



Release Notes

DBLv5.4 software, version 5.4.0

March 29, 2019

Summary

The DBLv5.4 software release is available for ARC Series E-Class network adapter users. The purpose of this is to address the following:

- Bug fixes
- Known issues

This CSPI release is an aggregation of individual features, bug fixes, limitations, and some known issues. We recommend that users migrate to this release at their earliest convenience.

For more information, refer to the [DBLv5.4 User Guide](#).

Table of Contents

DBLv5.4 Product Description.....	3
Software Support Notice.....	4
New Features and Enhancements	5
Benchmarks.....	5
Linux UDP half-round trip performance.....	6
Linux TCP half-round trip performance.....	7
Linux tick-to-trade performance	8
Linux UDP half-round trip performance (using -Z with 1 adapter)	9
Windows UDP half-round trip performance	10
Windows TCP half-round trip performance.....	11
Windows Tick-to-trade performance.....	12
Windows UDP half-round trip performance (using -Z with 1 adapter)	13
Bug Fixes.....	14
Limitations.....	14
Known Issues.....	14
Technical Support.....	15

DBLv5.4 Product Description

DBL version 5.4.0 software (DBLv5.4) is an optional, user-level software interface for accelerating applications whose performance benefits from improved network latency, reduced jitter, and consumed data at higher message rates.

Applications benefitting from DBLv5.4 include:

- Algorithmic financial trading
- Voice over internet protocol (VoIP)
- Online gaming engines
- Applications where reduced network latency is essential

DBLv5.4 software and the ARC Series network adapters leverage user-level, kernel-bypass, messaging techniques – originally developed for high performance computing (HPC) applications – and apply these techniques to IP communication over Ethernet in a multicast and/or unicast environment. DBLv5.4 is inter-operable and wire-compatible with all standard TCP and UDP implementations.

IP communication offers a full set of IP networking services to hundreds of user application threads, but at the cost of latency in the protocol stack. DBLv5.4, on the other hand, takes advantage of kernel-bypass capabilities in ARC Series network adapters to allow high-priority user threads to send and receive IP frames directly.

The kernel-bypass approach improves performance to several metrics:

- DBLv5.4 user-level read latency (the amount of time it takes a packet to travel from source to destination) on modern hosts is less than one microsecond.

Software Support Notice

- This release is compatible only with ARC Series E-class adapters (10G-PCIE3-8E-2S) and requires Firmware Version 2.0.6 or later.
- Linux Support
 - CentOS 7.6 is recommended.
 - For non-RPM-based Linux distributions, a TGZ driver is provided and supported up to Linux kernel version 4.18.
- Windows Support (64-bit)
 - Windows Server 2016 is recommended.
 - Windows Server 2012 R2 is supported
 - Windows 7 and 10 are supported. For best performance Windows Server versions are recommended.

For more information on DBLv5.4, refer to the [DBLv5.4 User Guide](#).

New Features and Enhancements

1. (ID# 506, 507) The DBL API allows an application to directly access the packet buffers. This includes peeking at a packet without consuming it. For example, a trading application can peek at the packet to search for a specific sequence or symbol. The application can then consume packets, without overhead, until a meaningful sequence or symbol is found, which may improve overall latency. New functions available in the API include: `dbl_eventq_open`, `dbl_eventq_close`, `dbl_eventq_peek_head`, `dbl_eventq_peek_next`, `dbl_eventq_inspect`, and `dbl_eventq_consume`. Two new test programs are also included: `dbl_ring_access` and `dbl_eventq`.
2. (ID# 508) The DBL API allows an application to determine if further packets are pending. The API also provides a function that allows you to receive multiple packets in one call instead of receiving packets sequentially, one at a time, minimizing overhead. The new function is `dbl_ext_recvfrom`. A new test program, called `dbl_batch_recv`, is also provided to exercise the function.

