

# » VXFabric™ «



## Release Notes for VX6060, VX3035, VX3042 & VX3044

SD.DT.G18-0e - February 2013

## Revision History

Publication Title:		VXFabric™ Release Notes	
Doc. ID:		SD.DT.G18-0e	
Rev.	Brief Description of Changes		Date of Issue
0e	Initial Version		02-2013

Copyright © 2013 Kontron AG. All rights reserved. All data is for information purposes only and not guaranteed for legal purposes. Information has been carefully checked and is believed to be accurate; however, no responsibility is assumed for inaccuracies. Kontron and the Kontron logo and all other trademarks or registered trademarks are the property of their respective owners and are recognized. Specifications are subject to change without notice.

## Proprietary Note

This document contains information proprietary to Kontron. It may not be copied or transmitted by any means, disclosed to others, or stored in any retrieval system or media without the prior written consent of Kontron or one of its authorized agents.

The information contained in this document is, to the best of our knowledge, entirely correct. However, Kontron cannot accept liability for any inaccuracies or the consequences thereof, or for any liability arising from the use or application of any circuit, product, or example shown in this document.

Kontron reserves the right to change, modify, or improve this document or the product described herein, as seen fit by Kontron without further notice.

## Trademarks

This document may include names, company logos and trademarks, which are registered trademarks and, therefore, proprietary to their respective owners.

## Environmental Protection Statement

This product has been manufactured to satisfy environmental protection requirements where possible. Many of the components used (structural parts, printed circuit boards, connectors, batteries, etc.) are capable of being recycled.

Final disposition of this product after its service life must be accomplished in accordance with applicable country, state, or local laws or regulations.



**Environmental protection is a high priority with Kontron.**

**Kontron follows the DEEE/WEEE directive.**

**You are encouraged to return our products for proper disposal.**

The Waste Electrical and Electronic Equipment (WEEE) Directive aims to:

- reduce waste arising from electrical and electronic equipment (EEE)
- make producers of EEE responsible for the environmental impact of their products, especially when they become waste
- encourage separate collection and subsequent treatment, reuse, recovery, recycling and sound environmental disposal of EEE
- improve the environmental performance of all those involved during the lifecycle of EEE

## Conventions

This guide uses several types of notice: Note, Caution, ESD.



Note: this notice calls attention to important features or instructions.



Caution: this notice alert you to system damage, loss of data, or risk of personal injury.



ESD: This banner indicates an Electrostatic Sensitive Device.

All numbers are expressed in decimal, except addresses and memory or register data, which are expressed in hexadecimal. The prefix `0x` shows a hexadecimal number, following the `C` programming language convention.

The multipliers `k`, `M` and `G` have their conventional scientific and engineering meanings of  $*10^3$ ,  $*10^6$  and  $*10^9$  respectively. The only exception to this is in the description of the size of memory areas, when `K`, `M` and `G` mean  $*2^{10}$ ,  $*2^{20}$  and  $*2^{30}$  respectively.



When describing transfer rates, `k` `M` and `G` mean  $*10^3$ ,  $*10^6$  and  $*10^9$  *not*  $*2^{10}$   $*2^{20}$  and  $*2^{30}$ .

In PowerPC terminology, multiple bit fields are numbered from 0 to n, where 0 is the MSB and n is the LSB. PCI and CompactPCI terminology follows the more familiar convention that bit 0 is the LSB and n is the MSB.

Signal names ending with an asterisk (\*) or a hash (#) denote active low signals; all other signals are active high.

Signal names follow the PICMG 2.0 R3.0 CompactPCI Specification and the PCI Local Bus 2.3 Specification.

## For Your Safety

Your new Kontron product was developed and tested carefully to provide all features necessary to ensure its compliance with electrical safety requirements. It was also designed for a long fault-free life. However, the life expectancy of your product can be drastically reduced by improper treatment during unpacking and installation. Therefore, in the interest of your own safety and of the correct operation of your new Kontron product, you are requested to conform with the following guidelines.

### High Voltage Safety Instructions



**Warning!**

All operations on this device must be carried out by sufficiently skilled personnel only.



