sd_bus_start

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

systemd

sd_bus_start

3

sd_bus_start

Initiate a bus connection to the D-bus broker daemon

#include <systemd/sd-bus.h>

int

sd_bus_start

sd_bus *

bus

Description

`sd_bus_start()` connects an existing bus connection object to the D-Bus broker daemon, usually `dbus-daemon(1)` or `dbus-broker(1)`. The mechanism to use for the connection must be configured before the call to `sd_bus_start()`, using one of `sd_bus_set_address(3)`, `sd_bus_set_fd(3)`, or `sd_bus_set_exec(3)`. `sd_bus_start()` will open the connection socket or spawn the executable as needed, and asynchronously start a `org.freedesktop.DBus.Hello()` call. The answer to the Hello call will be processed later from `sd_bus_process(3)`. If opening of the connection or queuing of the

asynchronous call fail, the connection will be closed with `sd_bus_close(3)`.

In most cases, it is better to use `sd_bus_default_user(3)`, `sd_bus_default_system(3)` or related calls instead of the more low-level `sd_bus_new()` and `sd_bus_start()`. The higher-level functions not only allocate a bus object but also start the connection to a well-known bus in a single function call.

Return Value

On success, this function returns a non-negative integer. On failure, it returns a negative errno-style error code.

Errors

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

`-ENOPKG`
Bus object `bus` could not be resolved.

`-EPERM`
The input parameter `bus` is in a wrong state (`sd_bus_start()` may only be called once on a newly-created bus object).

`-ECHILD`
The bus object `bus` was created in a different process.

In addition, other connection-related errors may be returned. See `sd_bus_send(3)`.

History

`sd_bus_start()` was added in version 246.

See Also

`systemd(1)`, `sd-bus(3)`, `sd_bus_default(3)`, `sd_bus_call_async(3)`

□□□ #2
□ □□□ □□□ 2026-10-31 14:53:22 UTC

□ □□□ □□□ 2026-10-31 14:54:11 UTC