XDR Manual Pages

This appendix lists the XDR library calls in UNIX-style manual page format.

XDR Library Calls

The XDR library calls are listed in alphabetical order.

Following a brief introductory statement summarizing its use, each function is described using the documentation style of UNIX. This table lists the basic components of each function description:

Table B-1 XDR Library Calls

Call	Description
Synopsis	A synopsis of the function is given in C language format. The function prototype statement is listed showing all function arguments
Description	A description of the function is given, including any special rules for specifying arguments, alternative uses of the function, and any results returned.
Parameters	Each parameter for the call is described.
Files	Any required include files are listed in this section.
See Also	References to related functions are given.

xdr_array()

Translate between arrays and their external representations.

Synopsis

```
bool_t xdr_array(xdrs, arrp, sizep, maxsize, elsize, elproc)
XDR         *xdrs;
char        **arrp;
u_int       *sizep, maxsize, elsize;
xdrproc_t   elproc;
```

Description

xdr_array() is a filter primitive that translates between variable-length arrays and their corresponding external representations. The parameter arrp is the address of the pointer to the array, while sizep is the address of the element count of the array; this element count cannot exceed maxsize. The parameter elsize is the sizeof each of the array's elements, and elproc is an XDR filter that translates between the array elements' C form and their external representation. This routine returns TRUE if it succeeds, FALSE otherwise.

Parameters

xdrs	XDR stream
arrp	The address of the pointer to the array
sizep	The address of the element count of the array; this element count cannot exceed maxsize.
maxsize	The maximum element count
elsize	The size of each of the array's elements
elproc	An XDR filter that translates between the array elements' C form and their external representation.

Files

rpc.h - RPC include file

See Also

xdr_bool(), xdr_bytes(), xdr_char().

xdr_bool()

Translate between booleans and their external representations.

Synopsis

```
bool_t xdr_bool(xdrs, bp)
XDR     *xdrs;
bool_t   *bp;
```

Description

xdr_bool() is a filter primitive that translates between booleans and their external representations. When encoding data, this filter produces values of either TRUE or FALSE. This routine returns TRUE if it succeeds, FALSE otherwise.

Parameters

xdrs	XDR stream
bp	The address of the boolean.

Files

rpc.h - RPC include file

See Also

xdr_bool(), xdr_bytes(), xdr_char().

xdr_bytes()

Translate between counted byte strings and their external representations.

Synopsis

```
bool_t xdr_bytes(xdrs, sp, sizep, maxsize)
XDR       *xdrs;
char      **sp;
u_int     *sizep, maxsize;
```

Description

xdr_bytes() is a filter primitive that translates between counted byte strings and their external representations. The parameter sp is the address of the string pointer. The length of the string is located at sizep; strings cannot be longer than maxsize. This routine returns TRUE if it succeeds, FALSE otherwise.

Parameters

xdrs	XDR stream
sp	The address of the string pointer.
sizep	The length of the string.
maxsize	Maximum length of string

Files

rpc.h - RPC include file

See Also

xdr_bool(), xdr_bytes(), xdr_char().

xdr_char()

Translate between C characters and their external representations.

Synopsis

```
bool_t xdr_char(xdrs, cp)
XDR      *xdrs;
char     *cp;
```

Description

xdr_char() is a filter primitive that translates between C characters and their external representations. This routine returns TRUE if it succeeds, FALSE otherwise.

Note Encoded characters are not packed, and occupy 4 bytes each. For arrays of characters, it is worthwhile to consider xdr_bytes(), xdr_opaque() or xdr_string().

Parameters

xdrs	XDR stream
cp	The address of the character

Files

rpc.h - RPC include file

See Also

xdr_bool(), xdr_bytes(), xdr_char(), xdr_opaque(), xdr_string().

xdr_destroy()

Destroy routine associated with XDR stream.

Synopsis

```
void xdr_destroy(xdrs)
XDR  *xdrs;
```

Description

xdr_destroy() is a macro that invokes the destroy routine associated with the XDR stream, xdrs. Destruction usually involves freeing private data structures associated with the stream. Using xdrs after invoking xdr_destroy() is undefined. This routine is called indirectly via the pointer stored in the structure XDR. This routine may also be called using the upper-case XDR_DESTROY().

Parameters

xdrs	XDR stream

Files

rpc.h - RPC include file

xdr_double()

Translate between C double precision numbers and their external representations.

Synopsis

```
bool_t xdr_double(xdrs, dp)
XDR      *xdrs;
double   *dp;
```

Description

xdr_double() is a filter primitive that translates between C double precision numbers and their external representations. This routine returns TRUE if it succeeds, FALSE otherwise.