**Caution, Electric Shock!**

Before installing a not hot-swappable Kontron product into a system always ensure that your mains power is switched off. This applies also to the installation of piggybacks. Serious electrical shock hazards can exist during all installation, repair and maintenance operations with this product. Therefore, always unplug the power cable and any other cables which provide external voltages before performing work.

## Special Handling and Unpacking Instructions



### ESD Sensitive Device!

Electronic boards and their components are sensitive to static electricity. Therefore, care must be taken during all handling operations and inspections of this product, in order to ensure product integrity at all times

Do not handle this product out of its protective enclosure while it is not used for operational purposes unless it is otherwise protected.

Whenever possible, unpack or pack this product only at EOS/ESD safe work stations. Where a safe work station is not guaranteed, it is important for the user to be electrically discharged before touching the product with his/her hands or tools. This is most easily done by touching a metal part of your system housing.

It is particularly important to observe standard anti-static precautions when changing piggybacks, ROM devices, jumper settings etc. If the product contains batteries for RTC or memory backup, ensure that the board is not placed on conductive surfaces, including anti-static plastics or sponges. They can cause short circuits and damage the batteries or conductive circuits on the board.

## General Instructions on Usage

In order to maintain Kontron's product warranty, this product must not be altered or modified in any way. Changes or modifications to the device, which are not explicitly approved by Kontron and described in this manual or received from Kontron's Technical Support as a special handling instruction, will void your warranty.

This device should only be installed in or connected to systems that fulfill all necessary technical and specific environmental requirements. This applies also to the operational temperature range of the specific board version, which must not be exceeded. If batteries are present, their temperature restrictions must be taken into account.

In performing all necessary installation and application operations, please follow only the instructions supplied by the present manual.

Keep all the original packaging material for future storage or warranty shipments. If it is necessary to store or ship the board, please re-pack it as nearly as possible in the manner in which it was delivered.

Special care is necessary when handling or unpacking the product. Please consult the special handling and unpacking instruction on the previous page of this manual.

---

## Table Of Contents

Chapter 1 - Overview .....	1
Chapter 2 - VXFabric™ Product Installation .....	2
Chapter 3 - VXFabric™ Product Description .....	3
3.1 Initial Configuration of the VXFabric™ .....	3
3.2 RAW Mode: Performing Simple DMA Access .....	5
3.3 Different Types of Memory Allocation .....	7
3.4 Test of the DMA Posted Mode .....	8
3.5 Option Related to the DMA Channel .....	9
3.6 Configuration of VXFabric™ in TCP/IP Mode .....	10
3.7 Use Different Standard Network Commands .....	12
3.8 Evaluate the TCP Performances Using Benchmark Iperf .....	13

## Chapter 1 - Overview

Kontron VXFabric™ is an open infrastructure which implements efficient inter-board communication at hardware speed. The architecture is compliant with the OpenVPX standard (VITA 65) which defines two main hardware topologies of the backplane: **distributed and centralized topologies**.

In its current form, VXFabric™ can simultaneously interconnect up to 12 nodes via PCI Express. The physical interconnect through the backplane can be made in various implementations according to the choice of backplane style:

- A distributed PCIe backplane can be used without the need of any PCIe switch. In this case, very compact architectures can be proposed, as long as the relevant data path between nodes is implemented in the backplane. This data path is mostly application dependent, making distributed backplane the right solution for cost effective deployments.
- When a higher bandwidth and more flexibility in the data plane is required, a centralized topology is recommended. With a PCIe switch, such as the Kontron 3U VPX PCIe and Ethernet Hybrid switch VX3905, it is possible to interconnect up to 12 nodes with the VXFabric™ via PCI Express and establish communications links between any nodes using the same backplane. This approach fits well the HPEC application domain, as well as lab use where multiple application data path can be evaluated with the same equipment.

The VXFabric™ product allows communication by DMA between boards such as the VX6060/VX3035/VX304x over the VPX backplane, or between two CPUs of the same board.

This means that all the boards in the fabric have a means of reading or writing to a remote at speeds limited only by the PCIe bus, which is extremely fast. When large amounts of data are to be transferred, a DMA engine is available on each board to handle the transactions, and so free up the CPU for other work.

Two modes are available:

a. The RAW mode:

This mode is performances and low latency oriented.

