User Manual
for the
Fibre Channel Adapter
VxWorks Enhanced Network Software Driver

© C²I² Systems The copyright of this document is the property of C²I² Systems. The document is issued for the sole purpose for which it is supplied, on the express terms that it may not be copied in whole or part, used by or disclosed to others except as authorised in writing by C²I² Systems.

Document prepared by C²I² Systems, Cape Town
<table>
<thead>
<tr>
<th>Name</th>
<th>Signature</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Completed by</td>
<td>pp. X. Keuger</td>
<td>2009-08-20</td>
</tr>
<tr>
<td>Signature</td>
<td>Project Engineer</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Board Level Products</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Cµ² Systems</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Accepted by</th>
<th>[Signature]</th>
<th>2009-08-28</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Manager</td>
<td>[Signature]</td>
<td></td>
</tr>
<tr>
<td>Board Level Products</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cµ² Systems</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Accepted by</th>
<th>[Signature]</th>
<th>2009-08-20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quality Assurance</td>
<td>[Signature]</td>
<td></td>
</tr>
<tr>
<td>Cµ² Systems</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Signature Sheet

<table>
<thead>
<tr>
<th>Name</th>
<th>Signature</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Completed by</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Project Engineer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Board Level Products</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C²I² Systems</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Accepted by</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Project Manager</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Board Level Products</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C²I² Systems</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Accepted by</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quality Assurance</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C²I² Systems</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Amendment History

<table>
<thead>
<tr>
<th>Issue</th>
<th>Description</th>
<th>Date</th>
<th>ECP No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>Initial Release.</td>
<td>2002-04-23</td>
<td>-</td>
</tr>
<tr>
<td>1.0</td>
<td>Baseline Document.</td>
<td>2002-10-24</td>
<td>-</td>
</tr>
<tr>
<td>1.1</td>
<td>Fixed up document and added description of initialisation string.</td>
<td>2005-07-21</td>
<td>CCII/FC/6-ECP/011</td>
</tr>
<tr>
<td>1.2</td>
<td>Improve document naming consistency.</td>
<td>2009-08-20</td>
<td>CCII/FC/6-ECP/016</td>
</tr>
</tbody>
</table>
Contents

1. **Scope** ................................................................................................................. 1
  1.1 Identification ................................................................................................. 1
  1.2 Introduction ................................................................................................... 1

2. **Applicable and Reference Documents** ............................................................ 2
  2.1 Applicable Documents ................................................................................ 2
  2.2 Reference Documents .................................................................................. 2

3. **Getting Started** .............................................................................................. 3
  3.1 Loading the Driver ....................................................................................... 3
  3.2 Loading the FDDI Shim Module ................................................................... 4

4. **Installation Procedure** ................................................................................... 5
  4.1 Building the FC END VxWorks Software Driver into the VxWorks Kernel ............... 5
  4.2 Loading the Driver Separately .................................................................... 6
  4.3 Starting the Driver ....................................................................................... 6
    4.3.1 Command Line Passing of FC Settings .............................................. 6
  4.4 Attaching Driver to TCP/IP stack .................................................................. 6
  4.5 Loading the FDDI Shim Module ................................................................... 7

5. **Contact Details** ............................................................................................... 8
  5.1 Contact Person ............................................................................................ 8
  5.2 Physical Address .......................................................................................... 8
  5.3 Postal Address ............................................................................................. 8
  5.4 Voice and Electronic Contacts .................................................................... 8
  5.5 Product Support .......................................................................................... 8

Annexure A ............................................................................................................. 9
  FDDI vs FC ......................................................................................................... 9

Annexure B ............................................................................................................. 10
  Comparing Use of the FC and FDDI END VxWorks Software Drivers ..................... 10
### Abbreviations and Acronyms

