#define X_PROTOCOL_REVISION 0            /* current minor version */

/* Resources */

typedef unsigned long XID;

typedef XID Window;
typedef XID Drawable;
typedef XID Font;
typedef XID Pixmap;
typedef XID Cursor;
typedef XID Colormap;
typedef XID GCContext;
typedef XID KeySym;

typedef unsigned long Mask;

typedef unsigned long Atom;

typedef unsigned long VisualID;

typedef unsigned long Time;

typedef unsigned char KeyCode;

/*****
 * RESERVED RESOURCE AND CONSTANT DEFINITIONS
 *****/

#define None            0L /* universal null resource or null atom */

#define ParentRelative  1L /* background pixmap in CreateWindow
                           and ChangeWindowAttributes */

#define CopyFromParent  0L /* border pixmap in CreateWindow
                           and ChangeWindowAttributes
                           special VisualID and special window
                           class passed to CreateWindow */

#define PointerWindow    0L /* destination window in SendEvent */
#define InputFocus      1L /* destination window in SendEvent */

#define PointerRoot      1L /* focus window in SetInputFocus */

#define AnyPropertyType  0L /* special Atom, passed to GetProperty */

#define AnyKey           0L /* special Key Code, passed to GrabKey */

#define AnyButton        0L /* special Button Code, passed to GrabButton */

```

```

#define AllTemporary          0L /* special Resource ID passed to KillClient */

#define CurrentTime           0L /* special Time */

#define NoSymbol              0L /* special KeySym */

/*****
 * EVENT DEFINITIONS
 *****/

/* Input Event Masks. Used as event-mask window attribute and as arguments
   to Grab requests. Not to be confused with event names. */

#define NoEventMask           0L
#define KeyPressMask          (1L<<0)
#define KeyReleaseMask        (1L<<1)
#define ButtonPressMask       (1L<<2)
#define ButtonReleaseMask     (1L<<3)
#define EnterWindowMask       (1L<<4)
#define LeaveWindowMask       (1L<<5)
#define PointerMotionMask     (1L<<6)
#define PointerMotionHintMask (1L<<7)
#define Button1MotionMask     (1L<<8)
#define Button2MotionMask     (1L<<9)
#define Button3MotionMask     (1L<<10)
#define Button4MotionMask     (1L<<11)
#define Button5MotionMask     (1L<<12)
#define ButtonMotionMask      (1L<<13)
#define KeymapStateMask       (1L<<14)
#define ExposureMask          (1L<<15)
#define VisibilityChangeMask  (1L<<16)
#define StructureNotifyMask   (1L<<17)
#define ResizeRedirectMask    (1L<<18)
#define SubstructureNotifyMask (1L<<19)
#define SubstructureRedirectMask (1L<<20)
#define FocusChangeMask       (1L<<21)
#define PropertyChangeMask    (1L<<22)
#define ColormapChangeMask    (1L<<23)
#define OwnerGrabButtonMask   (1L<<24)

/* Event names. Used in "type" field in XEvent structures. Not to be
   confused with event masks above. They start from 2 because 0 and 1
   are reserved in the protocol for errors and replies. */

#define KeyPress               2
#define KeyRelease             3
#define ButtonPress            4
#define ButtonRelease          5
#define MotionNotify           6
#define EnterNotify            7
#define LeaveNotify            8
#define FocusIn                9
#define FocusOut               10
#define KeymapNotify           11

```

```

#define Expose                12
#define GraphicsExpose        13
#define NoExpose              14
#define VisibilityNotify      15
#define CreateNotify          16
#define DestroyNotify         17
#define UnmapNotify           18
#define MapNotify             19
#define MapRequest            20
#define ReparentNotify        21
#define ConfigureNotify       22
#define ConfigureRequest      23
#define GravityNotify         24
#define ResizeRequest         25
#define CirculateNotify       26
#define CirculateRequest      27
#define PropertyNotify        28
#define SelectionClear        29
#define SelectionRequest      30
#define SelectionNotify       31
#define ColormapNotify        32
#define ClientMessage         33
#define MappingNotify         34
#define LASTEvent             35      /* must be bigger than any event # */

```

```

/* Key masks. Used as modifiers to GrabButton and GrabKey, results of
   QueryPointer, state in various key-, mouse-, and button-related events. */

```

```

#define ShiftMask             (1<<0)
#define LockMask              (1<<1)
#define ControlMask           (1<<2)
#define Mod1Mask              (1<<3)
#define Mod2Mask              (1<<4)
#define Mod3Mask              (1<<5)
#define Mod4Mask              (1<<6)
#define Mod5Mask              (1<<7)

```

```

/* modifier names. Used to build a SetModifierMapping request or
   to read a GetModifierMapping request. These correspond to the
   masks defined above. */

```

```

#define ShiftMapIndex         0
#define LockMapIndex          1
#define ControlMapIndex       2
#define Mod1MapIndex          3
#define Mod2MapIndex          4
#define Mod3MapIndex          5
#define Mod4MapIndex          6
#define Mod5MapIndex          7

```

```

/* button masks. Used in same manner as Key masks above. Not to be confused
   with button names below. */

```

```

#define Button1Mask            (1<<8)

```

```
#define Button2Mask          (1<<9)
#define Button3Mask          (1<<10)
#define Button4Mask          (1<<11)
#define Button5Mask          (1<<12)

#define AnyModifier          (1<<15)  /* used in GrabButton, GrabKey */

/* button names. Used as arguments to GrabButton and as detail in ButtonPress
   and ButtonRelease events. Not to be confused with button masks above.
   Note that 0 is already defined above as "AnyButton". */

#define Button1              1
#define Button2              2
#define Button3              3
#define Button4              4
#define Button5              5

/* Notify modes */

#define NotifyNormal          0
#define NotifyGrab           1
#define NotifyUngrab         2
#define NotifyWhileGrabbed   3

#define NotifyHint            1        /* for MotionNotify events */

/* Notify detail */

#define NotifyAncestor        0
#define NotifyVirtual         1
#define NotifyInferior        2
#define NotifyNonlinear       3
#define NotifyNonlinearVirtual 4
#define NotifyPointer         5
#define NotifyPointerRoot     6
#define NotifyDetailNone      7

/* Visibility notify */

#define VisibilityUnobscured   0
#define VisibilityPartiallyObscured 1
#define VisibilityFullyObscured 2

/* Circulation request */

#define PlaceOnTop             0
#define PlaceOnBottom         1

/* Property notification */

#define PropertyNewValue       0
#define PropertyDelete         1
```

```

/* Color Map notification */

#define ColormapUninstalled    0
#define ColormapInstalled     1

/* GrabPointer, GrabButton, GrabKeyboard, GrabKey Modes */

#define GrabModeSync          0
#define GrabModeAsync         1

/* GrabPointer, GrabKeyboard reply status */

#define GrabSuccess            0
#define AlreadyGrabbed         1
#define GrabInvalidTime       2
#define GrabNotViewable       3
#define GrabFrozen             4

/* AllowEvents modes */

#define AsyncPointer           0
#define SyncPointer            1
#define ReplayPointer          2
#define AsyncKeyboard          3
#define SyncKeyboard           4
#define ReplayKeyboard         5
#define AsyncBoth              6
#define SyncBoth               7

/* Used in SetInputFocus, GetInputFocus */

#define RevertToNone           (int)None
#define RevertToPointerRoot    (int)PointerRoot
#define RevertToParent         2

/*****
 * ERROR CODES
 *****/

#define Success                0    /* everything's okay */
#define BadRequest             1    /* bad request code */
#define BadValue               2    /* int parameter out of range */
#define BadWindow              3    /* parameter not a Window */
#define BadPixmap              4    /* parameter not a Pixmap */
#define BadAtom                5    /* parameter not an Atom */
#define BadCursor              6    /* parameter not a Cursor */
#define BadFont                7    /* parameter not a Font */
#define BadMatch               8    /* parameter mismatch */
#define BadDrawable            9    /* parameter not a Pixmap or Window */
#define BadAccess              10   /* depending on context:
    - key/button already grabbed
    - attempt to free an illegal
      cmap entry
    - attempt to store into a read-only

```

