NAME

rpc_control – library routine for manipulating global RPC attributes for client and server applications

SYNOPSIS

bool_t rpc_control(int *op*,*void* ∗**info***);*

MT-LEVEL

MT-Safe

DESCRIPTION

This RPC library routine allows applications to set and modify global RPC attributes that apply to clients as well as servers. At present, it supports only server side operations. This function allows applications to set and modify global attributes that apply to client as well as server functions. *op* indicates the type of operation, and *info* is a pointer to the operation specific information. The supported values of *op* and their argument types, and what they do are:

RPC_SVC_MTMODE_SET	**int** ∗	set multithread mode
RPC_SVC_MTMODE_GET	**int** ∗	get multithread mode
RPC_SVC_THRMAX_SET	**int** ∗	set maximum number of threads
RPC_SVC_THRMAX_GET	**int** ∗	get maximum number of threads
RPC_SVC_THRTOTAL_GET	**int** ∗	get number of active threads
RPC_SVC_THRCREATES_GET	**int** ∗	get number of threads created
RPC_SVC_THRERRORS_GET	**int** ∗	get number of thread create errors

There are three multithread (MT) modes. These are:

RPC_SVC_MT_NONE	Single threaded mode	(default)
RPC_SVC_MT_AUTO	Automatic MT mode	
RPC_SVC_MT_USER	User MT mode	

Unless the application sets the Automatic or User MT modes, it will stay in the default (single threaded) mode. See the Network Interfaces Programming Guide for the meanings of these modes and programming examples. Once a mode is set, it cannot be changed.

By default, the maximum number of threads that the server will create at any time is 16. This allows the service developer to put a bound on thread resources consumed by a server. If a server needs to process more than 16 client requests concurrently, the maximum number of threads must be set to the desired number. This parameter may be set at any time by the server.

Set and get operations will succeed even in modes where the operations don't apply. For example, you can set the maximum number of threads in any mode, even though it makes sense only for the Automtatic MT mode. All of the get operations except **RPC_SVC_MTMODE_GET** apply only to the Automatic MT mode, so values returned in other modes may be undefined.

RETURN VALUES

This routine returns **TRUE** if the operation was successful, and **FALSE** otherwise.

SEE ALSO

rpc(3N), **rpc_svc_calls**(3N), **rpcbind**(1M)