Parameters

xdrs	XDR stream
dp	Address of double precision number

Files

rpc.h - RPC include file

See Also

xdr_bool(), xdr_bytes(), xdr_char(), xdr_opaque(), xdr_string().

xdr_enum()

Translate between C enums and their external representations.

Synopsis

```
bool_t xdr_double(xdrs, ep)
XDR       *xdrs;
enum_t    *ep;
```

Description

xdr_enum() is a filter primitive that translates between C enums and their external representations. This routine returns TRUE if it succeeds, FALSE otherwise.

Note Enums with the IBM C/370 compiler are scaled by the size of the maximum value defined for the enum. Therefore an enum may be a byte, two bytes or four bytes long. This routine assumes that an enum will be four bytes long.

Parameters

xdrs	XDR stream
ep	Enum

Files

rpc.h - RPC include file

See Also

xdr_bool(), xdr_bytes(), xdr_char(), xdr_opaque(), xdr_string().

xdr_float()

Translate between C floats and their external representations.

Synopsis

```
bool_t xdr_float(xdrs, fp)
XDR      *xdrs;
float    *fp;
```

Description

xdr_float() is a filter primitive that translates between C floats and their external representations. This routine returns TRUE if it succeeds, FALSE otherwise.

Parameters

xdrs	XDR stream
fp	Float

Files

rpc.h - RPC include file

See Also

xdr_bool(), xdr_bytes(), xdr_char(), xdr_opaque(), xdr_string().

xdr_free()

Generic freeing routine.

Synopsis

```
void xdr_free(proc, objp)
xdrproc_t    proc;
char         *objp;
```

Description

xdr_free() is a generic freeing routine. The first argument is the XDR routine for the object being freed. The second argument is a pointer to the object itself.

The pointer passed to this routine is not freed, but what it points to is freed (recursively), such that objects pointed to are also freed, for example, limited lists.

Parameters

xdrproc	XDR routine
objp	Object to be freed

Files

rpc.h - RPC include file

See Also

xdr_bool(), xdr_bytes(), xdr_char(), xdr_opaque(), xdr_string().

xdr_getpos()

Invoke get-position routine.

Synopsis

```
u_int xdr_getpos(xdrs)
XDR  *xdrs;
```

Description

xdr_getpos() is a macro that invokes the get-position routine associated with the XDR stream, xdrs. The routine returns an unsigned integer, which indicates the position of the XDR byte stream. A desirable feature of XDR streams is that simple arithmetic works with this number, although the XDR stream instances need not guarantee this. This routine is called indirectly via the pointer stored in the structure XDR. This routine may also be called using the upper-case XDR_GETPOS().

Parameters

xdrs	XDR stream

Files

rpc.h - RPC include file

See Also

xdr_inline().

xdr_inline()

Invoke in-line routine.

Synopsis

```
xdr_inline(xdrs, len)
XDR    *xdrs;
int    len;
```

Description

xdr_inline() is a macro that invokes the in-line routine associated with the XDR stream, xdrs. The routine returns a pointer to a contiguous piece of the stream's buffer; len is the byte length of the desired buffer.

The pointer is cast to long *. This routine is called indirectly via the pointer stored in the structure XDR. This routine may also be called using the upper-case XDR_GETPOS().

xdr_inline() may return NULL (0) if it cannot allocate a contiguous piece of a buffer. Therefore the behavior may vary among stream instances; it exists for the sake of efficiency.

Parameters

xdrs	XDR stream
len	Length of desired buffer

Files

rpc.h - RPC include file

See Also

xdr_getpos().

xdr_int()

Translate between C integers and their external representations.

Synopsis

```
bool_t xdr_int(xdrs, ip)
XDR    *xdrs;
int    *ip;
```

Description

xdr_int() is a filter primitive that translates between C integers and their external representations. This routine returns TRUE if it succeeds, FALSE otherwise.

Parameters

xdrs	XDR stream
ip	Integer

Files

rpc.h - RPC include file

See Also

xdr_bool(), xdr_bytes(), xdr_char()cc, xdr_opaque(), xdr_string().

xdr_long()

Translate between C long integers and their external representations.

Synopsis

```
bool_t xdr_long(xdrs, lp)
XDR     *xdrs;
long    *lp;
```

Description

xdr_long() is a filter primitive that translates between C long integers and their external representations. This routine returns TRUE if it succeeds, FALSE otherwise.

Parameters

xdrs	XDR stream
lp	Long

Files

rpc.h - RPC include file

See Also

xdr_bool(), xdr_bytes(), xdr_char(), xdr_opaque(), xdr_string().

xdr_opaque()

Translate between fixed size opaque data and its external representation.

