NAME

Net::Bonjour − Module for DNS service discovery (Apple's Bonjour)

SYNOPSIS

```
use Net::Bonjour;

my $res = Net::Bonjour->new(<service>[, <protocol>]);

$res->discover;

foreach my $entry ( $res->entries ) {
        printf "%s %s:%s\n", $entry->name, $entry->address, $entry->port;
}
```

Or the cyclical way:

```
use Net::Bonjour;

my $res = Net::Bonjour->new(<service>[, <protocol>]);

$res->discover;

while ( 1 ) {
   foreach my $entry ( $res->entries ) {
           print $entry->name, "\n";
   }
   $res->discover;
}
```

DESCRIPTION

Net::Bonjour is a set of modules that allow one to discover local services via multicast DNS (mDNS) or enterprise services via traditional DNS. This method of service discovery has been branded as Bonjour by Apple Computer.

Base Object

The base object would be of the Net::Bonjour class. This object contains the resolver for DNS service discovery.

Entry Object

The base object (Net::Bonjour) will return entry objects of the class Net::Bonjour::Entry.

METHODS

new([<service>, <protocol>, <domain>])

Creates a new Net::Bonjour discovery object. First argument specifies the service to discover, e.g. http, ftp, afpovertcp, and ssh. The second argument specifies the protocol, i.e. tcp or udp. *The default protocol is TCP*. The third argument specifies the discovery domain, the default is 'local'.

If no arguments are specified, the resulting Net::Bonjour object will be empty and will not perform an automatic discovery upon creation.

all_services([<domain>])

Returns an array of new Net::Renedezvous objects for each service type advertised in the domain. The argument specifies the discovery domain, the default is 'local'. Please note that the resulting Net::Bonjour objects will not have performed a discovery during the creation. Therefore, the discovery process will need to be run prior to retriving a list of entries for that Net::Bonjour object.

domain([<domain>])

Get/sets current discovery domain. By default, the discovery domain is 'local'. Discovery for the 'local' domain is done via MDNS while all other domains will be done via traditional DNS.

discover

Repeats the discovery process and reloads the entry list from this discovery.

entries

Returns an array of Net::Renedezvous::Entry objects for the last discovery.

protocol([<protocol>])
Get/sets current protocol of the service type, i.e. TCP or UDP. Please note that this is not the protocol for DNS connection.

service([<service type>])
Get/sets current service type.

shift_entry
Shifts off the first entry of the last discovery. The returned object will be a Net::Bonjour::Entry object.

EXAMPLES

Print out a list of local websites

```
print "<HTML><TITLE>Local Websites</TITLE>";

use Net::Bonjour;

my $res = Net::Bonjour->new('http');
$res->discover;

foreach my $entry ( $res->entries) {
        printf "<A HREF='http://%s%s'>%s</A><BR>", $entry->address,
                $entry->attribute('path'), $entry->name;
}

print "</HTML>";
```

Find a service and connect to it

```
use Socket;
use Net::Bonjour;

my $res = Net::Bonjour->new('custom');
$res->discover;

my $entry = $res->shift_entry;

socket SOCK, PF_INET, SOCK_STREAM, scalar(getprotobyname('tcp'));

connect SOCK, $entry->sockaddr;

print SOCK "Send a message to the service";

while ($line = <SOCK>) { print $line; }

close SOCK;
```

Find all service types and print.

```
use Net::Bonjour;

foreach my $res ( Net::Bonjour->all_services ) {
        printf "%s (%s)\n", $res->service, $res->protocol;
}
```

Find and print all service types and entries.

```
use Net::Bonjour;

foreach my $res ( Net::Bonjour->all_services ) {
        printf "-- %s (%s) ---\n", $res->service, $res->protocol;
        $res->discover;
        foreach my $entry ( $res->entries) {
                printf "\t%s (%s:%s)\n", $entry->name, $entry->address, $
        }
}
```

SEE ALSO

Net::Bonjour::Entry

COPYRIGHT

AUTHORS

The Net::Bonjour module was created by George Chlipala <george AT walnutcs DOT com>