

I XDP2800 Firmware Release Notes

BETA Release Build 310D
Mar 12, 2004

NOTE:

- 1) Baud Rate is set to 9600 for this release.
2) Ensure that BOTH Ingress & Egress processors have identical versions of software.
3) Please read the Changes & Known Issues Sections at the beginning of each block below.
4) The build number displayed by the visionWARE boot manager will be 310D.
5) The build number displayed by the vxWorks BSP will be 310D.
6) The build number for all other components in this release is 310D.

* -----*
* VisionWare Boot Manager/Diagnostics *
* -----*

Changes made for this release:

- #1 All media loopback test "MAC" commands function, are repeatable, and no longer lock Ingress console.
#2 Uart -q -p commands were removed as test coverage is provided by UART interrupt tests.
#3 Timer -g test looks for B0 revision of IXP28x0 and no longer reports failure.
#4 TCAM test properly echos keystrokes.
#5 Media loopback test failures now give packet "expected" and "found" details that enhance debug.
#6 The i2c_pg test is operational and functions properly.
#7 The CSIX msf -c and -u tests check for CBUS flow control training patterns.
#8 All diagnostic tests return pass/fail for VisionWARE script usage. This gives us stop on error support for chip and systems testing.
#9 Improved QDR initialization algorithm, updated to adhere to SRAM Initialization application note.
#10 Changed the default message scrolled out to the alphanumeric display from "Deer Island" to "IXDP2800".
#11 Optimized the QDR SRAM initialization code for more generic application.
#12 Removed crypto command from the diagnostics command menu.
#13 Lengthened the flash write timeout due to 28F128 spec update erratta 1 which states the write complete time to be TBD.
#14 Added new IDT TCAM diagnostic files Idt0sSupport.cpp and Idt0sSupport.h

#15 Updated Licensing information in the following files
 ethernet.c
 ether_net.h
 copy2flash.h

#16 Removed unused source code files IXP2000gpio.h and IXP2000i2c.h

Known issues:

- #1 The temperature display on the Ingress NPU has not been calibrated for this board, there may be a temperature offset of +/- 15 degrees C.
- #2 The "shell" command always displays the MAC address as "11 22 33 44 55 66".

#####

Tool Chain:

The tool chain needed to compile visionWARE:

visionWARE:

visionWARE Development Kit Version 2.1

IXA_SDK_3.5:

IXA_SDK_3.5 me tools for ucode compilation

Instruction to upgrade visionWARE(Manager\Diagnosis):

Over NIC (TFTP) - it is assumed that TFTP server is setup:

For both IXP2800 and IXP2850:

```
Egress> copy \\<TFTP server IP address>\vWARE28x0.bin  
Ingress> copy \\<TFTP server IP address>\vWARE28x0.bin  
Egress> update vWARE28x0.bin 10000000  
Ingress> update vWARE28x0.bin 10000000
```

NOTE: When updating visionWARE update BOTH the master\egress and the slave\ingress NPUs BEFORE rebooting the platform. Failure to do so may require that FLASH devices be programmed out of the system on an external ROM programmer. Always delete the boot.ini file in both NPUs flash file system BEFORE using the update command.

Over Serial Port:

not supported.

Rebuilding visionWARE:

For both IXP2800 and IXP2850:

#1 In the VisionWARE build environment click on projects and then point to the project file contained in this release. This will be found under the boardsupport\visionWARE\projects\IXP28x0 directory as IXP28x0.vxp

#2 To build, click the "clean" icon, then click the "build" icon.

#3 Build output will be found under boardsupport\visionWARE\projects\builds\

I XDP2800FW3.51FCSReleaseNotes310D.txt
I XMB2800\ROM as "vware28x0.bin"

#4 Delete the boot.ini file from flash prior to running the flash "update" command from the VisionWARE prompt.

#5 Build output for flash programmers will be found under boardsupport\visionWARE\projects\builds\I XMB2800\ROM as "vware_swap28x0.bin"

```
*****  
*-----*  
*                               *  
*                               *  
*                               *  
*                               *  
*-----*  
*****
```

Changes made in this release:

- #1 This release of the vxWorks BSP requires the Tornado 2.2.1 build environment inclusive of the drivers and x-bit patches.
- #2 Added the setting of the x bit extended cache attribute to disable write coalescing for access to the SRAM extended memory areas.
- #3 Included in this release of the vxworks bsp the serial and ethernet rom images are included.
- #4 Updated licensing information for the following files:

- sysEnd.c
- usrNetwork.c
- usrExtra.c
- usrConfig.c
- ixdp2000Pci.h
- flashMem.c
- configNet.h
- bootInit.c
- bootConfig.c
- pciIomapShow.c
- pciIomapLib.h
- ixdp2000Sio.h
- sysLib.s
- target.nr
- sysSerial.c
- sysLib.c
- ixdp2000Sio.c
- ixdp2000Pci.c
- ixdp2000IntrCtl.c
- ixdp2000.h
- config.h
- romInit.s
- pciIomapLib.c
- Makefile
- ixdp2000Timer.c
- ixdp2000Timer.h
- ixdp2000.c
- ixdp2800.h
- ixdp2000I2c.h
- ixdp2000I2c.c
- ixdp2000misc.h

Code changes made to BSP between build 219A and 310D:

NOTE: The lines numbers referenced below may be different for the source files from WindRiver. These line numbers are provided here as a reference.

- 1) config.h
lines 62 and 64: changed the BSP rev
- 2) ixdp2000.h
line 48: changed the build number (IXP_VERSION) to 3.50.310D
- 3) makefile
line 71: changed ROM_SIZE to 2 megabytes
- 4) syslib.c
added lines 126-129 as follows:

```
#define ATTR_XWCB (ATTR_WC | VM_STATE_EX_BUFFERABLE)
#define ATTR_XWNB (ATTR_WN | VM_STATE_EX_BUFFERABLE)
#define ATTR_XRCB (ATTR_RC | VM_STATE_EX_BUFFERABLE)
#define ATTR_XRNB (ATTR_RN | VM_STATE_EX_BUFFERABLE)
```

Changed lines 226-224 as follows:

```
MEM_DESC(SRAM_CHAN0_BASE, SRAM_CHAN0_BASE, SZ_64M, ATTR_WNN),
MEM_DESC(SRAM_CHO_BIT_SET_OPS_BASE, SRAM_CHO_BIT_SET_OPS_BASE, SZ_64M,
ATTR_XWNB),
MEM_DESC(SRAM_CHO_BIT_CLR_OPS_BASE, SRAM_CHO_BIT_CLR_OPS_BASE, SZ_64M,
ATTR_XWNB),
MEM_DESC(SRAM_CHO_ADD_OPS_BASE, SRAM_CHO_ADD_OPS_BASE, SZ_64M, ATTR_XWNB),

MEM_DESC(SRAM_CHAN1_BASE, SRAM_CHAN1_BASE, SZ_64M, ATTR_WNN),
MEM_DESC(SRAM_CH1_BIT_SET_OPS_BASE, SRAM_CH1_BIT_SET_OPS_BASE, SZ_64M,
ATTR_XWNB),
MEM_DESC(SRAM_CH1_BIT_CLR_OPS_BASE, SRAM_CH1_BIT_CLR_OPS_BASE, SZ_64M,
ATTR_XWNB),
MEM_DESC(SRAM_CH1_ADD_OPS_BASE, SRAM_CH1_ADD_OPS_BASE, SZ_64M, ATTR_XWNB),

MEM_DESC(SRAM_CHAN2_BASE, SRAM_CHAN2_BASE, SZ_64M, ATTR_WNN),
MEM_DESC(SRAM_CH2_BIT_SET_OPS_BASE, SRAM_CH2_BIT_SET_OPS_BASE, SZ_64M,
ATTR_XWNB),
MEM_DESC(SRAM_CH2_BIT_CLR_OPS_BASE, SRAM_CH2_BIT_CLR_OPS_BASE, SZ_64M,
ATTR_XWNB),
MEM_DESC(SRAM_CH2_ADD_OPS_BASE, SRAM_CH2_ADD_OPS_BASE, SZ_64M, ATTR_XWNB),

MEM_DESC(SRAM_CHAN3_BASE, SRAM_CHAN3_BASE, SZ_64M, ATTR_WNN),
MEM_DESC(SRAM_CH3_BIT_SET_OPS_BASE, SRAM_CH3_BIT_SET_OPS_BASE, SZ_64M,
ATTR_XWNB),
MEM_DESC(SRAM_CH3_BIT_CLR_OPS_BASE, SRAM_CH3_BIT_CLR_OPS_BASE, SZ_64M,
ATTR_XWNB),
MEM_DESC(SRAM_CH3_ADD_OPS_BASE, SRAM_CH3_ADD_OPS_BASE, SZ_64M, ATTR_XWNB),
```

- 5) In addition to the above, the following files have been modified by WindRiver Systems, as part of their porting efforts to Tornado 2.2.1. No line numbers are provided here as there are too many changes. It is suggested to do a 'diff' between the current file and the new one to find the changes that have gone into these files.

