



Hiveware® for Ipv4Ipv6MFC

The **Hiveware® for Ipv4Ipv6MFC** library is a part of the Hiveware engine. It was anticipated during the development of the Hiveware engine, that Microsoft would be upgrading their aging C++ [MFC](#) framework to work with Ipv6. As of this writing, that has not happened. Since Hiveware is designed to operate with ordinary customers (public residential) as well as for enterprises that develop their own apps using the Hiveware framework, Hiveware Inc was forced to schedule and accomplish its development. As you will see below, during the year-long development, it was decided to include the implementation of [IETF's RFC7217](#) standard. This is essential to maintaining Hiveware's 3-pronged functionality: component-level security, privacy and ownership.

FOR SALE

The **Hiveware® for Ipv4Ipv6MFC** library, which is a part of the [Hiveware](#) engine. Since existing and future MFC products will all need this functionality, Hiveware Inc has decided to offer this Ipv4/Ipv6 MFC communications library to any customer wishing to protect their MFC application – legacy or new – from being left behind when Ipv6 deployment surpasses Ipv4.

Source code is delivered with continual upgrades using the Hiveware framework technology. Use of the library is royalty free and the purchase price plus optional yearly maintenance fees are modest. The library is licensed individually to each developer.

OVERVIEW

Given that MFC is almost 30 years old and given that Microsoft has been focusing on cloud and web technologies, they have not updated MFC properly, which is why current MFC is still based on Ipv4-only [Winsock](#) 1.0. This has not been a problem, however, until recently, where world adoption of Ipv6 is approaching 50%. It is widely proclaimed that there will be a tipping point when Ipv4 will be deprecated.

Existing MFC enterprise solutions are not ready for that, and in fact, may be a large part of why Ipv6 adoption has taken so long. These legacy MFC apps constitute considerable inertia regarding Ipv6 adoption.

Hiveware development as such last year had to be suspended. MFC's Ipv6 problem had to be resolved. Now that development and testing has been completed.

MARKET

An alternate future to the current web client-server, centralized architecture is decentralized applications. In practice, such a market would include residential as well as business clients. [NAT](#) and [Ipv4 port forwarding](#) could not be allowed to stop residential customers from signing up for Hiveware [dapp](#) products. MFC had to work with Ipv6 or go out of business. Although billions of investment capital are today being thrown at decentralized applications for various reasons, that is not the intended market for Ipv4Ipv6MFC. Rather, it is the many existing C++-based MFC-based so-called legacy applications that in the US run many banks and utilities today. Also, new apps that want to build their own point-2-point communications capabilities will be able to skip this time-and-money expensive - not to mention error-prone – comms development.

THE SOLUTION

The development task boiled down to eliminating MFC's dependency on [CAsyncSocket](#). For Ipv6 security, [RFC7217](#) was also implemented. To make the library durable across all [ISPs](#) with their different (some non-existent) and changing implementations of Ipv6, the library was designed to gracefully degrade at runtime, step by step back to Ipv4 depending on the runtime quality of the customer's internet service.

Features

- Enables enterprises to upgrade their legacy MFC apps to obtain Ipv6 functionality.
- Enable IoT components to be directly accessible via Global Unique TCP Addresses without the limitation of servers.
- Eliminates MFC's Ipv4-bound [CAsyncSocket](#) class.
- Creates Global Unique Addresses (GUA) for the TCP/IP protocol.
- Asynchronous (optional) logging.
- 128-bit application defined value (secret key) which means you own the data being sent to the receiver and not the Windows operating system.
- Implemented using WinSock 2.2 generic send/receive calls: `::recv`, `::send`, `::bind`, `::select`, `::listen`, `::connect`. Does not use [WSASocket](#) functions because the code needs to be easily upgradable to the upcoming standard **ISO C++23 Networking** when it emerges.
- Takes advantage of Microsoft's ip-family-agnostic functions, that is, socket calls that operate with either the `AF_INET` or `AF_INET6` address families.
- The library has been stress tested to over 100K round trip iterations (Debug and Release).

- The library has been partially tested using the massively multi-threaded Hiveware framework.
- The library is able to gracefully degrade its connections at runtime depending on the individual user's network capabilities.
- The library is able to specify as a property which internet types to prioritize (like Ethernet over Wireless when available) when runtime iterating through interfaces.
- Implements the [RFC7217](#) IETF standard.
- Ipv4Ipv6MFC library delivers continually released source code for a yearly fee.

CONTACT US

For more information contact us at ipv4ipv6mfc@hiveware.com.