

NAME

Net::DBus::Error – Error details for remote method invocation

SYNOPSIS

```
package Music::Player::UnknownFormat;

use base qw(Net::DBus::Error);

# Define an error type for unknown track encoding type
# for a music player service
sub new {
    my $proto = shift;
    my $class = ref($proto) || $proto;
    my $self = $class->SUPER::new(name => "org.example.music.UnknownFormat",
                                   message => "Unknown track encoding format");
}

package Music::Player::Engine;

...snip...

# Play either mp3 or ogg music tracks, otherwise
# thrown an error
sub play {
    my $self = shift;
    my $url = shift;

    if ($url =~ /\.(mp3|ogg)$/ ) {
        ...play the track
    } else {
        die Music::Player::UnknownFormat->new();
    }
}
```

DESCRIPTION

This objects provides for strongly typed error handling. Normally a service would simply call

```
die "some message text"
```

When returning the error condition to the calling DBus client, the message is associated with a generic error code or “org.freedesktop.DBus.Failed”. While this suffices for many applications, occasionally it is desirable to be able to catch and handle specific error conditions. For such scenarios the service should create subclasses of the `Net::DBus::Error` object providing in a custom error name. This error name is then sent back to the client instead of the generic “org.freedesktop.DBus.Failed” code.

METHODS

```
my $error = Net::DBus::Error->new(name => $error_name, message => $description);
```

Creates a new error object whose name is given by the `name` parameter, and long descriptive text is provided by the `message` parameter. The `name` parameter has certain formatting rules which must be adhered to. It must only contain the letters ‘a’-‘Z’, ‘0’-‘9’, ‘-’, ‘_’ and ‘.’. There must be at least two components separated by a ‘.’, For example a valid name is ‘org.example.Music.UnknownFormat’.

```
$error->name
```

Returns the DBus error name associated with the object.

```
$error->message
```

Returns the descriptive text/message associated with the error condition.

`$error->stringify`

Formats the error as a string in a manner suitable for printing out / logging / displaying to the user, etc.

AUTHOR

Daniel P. Berrange

COPYRIGHT

Copyright (C) 2005–2011 Daniel P. Berrange

SEE ALSO

Net::DBus, Net::DBus::Object