Basically, it allows to transfer data, from one board to the other one, using DMA engine.

b. The TCP/IP mode:

This mode is standard oriented. So the communication between the boards can be established using all standard network protocols such as TCP, UDP, IP... through the socket interface as well as the standard command such as `ssh`, `scp`, `ping`...

The implementation consists of emulating a complete Ethernet interface on the top of the PCI express. (similar to implementations of pseudo-Ethernet found in virtual machines).

As a result, on each node present into the VXFabric™ configuration, an Ethernet interface named `vxeth0` will be created.

## Chapter 2 - VXFabric™ Product Installation

Follow the procedure below to install the VXFabric™ package onto an existing bootable media (SATA disk or USB flash):

- ▶ Insert the Kontron VXFabric™ CD-ROM into the USB DVD-ROM drive.
- ▶ Boot the VX304x board on the SATA disk previously installed with the Fedora 16 64 bits.
- ▶ Login as root and run following commands:

```
# mount /dev/sr0 /mnt
# cd /mnt
rpm -i vxfabric-<kernel_version>ki7.fc16.x86_64.<package_version>.x86_64.rpm
```

To verify the installation has been successful, run the following command:

```
# rpm -ql vxfabric
```

The VXFabric™ product is ready to be used.

## Chapter 3 - VXFabric™ Product Description

In the next chapters, it is assumed that the machine is based on two VX304x boards plugged into the same VPX rack (respectively slot 1 and slot 2). The VPX backplane features the x8 gen3 mode.

### 3.1 Initial Configuration of the VXFabric™

Boot both of the boards and verify the presence of VXFabric™ component by running:

```
# service vxfabric start
```

Then, verify the VXFabric™ status by running:

```
# vxfabric -a
-----
VX FABRIC 0 configuration
-----
N H S S C S P P M A ( I A (
O O L T P Y C C E R S 0 R S
D S O A U S I I M E I E I
E T T T C B D O A Z A Z
I U O U E R E E
D S N S V Y ) )
-----
1 * 1 CONNECTED A * 0x064001000(0x3fff000) ( )
3 2 CONNECTED A 6 0 0x080000000(0x4000000) 0x0f4000000(0x40000)
-----
```

The status **CONNECTED** reveals both of the targets have been recognized and configured correctly. It indicates also the current mode selected is the TCP/IP mode. If the RAW mode would have been selected, the displayed status would have been **READY**. To select the RAW mode instead of the TCP/IP mode, simply edit the `/etc/sysconfig/vxfabric` file and comment the line `VXFABRIC_IP_MODE=yes`

So the resulting `/etc/sysconfig/vxfabric` should look like:

```
# cat /etc/sysconfig/vxfabric
# force the HOST to be a VPX SYSTEM BOARD controller :
# VXFABRIC_SYSTEMBOARD=1
# Set the size of the BACKPLANE memory area to 0x4000000 ( default value )
# VXFABRIC_BACKPLANEMEMAREASIZE=0x4000000
# Set the size of the LOCAL memory area to 0x1000000 ( default value )
# VXFABRIC_LOCALMEMAREASIZE=0x1000000
# Set the size of the RING of buffers to 0x1000000 ( default value )
# VXFABRIC_RINGSIZE=0x1000000
# Set the POSTED mode to ENABLE ( default value )
# VXFABRIC_POSTED_MODE=1
# Set the maximum number of POSTED requests to 6 ( default value )
# VXFABRIC_MAX_POSTED_REQUEST_NUM=6
# Set the minimum POSTED packet size to 0xF000 ( default value )
# VXFABRIC_MIN_POSTED_PACKET_SIZE=0xF000
# Select the TCP/IP mode
# VXFABRIC_IP_MODE=yes
# Select the DMA mode for TCP/IP ( default value )
# VXFABRIC_IP_DMAMODE=1
# Set the MTU to 64472 ( default value )
# VXFABRIC_IP_MTU=64472
# Select the SKB no copy mode ( default value )
# VXFABRIC_IP_SKBNOCOPY=1
VXFABRIC_PERF_OPTIM=1
```



The default parameters are optimized for most of the configurations. So do not change any of them except if it is suggested in the present document.

Then reboot the machine to take into account this change.

