sd_bus_send

- Description
- Return Value
  - Errors
- History
- See Also

systemd

sd_bus_send

3

sd_bus_send

sd_bus_send_to

sd_bus_message_send

Queue a D-Bus message for transfer

#include <systemd/sd-bus.h>

int

sd_bus_send

sd_bus *

bus

sd_bus_message *

m

uint64_t *

cookie

int

sd_bus_send_to

sd_bus *

bus

sd_bus_message *

m

const char *

destination

uint64_t *

cookie

int sd_bus_message_send

sd_bus_message *

m

Description

`sd_bus_send()` queues the bus message object `m` for transfer. If `bus` is `NULL`, the bus that `m` is attached to is used. `bus` only needs to be set when the message is sent to a different bus than the one it is attached to, for example when forwarding messages. If the output parameter `cookie` is not `NULL`, it is set to the message identifier. This value can later be used to match incoming replies to their corresponding messages. If `cookie` is set to `NULL` and the message is not sealed, `sd_bus_send()` assumes the message `m` does not expect a reply and adds the necessary headers to indicate this.

Note that in most scenarios, `sd_bus_send()` should not be called directly. Instead, use higher level functions such as `sd_bus_call_method(3)` and `sd_bus_reply_method_return(3)` which call `sd_bus_send()` internally.

`sd_bus_send_to()` is a shorthand for sending a message to a specific destination. It's main use case is to simplify sending unicast signal messages (signals that only have a single receiver). It's behavior is similar to calling `sd_bus_message_set_destination(3)` followed by calling `sd_bus_send()`.

`sd_bus_send()`/`sd_bus_send_to()` will write the message directly to the underlying transport (e.g. kernel socket buffer) if possible. If the connection is not set up fully yet the message is queued locally. If the transport buffers are congested any unwritten message data is queued locally, too. If the connection has been closed or is currently being closed the call fails. `sd_bus_process(3)` should be invoked to write out any queued message data to the transport.

`sd_bus_message_send()` is the same as `sd_bus_send()` but without the first and last argument. `sd_bus_message_send(m)` is equivalent to `sd_bus_send(sd_bus_message_get_bus(m), m, NULL)`.

Return Value

On success, these functions return a non-negative integer. On failure, they return a negative errno-style error code.

Errors

Returned errors may indicate the following problems:

`-EINVAL`
The input parameter `m` is `NULL`.

`-EOPNOTSUPP`
The bus connection does not support sending file descriptors.

`-ECHILD`
The bus connection was allocated in a parent process and is being reused in a child process after `fork()`.

`-ENOBUFS`
The bus connection's write queue is full.

`-ENOTCONN`
The input parameter `bus` is `NULL` or the bus is not connected.

`-ECONNRESET`
The bus connection was closed while waiting for the response.

`-ENOMEM`
Memory allocation failed.

History

`sd_bus_send()` and `sd_bus_send_to()` were added in version 246.

See Also

`systemd(1)`, `sd-bus(3)`, `sd_bus_call_method(3)`, `sd_bus_message_set_destination(3)`, `sd_bus_reply_method_return(3)`, `sd_bus_process(3)`

□□□ #2
□ □□□ □□□ 2026-10-31 14:52:09 UTC
□ □□□ □□□ 2026-10-31 14:52:10 UTC