

ost::SocketService(3)

ost::SocketService(3)

NAME

ost::SocketService –

The **SocketService** is a thread pool object that is meant to service attached socket ports.

SYNOPSIS

```
#include <socketport.h>
```

Inherits **ost::Thread**, and **ost::Mutex**.

Public Member Functions

void **update** (unsigned char flag=0xff)

Notify service thread that a port has been added or removed, or a timer changed, so that a new schedule can be computed for expiring attached ports.

SocketService (int pri=0, size_t stack=0, const char *id=NULL)

Create a service thread for attaching socket ports.

virtual ~**SocketService** ()

Terminate the thread pool and eliminate any attached socket ports.

int **getCount** (void) const

Get current reference count.

Protected Member Functions

virtual void **onUpdate** (unsigned char buf)

Handles all requests other than 'termination'.

virtual void **onEvent** (void)

Called once each time the service thread is rescheduled.

virtual void **onCallback** (**SocketPort** *port)

Called for each port that is being processed in response to an event.

Friends

class **SocketPort**

Detailed Description

The **SocketService** is a thread pool object that is meant to service attached socket ports.

Multiple pool objects may be created and multiple socket ports may be attached to the same thread of execution. This allows one to balance threads and sockets they service rather than either using a single thread for all connections or a separate thread for each connection. Features can be added through supported virtual methods.

Author:

David Sugar <dyfet AT ostel DOT com> **Thread** pool service object for socket ports.

Examples:

tcpservice.cpp.

Constructor & Destructor Documentation

ost::SocketService::SocketService (int pri = 0, size_t stack = 0, const char * id = NULL)

Create a service thread for attaching socket ports. The thread begins execution with the first attached socket.

Parameters:

pri of this thread to run under.

stack stack size.

id thread ID.

virtual **ost::SocketService::~SocketService** () [virtual]

Terminate the thread pool and eliminate any attached socket ports.

Member Function Documentation

int **ost::SocketService::getCount** (void) const [inline]

Get current reference count. This can be used when selecting the least used service handler from a pool.

Returns:

count of active ports.



ost::SocketService(3)

ost::SocketService(3)

virtual void ost::SocketService::onCallback (SocketPort * port) [protected, virtual]

Called for each port that is being processed in response to an event. This can be used to add additional notification options during callback in combination with **update()**.

Parameters:

port **SocketPort** who's callback events are being evaluated.

virtual void ost::SocketService::onEvent (void) [protected, virtual]

Called once each time the service thread is rescheduled. This is called after the mutex is locked and can be used to slip in additional processing.

virtual void ost::SocketService::onUpdate (unsigned char buf) [protected, virtual]

Handles all requests other than 'termination'. **Parameters:**

buf request id as posted from **update()**.

void ost::SocketService::update (unsigned char flag = 0xff)

Notify service thread that a port has been added or removed, or a timer changed, so that a new schedule can be computed for expiring attached ports. A '0' is used to terminate the service thread, and additional values can be specified which will be 'caught' in the **onUpdate()** handler.

Parameters:

flag update flag value.

Friends And Related Function Documentation**friend class SocketPort [friend]****Author**

Generated automatically by Doxygen for GNU CommonC++ from the source code.