Benchmarks

Operating Systems:	CentOS Linux release 7.3.1611 Windows 2012 R2
CSPi Network Adapters:	ARC Series E-class adapters (10G-PCIE3-8E-2S), Firmware Version 2.0.6.
CSPi Driver:	DBL Release 5.4.0
Host 1:	i7-4790K Devil's Canyon Processor @ 4.00 GHz (Frequency scaled to 4.20 GHz), 4 cores, 2x8GB DDR3-1333, Asus H97I-PLUS motherboard
Host 2:	i7-4790K Devil's Canyon Processor @ 4.00 GHz (Frequency scaled to 4.20 GHz), 4 cores, 2x8GB DDR3-1333, Asus H97I-PLUS motherboard
PCIe:	Gen 3 (8GT/s) x8, (Gen 3 x16 capable slot)
Topology:	point-to-point (switchless)
Tuning:	Linux: tuned-adm latency-performance Windows: powercfg.exe -setactive SCHEME_MIN (High Performance) Hyper-threading disabled CPU C-states disabled CPU Affinity set to specified core

Linux UDP half-round trip performance

Command lines:

```
$ taskset -c 1 /opt/dbl/bin/tests/dbl_pingpong -s -l 10.0.0.1
```

```
$ taskset -c 1 /opt/dbl/bin/tests/dbl_pingpong -l 10.0.0.2 -h 10.0.0.1
```

Output:

```
Half Round-Trip latency benchmark
Size 1:      1.27 us      (mean=1.35 median=1.35, 99%=1.49 max=1.72, std_dev=0.04)
Size 2:      1.27 us      (mean=1.35 median=1.35, 99%=1.50 max=1.81, std_dev=0.04)
Size 4:      1.27 us      (mean=1.35 median=1.34, 99%=1.52 max=1.85, std_dev=0.05)
Size 8:      1.27 us      (mean=1.35 median=1.35, 99%=1.50 max=2.16, std_dev=0.05)
Size 16:     1.27 us      (mean=1.35 median=1.34, 99%=1.50 max=1.80, std_dev=0.05)
Size 32:     1.29 us      (mean=1.36 median=1.36, 99%=1.50 max=1.96, std_dev=0.05)
Size 64:     1.30 us      (mean=1.39 median=1.38, 99%=1.54 max=1.95, std_dev=0.05)
Size 128:    1.36 us      (mean=1.45 median=1.45, 99%=1.57 max=2.13, std_dev=0.04)
Size 256:    1.57 us      (mean=1.65 median=1.64, 99%=1.78 max=3.07, std_dev=0.04)
Size 512:    1.88 us      (mean=1.98 median=1.97, 99%=2.11 max=2.60, std_dev=0.04)
Size 1024:   2.56 us      (mean=2.67 median=2.67, 99%=2.80 max=3.41, std_dev=0.04)
Size 2048:   3.93 us      (mean=4.05 median=4.05, 99%=4.23 max=11.91, std_dev=0.09)
```

Linux TCP half-round trip performance

Command lines:

```
$ taskset -c 1 /opt/dbl/bin/tests/dbltcp_pingpong -s -l 10.0.0.1
```

```
$ taskset -c 1 /opt/dbl/bin/tests/dbltcp_pingpong -l 10.0.0.2 -h 10.0.0.1
```

Output:

```
Test will be using Type DBL_TCP, Proto DBL_BSD
Opening the DBL device 10.0.0.129:3333
*** Enabling DBL TCP SBL enhancement ***
Connected
Size 1:      1.85 us      (mean=1.94 median=1.94, 99%=2.12 max=9.21, std_dev=0.10)
Size 2:      1.85 us      (mean=1.94 median=1.93, 99%=2.12 max=2.70, std_dev=0.07)
Size 4:      1.85 us      (mean=1.94 median=1.94, 99%=2.12 max=2.83, std_dev=0.06)
Size 8:      1.86 us      (mean=1.94 median=1.94, 99%=2.10 max=2.79, std_dev=0.06)
Size 16:     1.85 us      (mean=1.95 median=1.94, 99%=2.08 max=2.60, std_dev=0.05)
Size 32:     1.89 us      (mean=1.95 median=1.95, 99%=2.04 max=2.56, std_dev=0.04)
Size 64:     1.92 us      (mean=1.99 median=1.98, 99%=2.14 max=2.65, std_dev=0.05)
Size 128:    1.97 us      (mean=2.04 median=2.04, 99%=2.23 max=2.94, std_dev=0.06)
Size 256:    2.12 us      (mean=2.20 median=2.19, 99%=2.36 max=9.44, std_dev=0.10)
Size 512:    2.48 us      (mean=2.56 median=2.55, 99%=2.74 max=3.30, std_dev=0.06)
Size 1024:   3.20 us      (mean=3.32 median=3.30, 99%=3.49 max=4.09, std_dev=0.07)
Size 2048:   4.65 us      (mean=6.26 median=7.56, 99%=7.89 max=9.26, std_dev=1.44)
```

