CFMachPort Reference

Core Foundation

2008-04-08

Contents

CFMachPort Reference

Derived From:	*CFType Reference*
Framework:	CoreFoundation/CoreFoundation.h
Declared in	CFMachPort.h

Overview

A CFMachPort object is a wrapper for a native Mach port (`mach_port_t`). Mach ports are the native communication channel for the Mac OS X kernel.

CFMachPort does not provide a function to send messages, so you primarily use a CFMachPort object if you need to listen to a Mach port that you obtained by other means. You can get a callback when a message arrives on the port or when the port becomes invalid, such as when the native port dies.

To listen for messages you need to create a run loop source with `CFMachPortCreateRunLoopSource` (page 7) and add it to a run loop with `CFRunLoopAddSource`.

Important: If you want to tear down the connection, you must invalidate the port (using `CFMachPortInvalidate` (page 10)) before releasing the runloop source and the Mach port object.

To send data, you must use the Mach APIs with the native Mach port, which is not described here. Alternatively, you can use a *CFMessagePort Reference* object, which can send arbitrary data.

Mach ports only support communication on the local machine. For network communication, you have to use a *CFSocket Reference* object.

Functions by Task

Creating a CFMachPort Object

`CFMachPortCreate` (page 6)
    Creates a CFMachPort object with a new Mach port.

`CFMachPortCreateWithPort` (page 8)
    Creates a CFMachPort object for a pre-existing native Mach port.

Configuring a CFMachPort Object

`CFMachPortInvalidate` (page 10)
: Invalidates a CFMachPort object, stopping it from receiving any more messages.

`CFMachPortCreateRunLoopSource` (page 7)
: Creates a CFRunLoopSource object for a CFMachPort object.

`CFMachPortSetInvalidationCallBack` (page 11)
: Sets the callback function invoked when a CFMachPort object is invalidated.

Examining a CFMachPort Object

`CFMachPortGetContext` (page 8)
: Returns the context information for a CFMachPort object.

`CFMachPortGetInvalidationCallBack` (page 9)
: Returns the invalidation callback function for a CFMachPort object.

`CFMachPortGetPort` (page 9)
: Returns the native Mach port represented by a CFMachPort object.

`CFMachPortIsValid` (page 11)
: Returns a Boolean value that indicates whether a CFMachPort object is valid and able to receive messages.

Getting the CFMachPort Type ID

`CFMachPortGetTypeID` (page 10)
: Returns the type identifier for the CFMachPort opaque type.

Functions

CFMachPortCreate

Creates a CFMachPort object with a new Mach port.

```
CFMachPortRef CFMachPortCreate (
   CFAllocatorRef allocator,
   CFMachPortCallBack callout,
   CFMachPortContext *context,
   Boolean *shouldFreeInfo
);
```

Parameters

allocator
: The allocator to use to allocate memory for the new object. Pass `NULL` or `kCFAllocatorDefault` to use the current default allocator.

callout
: The callback function invoked when a message is received on the new Mach port.

context

A structure holding contextual information for the new Mach port. The function copies the information out of the structure, so the memory pointed to by *context* does not need to persist beyond the function call.

shouldFreeInfo

A flag set by the function to indicate whether the `info` member of *context* should be freed. The flag is set to `true` on failure, `false` otherwise. *shouldFreeInfo* can be `NULL`.

Return Value

The new CFMachPort object or `NULL` on failure. The CFMachPort object has both send and receive rights. Ownership follows the Create Rule.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`CFMachPort.h`

CFMachPortCreateRunLoopSource

Creates a CFRunLoopSource object for a CFMachPort object.

```
CFRunLoopSourceRef CFMachPortCreateRunLoopSource (
   CFAllocatorRef allocator,
   CFMachPortRef port,
   CFIndex order
);
```

Parameters

allocator

The allocator to use to allocate memory for the new object. Pass `NULL` or `kCFAllocatorDefault` to use the current default allocator.

port

The Mach port for which to create a CFRunLoopSource object.

order

A priority index indicating the order in which run loop sources are processed. *order* is currently ignored by CFMachPort run loop sources. Pass `0` for this value.

Return Value

The new CFRunLoopSource object for *port*. Ownership follows the Create Rule.

