It is customary in an application to give the user some feedback that a needed device is plugged in and ready to be used. The standard J2534 API does not effectively address this issue. It is possible to use the PassThruOpen call and look for for success or failure to determine if a device is present, but this is undesirable for a number of reasons.

- You end up with a opened device when you may have just wanted to know if the device is there - even though you are not ready to use it.
- You must close this device with PassThruClose if your PassThruOpen was successful.
- Opening the device incurs some overhead (e.g. starting new threads within the DLL)- you may not be able to poll this way very frequently.
- A device is opened exclusively, so if another process (or even your process) already has the device open, a failed attempt to re-open the device will be indistinguishable from there being no device at all.
- A device is opened exclusively, so if another process (or even your process) is also checking for the presence of a device, you may encounter a race condition where the other process has the device open briefly, and you will therefore detect nothing.

For these reasons and more, we have added a special Tactrix-specific IOCTL to simplify the detection of our hardware. TX_IOCTL_GET_DEVICE_INSTANCES accepts a pointer to a wide character string as output data which will hold the device path to an available device, if one exists. You do not really need to examine or use the contents of this string for most purposes - it is enough to know that the IOCTL has returned success to know that a device is present, and usable. The following example demonstrates this method:

```
#define TX_IOCTL_BASE                   0x70000
#define TX_IOCTL_GET_DEVICE_INSTANCES   (TX_IOCTL_BASE+4)

wchar_t name[512];

// use ioctl call to get available instances

if (!j2534.PassThruIoctl(NULL,TX_IOCTL_GET_DEVICE_INSTANCES,NULL,name))
```

```
{
   printf("usable OpenPort 2.0 on %Sn",name);
}
```