```

        color map entry.
        - attempt to modify the access control
          list from other than the local host.
    */
#define BadAlloc      11    /* insufficient resources */
#define BadColor      12    /* no such colormap */
#define BadGC         13    /* parameter not a GC */
#define BadIDChoice   14    /* choice not in range or already used */
#define BadName       15    /* font or color name doesn't exist */
#define BadLength     16    /* Request length incorrect */
#define BadImplementation 17 /* server is defective */

#define FirstExtensionError 128
#define LastExtensionError 255

/*****
 * WINDOW DEFINITIONS
 *****/

/* Window classes used by CreateWindow */
/* Note that CopyFromParent is already defined as 0 above */

#define InputOutput      1
#define InputOnly        2

/* Window attributes for CreateWindow and ChangeWindowAttributes */

#define CWBackPixmap      (1L<<0)
#define CWBackPixel       (1L<<1)
#define CWBorderPixmap    (1L<<2)
#define CWBorderPixel     (1L<<3)
#define CWBitGravity      (1L<<4)
#define CWWinGravity       (1L<<5)
#define CWBackingStore     (1L<<6)
#define CWBackingPlanes   (1L<<7)
#define CWBackingPixel     (1L<<8)
#define CWOVERRIDERedirect (1L<<9)
#define CWSaveUnder       (1L<<10)
#define CWEventMask        (1L<<11)
#define CWDontPropagate    (1L<<12)
#define CWColormap         (1L<<13)
#define CWCursor           (1L<<14)

/* ConfigureWindow structure */

#define CWX                (1<<0)
#define CWY                (1<<1)
#define CWWidth            (1<<2)
#define CWHeight           (1<<3)
#define CWBorderWidth      (1<<4)
#define CWSibling          (1<<5)
#define CWStackMode        (1<<6)

```

```

/* Bit Gravity */

#define ForgetGravity          0
#define NorthWestGravity      1
#define NorthGravity          2
#define NorthEastGravity      3
#define WestGravity           4
#define CenterGravity          5
#define EastGravity           6
#define SouthWestGravity      7
#define SouthGravity          8
#define SouthEastGravity      9
#define StaticGravity         10

/* Window gravity + bit gravity above */

#define UnmapGravity          0

/* Used in CreateWindow for backing-store hint */

#define NotUseful              0
#define WhenMapped            1
#define Always                 2

/* Used in GetWindowAttributes reply */

#define IsUnmapped             0
#define IsUnviewable          1
#define IsViewable             2

/* Used in ChangeSaveSet */

#define SetModeInsert          0
#define SetModeDelete          1

/* Used in ChangeCloseDownMode */

#define DestroyAll              0
#define RetainPermanent        1
#define RetainTemporary        2

/* Window stacking method (in configureWindow) */

#define Above                   0
#define Below                   1
#define TopIf                   2
#define BottomIf                3
#define Opposite                4

/* Circulation direction */

#define RaiseLowest             0
#define LowerHighest            1

```

```

/* Property modes */

#define PropModeReplace      0
#define PropModePrepend     1
#define PropModeAppend      2

/*****
 * GRAPHICS DEFINITIONS
 *****/

/* graphics functions, as in GC.alu */

#define GXclear              0x0      /* 0 */
#define GXand                0x1      /* src AND dst */
#define GXandReverse         0x2      /* src AND NOT dst */
#define GXcopy               0x3      /* src */
#define GXandInverted         0x4      /* NOT src AND dst */
#define GXnoop               0x5      /* dst */
#define GXxor                 0x6      /* src XOR dst */
#define GXor                  0x7      /* src OR dst */
#define GXnor                 0x8      /* NOT src AND NOT dst */
#define GXequiv               0x9      /* NOT src XOR dst */
#define GXinvert              0xa      /* NOT dst */
#define GXorReverse           0xb      /* src OR NOT dst */
#define GXcopyInverted        0xc      /* NOT src */
#define GXorInverted           0xd      /* NOT src OR dst */
#define GXnand                0xe      /* NOT src OR NOT dst */
#define GXset                  0xf      /* 1 */

/* LineStyle */

#define LineSolid              0
#define LineOnOffDash          1
#define LineDoubleDash         2

/* capStyle */

#define CapNotLast              0
#define CapButt                 1
#define CapRound                 2
#define CapProjecting            3

/* joinStyle */

#define JoinMiter                0
#define JoinRound                1
#define JoinBevel                 2

/* fillStyle */

#define FillSolid                0
#define FillTiled                 1
#define FillStippled              2
#define FillOpaqueStippled        3

```

```

/* fillRule */

#define EvenOddRule          0
#define WindingRule          1

/* subwindow mode */

#define ClipByChildren        0
#define IncludeInferiors      1

/* SetClipRectangles ordering */

#define Unsorted              0
#define YSorted               1
#define YXSorted              2
#define YXBanded              3

/* CoordinateMode for drawing routines */

#define CoordModeOrigin       0      /* relative to the origin */
#define CoordModePrevious     1      /* relative to previous point */

/* Polygon shapes */

#define Complex                0      /* paths may intersect */
#define Nonconvex              1      /* no paths intersect, but not convex */
#define Convex                 2      /* wholly convex */

/* Arc modes for PolyFillArc */

#define ArcChord               0      /* join endpoints of arc */
#define ArcPieSlice            1      /* join endpoints to center of arc */

/* GC components: masks used in CreateGC, CopyGC, ChangeGC, OR'ed into
   GC.stateChanges */

#define GCFunction              (1L<<0)
#define GCPlaneMask            (1L<<1)
#define GCForeground            (1L<<2)
#define GCBackground           (1L<<3)
#define GCLineWidth            (1L<<4)
#define GCLineStyle             (1L<<5)
#define GCCapStyle              (1L<<6)
#define GCJoinStyle            (1L<<7)
#define GCFillStyle            (1L<<8)
#define GCFillRule             (1L<<9)
#define GCTile                  (1L<<10)
#define GCStipple               (1L<<11)
#define GCTileStipXOrigin       (1L<<12)
#define GCTileStipYOrigin       (1L<<13)
#define GCFont                  (1L<<14)
#define GCSubwindowMode         (1L<<15)
#define GCGraphicsExposures     (1L<<16)

```

```

#define GCclipXOrigin      (1L<<17)
#define GCclipYOrigin      (1L<<18)
#define GCclipMask         (1L<<19)
#define GCDashOffset       (1L<<20)
#define GCDashList         (1L<<21)
#define GCArcMode          (1L<<22)

#define GCLastBit           22
/*****
 * FONTS
 *****/

/* used in QueryFont -- draw direction */

#define FontLeftToRight     0
#define FontRightToLeft    1

#define FontChange          255

/*****
 * IMAGING
 *****/

/* ImageFormat -- PutImage, GetImage */

#define XYBitmap            0      /* depth 1, XYFormat */
#define XYPixmap            1      /* depth == drawable depth */
#define ZPixmap             2      /* depth == drawable depth */

/*****
 * COLOR MAP STUFF
 *****/

/* For CreateColormap */

#define AllocNone           0      /* create map with no entries */
#define AllocAll            1      /* allocate entire map writeable */

/* Flags used in StoreNamedColor, StoreColors */

#define DoRed               (1<<0)
#define DoGreen            (1<<1)
#define DoBlue             (1<<2)

/*****
 * CURSOR STUFF
 *****/

/* QueryBestSize Class */

#define CursorShape         0      /* largest size that can be displayed */
#define TileShape           1      /* size tiled fastest */
#define StippleShape        2      /* size stippled fastest */

```

```

/*****
 * KEYBOARD/POINTER STUFF
 *****/

#define AutoRepeatModeOff      0
#define AutoRepeatModeOn      1
#define AutoRepeatModeDefault  2

#define LedModeOff             0
#define LedModeOn              1

/* masks for ChangeKeyboardControl */

#define KBKeyClickPercent      (1L<<0)
#define KBBellPercent          (1L<<1)
#define KBBellPitch            (1L<<2)
#define KBBellDuration         (1L<<3)
#define KBLed                  (1L<<4)
#define KBLedMode              (1L<<5)
#define KBKey                   (1L<<6)
#define KBAutoRepeatMode       (1L<<7)

