### **NAME**

Net::DBus::ASyncReply - asynchronous method reply handler

### **SYNOPSIS**

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

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

# List processes & get on with other work until

# the list is returned.

my $asyncreply = $object->list_processes(dbus_call_async, "someuser");

while (!$asyncreply->is_ready) {
    ... do some background work..
}

my $processes = $asyncreply->get_result;
```

### DESCRIPTION

This object provides a handler for receiving asynchronous method replies. An asynchronous reply object is generated when making remote method call with the dbus\_call\_async annotation set.

# **METHODS**

```
$asyncreply->discard_result;
```

Indicates that the caller is no longer interested in receiving the reply & that it should be discarded. After calling this method, this object should not be used again.

```
$asyncreply->wait_for_result;
```

Blocks the caller waiting for completion of the of the asynchronous reply. Upon returning from this method, the result can be obtained with the get result method.

```
my $boolean = $asyncreply->is_ready;
```

Returns a true value if the asynchronous reply is now complete (or a timeout has occurred). When this method returns true, the result can be obtained with the get\_result method.

```
$asyncreply->set_notify($coderef);
```

Sets a notify function which will be invoked when the asynchronous reply finally completes. The callback will be invoked with a single parameter which is this object.

```
my @data = $asyncreply->get_result;
```

Retrieves the data associated with the asynchronous reply. If a timeout occurred, then this method will throw an exception. This method can only be called once the reply is complete, as indicated by the <code>is\_ready</code> method returning a true value. After calling this method, this object should no longer be used.

## **AUTHOR**

Daniel Berrange < dan@berrange.com>

# **COPYRIGHT**

Copright (C) 2006–2011, Daniel Berrange.

# **SEE ALSO**

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