If you want to force the start of the service at boot step, run:

```
# chkconfig vxfabric on
```

If you want to force the start of the service at boot step, run:

```
# chkconfig vxfabric on
```

If you need detailed information related to the different VXFabric™ components, read the following user's manuals:

```
# man vx
# man vxfabric
# man vxfabric_api
# man vxdma
```

### 3.2 RAW Mode: Performing Simple DMA Access

At first, verify that the targets are available and configured in the right mode.

```
# vxfabric -a
-----
VX FABRIC 0 configuration
-----
N H S S          C S P P M A (      I A (
O O L T          P Y C C E R S      O R S
D S O A          U S I I M E I      E I
E T T T          C B D O A Z          A Z
I      U          O U E R E          E
D      S          N S V Y          )
-----
1 * 1  READY      A *          0x064001000(0x3fff000)      ( )
3   2  READY      A   6 0 0x080000000(0x4000000)  0x0f4000000(0x40000)
-----
```

A status other than READY for one or more targets indicates something is wrong.

The following list indicates all the possible values for the target status.

- NOTAVAIL : Target no more accessible.
- FIRM- : Firmware bringup step.
- FIRM : FIRMWARE booted.
- BOOTING : OS booting step
- BOOTED : OS booted.
- REBOOT : OS rebooting step.
- HINIT : VXFabric™ hardware initialized.
- SINIT : VXFabric™ software initialized.
- READY : VXFabric™ RAW mode available.
- CONNECTED : VXFabric™ TCP/IP mode available.

The System Controller board has an essential role which consists of probing, recognizing all the boards present on the VPX backplane.

It is also in charge of tracking down all the potential status changes of these targets.

So at this step, access to the System Controller board (slot 1) and initiate a DMA transfer by running:

```
# vxfabric -n 1 -N 3 -B 0x4000000
```

with

**n 1** referencing the node number 1 as source node (the node number appears into the first column of VXFabric™ -a output).

**-N 3** referencing the node number 3 as destination node.

**-B 0x4000000** requests a transfer size of 0x4000000

The output of the previous command should look like:

```
NODEID 1 PHYSICAL MEM SOURCE addr = 0x64001000  
NODEID 3 PHYSICAL MEM DESTINATION addr = 0x80000000  
Write of 0x4000000 bytes in 12749 usec (5263 MB/s)
```



By default a memory area of 4 megabytes is allocated on each node present on the VXFabric™. This memory area can be used in a free way in the RAW mode. On the contrary, in the TCP/IP mode, this memory area is directly managed by the vxeth interface (not accessible to the user).

### 3.3 Different Types of Memory Allocation

VXFabric™ proposes different options concerning the allocation of memory areas.

These areas used as source or destination buffer during the transfer of data can be allocated in a user virtual space or kernel space, and can be contiguous or not in the physical address space.

In this chapter, all the options of VXFabric™ related to the memory area allocations using the VXFabric™ utility are presented.

Of course all these capabilities are available through the VXFabric™ API (in this case the VXFabric™ utility plays the role of the USER application).

- a. `-s 0`: implies to use an existant physical address.

At first, use the following command in order to identify the base address of the physical memory area allocated by the VXFabric™ driver:

```
# base_address=`vxfabric -a | grep '\*' | awk '{print $7}'` ; echo $base_address
0x064001000(0x3fff000)
```

Then, use the resulting address in this command:

```
# vxfabric -a -S 0x064001000 -s 0 -N 3 -B 0x4000000
NODEID 1 PHYSICAL MEM SOURCE addr = 0x64001000
NODEID 3 PHYSICAL MEM DESTINATION addr = 0x80000000
Write of 0x4000000 bytes in 12749 usec (5263 MB/s)
```

In this case, this command is equivalent to the previous one.

No new memory area has been allocated. The memory area used is the one already allocated by the VX driver for the current node.

- b. `-s 1`: implies to allocate and use a virtual buffer.

No address should be given in this case.

So run this command:

```
# vxfabric -s 1 -N 3 -B 0x80000
VIRTUAL SOURCE addr = 0x7f4809b65000
NODEID 3 PHYSICAL MEM DESTINATION addr = 0x80000000
Write of 0x80000 bytes in 225 usec (2330 MB/s)
```

In this case, the buffer has been allocated into the USER space (by the VXFabric™ tool itself). As a result the memory area is not contiguous.