Synopsis

```
bool_t xdr_opaque(xdrs, cp, cnt)
XDR      *xdrs;
char     *cp;
u_int    cnt;
```

Description

xdr_opaque() is a filter primitive that translates between fixed size opaque data and its external representation. The parameter cp is the address of the opaque object, and cnt is its size in bytes. This routine returns TRUE if it succeeds, FALSE otherwise.

Parameters

xdrs	XDR stream
cp	Opaque object
cn	Size of object

Files

rpc.h - RPC include file

See Also

xdr_bool(), xdr_bytes(), xdr_char(), xdr_opaque(), xdr_string().

xdr_pointer()

Provide pointer chasing within structures.

Synopsis

```
bool_t xdr_pointer(xdrs, objpp, objsize, xdrobj)
XDR          *xdrs;
char         **objpp;
u_int        objsize;
xdrproc_t    xdrobj;
```

Description

xdr_pointer() provides pointer chasing within structures. xdr_pointer() is like xdr_reference() except that it serializes NULL pointers, whereas xdr_reference() does not. Thus, xdr_pointer() can represent recursive data structures, such as binary trees or linked lists. This routine returns TRUE if it succeeds, FALSE otherwise.

Parameters

xdrs	XDR stream
objpp	Address of object pointer
objsize	Size of object
xdrobj	XDR procedure that filters the structure between its C form and its external representation

Files

rpc.h - RPC include file

See Also

xdr_reference()

xdr_reference()

Provides pointer chasing within structures.

Synopsis

```
bool_t xdr_reference(xdrs, pp, size, proc)
XDR          *xdrs;
char         **pp;
u_int        size;
xdrproc_t    proc;
```

Description

xdr_reference() is a primitive that provides pointer chasing within structures. The parameter pp is the address of the pointer; size is the size of the structure that *pp points to; and proc is an XDR procedure that filters the structure between its C form and its external representation. This routine returns TRUE if it succeeds, FALSE otherwise.

This routine does not understand NULL pointers. Use xdr_pointer() instead.

Parameters

xdrs	XDR stream
pp	The address of the pointer
size	Size of the structure that *pp points to
proc	XDR procedure that filters the structure between its C form and its external representation.

Files

rpc.h - RPC include file

See Also

xdr_pointer().

xdr_setpos()

Invoke set position routine.

Synopsis

```
bool_t xdr_setpos(xdrs, pos)
XDR      *xdrs;
u_int    pos;
```

Description

xdr_setpos() is a macro that invokes the set position routine associated with the XDR stream xdrs. The parameter pos is a position value obtained from xdr_getpos(). This routine returns TRUE if the XDR stream could be repositioned, and FALSE otherwise. This routine is called indirectly via the pointer stored in the structure XDR. This routine may also be called with the upper-case XDR_SETPOS.

It is difficult to reposition some types of XDR streams, so this routine may fail with one type of stream and succeed with another.

Parameters

xdrs	XDR stream
pos	Position value obtained from xdr_getpos()

Files

rpc.h - RPC include file

See Also

xdr_getpos().

xdr_short()

Translate between C short integers and their external representations.

Synopsis

```
bool_t xdr_short(xdrs, sp)
XDR      *xdrs;
short    *sp;
```

Description

xdr_short() is a filter primitive that translates between C short integers and their external representations. This routine returns TRUE if it succeeds, FALSE otherwise.

Parameters

xdrs	XDR stream
sp	pointer to short

Files

rpc.h - RPC include file

See Also

xdr_long().

xdr_string()

Translate between C strings and their external representations.

Synopsis

```
bool_t xdr_string(xdrs, sp, maxsize)
XDR       *xdrs;
char      **sp;
u_int     maxsize;
```

Description

xdr_string() is a filter primitive that translates between C strings and their corresponding external representations. Strings cannot be longer than maxsize.

*sp is the address of the string's pointer. While decoding, if *sp is NULL, then the necessary storage is allocated to hold this null-terminated string and *sp is set to point to this. Use xdr_free() to free this storage.

This routine returns TRUE if it succeeds, FALSE otherwise.

Parameters

xdrs	XDR stream
sp	Address of the string's pointer
maxsize	Maximum size of string

Files

rpc.h - RPC include file

See Also

xdr_char().

xdr_u_char()

Translate between unsigned C chars and their external representations.

Synopsis

```
bool_t xdr_u_char(xdrs, ucp)
XDR             *xdrs;
unsigned char   *ucp;
```

Description

xdr_u_char() is a filter primitive that translates between unsigned C characters and their external representations. This routine returns TRUE if it succeeds, FALSE otherwise.