#define MappingSuccess          0
#define MappingBusy             1
#define MappingFailed           2

#define MappingModifier         0
#define MappingKeyboard         1
#define MappingPointer          2

/*****
 * SCREEN SAVER STUFF
 *****/

#define DontPreferBlanking      0
#define PreferBlanking          1
#define DefaultBlanking         2

#define DisableScreenSaver      0
#define DisableScreenInterval   0

#define DontAllowExposures      0
#define AllowExposures          1
#define DefaultExposures        2

/* for ForceScreenSaver */

#define ScreenSaverReset 0
#define ScreenSaverActive 1

/*****
 * HOSTS AND CONNECTIONS
 *****/

```

```
/* for ChangeHosts */

#define HostInsert          0
#define HostDelete         1

/* for ChangeAccessControl */

#define EnableAccess        1
#define DisableAccess       0

/* Display classes used in opening the connection
 * Note that the statically allocated ones are even numbered and the
 * dynamically changeable ones are odd numbered */

#define StaticGray          0
#define GrayScale           1
#define StaticColor         2
#define PseudoColor         3
#define TrueColor           4
#define DirectColor         5

/* Byte order used in imageByteOrder and bitmapBitOrder */

#define LSBFirst             0
#define MSBFirst            1
```

Figure 9. Manifest Constants from <X11/Xatom.h>

```

/* THIS IS A GENERATED FILE
 *
 * Do not change!  Changing this file implies a protocol change!
 */

#define XA_PRIMARY ((Atom) 1)
#define XA_SECONDARY ((Atom) 2)
#define XA_ARC ((Atom) 3)
#define XA_ATOM ((Atom) 4)
#define XA_BITMAP ((Atom) 5)
#define XA_CARDINAL ((Atom) 6)
#define XA_COLORMAP ((Atom) 7)
#define XA_CURSOR ((Atom) 8)
#define XA_CUT_BUFFER0 ((Atom) 9)
#define XA_CUT_BUFFER1 ((Atom) 10)
#define XA_CUT_BUFFER2 ((Atom) 11)
#define XA_CUT_BUFFER3 ((Atom) 12)
#define XA_CUT_BUFFER4 ((Atom) 13)
#define XA_CUT_BUFFER5 ((Atom) 14)
#define XA_CUT_BUFFER6 ((Atom) 15)
#define XA_CUT_BUFFER7 ((Atom) 16)
#define XA_DRAWABLE ((Atom) 17)
#define XA_FONT ((Atom) 18)
#define XA_INTEGER ((Atom) 19)
#define XA_PIXMAP ((Atom) 20)
#define XA_POINT ((Atom) 21)
#define XA_RECTANGLE ((Atom) 22)
#define XA_RESOURCE_MANAGER ((Atom) 23)
#define XA_RGB_COLOR_MAP ((Atom) 24)
#define XA_RGB_BEST_MAP ((Atom) 25)
#define XA_RGB_BLUE_MAP ((Atom) 26)
#define XA_RGB_DEFAULT_MAP ((Atom) 27)
#define XA_RGB_GRAY_MAP ((Atom) 28)
#define XA_RGB_GREEN_MAP ((Atom) 29)
#define XA_RGB_RED_MAP ((Atom) 30)
#define XA_STRING ((Atom) 31)
#define XA_VISUALID ((Atom) 32)
#define XA_WINDOW ((Atom) 33)
#define XA_WM_COMMAND ((Atom) 34)
#define XA_WM_HINTS ((Atom) 35)
#define XA_WM_CLIENT_MACHINE ((Atom) 36)
#define XA_WM_ICON_NAME ((Atom) 37)
#define XA_WM_ICON_SIZE ((Atom) 38)
#define XA_WM_NAME ((Atom) 39)
#define XA_WM_NORMAL_HINTS ((Atom) 40)
#define XA_WM_SIZE_HINTS ((Atom) 41)
#define XA_WM_ZOOM_HINTS ((Atom) 42)
#define XA_MIN_SPACE ((Atom) 43)
#define XA_NORM_SPACE ((Atom) 44)
#define XA_MAX_SPACE ((Atom) 45)
#define XA_END_SPACE ((Atom) 46)

```

```
#define XA_SUPERSCRIPT_X ((Atom) 47)
#define XA_SUPERSCRIPT_Y ((Atom) 48)
#define XA_SUBSCRIPT_X ((Atom) 49)
#define XA_SUBSCRIPT_Y ((Atom) 50)
#define XA_UNDERLINE_POSITION ((Atom) 51)
#define XA_UNDERLINE_THICKNESS ((Atom) 52)
#define XA_STRIKEOUT_ASCENT ((Atom) 53)
#define XA_STRIKEOUT_DESCENT ((Atom) 54)
#define XA_ITALIC_ANGLE ((Atom) 55)
#define XA_X_HEIGHT ((Atom) 56)
#define XA_QUAD_WIDTH ((Atom) 57)
#define XA_WEIGHT ((Atom) 58)
#define XA_POINT_SIZE ((Atom) 59)
#define XA_RESOLUTION ((Atom) 60)
#define XA_COPYRIGHT ((Atom) 61)
#define XA_NOTICE ((Atom) 62)
#define XA_FONT_NAME ((Atom) 63)
#define XA_FAMILY_NAME ((Atom) 64)
#define XA_FULL_NAME ((Atom) 65)
#define XA_CAP_HEIGHT ((Atom) 66)
#define XA_WM_CLASS ((Atom) 67)
#define XA_WM_TRANSIENT_FOR ((Atom) 68)

#define XA_LAST_PREDEFINED ((Atom) 68)
```

Figure 10. Manifest Constants and Data Types from <X11/Xresource.h>

```

/*****
 *
 * Quark Management
 *
 *****/

typedef int      XrmQuark, *XrmQuarkList;
#define NULLQUARK ((XrmQuark) 0)

typedef char *XrmString;
#define NULLSTRING ((XrmString) 0)

/*****
 *
 * Conversion of Strings to Lists
 *
 *****/

typedef enum {XrmBindTightly, XrmBindLoosely} XrmBinding, *XrmBindingList;

/*****
 *
 * Name and Class lists.
 *
 *****/

typedef XrmQuark      XrmName;
typedef XrmQuarkList  XrmNameList;

typedef XrmQuark      XrmClass;
typedef XrmQuarkList  XrmClassList;

/*****
 *
 * Resource Representation Types and Values
 *
 *****/

typedef XrmQuark      XrmRepresentation;

typedef struct {
    unsigned int      size;
    caddr_t           addr;
} XrmValue, *XrmValuePtr;

/*****
 *
 * Resource Manager Functions
 *
 *****/

```

```
typedef struct _XrmHashBucketRec *XrmHashBucket;
typedef XrmHashBucket *XrmHashTable;
typedef XrmHashTable XrmSearchList[];
typedef struct _XrmHashBucketRec *XrmDatabase;

/*****
 *
 * Resource Database Management
 *
 *****/

#define XrmEnumAllLevels 0
#define XrmEnumOneLevel 1

/*****
 *
 * Command line option mapping to resource entries
 *
 *****/

typedef enum {
    XrmoptionNoArg,      /* Value is specified in OptionDescRec.value */
    XrmoptionIsArg,      /* Value is the option string itself */
    XrmoptionStickyArg,  /* Value is characters immediately following option */
    XrmoptionSepArg,     /* Value is next argument in argv */
    XrmoptionResArg,     /* Resource and value in next argument in argv */
    XrmoptionSkipArg,    /* Ignore this option and the next argument in argv */
    XrmoptionSkipLine,   /* Ignore this option and the rest of argv */
    XrmoptionSkipNArgs   /* Ignore this option and the next
                        OptionDescRes.value arguments in argv */
} XrmOptionKind;

typedef struct {
    char      *option;      /* Option abbreviation in argv */
    char      *specifier;   /* Resource specifier */
    XrmOptionKind argKind;   /* Which style of option it is */
    caddr_t   value;        /* Value to provide if XrmoptionNoArg */
} XrmOptionDescRec, *XrmOptionDescList;
```

Figure 11. Manifest Constants and Data Types from <X11/Xutil.h>