Discussion

The run loop source is not automatically added to a run loop. To add the source to a run loop, use `CFRunLoopAddSource`.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

NotifyTool

Declared In

`CFMachPort.h`

CFMachPortCreateWithPort

Creates a CFMachPort object for a pre-existing native Mach port.

```
CFMachPortRef CFMachPortCreateWithPort (
   CFAllocatorRef allocator,
   mach_port_t portNum,
   CFMachPortCallBack callout,
   CFMachPortContext *context,
   Boolean *shouldFreeInfo
);
```

Parameters

allocator

The allocator to use to allocate memory for the new object. Pass `NULL` or `kCFAllocatorDefault` to use the current default allocator.

portNum

The native Mach port to use.

callout

The callback function invoked when a message is received on the Mach port.

context

A structure holding contextual information for the Mach port. The function copies the information out of the structure, so the memory pointed to by *context* does not need to persist beyond the function call.

shouldFreeInfo

A flag set by the function to indicate whether the `info` member of *context* should be freed. The flag is set to `true` on failure or if a CFMachPort object already exists for *portNum*, `false` otherwise. *shouldFreeInfo* can be `NULL`.

Return Value

The new CFMachPort object or `NULL` on failure. If a CFMachPort object already exists for *portNum*, the function returns the pre-existing object instead of creating a new object; the *context* and *callout* parameters are ignored in this case. Ownership follows the Create Rule.

Discussion

The CFMachPort object does not take full ownership of the send and receive rights of the Mach port *portNum*. It is the caller's responsibility to deallocate the Mach port rights after the CFMachPort object is no longer needed and has been invalidated.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

NotifyTool

Declared In

`CFMachPort.h`

CFMachPortGetContext

Returns the context information for a CFMachPort object.

```
void CFMachPortGetContext (
   CFMachPortRef port,
   CFMachPortContext *context
);
```

Parameters

port

The CFMachPort object to examine.

context

A pointer to the structure into which the context information for *port* is to be copied. The information being returned is usually the same information you passed to `CFMachPortCreate` (page 6) or `CFMachPortCreateWithPort` (page 8) when creating *port*. However, if `CFMachPortCreateWithPort` (page 8) returned a cached CFMachPort object instead of creating a new object, *context* is filled with information from the original CFMachPort object instead of the information you passed to the function.

Discussion

The context version number for CFMachPort objects is currently `0`. Before calling this function, you need to initialize the `version` member of *context* to `0`.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`CFMachPort.h`

CFMachPortGetInvalidationCallBack

Returns the invalidation callback function for a CFMachPort object.

```
CFMachPortInvalidationCallBack CFMachPortGetInvalidationCallBack (
   CFMachPortRef port
);
```

Parameters

port

The CFMachPort object to examine.

Return Value

The callback function invoked when *port* is invalidated. `NULL` if no callback has been set with `CFMachPortSetInvalidationCallBack` (page 11).

Availability

Available in Mac OS X v10.0 and later.

Declared In

`CFMachPort.h`

CFMachPortGetPort

Returns the native Mach port represented by a CFMachPort object.

```
mach_port_t CFMachPortGetPort (
   CFMachPortRef port
);
```

Parameters

`port`

The CFMachPort object to examine.

Return Value

The native Mach port represented by *`port`*.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`CFMachPort.h`

CFMachPortGetTypeID

Returns the type identifier for the CFMachPort opaque type.

```
CFTypeID CFMachPortGetTypeID (
   void
);
```

Return Value

The type identifier for the CFMachPort opaque type.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`CFMachPort.h`

CFMachPortInvalidate

Invalidates a CFMachPort object, stopping it from receiving any more messages.

```
void CFMachPortInvalidate (
   CFMachPortRef port
);
```

Parameters

`port`

The CFMachPort object to invalidate.

Discussion

Invalidating a CFMachPort object prevents the port from ever receiving any more messages. The CFMachPort object is not deallocated, though. If the port has not already been invalidated, the port's invalidation callback function is invoked, if one has been set with `CFMachPortSetInvalidationCallBack` (page 11). The `CFMachPortContext` (page 13) `info` information for *`port`* is also released, if a release callback was specified in the port's context structure. Finally, if a run loop source was created for *`port`*, the run loop source is invalidated, as well.