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BSD</td>
<td>Berkley Software Distribution</td>
</tr>
<tr>
<td>BSP</td>
<td>Board Support Package</td>
</tr>
<tr>
<td>END</td>
<td>Enhanced Network Driver</td>
</tr>
<tr>
<td>FC</td>
<td>Fibre Channel</td>
</tr>
<tr>
<td>FDDI</td>
<td>Fibre Distributed Data Interface</td>
</tr>
<tr>
<td>FTP</td>
<td>File Transfer Protocol</td>
</tr>
<tr>
<td>HCC</td>
<td>Host Carrier Card</td>
</tr>
<tr>
<td>ICMP</td>
<td>Internet Control Message Protocol</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>LAN</td>
<td>Local Area Network</td>
</tr>
<tr>
<td>MIB</td>
<td>Management Information Base</td>
</tr>
<tr>
<td>NIC</td>
<td>Network Interface Card</td>
</tr>
<tr>
<td>PC</td>
<td>Personal Computer</td>
</tr>
<tr>
<td>PCI</td>
<td>Peripheral Component Interconnect</td>
</tr>
<tr>
<td>PMC</td>
<td>Peripheral Component Interconnect Mezzanine Card</td>
</tr>
<tr>
<td>SBC</td>
<td>Single Board Computer</td>
</tr>
<tr>
<td>SENS</td>
<td>Scalable Enhanced Network Stack</td>
</tr>
<tr>
<td>TCP</td>
<td>Transmission Control Protocol</td>
</tr>
<tr>
<td>UDP</td>
<td>User Datagram Protocol</td>
</tr>
<tr>
<td>VME</td>
<td>Versa Module Eurocard</td>
</tr>
</tbody>
</table>
1. **Scope**

1.1 **Identification**

This document is the User Manual for the Fibre Channel (FC) Enhanced Network Driver (END) VxWorks Software Driver.

1.2 **Introduction**

The FC END VxWorks Software Driver was developed to support the Intel X86 PC, MVME5100 PPC and the SVME179 PPC platforms. As such the drivers binaries are provided with explicit installation instructions. The FC END VxWorks Software Driver conforms to VxWorks 5.4 END driver model and the driver will interface to VxWorks using this standard.

The FC Adapter is a dual channel PCI Mezzanine Card to FC protocol controller, attaching computers to 1 Gbit/s FC networks using copper cable and 2 Gbit/s using fibre optic cable.

At present, C²I² Systems FC Adapters are available in PMC formfactor.

The driver software distribution consists of the following files:

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ccfEndxxx.a</td>
<td>FC END VxWorks Software Driver object file</td>
</tr>
<tr>
<td>ccFddShim.o</td>
<td>Shim module for FDDI compatibility</td>
</tr>
<tr>
<td>ccFddShim.c</td>
<td>Shim source file</td>
</tr>
<tr>
<td>ccMib.h</td>
<td>Header file for FDDI END VxWorks Software Driver SMT 7.3 MIB</td>
</tr>
<tr>
<td>Readme.txt</td>
<td>Installation notes</td>
</tr>
<tr>
<td>Release.txt</td>
<td>Release notes and revision history. Please check this file for information on the latest updates.</td>
</tr>
</tbody>
</table>

1. xxx=X86 for Intel X86 PC, DY4PPC for DY4179/181/182 or 5100 for MV5100 PPC
2. **Applicable and Reference Documents**

2.1 **Applicable Documents**


2.1.2 DI-IPSC-81443, *Data Item Description for a Software User Manual*, dated Apr 89.


2.2 **Reference Documents**

None.
3. **Getting Started**

This section is a quick start guide for getting the FC END VxWorks Software Driver up and running.

3.1 **Loading the Driver**

1. Boot VxWorks and load the ccfcEndxxx.a lib with `ld < ccfcEndxxx.a`.

2. Type `muxDevStart ( muxDevLoad (unitno, ccfcLoad,"",0,0 ))` to attach and start the FC END VxWorks Software Driver. Verify by using `muxShow`. The `unitno` must be 0 for Function[0] and 1 for Function[1] on the multi-function FC Adapter.

3. Attach the FC END VxWorks Software Driver to the TCP/IP protocol stack. Type `ipAttach unitno, "ccfc"` and use `muxShow` and `ifShow` to confirm that the `ccfcunitno` exists and was attached to TCP/IP.

4. Set the IP address using `ifAddrSet`. Refer to the VxWorks Network Programmers Guide paragraph ’4.4 Overview of TCP/IP’ for setting up IP host names and IP routing.

