NAME

Net::DBus::Binding::Bus − Handle to a well−known message bus instance

SYNOPSIS

```
use Net::DBus::Binding::Bus;

# Get a handle to the system bus
my $bus = Net::DBus::Binding::Bus->new(type => &Net::DBus::Binding::Bus::SYSTEM
```

DESCRIPTION

This is a specialization of the Net::DBus::Binding::Connection module providing convenience constructor for connecting to one of the well-known bus types. There is no reason to use this module directly, instead get a handle to the bus with the `session` or `system` methods in Net::DBus.

METHODS

my `$bus` = Net::DBus::Binding::Bus−>new(type => `$type`);

my `$bus` = Net::DBus::Binding::Bus−>new(address => `$addr`);

Open a connection to a message bus, either a well known bus type specified using the `type` parameter, or an arbitrary bus specified using the `address` parameter. If the `private` parameter is set to a true value, then a private connection to the bus is obtained. The caller must explicitly disconnect this bus instance before releasing the last instance of the object.

`$bus`−>request_name($service_name)

Send a request to the bus registering the well known name specified in the `$service_name` parameter. If another client already owns the name, registration will be queued up, pending the exit of the other client.

my `$name` = `$bus`−>get_unique_name

Returns the unique name by which this processes' connection to the bus is known. Unique names are never re-used for the entire lifetime of the bus daemon.

`$bus`−>add_match($rule)

Register a signal match rule with the bus controller, allowing matching broadcast signals to routed to this client.

`$bus`−>remove_match($rule)

Unregister a signal match rule with the bus controller, preventing further broadcast signals being routed to this client

AUTHOR

Daniel P. Berrange

COPYRIGHT

SEE ALSO

Net::DBus::Binding::Connection, Net::DBus