If the underlying Mach port is destroyed, the CFMachPort object is automatically invalidated.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`CFMachPort.h`

CFMachPortIsValid

Returns a Boolean value that indicates whether a CFMachPort object is valid and able to receive messages.

```
Boolean CFMachPortIsValid (
   CFMachPortRef port
);
```

Parameters

`port`

The CFMachPort object to examine.

Return Value
`true` if *`port`* can be used for communication, otherwise `false`.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`CFMachPort.h`

CFMachPortSetInvalidationCallBack

Sets the callback function invoked when a CFMachPort object is invalidated.

```
void CFMachPortSetInvalidationCallBack (
   CFMachPortRef port,
   CFMachPortInvalidationCallBack callout
);
```

Parameters

`port`

The CFMachPort object to modify.

`callout`

The callback function to invoke when *`port`* is invalidated. Pass `NULL` to remove a callback.

Discussion
If *`port`* is already invalid, *`callout`* is invoked immediately.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`CFMachPort.h`

Callbacks

CFMachPortCallBack

Callback invoked to process a message received on a CFMachPort object.

```
typedef void (*CFMachPortCallBack) (
    CFMachPortRef port,
    void *msg,
    CFIndex size,
    void *info
);
```

If you name your function `MyCallBack`, you would declare it like this:

```
void MyCallBack (
    CFMachPortRef port,
    void *msg,
    CFIndex size,
    void *info
);
```

Parameters

`port`
: The CFMachPort object on which the message *`msg`* was received.

`msg`
: The Mach message received on *`port`*. The pointer is to a `mach_msg_header_t` structure.

`size`
: Size of the Mach message *`msg`*, excluding the message trailer.

`info`
: The `info` member of the `CFMachPortContext` (page 13) structure used when creating *`port`*.

Discussion

You specify this callback when creating a CFMachPort object with either `CFMachPortCreate` (page 6) or `CFMachPortCreateWithPort` (page 8). To receive messages on a CFMachPort object (and have this callback invoked), you must create a run loop source for the port and add it to a run loop.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`CFMachPort.h`

CFMachPortInvalidationCallBack

Callback invoked when a CFMachPort object is invalidated.

```
typedef void (*CFMachPortInvalidationCallBack) (
    CFMachPortRef port,
    void *info
);
```

If you name your function `MyCallBack`, you would declare it like this:

```
void MyCallBack (
    CFMachPortRef port,
    void *info
);
```

Parameters

`port`

The CFMachPort object that has been invalidated.

`info`

The `info` member of the `CFMachPortContext` (page 13) structure used when creating *`port`*.

Discussion

Your callback should free any resources allocated for *`port`*.

You specify this callback with `CFMachPortSetInvalidationCallBack` (page 11).

Availability

Available in Mac OS X v10.0 and later.

Declared In

`CFMachPort.h`

Data Types

CFMachPortContext

A structure that contains program-defined data and callbacks with which you can configure a CFMachPort object's behavior.

```
struct CFMachPortContext {
    CFIndex version;
    void *info;
    CFAllocatorRetainCallBack retain;
    CFAllocatorReleaseCallBack release;
    CFAllocatorCopyDescriptionCallBack copyDescription;
};
typedef struct CFMachPortContext CFMachPortContext;
```

Fields

`version`

Version number of the structure. Must be `0`.

`info`

An arbitrary pointer to program-defined data, which can be associated with the CFMachPort object at creation time. This pointer is passed to all the callbacks defined in the context.

`retain`
: A retain callback for your program-defined `info` pointer. Can be `NULL`.

`release`
: A release callback for your program-defined `info` pointer. Can be `NULL`.

`copyDescription`
: A copy description callback for your program-defined `info` pointer. Can be `NULL`.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`CFMachPort.h`

CFMachPortRef

A reference to a CFMachPort object.

```
typedef struct __CFMachPort *CFMachPortRef;
```

Availability
Available in Mac OS X v10.0 and later.

Declared In
`CFMachPort.h`

Document Revision History

This table describes the changes to *CFMachPort Reference*.

Date	Notes
2008-04-08	Corrected description of CFMachPortCallBack's "msg" parameter.
2007-03-20	Added note regarding invalidation of CFMachPort object.
2006-02-07	Made formatting changes.
2005-11-09	Removed reference to retired document.
2005-08-11	Cosmetic changes to conform to documentation guidelines.
2003-01-01	First version of this document.

Index