5. Confirm the FC LAN connection using ping. An example is shown below.

Note: This example assumes there is already a Network Interface Card (NIC) on the FC LAN which has been set up with an IP address of 10.0.0.1.

```
-> muxDevStart ( muxDevLoad ( 0,ccfcLoad, "", 0, 0 ))
-> ipAttach 0, "ccfc"

-> ifShow
ppp (unit number 0):
  Flags: (0x71) UP POINT-TO-POINT ARP RUNNING
  Internet address: 172.16.0.2
  Destination Internet address: 172.16.0.1
  Netmask 0xffff0000 Subnetmask 0xffff0000
  Metric is 0
  Maximum Transfer Unit size is 1 500
  5 packets received; 5 packets sent
  0 input errors; 0 output errors
  0 collisions

lo (unit number 0):
  Flags: (0x69) UP LOOPBACK ARP RUNNING
  Internet address: 127.0.0.1
  Netmask 0xff000000 Subnetmask 0xff000000
  Metric is 0
  Maximum Transfer Unit size is 4 096
  0 packets received; 0 packets sent
  0 input errors; 0 output errors
  0 collisions

ccfc (unit number 0):
  Flags: (0x63) UP BROADCAST ARP RUNNING
  Netmask 0xffffffff Subnetmask 0xffffffff
  Ethernet address is 00:50:C2:18:D0:00
  Metric is 0
  Maximum Transfer Unit size is 16 384
  18 packets received; 0 packets sent
  0 input errors; 0 output errors
  0 collisions

-> ifAddrSet "ccfc0","10.0.0.4"
```

---

1. `xxx=X86 for Intel X86 PC, DY4PPC for DY4179/181/182 or 5100 for MV5100 PPC`
3.2 Loading the FDDI Shim Module

In some instances, the FC Adapter will be used to replace the FDDI Adapter in an existing system. In this case, there is no need to change the user applications. The only difference between the implementation of the FC and FDDI END VxWorks Software Drivers are that the fddiGetStats() routine is replaced by the ccfcGetStats() routine. The shim is present to restore the difference. Figure 1 below shows a graphical layout of the difference between the two drivers. Users of the FDDI END VxWorks Software Driver will be familiar with the structure `cc_fddi_mib_type` and the following routines:

- `void ccfdiGetStats(struct cc_fddi_mib_type *data)`
- `void ccfdiClrStats(void)`

Although the FC END VxWorks Software Driver is not compatible with the FDDI MIB, calls to the MIB routines will not result in an error as long as the FDDI shim module is loaded. In the FDDI structure `cc_fddi_mib_type`, some fields may or may not correspond with fields in structures of the FC END VxWorks Software Driver. As per default, all the fields will be set to NULL. If any fields in the FDDI MIB needs to be changed to ensure compatibility with an application, the values can be changed in the `ccFddiShim.c` source file and recompiled.

A comparative layout of the difference between the drivers are shown below.

Load the FDDI shim with 'ld < ccFddiShim.o'.
4. **Installation Procedure**

4.1 **Building the FC END VxWorks Software Driver into the VxWorks Kernel**

This section describes the installation instructions for building the FC END VxWorks Software Driver into the VxWorks kernel.

1. Copy ccfcEndxxx.a\(^1\) to your BSP library directory (eg /tornado/target/config/X86/lib).
2. Edit the Makefile in the BSP directory (eg /tornado/target/config/X86).
   
   Find the line
   \[
   \text{MACH}\_\text{EXTRA} =
   \]
   and replace with
   \[
   \text{MACH}\_\text{EXTRA} = ./lib/ccfcEndxxx.a ./lib/ccfcShim.o^2
   \]
3. Add the following code fragment to config.h.
   (before "\#define DEFAULT\_BOOT\_LINE\(\)"
   
   /*Added by CCII for FC driver*/
   \#define INCLUDE\_CCFC\_END /* CCII FC END DRIVER */
   /* CCII Modification */
   \#ifdef INCLUDE\_CCFC\_END
   \#ifndef INCLUDE\_PCI
   \#define INCLUDE\_PCI
   \#endif
   \#ifdef INCLUDE\_BSD
   \#undef INCLUDE\_BSD
   \#endif
   \#ifndef INCLUDE\_END
   \#define INCLUDE\_END
   \#endif
   \#endif
   /*End CCII Modification */
   
   4. In config.h, change the DEFAULT\_BOOT\_LINE to use the ccfc driver.
5. Add the following to configNet.h.
   (Before the start of the endDevTbl[ ] declaration.)
   IMPORT END\_OBJ * ccfcLoad (char *initString);
   (Add the following segment to the endDevTbl[ ] , before the default last entry.)
   \#ifdef INCLUDE\_CCFC\_END
   \{ 0, ccfcLoad, "", NULL, NULL, FALSE},
   \#endif /* INCLUDE\_CCFC\_END */
6. Rebuild all VxWorks images.