However, the VXFabric™ driver features DMA engines which manage the non contiguous memory areas and ensure good transfer realisation.

- c. `-s 2`: allocate and use some physically contiguous memory.

```
# vxfabric -s 2 -N 3 -B 0x4000000
Allocating the SOURCE DMA buffer (CONTIGUOUS MEMORY) Size = 0x4000000
PHYSICAL MEM SOURCE addr = 0x68000000
NODEID 3 PHYSICAL MEM DESTINATION addr = 0x80000000
Write of 0x4000000 bytes in 12745 usec (5265 MB/s)
```

In this case, the buffer has been allocated into the KERNEL space, authorizing physical contiguous memory allocation, and has been mapped directly into the user application context (VXFabric™ tool).

## 3.4 Test of the DMA Posted Mode

By default, when using VXFabric™ tool, the DMA Posted Mode is disabled. When several DMA requests are initiated (-L parameter), they will all be serialized.

To understand the impact of the DMA Posted Mode, the two following commands can be run.

First command:

```
# vxfabric -n 1 -N 3 -B 0x10000 -L 0x100
NODEID 1 PHYSICAL MEM SOURCE addr = 0x64001000
NODEID 3 PHYSICAL MEM DESTINATION addr = 0x80000000
Write of 0x1000000 bytes in 10087 usec (1663 MB/s)
```

Then same quantity of transfers but in DMA Posted Mode:

```
# vxfabric -n 1 -N 3 -B 0x10000 -L 0x100 -P
NODEID 1 PHYSICAL MEM SOURCE addr = 0x64001000
NODEID 3 PHYSICAL MEM DESTINATION addr = 0x80000000
Write of 0x1000000 bytes in 3182 usec (5272 MB/s)
```

This example shows dramatic performance differences when the DMA posted mode is disabled or enabled.

## 3.5 Option Related to the DMA Channel

The VXFabric™ driver gives access to 4 DMA channels.

By default, VXFabric™ works in Round Robin mode. So, a different channel is used for each new request.

To select explicitly one particular channel use the `-C` option.

For example:

```
# vxfabric -n 1 -N 3 -C 3 -B 0x1000000
NODEID 1 PHYSICAL MEM SOURCE addr = 0x64001000
NODEID 3 PHYSICAL MEM DESTINATION addr = 0x80000000
Write of 0x1000000 bytes in 3274 usec (5124 MB/s)
```

will cause a transfer using the channel number 3.

Using the stats command, verify the right channel has been used.

```
# cat /proc/vxfabric_dma0
0: 0 0 0 0
1: 0 0 0 0
2: 0 0 0 0
3: 0 16777216 0 0
```

### 3.6 Configuration of VXFabric™ in TCP/IP Mode

At first, select the TCP mode of VXFabric™ by adding the line:

```
# VXFABRIC_IP_MODE=yes
```

to the `/etc/sysconfig/vxfabric` file.

Then start the VXFabric™ service by:

```
# service vxfabric start
```

Rerun ( both sides )

```
# vxfabric -a
```

and verify the status of all the targets are set to `CONNECTED`.

The output should look like:

```
-----
VX FABRIC 0 configuration
-----
N H S S          C S P P M A (          I A (
O O L T          P Y C C E R S          O R S
D S O A          U S I I M E I          E I
E T T T          C B D O A Z          A Z
I          U          O U E R          E          E
D          S          N S V Y          )          )
-----
1 * 1  CONNECTED A *          0x064001000(0x3fff000)          (          )
3   2  CONNECTED A   6 0 0x080000000(0x4000000)  0x0f4000000(0x40000)
-----
```

Control the presence of the `vxeth0` network interface by running:

```
# ifconfig vxeth0
```

The output should be similar to:

```
vxeth0 Link encap:Ethernet HWaddr 02:00:00:00:00:01
        inet addr:192.168.10.1 Bcast:192.168.10.255 Mask:255.255.255.0
        inet6 addr: fe80::ff:fe00:1/64 Scope:Link
        UP BROADCAST RUNNING MTU:64472 Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:0 (0.0 b) TX bytes:0 (0.0 b)
```

This interface conforms strictly to a standard network interface.

