## NAME

**rfcomm_sppd** - RFCOMM Serial Port Profile daemon

## SYNOPSIS

**rfcomm_sppd** [**-bhtS**] **-a** *address* **-c** *channel*

## DESCRIPTION

The **rfcomm_sppd** utility is a Serial Port Profile daemon. It can operate in two modes: client and server.

In client mode, **rfcomm_sppd** opens RFCOMM connection to the specified *address* server and *channel*. Once connection is established, the **rfcomm_sppd** utility provides access to the server's remote serial port via stdin/stdout or via pts(4) interface if **-t** option was specified.

If the **-S** option is specified, **rfcomm_sppd** will operate in server mode and act as RFCOMM server, listening on ANY address and advertising a virtual serial port via the sdpd(8) daemon. If **-t** options was specified, the server side of the virtual serial port is attached to a pseudo-terminal. Otherwise the virtual serial port is attached to the stdin/stdout. **rfcomm_sppd** should be run as root in order to communicate with sdpd(8) in this case.

The **rfcomm_sppd** utility opens both master and slave pseudo terminals. This is done to ensure that RFCOMM connection stays open until **rfcomm_sppd** is terminated. The data received from the master pseudo terminal are sent over the RFCOMM connection. The data received from the RFCOMM connection are written into master pseudo terminal. The application in its turn opens the slave pseudo terminal and operates on it just like it would operate over the standard serial port.

The options are as follows:

**-a** *address*

In client mode, this required option specifies the address of the remote RFCOMM server. If this option is specified in server mode, **rfcomm_sppd** will only accept connections from the Bluetooth device with address *address*. The address can be specified as BD_ADDR or name. If name was specified then **rfcomm_sppd** utility will attempt to resolve the name via bt_gethostbyname(3).

**-b** Detach from the controlling terminal, i.e., run in background.

**-c** *channel*

In both client and server mode, this option specifies the RFCOMM channel to connect to or listen on. In server mode, the channel should be a number between 1 and 30. If not specified, **rfcomm_sppd** will try to bind to "wildcard" RFCOMM channel number. The actual RFCOMM channel will be obtained via getsockname(2) call and will be used to register Serial Port service with sdpd(8). In client mode, the channel could either be a number between 1 and 30 or a service name. Supported service names are: **DUN** (for DialUp Networking service), **FAX** (for Fax service), **LAN** (for LAN Access Using PPP service) and **SP** (for Serial Port service). If channel was not specified then **rfcomm_sppd** utility will try to obtain RFCOMM channel for Serial Port service via Service Discovery Protocol from the server.

**-h** Display usage message and exit.

**-S** Server mode; see *DESCRIPTION*.

**-t** Use slave pseudo tty. If not set stdin/stdout will be used. This option is required if **-b** option was specified.

## FILES

*/dev/pts/[num]* slave pseudo terminals

## EXIT STATUS

The **rfcomm_sppd** utility exits 0 on success, and >0 if an error occurs.

## EXAMPLES

```
rfcomm_sppd -a 00:01:02:03:04:05 -c 1 -t
```

Will start the **rfcomm_sppd** utility and open RFCOMM connection to the server at 00:01:02:03:04:05 and channel 1. Once the connection has been established, a pts(4) can be used to talk to the remote serial port on the server. **rfcomm_sppd** prints the name of the pts(4) to use on stdout.

## SEE ALSO

bluetooth(3), ng_btsocket(4), pts(4), rfcomm_pppd(8), sdpd(8)

## AUTHORS

Maksim Yevmenkin <*m_evmenkin@yahoo.com*>

## BUGS

Please report if found.