Linux tick-to-trade performance

Command lines:

```
$ taskset -c 1 /opt/dbl/bin/tests/dbltcp_tick_to_trade -U 3 -T 5 -1  
10.0.0.1 -s
```

```
$ taskset -c 1 /opt/dbl/bin/tests/dbltcp_tick_to_trade -U 3 -T 5 -1  
10.0.0.2 -h 10.0.0.1
```

Output:

```
Opening the DBL device 10.0.0.129:3333  
*** Enabling DBL TCP SBL enhancement ***  
Connecting.  
Warming up with 10000 iterations followed by 10000 measurements  
tick-to-trade iteration 1: tcp size: 5, udp size 12  
tick-to-trade 1434 ns (mean=1555, median=1516, 99%=1778, max=240933, std_dev=2394)  
tick-to-trade iteration 2: tcp size: 38, udp size 12  
tick-to-trade 1452 ns (mean=1588, median=1528, 99%=1788, max=278381, std_dev=3372)  
tick-to-trade iteration 3: tcp size: 64, udp size 12  
tick-to-trade 1436 ns (mean=1548, median=1525, 99%=1790, max=54488, std_dev=534)  
tick-to-trade iteration 4: tcp size: 128, udp size 12  
tick-to-trade 1468 ns (mean=1599, median=1549, 99%=1816, max=249026, std_dev=2541)  
tick-to-trade iteration 5: tcp size: 400, udp size 12  
tick-to-trade 1615 ns (mean=1732, median=1693, 99%=1950, max=146298, std_dev=1679)  
tick-to-trade iteration 6: tcp size: 5, udp size 38  
tick-to-trade 1427 ns (mean=1544, median=1527, 99%=1783, max=88769, std_dev=875)  
tick-to-trade iteration 7: tcp size: 38, udp size 38  
tick-to-trade 1441 ns (mean=1586, median=1539, 99%=1781, max=252037, std_dev=2994)  
tick-to-trade iteration 8: tcp size: 64, udp size 38  
tick-to-trade 1442 ns (mean=1559, median=1538, 99%=1789, max=117597, std_dev=1162)  
tick-to-trade iteration 9: tcp size: 128, udp size 38  
tick-to-trade 1475 ns (mean=1583, median=1563, 99%=1801, max=86043, std_dev=947)  
tick-to-trade iteration 10: tcp size: 400, udp size 38  
tick-to-trade 1615 ns (mean=1715, median=1707, 99%=1952, max=16735, std_dev=164)  
tick-to-trade iteration 11: tcp size: 5, udp size 64  
tick-to-trade 1460 ns (mean=1585, median=1551, 99%=1805, max=254823, std_dev=2533)  
tick-to-trade iteration 12: tcp size: 38, udp size 64  
tick-to-trade 1474 ns (mean=1576, median=1565, 99%=1809, max=46668, std_dev=455)  
tick-to-trade iteration 13: tcp size: 64, udp size 64  
tick-to-trade 1468 ns (mean=1600, median=1564, 99%=1815, max=166636, std_dev=1839)  
tick-to-trade iteration 14: tcp size: 128, udp size 64  
tick-to-trade 1498 ns (mean=1622, median=1592, 99%=1835, max=214115, std_dev=2126)  
tick-to-trade iteration 15: tcp size: 400, udp size 64  
tick-to-trade 1644 ns (mean=1772, median=1736, 99%=1957, max=146108, std_dev=1981)
```


Linux UDP half-round trip performance (using -Z with 1 adapter)

Command lines:

```
$ taskset -c 1 /opt/dbl/bin/tests/udp_pingpong -l 10.0.0.1 -s
```

```
$ taskset -c 1 /opt/dbl/bin/tests/dbl_pingpong -l 10.0.0.2 -h 10.0.0.1 -Z
```