Parameters

xdrs	XDR stream
ucp	Pointer to unsigned character

Files

rpc.h - RPC include file

See Also

xdr_char().

xdr_u_int()

Translate between unsigned C integers and their external representations.

Synopsis

```
bool_t xdr_u_int(xdrs, up)
XDR            *xdrs;
unsigned int   *up;
```

Description

xdr_u_int() is a filter primitive that translates between C unsigned integers and their external representations. This routine returns TRUE if it succeeds, FALSE otherwise.

Parameters

xdrs	XDR stream
up	Pointer to unsigned integer

Files

rpc.h - RPC include file

See Also

xdr_int().

xdr_u_long()

Translate between unsigned C long integers and their external representations.

Synopsis

```
bool_t xdr_u_long(xdrs, ulp)
XDR             *xdrs;
unsigned long   *ulp;
```

Description

xdr_u_long() is a filter primitive that translates between C unsigned long integers and their external representations. This routine returns TRUE if it succeeds, FALSE otherwise.

Parameters

xdrs	XDR stream
ulp	pointer to unsigned long integer

Files

rpc.h - RPC include file

See Also

xdr_int()

xdr_u_short()

Translate between unsigned C short integers and their external representations.

Synopsis

```
bool_t xdr_u_short(xdrs, usp)
XDR               *xdrs;
unsigned short    *usp;
```

Description

xdr_u_short() is a filter primitive that translates between C unsigned short integers and their external representations. This routine returns TRUE if it succeeds, FALSE otherwise.

Parameters

xdrs	XDR stream
usp	pointer to unsigned short integer

Files

rpc.h - RPC include file

See Also

xdr_int().

xdr_union()

Translate between discriminated C union and its external representation.

Synopsis

```
bool_t xdr_union(xdrs, dscmp, unp, choices, defaultarm)
XDR                   *xdrs;
int                   *dscmp;
char                  *unp
struct xdr_discrim    *choices;
bool_t               (*defaultarm)();                    /* may equal NULL */
```

Description

xdr_union() is a filter primitive that translates between a discriminated C union and its external representation. It first translates the discriminant of the union located at dscmp. This discriminant is always an enum_t. Next the union located at unp is translated. The parameter choices is a pointer to an array of xdr_discrim() structures. Each structure contains an ordered pair of [value, proc]. If the union's discriminant is equal to the associated value, then the proc is called to translate the union. The end of the xdr_discrim() structure array is denoted by a routine value of NULL. If the discriminant is not found in the choices array, then the defaultarm procedure is called (if it is not NULL). This routine returns TRUE if it succeeds, FALSE otherwise.

Parameters

xdrs	XDR stream
dscmp	Union discriminant
unp	Address of union
choices	Pointer to array of xdr_discrim() structures
defaultarm	Procedure called if discriminant not found in choices array

Files

rpc.h - RPC include file

See Also

xdr_discrim()

xdr_vector()

Translate between fixed length arrays and their external representations.

Synopsis

```
bool_t xdr_vector(xdrs, arrp, size, elsize, elproc)
XDR          *xdrs;
char         *arrp;
u_int        size, elsize;
xdrproc_t    elproc;
```

Description

xdr_vector() is a filter primitive that translates between fixed length arrays and their external representations. The parameter arrp is the address of the pointer to the array, while size is the element count of the array. The parameter elsize is the size of each of the array's elements, and elproc is an XDR filter that translates between the array elements' C form, and their external representation. This routine returns TRUE if it succeeds, FALSE otherwise.

Parameters

xdrs	XDR stream
arrp	Address of the pointer to the array
size	Element count of the array
elsize	Size of each of the array's elements
elproc	XDR filter

Files

rpc.h - RPC include file

See Also

xdr_array()

xdr_void()

Routine which always returns one.

Synopsis

```
bool_t xdr_void()
```

Description

xdr_void() always returns one. It may be passed to RPC routines that require a function parameter, where nothing is to be done.

Files

rpc.h - RPC include file

xdr_wrapstring()

Call xdr_string().

Synopsis

```
bool_t xdr_wrapstring(xdrs, sp)
XDR     *xdrs;
char    **sp;
```

Description

xdr_wrapstring() is a primitive that calls xdr_string(xdrs, sp, MAXUN.UNSIGNED); where MAXUN.UNSIGNED is the maximum value of an unsigned integer. xdr_wrapstring() is handy because the RPC package passes a maximum of two XDR routines as parameters and xdr_string(), one of the most frequently used primitives, requires three. Returns TRUE if it succeeds, FALSE otherwise.

