NAME

Net::DBus::Annotation − annotations for changing behaviour of APIs

SYNOPSIS

```
use Net::DBus::Annotation qw(:call);

my $object = $service->get_object("/org/example/systemMonitor");

# Block until processes are listed
my $processes = $object->list_processes("someuser");

# Just throw away list of processes, pretty pointless
# in this example, but useful if the method doesn't have
# a return value
$object->list_processes(dbus_call_noreply, "someuser");

# List processes & get on with other work until
# the list is returned.
my $asyncreply = $object->list_processes(dbus_call_async, "someuser");

... some time later...
my $processes = $asyncreply->get_data;


# List processes, with a shorter 10 second timeout, instead of
# the default 60 seconds
my $object->list_processes(dbus_call_timeout, 10 * 1000, "someuser");
```

DESCRIPTION

This module provides a number of annotations which will be useful when dealing with the DBus APIs. There are annotations for switching remote calls between sync, async and no-reply mode. More annotations may be added over time.

METHODS

dbus_call_sync

Requests that a method call be performed synchronously, waiting for the reply or error return to be received before continuing.

dbus_call_async

Requests that a method call be performed a−synchronously, returning a pending call object, which will collect the reply when it eventually arrives.

dbus_call_noreply

Requests that a method call be performed a−synchronously, discarding any possible reply or error message.

dbus_call_timeout

Indicates that the next parameter for the method call will specify the time to wait for a reply in milliseconds. If omitted, then the default timeout for the object will be used

AUTHOR

Daniel Berrange <dan@berrange.com>

COPYRIGHT

Copright (C) 2006−2011, Daniel Berrange.

SEE ALSO

Net::DBus, Net::DBus::RemoteObject