NAME

rpcbind, rpcb_getmaps, rpcb_getaddr, rpcb_gettime, rpcb_rmtcall, rpcb_set, rpcb_unset – library routines for RPC bind service

MT-LEVEL

MT-Safe

DESCRIPTION

These routines allow client C programs to make procedure calls to the RPC binder service. **rpcbind** (see **rpcbind**(1M)) maintains a list of mappings between programs and their universal addresses.

Routines

#include <rpc/rpc.h>

struct rpcblist *rpcb_getmaps(const struct netconfig **netconf*, **const char** **host*);

An interface to the **rpcbind** service, which returns a list of the current RPC program-to-address mappings on *host*. It uses the transport specified through *netconf* to contact the remote **rpcbind** service on *host*. This routine will return **NULL**, if the remote **rpcbind** could not be contacted.

bool_t rpcb_getaddr(const u_long *prognum*, **const u_long** *versnum*, **const struct netconfig** **netconf*, **struct netbuf** **svcaddr*, **const char** **host*);

An interface to the **rpcbind** service, which finds the address of the service on *host* that is registered with program number *prognum*, version *versnum*, and speaks the transport protocol associated with *netconf*. The address found is returned in *svcaddr*. *svcaddr* should be preallocated. This routine returns **TRUE** if it succeeds. A return value of **FALSE** means that the mapping does not exist or that the RPC system failed to contact the remote **rpcbind** service. In the latter case, the global variable **rpc_createerr** (see **rpc_clnt_create**(3N)) contains the RPC status.

bool_t rpcb_gettime(const char **host*, **time_t** **timep*);

This routine returns the time on *host* in *timep*. If *host* is **NULL**, **rpcb_gettime( )** returns the time on its own machine. This routine returns **TRUE** if it succeeds, **FALSE** if it fails. **rpcb_gettime( )** can be used to synchronize the time between the client and the remote server. This routine is particularly useful for secure RPC.

enum clnt_stat rpcb_rmtcall(const struct netconfig **netconf*, **const char** **host*, **const u_long** *prognum*, **const u_long** *versnum*, **const u_long** *procnum*, **const xdrproc_t** *inproc*, **const caddr_t** *in*, **const xdrproc_t** *outproc*, **caddr_t** *out*, **const struct timeval** *tout*, **struct netbuf** **svcaddr*);

An interface to the **rpcbind** service, which instructs **rpcbind** on *host* to make an RPC call on your behalf to a procedure on that host. The **netconfig** structure should correspond to a connectionless transport. The parameter **svcaddr* will be modified to the server's address if the procedure succeeds (see **rpc_call( )** and **clnt_call( )** in **rpc_clnt_calls**(3N) for the definitions of other parameters).

This procedure should normally be used for a "ping" and nothing else. This routine allows programs to do lookup and call, all in one step.

Note: Even if the server is not running **rpcbind** does not return any error messages to the caller. In such a case, the caller times out.

Note: **rpcb_rmtcall( )** is only available for connectionless transports.

bool_t rpcb_set(const u_long *prognum***, const u_long** *versnum***,**
const struct netconfig **netconf***, const struct netbuf** **svcaddr***);**

An interface to the **rpcbind** service, which establishes a mapping between the triple [*prognum*, *versnum*, *netconf*→*nc_netid*] and *svcaddr* on the machine's **rpcbind** service. The value of *nc_netid* must correspond to a network identifier that is defined by the netconfig database. This routine returns **TRUE** if it succeeds, **FALSE** otherwise. (See also **svc_reg( )** in **rpc_svc_calls**(3N)). If there already exists such an entry with **rpcbind**, **rpcb_set( )** will fail.

bool_t rpcb_unset(const u_long *prognum***, const u_long** *versnum***,**
const struct netconfig **netconf***);**

An interface to the **rpcbind** service, which destroys the mapping between the triple [*prognum*, *versnum*, *netconf*→*nc_netid*] and the address on the machine's **rpcbind** service. If *netconf* is **NULL**, **rpcb_unset( )** destroys all mapping between the triple [*prognum*, *versnum*, *all-transports*] and the addresses on the machine's **rpcbind** service. This routine returns **TRUE** if it succeeds, **FALSE** otherwise. Only the owner of the service or the super-user can destroy the mapping. (See also **svc_unreg( )** in **rpc_svc_calls**(3N)).

SEE ALSO

rpcbind(1M), **rpcinfo**(1M), **rpc_clnt_calls**(3N), **rpc_svc_calls**(3N)