Output:

```
Half Round-Trip latency benchmark
Size 1:      2.98 us      (mean=3.23 median=3.21, 99%=3.46 max=3.81, std_dev=0.08)
  turnaround-time avg: 1.33 us min: 1.21 us
Size 2:      2.99 us      (mean=3.22 median=3.20, 99%=3.48 max=3.94, std_dev=0.08)
  turnaround-time avg: 1.33 us min: 1.22 us
Size 4:      2.95 us      (mean=3.20 median=3.17, 99%=3.44 max=4.12, std_dev=0.08)
  turnaround-time avg: 1.33 us min: 1.22 us
Size 8:      2.96 us      (mean=3.19 median=3.17, 99%=3.44 max=5.70, std_dev=0.09)
  turnaround-time avg: 1.33 us min: 1.22 us
Size 16:     2.99 us      (mean=3.22 median=3.20, 99%=3.48 max=4.00, std_dev=0.08)
  turnaround-time avg: 1.32 us min: 1.22 us
Size 32:     3.00 us      (mean=3.23 median=3.21, 99%=3.48 max=3.89, std_dev=0.08)
  turnaround-time avg: 1.35 us min: 1.24 us
Size 64:     3.04 us      (mean=3.26 median=3.24, 99%=3.51 max=5.84, std_dev=0.08)
  turnaround-time avg: 1.37 us min: 1.26 us
Size 128:    3.16 us      (mean=3.37 median=3.34, 99%=3.62 max=4.60, std_dev=0.08)
  turnaround-time avg: 1.44 us min: 1.32 us
Size 256:    3.36 us      (mean=3.56 median=3.54, 99%=3.83 max=4.29, std_dev=0.08)
  turnaround-time avg: 1.61 us min: 1.45 us
Size 512:    3.77 us      (mean=3.97 median=3.95, 99%=4.24 max=4.82, std_dev=0.08)
  turnaround-time avg: 1.94 us min: 1.83 us
Size 1024:   4.67 us      (mean=4.86 median=4.83, 99%=5.15 max=5.68, std_dev=0.10)
  turnaround-time avg: 2.64 us min: 2.51 us
Size 2048:   6.62 us      (mean=6.81 median=6.80, 99%=7.09 max=14.35, std_dev=0.12)
  turnaround-time avg: 4.02 us min: 3.85 us
```

Windows UDP half-round trip performance

Command lines:

```
C:\DBL_PHX-10G\bin\tests>dbl_pingpong.exe -l 10.0.0.1 -s
```

```
C:\DBL_PHX-10G\bin\tests>dbl_pingpong.exe -l 10.0.0.2 -h 10.0.0.1
```

Output:

```
UAC enabled. Process is Elevated
Half Round-Trip latency benchmark
Size 1:      1.28 us      (mean=1.41 median=1.41, 99%=1.54 max=6.16, std_dev=0.08)
Size 2:      1.28 us      (mean=1.41 median=1.41, 99%=1.54 max=2.57, std_dev=0.06)
Size 4:      1.28 us      (mean=1.41 median=1.41, 99%=1.54 max=2.31, std_dev=0.06)
Size 8:      1.28 us      (mean=1.41 median=1.41, 99%=1.54 max=4.11, std_dev=0.07)
Size 16:     1.28 us      (mean=1.42 median=1.41, 99%=1.54 max=2.69, std_dev=0.06)
Size 32:     1.28 us      (mean=1.43 median=1.41, 99%=1.54 max=21.17, std_dev=0.21)
Size 64:     1.28 us      (mean=1.44 median=1.41, 99%=1.54 max=2.82, std_dev=0.06)
Size 128:    1.41 us      (mean=1.48 median=1.54, 99%=1.67 max=2.69, std_dev=0.07)
Size 256:    1.54 us      (mean=1.64 median=1.67, 99%=1.80 max=2.82, std_dev=0.07)
Size 512:    1.92 us      (mean=1.99 median=2.05, 99%=2.18 max=2.31, std_dev=0.07)
Size 1024:   2.57 us      (mean=2.69 median=2.69, 99%=2.82 max=3.85, std_dev=0.06)
Size 2048:   3.85 us      (mean=4.05 median=4.11, 99%=4.23 max=16.94, std_dev=0.19)
```

Windows TCP half-round trip performance

Command lines:

```
C:\DBL_PHX-10G\bin\tests>dbltcp_pingpong.exe -l 10.0.0.1 -s
```

```
C:\DBL_PHX-10G\bin\tests>dbltcp_pingpong.exe -l 10.0.0.2 -h 10.0.0.1
```

