

**NAME**

`dbus-send` – Send a message to a message bus

**SYNOPSIS**

```
dbus-send [--system | --session | --bus=ADDRESS | --peer=ADDRESS] [--dest=NAME]
          [--print-reply [=literal]] [--reply-timeout=MSEC] [--type=TYPE] OBJECT_PATH
          INTERFACE.MEMBER [CONTENTS...]
```

**DESCRIPTION**

The `dbus-send` command is used to send a message to a D-Bus message bus. See <http://www.freedesktop.org/software/dbus/> for more information about the big picture.

There are two well-known message buses: the systemwide message bus (installed on many systems as the "messagebus" service) and the per-user-login-session message bus (started each time a user logs in). The `--system` and `--session` options direct `dbus-send` to send messages to the system or session buses respectively. If neither is specified, `dbus-send` sends to the session bus.

Nearly all uses of `dbus-send` must provide the `--dest` argument which is the name of a connection on the bus to send the message to. If `--dest` is omitted, no destination is set.

The object path and the name of the message to send must always be specified. Following arguments, if any, are the message contents (message arguments). These are given as type-specified values and may include containers (arrays, dicts, and variants) as described below.

```
<contents> ::= <item> | <container> [ <item> | <container>... ]
<item>     ::= <type>:<value>
<container> ::= <array> | <dict> | <variant>
<array>    ::= array:<type>:<value>[,<value>...]
<dict>     ::= dict:<type>:<type>:<key>,<value>[,<key>,<value>...]
<variant>  ::= variant:<type>:<value>
<type>     ::= string | int16 | uint16 | int32 | uint32 | int64 | uint64 | double | byte | boolean | objpath
```

D-Bus supports more types than these, but `dbus-send` currently does not. Also, `dbus-send` does not permit empty containers or nested containers (e.g. arrays of variants).

Here is an example invocation:

```
dbus-send --dest=org.freedesktop.ExampleName \
/org/freedesktop/sample/object/name \
org.freedesktop.ExampleInterface.ExampleMethod \
int32:47 string:'hello world' double:65.32 \
array:string:"1st item","next item","last item" \
dict:string:int32:"one",1,"two",2,"three",3 \
variant:int32:-8 \
objpath:/org/freedesktop/sample/object/name
```

Note that the interface is separated from a method or signal name by a dot, though in the actual protocol the interface and the interface member are separate fields.

**OPTIONS**

The following options are supported:

`--dest=NAME`

Specify the name of the connection to receive the message.

`--print-reply`

Block for a reply to the message sent, and print any reply received in a human-readable form. It also means the message type (`--type=`) is `method_call`.

`--print-reply=literal`

Block for a reply to the message sent, and print the body of the reply. If the reply is an object path or a string, it is printed literally, with no punctuation, escape characters etc.

**--reply-timeout=MSEC**

Wait for a reply for up to *MSEC* milliseconds. The default is implementation-defined, typically 25 seconds.

**--system**

Send to the system message bus.

**--session**

Send to the session message bus. (This is the default.)

**--bus=ADDRESS**

Register on a message bus at *ADDRESS*, typically a **dbus-daemon**.

**--peer=ADDRESS**

Send to a non-message-bus D-Bus server at *ADDRESS*. In this case **dbus-send** will not call the Hello method.

**--type=TYPE**

Specify **method\_call** or **signal** (defaults to "signal").

## AUTHOR

dbus-send was written by Philip Blundell.

## BUGS

Please send bug reports to the D-Bus mailing list or bug tracker, see <http://www.freedesktop.org/software/dbus/>