Contact WindRiver systems on any questions on BSP source or patches.

```
bootInit.c
bootConfig.c
sysEnd.c
usrConfig.c
usrExtra.c
usrNetwork.c
```

Known Issues:

When booting the vxWorks_rom_ser.bin images, the egress NPU will require an additional 1 minute to complete the boot sequence as compared to the ingress NPU. This will be fixed in the next release.

BSP VTS errors:

VTS testing completed successfully on this release of the vxWorks BSP.

#####

The following vxWorks images are included in this release:

bootrom.bin	(vxworks bootrom binary)
vxworks	(vxworks downloadable image)
vxWorks.sym	(vxworks symbol file)
vxworks_rom_ser	(vxworks serial ROM object)
vxworks_rom_ser.bin	(vxworks serial ROM binary)
vxworks_rom_eth	(vxworks ethernet ROM object)
vxworks_rom_eth.bin	(vxworks ethernet ROM binary)

To compile BSP:

NOTE: This release of the vxWorks BSP requires the Tornado 2.2.1 build environment.

- 1) Install Tornado 2.2.1 as per directions on the WindRiver website. The instructions are also included in the Intel IXA SDK 3.51 tools CD #1 release notes.
- 2) Run c:\WIND_BASE\host\x86-win32\bin\torvars.bat
- 3) Go to the ixdp2800_be directory on the PC where the BSP source files are located.
- 4) Depending on images you wish to build, choose one of the three below:
 - a) Type: make BOOTROM.bin and then make vxWorks
 - b) Type: make vxWorks_rom_ser.bin
 - c) Type: make vxWorks_rom_eth.bin

Procedure to flash and boot BOOTROM and VxWorks via Boot Manager:

Note: The BOOTROM.BM file is supplied on the SDK as part of the release configuration shipped for the IXDP2800 system. If you have not opted to rebuild the BOOTROM.BIN file, you may begin at step 4 below. If you choose to rebuild the bootrom, start at step 1.

- 1) Edit the makefile to NOT use the STANDALONE_BOOT addresses for ROM:
 - a) uncomment the following as shown in makefile:


```
# Use the following for Boot Manager launchable build
ROM_BASE_ADRS = c4800000 # base address of EPROM
ROM_TEXT_ADRS = c4808000 # code start addr in EPROM
ROM_SIZE = 00800000 # 8MB size of EPROM
```
 - b) comment OUT the following as shown in makefile:


```
# Use the following for STANDALONE_BOOT vxWorks build
#ROM_BASE_ADRS = c4000000 # base address of EPROM
#ROM_TEXT_ADRS = c4008000 # code start addr in EPROM
```

```
IXDP2800FW3.51FCSReleaseNotes310D.txt
#ROM_SIZE = 01000000 # 16MB size of EPROM
```

- 2) Add the following line to config.h:
#undef STANDALONE_BOOT
- 3) Build the BOOTROM.BIN
- 4) Copy the BOOTROM.bin from the TFTP host. At the Boot Manager Prompt, substituting the appropriate parameters for your environment, type:
"copy \\17.17.17.81\c:\ixdp2800_be\BOOTROM.bin BOOTROM.bm"
- 5) Execute a "dir" command at the boot manager prompt to confirm the file has been copied to the flash file system of the IXDP2800.
- 6) Load the file into memory. At the Boot Manager Prompt, type:
"load BOOTROM.bm 100000"
- 7) Copy the BOOTROM.bm image from memory to the free flash area. At the Boot Manager Prompt, type "copy2flash <sourceaddr> <destaddr> <length>":
"copy2flash 100000 c480000 40000"
- 8) Launch execution of the BOOTROM (slave(ingress) FIRST, then master(egress)). At the Boot Manager Prompt, type:
"launch c4808000"
- 9) At vxworks BOOTROM prompt, change the boot parameters on the master(egress).
- 10) Press '@' at prompt on master(egress).
- 11) At vxworks boot prompt, change the boot parameters on the slave(ingress).
- 12) Press '@' at prompt on the slave(ingress).
- 13) Once vxworks is booted on both the NPUs, use 'vxworks', as the corefile in target server configuration and connect to target server.

NOTE:

- 1) From Boot Manager launch the BOOTROM on the slave(ingress) BEFORE launching BOOTROM on the master(egress).
- 2) You must launch the BOOTROM on the master(egress) in order for the slave(ingress) to complete the BOOTROM load.
- 3) If loading and launching vxWorks via the bootrom on the slave(ingress) before doing so on the master(egress) requires that you wait until the slave(ingress) has completed the loading of the vxWorks symbol file before loading and launching vxWorks on the master(egress).
- 4) Loading and launching vxWorks via the bootrom on the master(egress) before doing so on the slave(ingress) imposes no timing requirement on loading and launching vxWorks on the slave(ingress).