Output:

```
Test will be using Type DBL_TCP, Proto DBL_BSD
UAC enabled. Process is Elevated
Opening the DBL device 10.0.0.1:3333
Connected
Size 1:      2.05 us      (mean=2.17 median=2.18, 99%=2.31 max=2.95, std_dev=0.06)
Size 2:      2.05 us      (mean=2.16 median=2.18, 99%=2.31 max=7.44, std_dev=0.08)
Size 4:      2.05 us      (mean=2.17 median=2.18, 99%=2.31 max=3.21, std_dev=0.06)
Size 8:      2.05 us      (mean=2.17 median=2.18, 99%=2.31 max=3.72, std_dev=0.07)
Size 16:     2.05 us      (mean=2.17 median=2.18, 99%=2.31 max=22.07, std_dev=0.21)
Size 32:     2.05 us      (mean=9.21 median=2.18, 99%=2.31 max=70008.51,
std_dev=700.03)
Size 64:     2.18 us      (mean=2.25 median=2.31, 99%=2.44 max=2.95, std_dev=0.07)
Size 128:    2.18 us      (mean=2.34 median=2.31, 99%=2.44 max=4.11, std_dev=0.07)
Size 256:    2.44 us      (mean=2.55 median=2.57, 99%=2.69 max=8.85, std_dev=0.09)
Size 512:    2.82 us      (mean=2.93 median=2.95, 99%=3.08 max=5.90, std_dev=0.07)
Size 1024:   3.59 us      (mean=3.71 median=3.72, 99%=3.85 max=4.88, std_dev=0.07)
Size 2048:   6.80 us      (mean=7.02 median=7.06, 99%=7.18 max=8.72, std_dev=0.08)
```

Windows Tick-to-trade performance

Command lines:

```
C:\DBL_PHX-10G\bin\tests>dbl_tcp_tick_to_trade -T 3 -U 5 -l 10.0.0.1 -s
```

```
C:\DBL_PHX-10G\bin\tests>dbl_tcp_tick_to_trade -T 3 -U 5 -l 10.0.0.2 -h  
10.0.0.1
```

Output:

```
UAC enabled. Process is Elevated  
Opening the DBL device 10.0.0.1:3333  
Connecting.  
Warming up with 10000 iterations followed by 10000 measurements  
tick-to-trade iteration 1: tcp size: 5, udp size 12  
tick-to-trade 1634 ns (mean=1787, median=1761, 99%=1982, max=44554, std_dev=439)  
tick-to-trade iteration 2: tcp size: 38, udp size 12  
tick-to-trade 1679 ns (mean=1830, median=1807, 99%=2127, max=14918, std_dev=170)  
tick-to-trade iteration 3: tcp size: 64, udp size 12  
tick-to-trade 1672 ns (mean=1834, median=1813, 99%=2156, max=3744, std_dev=108)  
tick-to-trade iteration 4: tcp size: 128, udp size 12  
tick-to-trade 1740 ns (mean=1919, median=1901, 99%=2234, max=4405, std_dev=126)  
tick-to-trade iteration 5: tcp size: 400, udp size 12  
tick-to-trade 1935 ns (mean=2088, median=2065, 99%=2360, max=4133, std_dev=99)  
tick-to-trade iteration 6: tcp size: 5, udp size 38  
tick-to-trade 1629 ns (mean=1770, median=1749, 99%=2024, max=4380, std_dev=100)  
tick-to-trade iteration 7: tcp size: 38, udp size 38  
tick-to-trade 1660 ns (mean=1810, median=1789, 99%=2127, max=6696, std_dev=119)  
tick-to-trade iteration 8: tcp size: 64, udp size 38  
tick-to-trade 1674 ns (mean=1819, median=1800, 99%=2139, max=2442, std_dev=98)  
tick-to-trade iteration 9: tcp size: 128, udp size 38  
tick-to-trade 1731 ns (mean=1887, median=1873, 99%=2181, max=3480, std_dev=103)  
tick-to-trade iteration 10: tcp size: 400, udp size 38  
tick-to-trade 1930 ns (mean=2074, median=2053, 99%=2329, max=2931, std_dev=89)  
tick-to-trade iteration 11: tcp size: 5, udp size 64  
tick-to-trade 1644 ns (mean=1791, median=1775, 99%=2048, max=4360, std_dev=102)  
tick-to-trade iteration 12: tcp size: 38, udp size 64  
tick-to-trade 1677 ns (mean=1831, median=1814, 99%=2162, max=5191, std_dev=112)  
tick-to-trade iteration 13: tcp size: 64, udp size 64  
tick-to-trade 1693 ns (mean=1840, median=1821, 99%=2155, max=4284, std_dev=99)  
tick-to-trade iteration 14: tcp size: 128, udp size 64  
tick-to-trade 1740 ns (mean=1924, median=1919, 99%=2221, max=36658, std_dev=365)  
tick-to-trade iteration 15: tcp size: 400, udp size 64  
tick-to-trade 1958 ns (mean=2107, median=2088, 99%=2377, max=15091, std_dev=164)
```