---

\(^1\) xxx=X86 for Intel X86 PC, DY4PPC for DY4179/181/182 or 5100 for MV5100 PPC

\(^2\) Refer to Section 3.2
4.2 Loading the Driver Separately

From the VxWorks shell type:

\[ \text{ld < ccfcEndxxx.a} \]

4.3 Starting the Driver

The driver is started and attached to the MUX with the VxWorks \text{muxDevLoad} and \text{muxDevStart} commands. The syntax is as follows:

\[ \text{muxDevStart ( muxDevLoad ( unitno, ccfcLoad, ",", 0, 0 ) )} \]

where unitno corresponds to Function[0] = 0 and Function[1] = 1. This is only necessary if the driver was not included in the \text{configNet.h} file as described in Section 4.1.

Type \text{muxShow} to see if the driver was installed.

4.3.1 Command Line Passing of FC Settings

The FC END VxWorks Software Driver supports the changing of some default parameters at startup.

\[ \text{e.g.muxDevStart (muxDevLoad ( unitno, ccfcLoad,"unitno:NumberOfClusters:MTU",0,0))} \]

where:

- \text{unitno} : corresponds to Function[0] = 0 and Function[1] = 1
- \text{NumberOfClusters} : Number of TX and RX clusters or buckets - default 227 clusters (RX = 100 and TX = 127 clusters), minimum 10 clusters and maximum 254 clusters.
- \text{MTU} : Max Transfer Unit size in Decimal Bytes - default 32 kBytes - 4 Bytes, minimum 1 kByte and maximum 64 kBytes.

If a parameter is not used a placeholder of 0 must be used. This is only applicable for \text{NumberOfClusters and MTU}.

\[ \text{e.g. To change only NumberOfClusters to 200:} \]

\[ \text{muxDevStart (muxDevLoad (0,ccfcLoad,"0:200:0",0,0))} \]

All parameters preceding the ones you want to change must be specified.

To keep all settings at default values use:

\[ \text{muxDevStart ( muxDevLoad ( unitno, ccfcLoad, ",", 0, 0 ) )} \]

4.4 Attaching Driver to TCP/IP stack

Attach the Fibre Channel protocol to the TCP/IP protocol stack with:

\[ \text{ipAttach unitno, "ccfc"} \]

Use \text{muxShow} and \text{ifShow} to confirm.

---

\(^1\) xxx=X86 for Intel X86 PC, DY4PPC for DY4179/181/182 or 5100 for MV5100 PPC
4.5 Loading the FDDI Shim Module

Note: As mentioned in Section 3.2, this module is only necessary in some instances.

Load the FDDI shim with `ld < ccFddiShim.o`. 
5. **Contact Details**

5.1 **Contact Person**

Direct all correspondence and / or support queries to the Project Manager at C²I² Systems.

5.2 **Physical Address**

C²I² Systems  
Unit 3, Rosmead Place, Rosmead Centre  
67 Rosmead Avenue  
Kenilworth  
Cape Town  
7708  
South Africa

5.3 **Postal Address**

C²I² Systems  
P.O. Box 171  
Rondebosch  
7701  
South Africa

5.4 **Voice and Electronic Contacts**

<table>
<thead>
<tr>
<th>Tel</th>
<th>(+27) (0)21 683 5490</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fax</td>
<td>(+27) (0)21 683 5435</td>
</tr>
<tr>
<td>Email</td>
<td><a href="mailto:info@ccii.co.za">info@ccii.co.za</a></td>
</tr>
<tr>
<td>Email</td>
<td><a href="mailto:support@ccii.co.za">support@ccii.co.za</a></td>
</tr>
<tr>
<td>URL</td>
<td><a href="http://www.ccii.co.za/">http://www.ccii.co.za/</a></td>
</tr>
</tbody>
</table>

5.5 **Product Support**

Support on C²I² Systems products is available telephonically between Monday and Friday from 09:00 to 17:00 CAT. Central African Time (CAT = GMT + 2).
## Annexure A