The only specific points to be noticed:

- Value of the MTU = 64472 not usual at all in the network area:

The advantage is clear in this case: improvement of the performance by increasing the maximal size of a network packet.

- The HWaddr field which has nothing to do with a standard MAC addr. 02 is a value specific to local embedded network and 01 is related to the nodeid of the current target.

The presence of this network interface vxeth0 reveals an essential aspect of the VXFabric™ component:

- A complete emulation of an Ethernet interface at the top of the VPX/PCI express bus.

As a result all the services/protocols available with a standard Ethernet interface are available also with VXFabric™. With the real advantage linked to the PCI express huge transfer speed.

So now, log on the System Controller (node 1) and verify the network communication is now established by running:

```
# ping 192.168.10.3
PING 192.168.10.3 (192.168.10.3) 56(84) bytes of data.
64 bytes from 192.168.10.3: icmp_req=1 ttl=64 time=1.14 ms
64 bytes from 192.168.10.3: icmp_req=2 ttl=64 time=0.509 ms
64 bytes from 192.168.10.3: icmp_req=3 ttl=64 time=0.356 ms
64 bytes from 192.168.10.3: icmp_req=4 ttl=64 time=0.490 ms
64 bytes from 192.168.10.3: icmp_req=5 ttl=64 time=0.507 ms
....
PING 192.168.10.3 (192.168.10.3) 56(84) bytes of data.
```

The ping -f command worked well without error.

Of course, at this step, all types of network oriented commands are usable.

## 3.7 Use Different Standard Network Commands

Run `ping -f` and verify no packet is lost:

```
./# ping -f 192.168.10.3
--- 192.168.10.3 ping statistics ---
2229 packets transmitted, 2229 received, 0% packet loss, time 614ms
rtt min/avg/max/mdev = 0.097/0.111/0.585/0.024 ms, ipg/ewma 0.276/0.109 ms
```

Then test the `scp` command by running:

```
#scp -r /boot/* 192.168.10.3:/tmp
```

The output should look like:

```
# scp -r /boot/* 192.168.10.3:/tmp
The authenticity of host '192.168.10.3 (192.168.10.3)' can't be established.
RSA key fingerprint is 60:ea:0a:e9:dd:09:3b:ad:11:26:7f:84:bb:37:18:8a.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.10.3' (RSA) to the list of known hosts.
root@192.168.10.3's password:
config-3.4.9-12283.ki7.fc16.x86_64          100% 120KB 119.7KB/s  00:00
splash.xpm.gz                             100%  66KB  65.7KB/s  00:00
afs_be.mod                                100%  7152    7.0KB/s  00:00
xnu_uuid.mod                              100%  2704    2.6KB/s  00:00
gcry_md5.mod                              100%  4676    4.6KB/s  00:00
.....
```

Finally, use the `ssh` command in this way:

```
# ssh 192.168.10.3
find .
```

notice that no error is detected.

Once again, there is no limitation concerning the usage of network protocol such as UDP, NFS... providing the VXFabric™ network interface emulates strictly a real one.

## 3.8 Evaluate the TCP Performances Using Benchmark Iperf

Install the iperf benchmark if needed:

```
# yum install iperf
```

Then run a server on one side (for example on the System controller):

```
# iperf -s -f M
```

And run the client on the other side:

```
# iperf -c 192.168.10.1 -f M
```

The output should be similar to:

```
Client connecting to 192.168.10.3, TCP port 5001
TCP window size: 0.62 MByte (default)
-----
[ 3] local 192.168.10.1 port 38214 connected with 192.168.10.3 port 5001

[ ID] Interval      Transfer    Bandwidth
[ 3]  0.0-10.0 sec 38178 MBytes 3818 MBytes/sec
```

Of course, the performance depends on the characteristics of the hardware configuration.

**MAILING ADDRESS**

Kontron Modular Computers S.A.S.  
150 rue Marcelin Berthelot - BP 244  
ZI TOULON EST  
83078 TOULON CEDEX - France

**TELEPHONE AND E-MAIL**

+33 (0) 4 98 16 34 00  
sales@kontron.com  
support-kom-sa@kontron.com

For further information about other Kontron products, please visit our Internet web site:  
[www.kontron.com](http://www.kontron.com).