# Tritium Tools vs Prohelion Profinity

The Tritium tools and Profinity provide similar functionality, but work in fundamentally different ways, here are some important things to understand.

## How the tools are similar

Both the Tritium tools and Profinity use the same CAN Bus messages to achieve their outcomes, they both use UDP and TCP but the manner in which they use them is different.

In many ways they are more similar than different, but some of the differences are important to understand if you are migrating from the Tritium tools to the Prohelion Profinity tools.

## How the Tritium and Profinity tools are different

Profinity is a ground up rewrite of the Tritium tools and as such while the underlying functionality is similar, the technical code used to achieve the outcomes is completely different.

The Tritium tools are in written in C++ and Prohelion tools in C#, the network libraries used to talk CAN Bus are also completely different.

Even though the Prohelion libraries have a common heritage with some of the Tritium tools, they have been heavily modified for the use at Prohelion and little of the original code remains.

There are also a couple of key ways in which the tools are completely different in their behaviours.

### Profinity is an all in one solution

Tritiums uses multiple tools that operate independently. Why is this important?

Say for example you are using the Tritium CAN Bus logger while updating your BMS. In the Tritium tools this is two different executables with two different network connections to the same device, running the Can Bridge Config tool as well, then you have a third connection.

In Profinity, all the tools are sharing a single connection and running within a single executable. The different is subtle, but also has many flow on effects on how things operate, particularly around networking and the issues outlined in the next point.

## Tritium tools exclusively lock network ports

The Prohelion tools do no not. This means that it is often not possible to run multiple copies of the Tritium tools at once, but it is possible to run multiple copies of Profinity at once. If Profinity is running, a Tritium tool will often not work because it is trying to grab exclusive use of a port that is already in use.

However, Profinity will generally run when a Tritium tool is already running as it is not concerned with exclusivity.

## Tritium tools have the concept of an ‘Active Adapter’

Profinity does not have active adapters, it uses all the active adapters configured in your current profile to operate.

## The Tritium tools connect to one PC network adapter

Unfortunately, it’s not always the right one.

Ever used the Tritium tools and found that you just can’t see data even though Profinity or WireShark can?

The issue is usually that the Tritium tools connect to just one network adapter based on a OS level recommendation and if they select the wrong adapter then they will not see any traffic.

For example if your CAN Bus traffic is coming in on Ethernet and the Tritium tools connect to your WiFi adapter, then they will not see any CAN Bus data.

Profinity was written right from the start to try and address this issue and it does so in two ways.

1. All UDP connections in Profinity connect to all Network Adapters until
2. CAN Bus packets from the correct bridge are discovered.

Through this discovery process the adapter is able to determine what Network adapter the traffic arrived from and hence is the correct one to talk to for that bridge. At which point it exclusively uses that adapter.

All this occurs behind the scenes and is not visible to the user, but will usually occur very rapidly. By the time an adapter has been auto discovered this process has already occurred.

## The Tritium tools focus on UDP primarily and use TCP when they must

Profinity treats both protocols as first class citizens and can use either interchangeably. The TCP performance of the Prohelion libraries is around 10x that of the Tritium C# libraries. While both protocols are treated as first class citizens, when doing updates we strongly recommend and in some cases insist on TCP as it is more reliable.

***Use Profinity wherever possible, it gets regular updates and has a lot of features designed to make your life easier and this whole experience of using the bridges a lot more reliable. The Tritium CAN Bus tools while not End of Life are functionally stabilised and unlikely to see further updates.***

Tritium Bridge Overview

Bridge Quickstart

Bridge Fundamentals

CAN & Ethernet Fundamentals

Supported Network Setups

Tritium Tools vs Profinity

Common Problems & Solutions

Supported Network Setups

Common Problems & Solutions

## Prohelion Documentation