Windows UDP half-round trip performance (using -Z with 1 adapter)

Command lines:

```
C:\DBL_PHX-10G\bin\tests>udp_pingpong.exe -l 10.0.0.1 -s
```

```
C:\DBL_PHX-10G\bin\tests>dbl_pingpong.exe -l 10.0.0.2 -h 10.0.0.1 -Z
```

Output:

```
UAC enabled. Process is Elevated
Half Round-Trip latency benchmark
Size 1:      3.85 us      (mean=4.00 median=3.98, 99%=5.26 max=83.39, std_dev=1.17)
    turnaround-time avg: 1.37 us min: 1.31 us
Size 2:      3.85 us      (mean=4.00 median=3.98, 99%=7.06 max=82.11, std_dev=1.17)
    turnaround-time avg: 1.37 us min: 1.31 us
Size 4:      3.72 us      (mean=3.97 median=3.98, 99%=4.88 max=81.85, std_dev=1.16)
    turnaround-time avg: 1.37 us min: 1.31 us
Size 8:      3.72 us      (mean=3.96 median=3.85, 99%=7.06 max=82.11, std_dev=1.16)
    turnaround-time avg: 1.37 us min: 1.31 us
Size 16:     3.85 us      (mean=4.00 median=3.98, 99%=5.26 max=82.11, std_dev=1.16)
    turnaround-time avg: 1.37 us min: 1.31 us
Size 32:     3.85 us      (mean=3.99 median=3.98, 99%=4.36 max=81.72, std_dev=0.97)
    turnaround-time avg: 1.39 us min: 1.32 us
Size 64:     3.85 us      (mean=4.02 median=3.98, 99%=5.00 max=81.85, std_dev=1.15)
    turnaround-time avg: 1.40 us min: 1.33 us
Size 128:    3.98 us      (mean=4.11 median=4.11, 99%=5.00 max=82.24, std_dev=1.16)
    turnaround-time avg: 1.45 us min: 1.38 us
Size 256:    4.11 us      (mean=4.34 median=4.23, 99%=5.26 max=82.75, std_dev=1.16)
    turnaround-time avg: 1.60 us min: 1.52 us
Size 512:    4.62 us      (mean=4.78 median=4.75, 99%=6.16 max=82.62, std_dev=1.16)
    turnaround-time avg: 1.96 us min: 1.86 us
Size 1024:   5.52 us      (mean=5.71 median=5.65, 99%=6.93 max=83.78, std_dev=1.16)
    turnaround-time avg: 2.69 us min: 2.54 us
Size 2048:   7.31 us      (mean=7.63 median=7.57, 99%=10.52 max=85.32, std_dev=1.17)
    turnaround-time avg: 4.00 us min: 3.88 us
```

Bug Fixes

1. (ID# 500) Linux: Added support for building on CentOS 7.5, Ubuntu 18.04, and Linux kernels up to version 4.18.

Limitations

- Refer to the Appendix 3: *DBLv5.4 Driver Restrictions and Limitations* of the [DBLv5.4 User Guide](#) for more information.

Known Issues

1. (ID# 160) Windows: There is no Makefile available to build the test programs.
2. (ID# 312) You must connect both Rx and Tx to the 10G-PCIE3-8E-2S adapter port. Disconnecting Tx may cause the adapter to hang.
3. (ID# 327) When using **dbltcp_tick_to_trade** and **sock_tick_to_trade** benchmarks together you may observe a large amount of jitter in the results. This has only been observed in device-under-test mode when started via **dblrn -b 0**.
4. (ID# 329) Running **sock_tick_to_trade** via **dblrn -b 0** may display false results.
5. (ID# 260, 360, 364) When switching transceiver or cable types, the link may not be detected without first reloading the driver.

Technical Support

DBLv5.4 software documentation, technical support, and downloads are available from the CSPi website at <https://www.cspi.com/ethernet-products/adapters>

If there are problems installing or using CSPi products, or if any bugs or possible enhancements are noticed, contact Technical Support via the CSPi Customer Portal * <https://www.cspi.com/ethernet-products/support>

CSPi email support at support@cspi.com

Before you contact our technical support staff, have the following information available:

- Your name, title, company name, phone number, and email address
- Operating system and version number
- Product name and release version
- Problem description

* Follow the instructions on the CSPi Customer Portal website to register for a CSPi Customer Support account