```

/*
 * Bitmask returned by XParseGeometry(). Each bit tells if the corresponding
 * value (x, y, width, height) was found in the parsed string.
 */
#define NoValue          0x0000
#define XValue           0x0001
#define YValue           0x0002
#define WidthValue       0x0004
#define HeightValue      0x0008
#define AllValues        0x000F
#define XNegative        0x0010
#define YNegative        0x0020

/*
 * new version containing base_width, base_height, and win_gravity fields;
 * used with WM_NORMAL_HINTS.
 */
typedef struct {
    long flags;          /* marks which fields in this structure are defined */
    int x, y;            /* obsolete for new window mgrs, but clients */
    int width, height;   /* should set so old wm's don't mess up */
    int min_width, min_height;
    int max_width, max_height;
    int width_inc, height_inc;
    struct {
        int x; /* numerator */
        int y; /* denominator */
    } min_aspect, max_aspect;
    int base_width, base_height; /* added by ICCCM version 1 */
    int win_gravity; /* added by ICCCM version 1 */
} XSizeHints;

/*
 * The next block of definitions are for window manager properties that
 * clients and applications use for communication.
 */

/* flags argument in size hints */
#define USPosition      (1L << 0) /* user specified x, y */
#define USSize          (1L << 1) /* user specified width, height */

#define PPosition       (1L << 2) /* program specified position */
#define PSize           (1L << 3) /* program specified size */
#define PMinSize        (1L << 4) /* program specified minimum size */
#define PMaxSize        (1L << 5) /* program specified maximum size */
#define PResizeInc      (1L << 6) /* program specified resize increments */
#define PAspect         (1L << 7) /* program specified min and max aspect ratios */
#define PBaseSize       (1L << 8) /* program specified base for incrementing */
#define PWinGravity      (1L << 9) /* program specified window gravity */

/* obsolete */

```

```
#define PAllHints (PPosition|PSize|PMinSize|PMaxSize|PResizeInc|PAspect)

typedef struct {
    long flags;          /* marks which fields in this structure are defined */
    Bool input;          /* does this application rely on the window manager to
                           get keyboard input? */
    int initial_state;    /* see below */
    Pixmap icon_pixmap;   /* pixmap to be used as icon */
    Window icon_window;    /* window to be used as icon */
    int icon_x, icon_y;   /* initial position of icon */
    Pixmap icon_mask;     /* icon mask bitmap */
    XID window_group;     /* id of related window group */
    /* this structure may be extended in the future */
} XWMHints;

/* definition for flags of XWMHints */

#define InputHint          (1L << 0)
#define StateHint          (1L << 1)
#define IconPixmapHint     (1L << 2)
#define IconWindowHint     (1L << 3)
#define IconPositionHint   (1L << 4)
#define IconMaskHint       (1L << 5)
#define WindowGroupHint    (1L << 6)
#define AllHints (InputHint|StateHint|IconPixmapHint|IconWindowHint| \
IconPositionHint|IconMaskHint|WindowGroupHint)

/* definitions for initial window state */
#define WithdrawnState 0    /* for windows that are not mapped */
#define NormalState 1      /* most applications want to start this way */
#define IconicState 3      /* application wants to start as an icon */

/*
 * new structure for manipulating TEXT properties; used with WM_NAME,
 * WM_ICON_NAME, WM_CLIENT_MACHINE, and WM_COMMAND.
 */
typedef struct {
    unsigned char *value;    /* same as Property routines */
    Atom encoding;          /* prop type */
    int format;             /* prop data format: 8, 16, or 32 */
    unsigned long nitems;    /* number of data items in value */
} XTextProperty;

#define XNoMemory -1
#define XLocaleNotSupported -2
#define XConverterNotFound -3

typedef enum {
    XStringStyle,          /* STRING */
    XCompoundTextStyle,    /* COMPOUND_TEXT */
    XTextStyle,            /* text in owner's encoding (current locale) */
    XStdICCTextStyle       /* STRING, else COMPOUND_TEXT */
} XICCEncodingStyle;
```

```

typedef struct {
    int min_width, min_height;
    int max_width, max_height;
    int width_inc, height_inc;
} XIconSize;

typedef struct {
    char *res_name;
    char *res_class;
} XClassHint;

/*
 * Compose sequence status structure, used in calling XLookupString.
 */
typedef struct _XComposeStatus {
    char *compose_ptr;          /* state table pointer */
    int chars_matched;          /* match state */
} XComposeStatus;

/*
 * opaque reference to Region data type
 */
typedef struct _XRegion *Region;

/* Return values from XRectInRegion() */

#define RectangleOut 0
#define RectangleIn 1
#define RectanglePart 2

/*
 * Information used by the visual utility routines to find desired visual
 * type from the many visuals a display may support.
 */

typedef struct {
    Visual *visual;
    VisualID visualid;
    int screen;
    int depth;
    int class;
    unsigned long red_mask;
    unsigned long green_mask;
    unsigned long blue_mask;
    int colormap_size;
    int bits_per_rgb;
} XVisualInfo;

#define VisualNoMask          0x0
#define VisualIDMask          0x1
#define VisualScreenMask      0x2
#define VisualDepthMask       0x4
#define VisualClassMask       0x8

```

```

#define VisualRedMaskMask      0x10
#define VisualGreenMaskMask    0x20
#define VisualBlueMaskMask     0x40
#define VisualColormapSizeMask 0x80
#define VisualBitsPerRGBMask   0x100
#define VisualAllMask          0x1FF

/*
 * This defines a window manager property that clients may use to
 * share standard color maps of type RGB_COLOR_MAP:
 */
typedef struct {
    Colormap colormap;
    unsigned long red_max;
    unsigned long red_mult;
    unsigned long green_max;
    unsigned long green_mult;
    unsigned long blue_max;
    unsigned long blue_mult;
    unsigned long base_pixel;
    VisualID visualid;           /* added by ICCCM version 1 */
    XID killid;                 /* added by ICCCM version 1 */
} XStandardColormap;

#define ReleaseByFreeingColormap ((XID) 1L) /* for killid field above */

/*
 * return codes for XReadBitmapFile and XWriteBitmapFile
 */
#define BitmapSuccess          0
#define BitmapOpenFailed      1
#define BitmapFileInvalid     2
#define BitmapNoMemory        3

/*
 * Declare the routines that don't return int.
 */

/*****
 *
 * Context Management
 *
 *****/

/* Associative lookup table return codes */

#define XCSUCCESS 0      /* No error. */
#define XCNOMEM  1      /* Out of memory */
#define XCNOENT  2      /* No entry in table */

typedef int XContext;

```

Figure 12. Manifest Constants and Data Types from <X11/Xcms.h>

```

/*
 * XCMS Status Values
 */
#define XcmsFailure                0
#define XcmsSuccess                1
#define XcmsSuccessWithCompression 2

/*
 * Color Space Format ID's
 *   Color Space ID's are of XcmsColorFormat type, which is an
 *   unsigned short (16 bits).
 *
 *   bit 15 (most significant bit):
 *       0 == Device-Independent
 *       1 == Device-Dependent
 *
 *   bit 14:
 *       0 == Registered with X Consortium
 *       1 == Unregistered
 */
#define XcmsUndefinedFormat        (XcmsColorFormat)0x00000000
#define XcmsCIEXYZFormat           (XcmsColorFormat)0x00000001
#define XcmsCIEuvYFormat           (XcmsColorFormat)0x00000002
#define XcmsCIExyYFormat           (XcmsColorFormat)0x00000003
#define XcmsCIELabFormat           (XcmsColorFormat)0x00000004
#define XcmsCIELuvFormat           (XcmsColorFormat)0x00000005
#define XcmsTekHVCFormat           (XcmsColorFormat)0x00000006
#define XcmsRGBFormat              (XcmsColorFormat)0x80000000
#define XcmsRGBiFormat             (XcmsColorFormat)0x80000001