Parameters

xdrs	XDR stream
sp	Address of the string

Files

rpc.h - RPC include file

See Also

xdr_array()

xdrmem_create()

Initialize XDR stream.

Synopsis

```
void xdrmem_create(xdrs, addr, size, op)
XDR             *xdrs;
char            *addr;
u_int           size;
enum xdr_op     op;
```

Description

xdrmem_create() initializes the XDR stream object pointed to by xdrs. The stream's data is written to, or read from, a chunk of memory at location addr whose length is no more than size bytes long. The op determines the direction of the XDR stream (either XDR_ENCODE, XDR_DECODE, or XDR_FREE).

Parameters

xdrs	XDR stream
addr	Object
size	Length of object
op	Direction of XDR stream

Files

rpc.h - RPC include file

See Also

xdr_bool(), xdr_bytes(), xdr_char(), xdr_opaque(), xdr_string().

xdrrec_create()

Initialize XDR stream object.

Synopsis

```
void xdrrec_create(xdrs, sendsize, recvsize, handle, readit, writeit)
XDR       *xdrs;
u_int     sendsize, recvsize;
char      *handle;
int       (*readit)(), (*writeit)();
```

Description

xdrmem_create() initializes the XDR stream object pointed to by xdrs. The stream's data is written to a buffer of size sendsize; a value of zero indicates the system should use a suitable default. The stream's data is read from a buffer of size recvsize; it too can be set to a suitable default by passing a zero value. When a stream's output buffer is full, writeit is called. Similarly, when a stream's input buffer is empty, readit is called. The behavior of these two routines is similar to the system calls read and write, except that handle is passed to the former routines as the first parameter.

Note The XDR stream's op field must be set by the caller.

This XDR stream implements an intermediate record stream. Therefore, there are additional bytes in the stream to provide record boundary information.

Parameters

xdrs	XDR stream
sendsize	Write buffer size
recvsize	Read buffer size
handle	Passed to readit and writeit routines
readit	Called when input buffer is empty
writeit	Called when output buffer is full

Files

rpc.h - RPC include file

See Also

read(), write()

xdrrec_endofrecord()

Mark data in buffer as completed record.

Synopsis

```
bool_t xdrrec_endofrecord(xdrs, sendnow)
XDR     *xdrs;
int     sendnow;
```

Description

xdrmem_endofrecord() can be invoked only on streams created by xdrrec_create(). The data in the output buffer is marked as a completed record, and the output buffer is optionally written out if sendnow is non-zero. This routine returns TRUE if it succeeds, FALSE otherwise.

Parameters

xdrs	XDR stream
sendnow	write out data (if set)

Files

rpc.h - RPC include file

See Also

`xdrrec_create(), xdrrec_eof(), xdrrec_skiprecord()`

xdrrec_eof()

Mark data in buffer as end of file.

Synopsis

```
bool_t xdrrec_eof(xdrs, empty)
XDR    *xdrs;
int    empty;
```

Description

xdrmem_eof() can be invoked only on streams created by xdrrec_create(). After consuming the rest of the current record in the stream, this routine returns TRUE if the stream has no more input, FALSE otherwise.

Parameters

xdrs	XDR stream
empty	no more data

Files

rpc.h - RPC include file

See Also

xdrrec_create(), xdrrec_endofrecord(), xdrrec_skiprecord()

xdrrec_skiprecord()

Discard rest of current record.

Synopsis

```
bool_t xdrrec_skiprecord(xdrs)
XDR *xdrs;
```

Description

xdrmem_skiprecord() can be invoked only on streams created by xdrrec_create(). It tells the XDR implementation that the rest of the current record in the stream's input buffer should be discarded. This routine returns TRUE if it succeeds, FALSE otherwise.

Parameters

xdrs	XDR stream

Files

rpc.h - RPC include file

See Also

xdrrec_create(), xdrrec_endofrecord(), xdrrec_eof().

xdrstdio_create()

Initialize XDR stream.

Synopsis

```
void xdrstdio_create(xdrs, file, op)
XDR          *xdrs;
FILE         *file;
enum xdr_op  op;
```

Description

xdrstdio_create() initializes the XDR stream object pointed to by xdrs. The XDR stream data is written to, or read from, the standard I/O stream file. The parameter op determines the direction of the XDR stream (either XDR_ENCODE, XDR_DECODE, or XDR_FREE).

The destroy routine associated with such XDR streams calls **fflush()** on the file stream, but never **fclose().**

Parameters

xdrs	XDR stream
file	Standard I/O stream
op	Direction of XDR stream

Files

rpc.h - RPC include file

See Also

xdrmem_create(), xdrrec_create().