

[Network Extension](#) / App proxy provider

API Collection

# App proxy provider

Implement a VPN client for a flow-oriented, custom VPN protocol.

## Overview

A virtual private network (VPN) is a form of network tunnel where a VPN client uses the public internet to create a connection to a VPN server and then passes private network traffic over that connection. If you want to build a VPN client that implements a flow-oriented, custom VPN protocol—one that works with the data passing through a transmission control protocol (TCP) connection rather than the packets used to transport that data—create an app proxy provider app extension.

When the system starts a VPN configuration that uses your app proxy provider, it performs the following steps:

- Launches your app extension.
- Instantiates your proxy provider subclass within that app extension.
- Starts forwarding flows to your provider.

Each flow represents either a TCP connection or a conversation over user datagram protocol (UDP). Your provider should to open a tunnel to a VPN server and forward each flow over that tunnel. Similarly, if your provider receives flow data from the tunnel, it should pass that back to the system through the appropriate flow.

App proxy providers are one form of per-app VPN, the other being a [Packet tunnel provider](#) in source application mode.

For detailed information about app proxy provider deployment options, see [TN3134: Network Extension provider deployment](#).

### Note

When a VPN configuration is active, connections use the VPN instead of iCloud Private Relay. Network Extension providers also don't use iCloud Private Relay.

# Topics

## Essentials

### Network Extensions Entitlement

The APIs an app can use to customize networking features.

## App proxy provider

`class NEAppProxyProvider`

The principal class for an app proxy provider app extension.

`class NETunnelProvider`

An abstract base class shared by NEPacketTunnelProvider and NEAppProxyProvider.

`class NEProvider`

An abstract base class for all NetworkExtension providers.

`class NETunnelNetworkSettings`

The configuration for a tunnel provider's virtual interface.

## Flow handling

`class NEAppProxyTCPFlow`

An object for reading and writing data to and from a TCP connection being proxied by the provider.

`class NEAppProxyUDPFlow`

An object for reading and writing data to and from a UDP conversation being proxied by the provider.

`class NEAppProxyFlow`

An abstract base class shared by NEAppProxyTCPFlow and NEAppProxyUDPFlow.

## `class NEFlowMetadata`

Additional information about data flowing through a per-app VPN provider.

## In-Provider Networking

Network APIs for use by all types of NetworkExtension providers and by hotspot helpers.

## Handling Flow Copying

Exchange data streams by using proxy-provider classes.

# VPN configuration

## `class NEAppProxyProviderManager`

An object to create and manage the app proxy provider's VPN configuration.

## `class NETunnelProviderManager`

An object to create and manage the tunnel provider's VPN configuration.

## `class NEVPNManager`

An object to create and manage a Personal VPN configuration.

## `class NETunnelProviderProtocol`

Configuration parameters for a VPN tunnel.

## `class NEAppRule`

The identity of an app whose traffic is to be routed through the tunnel.

## VPN On Demand Rules

Set up VPN On Demand.

# VPN control

## `class NETunnelProviderSession`

An object to start and stop a tunnel connection and get its status.

## `class NEVPNConnection`

An object to start and stop a Personal VPN connection and get its status.

# Transparent proxy configuration

## `class NETransparentProxyManager`

An object that configures and controls transparent proxies.

`class NETransparentProxyProvider`

An object that implements the client side of a custom transparent network proxy solution.

`class NETransparentProxyNetworkSettings`

A specification of what traffic to route through a transparent proxy.

`class NENetworkRule`

A rule to match attributes of network traffic.

---

## See Also

### Virtual private networks

 **Routing your VPN network traffic**

Configure your VPN to include and exclude some network traffic.

 **Personal VPN**

Create and manage a VPN configuration that uses one of the built-in VPN protocols (IPsec or IKEv2).

 **Packet tunnel provider**

Implement a VPN client for a packet-oriented, custom VPN protocol.