/*
 * State of XcmsPerScrnInfo
 */
#define XcmsInitNone                0x00    /* no initialization attempted */
#define XcmsInitSuccess             0x01    /* initialization successful */
#define XcmsInitDefault             0xff    /* initialization failed */

typedef unsigned int XcmsColorFormat;    /* Color Space Format ID */

typedef double XcmsFloat;

/*
 * Device RGB
 */
typedef struct {
    unsigned short red;                /* scaled from 0x0000 to 0xffff */
    unsigned short green;              /* scaled from 0x0000 to 0xffff */
    unsigned short blue;               /* scaled from 0x0000 to 0xffff */
} XcmsRGB;

/*

```

```

        * RGB Intensity
        */
typedef struct {
    XcmsFloat red;          /* 0.0 - 1.0 */
    XcmsFloat green;        /* 0.0 - 1.0 */
    XcmsFloat blue;         /* 0.0 - 1.0 */
} XcmsRGBi;

/*
 * CIE XYZ
 */
typedef struct {
    XcmsFloat X;
    XcmsFloat Y;
    XcmsFloat Z;
} XcmsCIEXYZ;

/*
 * CIE u'v'Y
 */
typedef struct {
    XcmsFloat u_prime;      /* 0.0 - 1.0 */
    XcmsFloat v_prime;      /* 0.0 - 1.0 */
    XcmsFloat Y;            /* 0.0 - 1.0 */
} XcmsCIEuvY;

/*
 * CIE xyY
 */
typedef struct {
    XcmsFloat x;            /* 0.0 - 1.0 */
    XcmsFloat y;            /* 0.0 - 1.0 */
    XcmsFloat Y;            /* 0.0 - 1.0 */
} XcmsCIExyY;

/*
 * CIE L*a*b*
 */
typedef struct {
    XcmsFloat L_star;       /* 0.0 - 100.0 */
    XcmsFloat a_star;
    XcmsFloat b_star;
} XcmsCIELab;

/*
 * CIE L*u*v*
 */
typedef struct {
    XcmsFloat L_star;       /* 0.0 - 100.0 */
    XcmsFloat u_star;
    XcmsFloat v_star;
} XcmsCIELuv;

/*

```

```

        * TekHVC
        */
typedef struct {
    XcmsFloat H;          /* 0.0 - 360.0 */
    XcmsFloat V;          /* 0.0 - 100.0 */
    XcmsFloat C;          /* 0.0 - 100.0 */
} XcmsTekHVC;

/*
 * PAD
 */
typedef struct {
    XcmsFloat pad0;
    XcmsFloat pad1;
    XcmsFloat pad2;
    XcmsFloat pad3;
} XcmsPad;

/*
 * XCMS Color Structure
 */
typedef struct {
    union {
        XcmsRGB RGB;
        XcmsRGBi RGBi;
        XcmsCIEXYZ CIEXYZ;
        XcmsCIEuvY CIEuvY;
        XcmsCIExyY CIExyY;
        XcmsCIELab CIELab;
        XcmsCIELuv CIELuv;
        XcmsTekHVC TekHVC;
        XcmsPad Pad;
    } spec;                /* the color specification */
    unsigned long pixel;    /* pixel value (as needed) */
    XcmsColorFormat format; /* the specification format */
} XcmsColor;

/*
 * XCMS Per Screen related data
 */

typedef struct _XcmsPerScrnInfo {
    XcmsColor screenWhitePt; /* Screen White point */
    XPointer functionSet;    /* pointer to Screen Color Characterization */
                                /* Function Set structure */
    XPointer screenData;    /* pointer to corresponding Screen Color */
                                /* Characterization Data */
    unsigned char state;    /* XcmsInitNone, XcmsInitSuccess, XcmsInitDefault */
    char pad[3];
} XcmsPerScrnInfo;

typedef struct _XcmsCCC *XcmsCCC;

```

```

typedef Status (*XcmsCompressionProc)();

typedef Status (*XcmsWhiteAdjustProc)();

/*
 * XCMS Color Conversion Context
 */
typedef struct _XcmsCCC {
    Display      *dpy;                /* X Display */
    int          screenNumber;        /* X screen number */
    Visual       *visual;             /* X Visual */
    XcmsColor    clientWhitePt;       /* Client White Point */
    XcmsCompressionProc gamutCompProc; /* Gamut Compression Function */
    XPointer     gamutCompClientData; /* Gamut Comp Func Client Data */
    XcmsWhiteAdjustProc whitePtAdjProc; /* White Point Adjustment Function */
    XPointer     whitePtAdjClientData; /* White Pt Adj Func Client Data */
    XcmsPerScrnInfo *pPerScrnInfo;    /* pointer to per screen information */
                                           /* associated with the above display */
                                           /* screenNumber */
} XcmsCCCRec;

typedef Status (*XcmsScreenInitProc)();

typedef void (*XcmsScreenFreeProc)();

/*
 * Function List Pointer -- pointer to an array of function pointers.
 * The end of list is indicated by a NULL pointer.
 */
typedef Status (*XcmsConversionProc)();
typedef XcmsConversionProc *XcmsFuncListPtr;

typedef int (*XcmsParseStringProc)();

/*
 * Color Space -- per Color Space related data (Device-Independent
 * or Device-Dependent)
 */
typedef struct _XcmsColorSpace {
    char *prefix;                    /* Prefix of string format. */
    XcmsColorFormat id;              /* Format ID number. */
    XcmsParseStringProc parseString; /* String format parsing function */
    XcmsFuncListPtr to_CIEXYZ;       /* Pointer to an array of function */
                                           /* pointers such that when the */
                                           /* functions are executed in sequence */
                                           /* will convert a XcmsColor structure */
                                           /* from this color space to CIEXYZ */
                                           /* space. */
    XcmsFuncListPtr from_CIEXYZ;     /* Pointer to an array of function */
                                           /* pointers such that when the */
                                           /* functions are executed in sequence */
                                           /* will convert a XcmsColor structure */

```

```

/* from CIEXYZ space to this color */
/* space. */
int inverse_flag; /* If 1, indicates that for 0 <= i < n */
/* where n is the number of function */
/* pointers in the lists to_CIEXYZ */
/* and from_CIEXYZ; for each function */
/* to_CIEXYZ[i] its inverse function */
/* is from_CIEXYZ[n - i]. */

} XcmsColorSpace;

/*
 * Screen Color Characterization Function Set -- per device class
 * color space conversion functions.
 */
typedef struct _XcmsFunctionSet {
    XcmsColorSpace **DDColorSpaces;
    /* Pointer to an array of pointers to */
    /* Device-DEPENDENT color spaces */
    /* understood by this SCCFuncSet. */
    XcmsScreenInitProc screenInitProc;
    /* Screen initialization function that */
    /* reads Screen Color Characterization*/
    /* Data off properties on the screen's*/
    /* root window. */
    XcmsScreenFreeProc screenFreeProc;
    /* Function that frees the SCCData */
    /* structures. */
} XcmsFunctionSet;

```

Figure 13. Manifest Constants from <X11/keysymdef.h>

```
#define XK_VoidSymbol          0xFFFFF      /* void symbol */

#ifdef XK_MISCELLANY
/*
 * TTY Functions, cleverly chosen to map to ascii, for convenience of
 * programming, but could have been arbitrary (at the cost of lookup
 * tables in client code.
 */

#define XK_BackSpace          0xFF08      /* back space, back char */
#define XK_Tab                0xFF09
#define XK_Linefeed           0xFF0A      /* Linefeed, LF */
#define XK_Clear              0xFF0B
#define XK_Return             0xFF0D      /* Return, enter */
#define XK_Pause              0xFF13      /* Pause, hold */
#define XK_Scroll_Lock        0xFF14
#define XK_Escape             0xFF1B
#define XK_Delete             0xFFFF      /* Delete, rubout */