### FDDI vs FC

<table>
<thead>
<tr>
<th>FDDI</th>
<th>FC</th>
</tr>
</thead>
<tbody>
<tr>
<td>ccFddiEndxxx.a</td>
<td>ccfcEndxxx.a</td>
</tr>
<tr>
<td>fddiLoad</td>
<td>ccfcLoad</td>
</tr>
<tr>
<td>-</td>
<td>ccfcShim.a</td>
</tr>
<tr>
<td>*<em>ccfddiGetStats(struct cc_fddi_mib_type <em>data)</em></em> - returns FDDI Management Information Base (MIB)</td>
<td>*<em>ccfddiGetStats(struct cc_fddi_mib_type <em>data)</em></em> - returns a structure with fields set to NULL</td>
</tr>
<tr>
<td><strong>ccfddiClrStats(void)</strong> - reset counters in the FDDI MIB</td>
<td><strong>ccfddiClrStats(void)</strong> - reset some fields related to FC</td>
</tr>
<tr>
<td><strong>struct cc_fddi_mib_type</strong> - all fields related to FDDI MIB</td>
<td><strong>struct cc_fddi_mib_type</strong> - all fields set to NULL except:</td>
</tr>
<tr>
<td>rx_packets;0</td>
<td>rx_packets;</td>
</tr>
<tr>
<td>tx_packets;</td>
<td>rx_errors;</td>
</tr>
<tr>
<td>tx_errors;</td>
<td></td>
</tr>
</tbody>
</table>
### Annexure B

#### Comparing Use of the FC and FDDI END VxWorks Software Drivers

<table>
<thead>
<tr>
<th>Getting Started With the FC END VxWorks Software Driver</th>
<th>Getting Started With the FDDI END VxWorks Software Driver</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Boot VxWorks and load the ccfcEndxxx.a¹ lib with 'ld &lt; ccfcEndxxx.a'</td>
<td>1. Boot VxWorks and load the ccFddiEndxxx.a¹ lib with 'ld &lt; ccFddiEndxxx.a'</td>
</tr>
<tr>
<td>2. Type 'muxDevStart ( muxDevLoad ( unitno, ccfcLoad,&quot;&quot;,0,0 ))' to attach and start the driver. Verify by using muxShow. The unitno should be 0 for Function[0] and 1 for Function[1] on the FC Adapter.</td>
<td>2. Type 'muxDevStart ( muxDevLoad ( unitno, fddiLoad,&quot;&quot;,0,0 ))' to attach and start the driver. Verify by using muxShow. The unitno should be 0 for first adapter and 1 for second FDDI Adapter.</td>
</tr>
<tr>
<td>3. Attach the Driver to the TCP/IP protocol stack. Type 'ipAttach unitno, &quot;ccfc&quot; Use muxShow and ifShow to confirm that the ccfcunitno exists and was attached to TCP/IP.</td>
<td>3. Attach the Driver to the TCP/IP protocol stack. Type 'ipAttach unitno, &quot;fddi&quot; Use muxShow and ifShow to confirm that the ccfcunitno exists and was attached to TCP/IP.</td>
</tr>
<tr>
<td>3. Set the IP address using 'ifAddrSet'. Refer to VxWorks Programmers Guide paragraph '5.2.5 TCP/IP Internet Protocols and Addresses' and paragraph '5.3 Configuring the Network' for setting up IP host names and IP routing.</td>
<td>3. Set the IP address using 'ifAddrSet'. Refer to VxWorks Programmers Guide paragraph '5.2.5 TCP/IP Internet Protocols and Addresses' and paragraph '5.3 Configuring the Network' for setting up IP host names and IP routing.</td>
</tr>
<tr>
<td>4. Confirm the FC LAN connection using ping. An example is shown on the next page. Note: This example assumes there is already a NIC on the FC LAN which has been set up with an IP address of 10.0.0.1.</td>
<td>4. Confirm the FDDI LAN connection using ping. An example is shown on the next page. Note: This example assumes there is already a NIC on the FDDI LAN which has been set up with an IP address of 10.0.0.1.</td>
</tr>
</tbody>
</table>

