

**NAME**

**rpcb\_getmaps, rpcb\_getaddr, rpcb\_gettime, rpcb\_rmtcall, rpcb\_set, rpcb\_unset** - library routines for RPC bind service

**LIBRARY**

Standard C Library (libc, -lc)

**SYNOPSIS**

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

*rpcblist* \*

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

*bool\_t*

```
rpcb_getaddr(const rpcprog_t prognum, const rpcvers_t versnum, const struct netconfig *netconf,  
             struct netbuf *svcaddr, const char *host);
```

*bool\_t*

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

*enum clnt\_stat*

```
rpcb_rmtcall(const struct netconfig *netconf, const char *host, const rpcprog_t prognum,  
             const rpcvers_t versnum, const rpcproc_t procnum, const xdrproc_t inproc, const caddr_t in,  
             const xdrproc_t outproc, const caddr_t out, const struct timeval tout, const struct netbuf *svcaddr);
```

*bool\_t*

```
rpcb_set(const rpcprog_t prognum, const rpcvers_t versnum, const struct netconfig *netconf,  
         const struct netbuf *svcaddr);
```

*bool\_t*

```
rpcb_unset(const rpcprog_t prognum, const rpcvers_t versnum, const struct netconfig *netconf);
```

**DESCRIPTION**

These routines allow client C programs to make procedure calls to the RPC binder service. (see [rpcbind\(8\)](#)) maintains a list of mappings between programs and their universal addresses.

**Routines**

**rpcb\_getmaps()**

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.

### **rpcb\_getaddr()**

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*. The *svcaddr* argument 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(3)*) contains the RPC status.

### **rpcb\_gettime()**

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. The **rpcb\_gettime()** function can be used to synchronize the time between the client and the remote server.

### **rpcb\_rmtcall()**

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 *svcaddr* argument will be modified to the server's address if the procedure succeeds (see **rpc\_call()** and **clnt\_call()** in *rpc\_clnt\_calls(3)* for the definitions of other arguments).

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 **rpcb\_rmtcall()** 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.

### **rpcb\_set()**

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(3)*.) If there already exists such an entry with rpcbind, **rpcb\_set()** will fail.

**rpcb\_unset()**

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-transport] 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(3)*.)

**SEE ALSO**

*rpc\_clnt\_calls(3)*, *rpc\_svc\_calls(3)*, *rpcbind(8)*, *rpcinfo(8)*