/* International & multi-key character composition */

#define XK_Multi_key          0xFF20      /* Multi-key character compose */

/* Japanese keyboard support */

#define XK_Kanji              0xFF21      /* Kanji, Kanji convert */
#define XK_Muhenkan           0xFF22      /* Cancel Conversion */
#define XK_Henkan_Mode        0xFF23      /* Start/Stop Conversion */
#define XK_Henkan             0xFF23      /* Alias for Henkan_Mode */
#define XK_Romaji             0xFF24      /* to Romaji */
#define XK_Hiragana           0xFF25      /* to Hiragana */
#define XK_Katakana           0xFF26      /* to Katakana */
#define XK_Hiragana_Katakana  0xFF27      /* Hiragana/Katakana toggle */
#define XK_Zenkaku            0xFF28      /* to Zenkaku */
#define XK_Hankaku           0xFF29      /* to Hankaku */
#define XK_Zenkaku_Hankaku    0xFF2A      /* Zenkaku/Hankaku toggle */
#define XK_Touroku            0xFF2B      /* Add to Dictionary */
#define XK_Massyo             0xFF2C      /* Delete from Dictionary */
#define XK_Kana_Lock          0xFF2D      /* Kana Lock */
#define XK_Kana_Shift         0xFF2E      /* Kana Shift */
#define XK_Eisu_Shift         0xFF2F      /* Alphanumeric Shift */
#define XK_Eisu_toggle        0xFF30      /* Alphanumeric toggle */

/* Cursor control & motion */

#define XK_Home               0xFF50
#define XK_Left               0xFF51      /* Move left, left arrow */
#define XK_Up                 0xFF52      /* Move up, up arrow */
#define XK_Right              0xFF53      /* Move right, right arrow */
```

```

#define XK_Down          0xFF54 /* Move down, down arrow */
#define XK_Prior         0xFF55 /* Prior, previous */
#define XK_Next          0xFF56 /* Next */
#define XK_End           0xFF57 /* EOL */
#define XK_Begin        0xFF58 /* BOL */

/* Misc Functions */

#define XK_Select        0xFF60 /* Select, mark */
#define XK_Print         0xFF61
#define XK_Execute       0xFF62 /* Execute, run, do */
#define XK_Insert        0xFF63 /* Insert, insert here */
#define XK_Undo          0xFF65 /* Undo, oops */
#define XK_Redo          0xFF66 /* redo, again */
#define XK_Menu          0xFF67
#define XK_Find          0xFF68 /* Find, search */
#define XK_Cancel        0xFF69 /* Cancel, stop, abort, exit */
#define XK_Help          0xFF6A /* Help, ? */
#define XK_Break         0xFF6B
#define XK_Mode_switch   0xFF7E /* Character set switch */
#define XK_script_switch 0xFF7E /* Alias for mode_switch */
#define XK_Num_Lock      0xFF7F

/* Keypad Functions, keypad numbers cleverly chosen to map to ascii */

#define XK_KP_Space      0xFF80 /* space */
#define XK_KP_Tab        0xFF89
#define XK_KP_Enter      0xFF8D /* enter */
#define XK_KP_F1         0xFF91 /* PF1, KP_A, ... */
#define XK_KP_F2         0xFF92
#define XK_KP_F3         0xFF93
#define XK_KP_F4         0xFF94
#define XK_KP_Equal      0xFFBD /* equals */
#define XK_KP_Multiply    0xFFAA
#define XK_KP_Add         0xFFAB
#define XK_KP_Separator   0xFFAC /* separator, often comma */
#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

```

```

/*
 * Auxilliary Functions; note the duplicate definitions for left and right
 * function keys; Sun keyboards and a few other manufactures have such
 * function key groups on the left and/or right sides of the keyboard.
 * We've not found a keyboard with more than 35 function keys total.
 */

#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 /* Left shift */
#define XK_Shift_R            0xFFE2 /* Right shift */
#define XK_Control_L          0xFFE3 /* Left control */
#define XK_Control_R          0xFFE4 /* Right control */
#define XK_Caps_Lock          0xFFE5 /* Caps lock */
#define XK_Shift_Lock         0xFFE6 /* Shift lock */

#define XK_Meta_L             0xFFE7 /* Left meta */
#define XK_Meta_R             0xFFE8 /* Right meta */
#define XK_Alt_L              0xFFE9 /* Left alt */
#define XK_Alt_R              0xFFEA /* Right alt */
#define XK_Super_L            0xFFEB /* Left super */
#define XK_Super_R            0xFFEC /* Right super */
#define XK_Hyper_L            0xFFED /* Left hyper */
#define XK_Hyper_R            0xFFEE /* Right hyper */
#endif /* XK_MISCELLANY */

/*
 * Latin 1
 * Byte 3 = 0
 */
#ifdef 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 /* left angle quotation mark */
#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 /* right angle quotation mark */

```

```

#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
 */

#ifdef 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_emacon                0x3ba
#define XK_gcedilla              0x3bb
#define XK_tslash                0x3bc
#define XK_ENG                   0x3bd
#define XK_eng                    0x3bf
#define XK_Amacron               0x3c0
#define XK_Iogonek               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 /* Alias for mode_switch */
#endif /* XK_KATAKANA */

/*
 * Arabic
 * Byte 3 = 5
 */

#ifdef 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_alefmaksura        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 /* Alias for mode_switch */
#endif /* XK_ARABIC */

/*
 * Cyrillic
 * Byte 3 = 6
 */
#ifdef XK_CYRILLIC
#define XK_Serbian_dje                0x6a1
#define XK_Macedonia_gje             0x6a2
#define XK_Cyrillic_io                0x6a3
#define XK_Ukrainian_ie              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
 */

#ifdef 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_IOTAdiaeresis 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 /* Alias for mode_switch */
#endif /* XK_GREEK */

/*
 * Technical
 * Byte 3 = 8
 */

#ifdef 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_greaterthanequal        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_leftt                 0x9f4
#define XK_rightt                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                0aaa
#define XK_signifblank           0aac
#define XK_ellipsis              0aae
#define XK_doubbaselinedot       0aaf
#define XK_onethird              0ab0
#define XK_twothirds             0ab1
#define XK_onefifth              0ab2
#define XK_twofifths             0ab3
#define XK_threefifths           0ab4
#define XK_fourfifths            0ab5
#define XK_onesixth              0ab6
#define XK_fivesixths            0ab7
#define XK_careof                0ab8
#define XK_figdash               0abb
#define XK_leftanglebracket      0abc
#define XK_decimalpoint          0abd
#define XK_rightanglebracket     0abe
#define XK_marker                0abf
#define XK_oneeighth             0ac3
#define XK_threeeighths          0ac4
#define XK_fiveeighths           0ac5
#define XK_seveneighths          0ac6
#define XK_trademark             0ac9

```

```

#define XK_signaturemark 0xaca
#define XK_trademarkincircle 0xacb
#define XK_leftopentriangle 0xacc
#define XK_rightopentriangle 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_enopencircbullet 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 0xae
#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        0bcf
#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 /* Alias for mode_switch */
```

Figure 14. Manifest Constants from <X11/cursorfont.h>

```
/* $XConsortium: cursorfont.h,v 1.2 88/09/06 16:44:27 jim Exp $ */
#define XC_num_glyphs 154
#define XC_X_cursor 0
#define XC_arrow 2
#define XC_based_arrow_down 4
#define XC_based_arrow_up 6
#define XC_boat 8
#define XC_bogosity 10
#define XC_bottom_left_corner 12
#define XC_bottom_right_corner 14
#define XC_bottom_side 16
#define XC_bottom_tee 18
#define XC_box_spiral 20
#define XC_center_ptr 22
#define XC_circle 24
#define XC_clock 26
#define XC_coffee_mug 28
#define XC_cross 30
#define XC_cross_reverse 32
#define XC_crosshair 34
#define XC_diamond_cross 36
#define XC_dot 38
#define XC_dotbox 40
#define XC_double_arrow 42
#define XC_draft_large 44
#define XC_draft_small 46
#define XC_draped_box 48
#define XC_exchange 50
#define XC_fleur 52
#define XC_gobbler 54
#define XC_gumby 56
#define XC_hand1 58
#define XC_hand2 60
#define XC_heart 62
#define XC_icon 64
#define XC_iron_cross 66
#define XC_left_ptr 68
#define XC_left_side 70
#define XC_left_tee 72
#define XC_leftbutton 74
#define XC_ll_angle 76
#define XC_lr_angle 78
#define XC_man 80
#define XC_middlebutton 82
#define XC_mouse 84
#define XC_pencil 86
#define XC_pirate 88
#define XC_plus 90
#define XC_question_arrow 92
#define XC_right_ptr 94
#define XC_right_side 96
```