¹ xxx=X86 for Intel X86 PC, DY4PPC for DY4179/181/182 or 5100 for MV5100 PPC
-> muxDevStart (muxDevLoad (0, ccfcLoad, "", 0, 0))

value = 0 = 0x0

-> ipAttach 0 , "ccfc"
value = 0 = 0x0

-> ifShow

  ppp (unit number 0):
  Flags: (0x71) UP POINT-TO-POINT ARP RUNNING
  Internet address: 172.16.0.2
  Destination Internet address: 172.16.0.1
  Netmask 0xffff0000 Subnetmask 0xffff0000
  Metric is 0
  Maximum Transfer Unit size is 1 500
  5 packets received; 5 packets sent
  0 input errors; 0 output errors
  0 collisions

  lo (unit number 0):
  Flags: (0x69) UP LOOPBACK ARP RUNNING
  Internet address: 127.0.0.1
  Netmask 0xff000000 Subnetmask 0xff000000
  Metric is 0
  Maximum Transfer Unit size is 4 096
  0 packets received; 0 packets sent
  0 input errors; 0 output errors
  0 collisions

  ccfc (unit number 0):
  Flags: (0x63) UP BROADCAST ARP RUNNING
  Netmask 0xffffffff Subnetmask 0xffffffff
  Ethernet address is 00:50:C2:18:D0:00
  Metric is 0
  Maximum Transfer Unit size is 16 384
  18 packets received; 0 packets sent
  0 input errors; 0 output errors
  0 collisions
value = 18 = 0x12

-> ifAddrSet "ccfc0","10.0.0.4"
value = 0 = 0x0

-> muxDevStart (muxDevLoad (0, fddiLoad, "", 0, 0))

value = 0 = 0x0

-> ipAttach 0 , "fddi"
value = 0 = 0x0

-> ifShow

  ppp (unit number 0):
  Flags: (0x71) UP POINT-TO-POINT ARP RUNNING
  Internet address: 172.16.0.2
  Destination Internet address: 172.16.0.1
  Netmask 0xffff0000 Subnetmask 0xffff0000
  Metric is 0
  Maximum Transfer Unit size is 1 500
  5 packets received; 5 packets sent
  0 input errors; 0 output errors
  0 collisions

  lo (unit number 0):
  Flags: (0x69) UP LOOPBACK ARP RUNNING
  Internet address: 127.0.0.1
  Netmask 0xff000000 Subnetmask 0xff000000
  Metric is 0
  Maximum Transfer Unit size is 4 096
  0 packets received; 0 packets sent
  0 input errors; 0 output errors
  0 collisions

  fddi (unit number 0):
  Flags: (0x63) UP BROADCAST ARP RUNNING
  Netmask 0xffffffff Subnetmask 0xffffffff
  Ethernet address is 00:00:5a:45:f0:46
  Metric is 0
  Maximum Transfer Unit size is 4 491
  18 packets received; 0 packets sent
  0 input errors; 0 output errors
  0 collisions
value = 18 = 0x12

-> ifAddrSet "fddi0","10.0.0.4"
value = 0 = 0x0
-> ping "10.0.0.1"
  PING 10.0.0.1: 56 Data Bytes
  64 Bytes from 10.0.0.1: icmp_seq=1.
    time=0. ms
  64 Bytes from 10.0.0.1: icmp_seq=2.
    time=0. ms
  64 Bytes from 10.0.0.1: icmp_seq=3.
    time=0. ms
  64 Bytes from 10.0.0.1: icmp_seq=4.
    time=0. ms
  64 Bytes from 10.0.0.1: icmp_seq=5.
    time=0. ms

Load the FDDI Shim

In some instances, the FC Adapter will be used to replace the FDDI Adapter in an existing system. In this case, there is no need to change the user applications. The only difference between the FC and FDDI END VxWorks Software Drivers are that the fddiGetStats() routine is replaced by the ccfcGetStats() routine. The shim is present to restore the difference. Figure 1 shows a graphical layout of the difference between the FDDI END VxWorks Software Driver and the FC END VxWorks Software Driver. Users of the FDDI END VxWorks Software Driver will be familiar with the structure cc_fddi_mib_type and the following routines:

- void ccfdiGetStats
  (struct cc_fddi_mib_type *data)
- void ccfdiClrStats(void)

Although the FC END VxWorks Software Driver is not compatible with the FDDI MIB, calls to the MIB routines will not result in an error as long as the FDDI shim module is loaded. In the FDDI structure cc_fddi_mib_type, some fields may or may not correspond with fields in structures of the FC END VxWorks Software Driver. As per default, all the fields will be set to NULL.

Load the FDDI shim with ‘ld < ccFddiShim.o’.