```
#define XC_right_tee 98
#define XC_rightbutton 100
#define XC_rtl_logo 102
#define XC_sailboat 104
#define XC_sb_down_arrow 106
#define XC_sb_h_double_arrow 108
#define XC_sb_left_arrow 110
#define XC_sb_right_arrow 112
#define XC_sb_up_arrow 114
#define XC_sb_v_double_arrow 116
#define XC_shuttle 118
#define XC_sizing 120
#define XC_spider 122
#define XC_spraycan 124
#define XC_star 126
#define XC_target 128
#define XC_tcross 130
#define XC_top_left_arrow 132
#define XC_top_left_corner 134
#define XC_top_right_corner 136
#define XC_top_side 138
#define XC_top_tee 140
#define XC_trek 142
#define XC_ul_angle 144
#define XC_umbrella 146
#define XC_ur_angle 148
#define XC_watch 150
#define XC_xterm 152
```

The X Toolkit

Overview

This section 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 2.2 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 2.2 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 and Table 11 have been included in SCD 2.2 because they are REQUIRED to be present on all compliant systems, in the dynamic library `/usr/lib/libXt.so.5`.

Table 12 contains a list of unsafe macros. These macros should be avoided by application programmers which are trying to build portable SPARC applications.

Table 13 contains the exported data which are also REQUIRED to be present in `libXt.so.5`. The format of these entries is: `data[size]`. Data without a size are opaque.

Conformant systems are also REQUIRED to have `/usr/lib/libXt.so.4` in order to support SPARC applications written to conform to versions 2.0 and 2.1 of the *SPARC Compliance Definition*. Since the X Version 11, Release 5 specification is a proper superset of the X Version 11, Release 4 specification, system vendors can provide this support by simply making a link to `/usr/lib/libXt.so.5` for the file `/usr/lib/libXt.so.4` as part of the system installation process. If an application executable references `/usr/lib/libXt.so.4` and does not reference `/usr/lib/libXt.so.5`, the application may not use any functionality defined by X11R5 but not defined by X11R4. See the *SPARC Compliance Definition 2.1* for a list of X11R4 components. The file name `/usr/lib/libXt.so.4` is a DEPRECATED interface effective November 1st, 1993. Also DEPRECATED effective November 1st, 1993 are the functions found in Table 11 on page 11. Any of the DEPRECATED interfaces may be removed from this specification as early as November 1st, 1996 so new applications should not be dependent on them.

Figures 15 through 23 detail the manifest constants and visible data structures associated with the X Toolkit library.

Figures 24 through 31 detail the manifest constants and visible data structures needed by widget programmers. Only widget programmers should use the information in these figures.

Table 10. Contents of libxt, Part 1 of 2

XtAddCallback	XtConfigureWidget
XtAddCallbacks	XtConvertAndStore
XtAddEventHandler	XtConvertCase
XtAddExposureToRegion	XtCreateApplicationContext
XtAddGrab	XtCreateManagedWidget
XtAddRawEventHandler	XtCreatePopupShell
XtAllocateGC	XtCreateWidget
XtAppAddActionHook	XtCreateWindow
XtAppAddActions	XtCvtColorToPixel
XtAppAddInput	XtCvtIntToBool
XtAppAddTimeOut	XtCvtIntToBoolean
XtAppAddWorkProc	XtCvtIntToColor
XtAppCreateShell	XtCvtIntToFloat
XtAppError	XtCvtIntToFont
XtAppErrorMsg	XtCvtIntToPixel
XtAppGetErrorDatabase	XtCvtIntToPixmap
XtAppGetErrorDatabaseText	XtCvtIntToShort
XtAppGetSelectionTimeout	XtCvtIntToUnsignedChar
XtAppInitialize	XtCvtStringToAcceleratorTable
XtAppMainLoop	XtCvtStringToAtom
XtAppNextEvent	XtCvtStringToBool
XtAppPeekEvent	XtCvtStringToBoolean
XtAppPending	XtCvtStringToCursor
XtAppProcessEvent	XtCvtStringToDimension
XtAppReleaseCacheRefs	XtCvtStringToDisplay
XtAppSetErrorHandler	XtCvtStringToFile
XtAppSetErrorMsgHandler	XtCvtStringToFloat
XtAppSetFallbackResources	XtCvtStringToFont
XtAppSetSelectionTimeout	XtCvtStringToFontSet
XtAppSetTypeConverter	XtCvtStringToFontStruct
XtAppSetWarningHandler	XtCvtStringToInitialState
XtAppSetWarningMsgHandler	XtCvtStringToInt
XtAppWarning	XtCvtStringToPixel
XtAppWarningMsg	XtCvtStringToShort
XtAugmentTranslations	XtCvtStringToTranslationTable
XtBuildEventMask	XtCvtStringToUnsignedChar
XtCallAcceptFocus	XtCvtStringToVisual
XtCallActionProc	XtDatabase
XtCallbackExclusive	XtDestroyApplicationContext
XtCallbackNone	XtDestroyWidget
XtCallbackNonexclusive	XtDisownSelection
XtCallbackPopdown	XtDispatchEvent
XtCallbackReleaseCacheRef	XtDisplay
XtCallbackReleaseCacheRefList	XtDisplayInitialize
XtCallCallbackList	XtDisplayOfObject
XtCallCallbacks	XtDisplayStringConversionWarning
XtCallConverter	XtDisplayToApplicationContext
XtCalloc	XtFindFile
XtClass	XtFree
XtCloseDisplay	XtGetActionKeysym

XtGetActionList	XtNameToWidget
XtGetApplicationNameAndClass	XtNewString
XtGetApplicationResources	XtOpenDisplay
XtGetConstraintResourceList	XtOverrideTranslations
XtGetGC	XtOwnSelection
XtGetKeysymTable	XtOwnSelectionIncremental
XtGetMultiClickTime	XtParent
XtGetResourceList	XtParseAcceleratorTable
XtGetSelectionRequest	XtParseTranslationTable
XtGetSelectionValue	XtPopdown
XtGetSelectionValueIncremental	XtPopup
XtGetSelectionValues	XtPopupSpringLoaded
XtGetSelectionValuesIncremental	XtQueryGeometry
XtGetSubresources	XtRealizeWidget
XtGetSubvalues	XtRealloc
XtGetValues	XtRegisterCaseConverter
XtGrabButton	XtRegisterGrabAction
XtGrabKey	XtReleaseGC
XtGrabKeyboard	XtRemoveActionHook
XtGrabPointer	XtRemoveAllCallbacks
XtHasCallbacks	XtRemoveCallback
XtInitializeWidgetClass	XtRemoveCallbacks
XtInsertEventHandler	XtRemoveEventHandler
XtInsertRawEventHandler	XtRemoveGrab
XtInstallAccelerators	XtRemoveInput
XtInstallAllAccelerators	XtRemoveRawEventHandler
XtIsApplicationShell	XtRemoveTimeOut
XtIsComposite	XtRemoveWorkProc
XtIsConstraint	XtResizeWidget
XtIsManaged	XtResizeWindow
XtIsObject	XtResolvePathname
XtIsOverrideShell	XtScreen
XtIsRealized	XtScreenDatabase
XtIsRectObj	XtScreenOfObject
XtIsSensitive	XtSetKeyboardFocus
XtIsShell	XtSetKeyTranslator
XtIsSubclass	XtSetLanguageProc
XtIsTopLevelShell	XtSetMappedWhenManaged
XtIsTransientShell	XtSetMultiClickTime
XtIsVendorShell	XtSetSensitive
XtIsWidget	XtSetSubvalues
XtIsWMShell	XtSetTypeConverter
XtKeysymToKeycodeList	XtSetValues
XtLastTimestampProcessed	XtSetWMColormapWindows
XtMakeGeometryRequest	XtSuperclass
XtMakeResizeRequest	XtToolkitInitialize
XtMalloc	XtTranslateCoords
XtManageChild	XtTranslateKey
XtManageChildren	XtTranslateKeycode
XtMapWidget	XtUngrabButton
XtMenuPopupAction	XtUngrabKey
XtMergeArgLists	XtUngrabKeyboard
XtMoveWidget	XtUngrabPointer
XtName	XtUninstallTranslations

XtUnmanageChild
XtUnmanageChildren
XtUnmapWidget
XtUnrealizeWidget
XtVaAppCreateShell
XtVaAppInitialize
XtVaCreateArgsList
XtVaCreateManagedWidget
XtVaCreatePopupShell
XtVaCreateWidget
XtVaGetApplicationResources
XtVaGetSubresources
XtVaGetSubvalues
XtVaGetValues
XtVaSetSubvalues
XtVaSetValues
XtWidgetToApplicationContext
XtWindow
XtWindowOfObject
XtWindowToWidget

Deprecated X Toolkit Functions

Table 11 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.

Table 11: Contents of libXt, Part 2 of 2

Obsolete Function	Superseded By
XtAddActions	XtAppAddActions
XtAddConverter	XtSetTypeConverter
XtAddInput	XtAppAddInput
XtAddTimeOut	XtAppAddTimeOut
XtAddWorkProc	XtAppAddWorkProc
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

Table 11: Contents of libXt, Part 2 of 2

Obsolete Function	Superseded By
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 12 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 12: Unsafe Macros

XtIsApplicationShell
XtIsComposite
XtIsConstraint
XtIsOverrideShell
XtIsRectObj
XtIsShell
XtIsTopLevelShell
XtIsTransientShell
XtIsVendorShell
XtIsWidget
XtIsWMShell

Table 13. 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]
_XtInherit1
_XtInheritTranslations2
XtShellStrings3
XtStrings4

```

1. `_XtInherit` is only for use by widget programmers.

2. `_XtInheritTranslations` is only for use by widget programmers.

3. `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.

4. See note number 3.

Figure 15. Manifest Constants and Data Types from <X11/Composite.h>

```
typedef struct _CompositeClassRec *CompositeWidgetClass;  
  
typedef Cardinal (*XtOrderProc)();
```

Figure 16. Manifest Constants and Data Types from <X11/Constraint.h>

```
typedef struct _ConstraintClassRec *ConstraintWidgetClass;
```

Figure 17. Manifest Constants and Data Types from <X11/Core.h>

```
typedef struct _WidgetClassRec *CoreWidgetClass;  
typedef struct _WidgetRec *CoreWidget;
```

Figure 18. 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

/*****
 *
 * System Dependent Definitions; see spec for specific range
 * requirements. Do not assume every implementation uses the
 * same base types!
 *
 *
 * XtArgVal ought to be a union of XtPointer, char *, long, int *, and proc *
 * but casting to union types is not really supported.
 *
 * So the typedef for XtArgVal should be chosen such that
 *
 *      sizeof (XtArgVal) >=      sizeof(XtPointer)
 *                                sizeof(char *)
 *                                sizeof(long)
 *                                sizeof(int *)
 *                                sizeof(proc *)
 *
 * ArgLists rely heavily on the above typedef.
 *
 *****/
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;

/* The type Opaque is NOT part of the Xt standard, do NOT use it. */
/* (It remains here only for backward compatibility.) */
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 {
/* address mode          parameter representation */
/* -----              -*/
    XtAddress,           /* address */
    XtBaseOffset,        /* offset */
    XtImmediate,         /* constant */
    XtResourceString,    /* resource name string */
    XtResourceQuark,     /* resource name quark */
    XtWidgetBaseOffset,  /* offset from ancestor */
    XtProcedureArg       /* procedure to invoke */
} 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;

/* Additions to Xlib geometry requests: ask what would happen, don't do it */
#define XtCWQueryOnly (1 << 7)

```

```

/* Additions to Xlib stack modes: don't change stack order */
#define XtSMDontChange 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)();

/*****
 *
 * Event Management
 *
 *****/

/* XtAllEvents is valid only for XtRemoveEventHandler and
 * XtRemoveRawEventHandler; don't use it to select events!
 */
#define XtAllEvents ((EventMask) -1L)

/*****
 *
 * Event Gathering Routines
 *
 *****/

#define XtIMXEvent          1
#define XtIMTimer           2
#define XtIMAlternateInput  4
#define XtIMAll (XtIMXEvent | XtIMTimer | XtIMAlternateInput)

/*****
 *
 * Vararg lists
 *
 *****/

#define XtVaNestedList  "XtVaNestedList"
#define XtVaTypedArg    "XtVaTypedArg"

/*****
 *
 * Toolkit initialization
 *
 *****/

#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"

/*****

```

```
*
* Selections
*
*****/

#define XT_CONVERT_FAIL (Atom)0x80000001
```

Figure 19. Manifest Constants and Data Types from <X11/Object.h>

```
typedef struct _ObjectRec *Object;  
typedef struct _ObjectClassRec *ObjectClass;
```

Figure 20. Manifest Constants and Data Types from <X11/RectObj.h>

```
typedef struct _RectObjRec *RectObj;  
typedef struct _RectObjClassRec *RectObjClass;
```

Figure 21. 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. */
/* Do not edit. */

#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 22. 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 XtNmenuEntry "menuEntry"
#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 XtCMenuEntry "MenuEntry"
#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 23. Manifest Constants and Data Types from <X11/Vendor.h>

```
typedef struct _VendorShellClassRec *VendorShellWidgetClass;
```

Subclassing Xt Widgets

Figures 24 through 31 are intended to be used by widget programmers only. This information is included so widget programmers can subclass Xt widgets. Ordinary application programmers should never write code which depends on the definitions found in these tables.

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 24. Manifest Constants and Data Types from <X11/CompositeP.h>

```

/*****
 *
 * Additional instance fields for widgets of (sub)class 'Composite'
 *
 *****/

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;

/*****
 *
 * Additional class fields for widgets of (sub)class 'Composite'
 *
 *****/

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 25. Manifest Constants and Data Types from <X11/ConstrainP.h>

```
typedef struct _ConstraintPart {
    XtPointer mumble;          /* No new fields, keep C compiler happy */
} 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 26. Manifest Constants and Data Types from <X11/CoreP.h>

```

#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)

/*****
 * Widget Core Data Structures
 *
 *****/

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;

```

```

/*****
 *
 * Core Class Structure. Widgets, regardless of their class, will have
 * these fields. All widgets of a given class will have the same values
 * for these fields. Widgets of a given class may also have additional
 * common fields. These additional fields are included in incremental
 * class structures, such as CommandClass.
 *
 * The fields that are specific to this subclass, as opposed to fields that
 * are part of the superclass, are called "subclass fields" below. Many
 * procedures are responsible only for the subclass fields, and not for
 * any superclass fields.
 *
 *****/

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_inited;         /* 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 27. 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)();

/* compress_exposure options*/
#define XtExposeNoCompress ((XtEnum)False)
#define XtExposeCompressSeries ((XtEnum)True)
#define XtExposeCompressMultiple 2
#define XtExposeCompressMaximal 3

/* modifiers */
#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 28. 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 29. 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_initied;        /* 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 30. 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;          /* pointer to extension record */
} ShellClassPart;

typedef struct {
    XtPointer next_extension; /* 1st 4 mandated for all extension records */
    XrmQuark record_type;    /* NULLQUARK; on ShellClassPart */
    long version;           /* must be XtShellExtensionVersion */
    Cardinal record_size;   /* sizeof(ShellClassExtensionRec) */
    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    client_specified; /* re-using old name */
#define _XtShellPositionValid ((Boolean)(1<<0))
#define _XtShellNotReparented ((Boolean)(1<<1))
#define _XtShellPPositionOK ((Boolean)(1<<2))
#define _XtShellGeometryParsed ((Boolean)(1<<3))
    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;          /* pointer to extension record */
} 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;          /* pointer to extension record */
} 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;          /* pointer to extension record */
} 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;          /* pointer to extension record */
} 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;          /* pointer to extension record */
} 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 31. Data Types from <X11/VendorP.h>

```

/* New fields for the VendorShell widget class record */

typedef struct {
    XtPointer      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;

```