Booting standalone BOOTROM (using 'BOOTROM.bin'):

- 1) Edit the makefile to use the STANDALONE_BOOT addresses for ROM:
 - a) comment out the following as shown in makefile:
Use the following for Boot Manager launchable build
#ROM_BASE_ADRS = c4800000 # base address of EPROM
#ROM_TEXT_ADRS = c4808000 # code start address in EPROM
#ROM_SIZE = 00800000 # 8MB size of EPROM
 - b) uncomment the following as shown in makefile:
Use the following for STANDALONE_BOOT vxWorks build
ROM_BASE_ADRS = c4000000 # base address of EPROM
ROM_TEXT_ADRS = c4008000 # code start address in EPROM
ROM_SIZE = 01000000 # 16MB size of EPROM
- 2) Add the following line to config.h:
#define STANDALONE_BOOT
- 3) Build the BOOTROM.BIN (for flashing over XSCALE JTAG tool) or BOOTROM_SWP.BIN (for external ROM burner flashing).
- 4) Install the BOOTROM image onto the system via the chosen flash method.
- 5) Power on the system and boot both the NPUs with BOOTROM.

- 6) At vxworks boot prompt, change the boot parameters on the master(egress).
- 7) Press '@' at prompt on master(egress).
- 8) At vxworks boot prompt, change the boot parameters on the slave(ingress).
- 9) Once vxworks is booted on both the NPUs, use 'vxworks', as the corefile in

targetserver configuration and connect to target server.

NOTE:

- 1) If loading and launching vxWorks via the bootrom on the slave(ingress) before doing so on the master(egress) requires that you wait until the slave(ingress) has completed the loading of the vxWorks symbol file before loading and launching vxWorks on the master(egress).
- 2) Loading and launching vxWorks via the bootrom on the master(egress) before doing so on the slave(ingress) imposes no timing requirement on loading and launching vxWorks on the slave(ingress).

-
Connecting to Tornado over Ethernet using the vxWorks_rom_eth image via Boot Manager:

-

- 1) Edit the makefile to NOT use the STANDALONE_BOOT addresses for ROM:
 - a) uncomment the following as shown in makefile:

```
# Use the following for Boot Manager launchable build
ROM_BASE_ADRS   = c4800000      # base address of EPROM
ROM_TEXT_ADRS   = c4808000      # code start addr in EPROM
ROM_SIZE        = 00800000      # 8MB size of EPROM
```
 - b) comment OUT the following as shown in makefile:

```
# Use the following for STANDALONE_BOOT vxWorks build
#ROM_BASE_ADRS  = c4000000      # base address of EPROM
#ROM_TEXT_ADRS  = c4008000      # code start addr in EPROM
#ROM_SIZE       = 01000000      # 16MB size of EPROM
```
- 2) Add the following line to config.h:

```
#undef STANDALONE_BOOT
#undef INCLUDE_NET_SYM_TBL
```
- 3) Change Master and Slave Default bootlines with the IP addresses (in config.h) for your environment.
- 4) Build the vxWorks_rom_eth.bin
- 5) Copy the vxWorks_rom_eth.bin from the TFTP host. At the Boot Manager Prompt, substituting the appropriate parameters for you environment, type:

```
"copy \\17.17.17.81\c:\ixdp2800_be\vxWorks_rom_eth.bin vxWorks_rom_eth.bm"
```
- 6) Execute a "dir" command at the boot manager prompt to confirm the file has been copied to the flash file system of the IXDP2800.
- 7) Load the file into memory. At the Boot Manager Prompt, type:

```
"load vxWorks_rom_eth.bm 100000"
```
- 8) Copy the vxWorks_rom_eth.bm image from memory to the free flash area. At the Boot Manager Prompt, type:

```
"copy2flash <sourceaddr> <destaddr> <length>": "copy2flash 100000
c4800000 <length>"
```
- 9) Launch execution of the vxWorks_rom_eth (slave(ingress) FIRST, then master(egress)). At the Boot Manager Prompt, type:

```
"launch c4808000"
```
- 10) Once vxworks is booted on both the NPUs, use 'vxworks_rom_eth', as the corefile in target server configuration and connect to target server.
- 11) Wait until you see "WDB READY" on the serial console
- 12) Connect the target server with the correct IP address and corefile and symbol file vxworks_rom_eth.
- 13) Launch the target server

14) Launch shell and load your application.

Connecting to Tornado over Ethernet using the vxWorks_rom_eth image in standalone mode:

1) Apply the procedure for Booting standalone BOOTROM (using 'BOOTROM.bin') above, and rebuild the vxWorks_rom_eth.bin image.

Connecting to Tornado over Serial using the vxWorks_rom_ser image via Boot Manager:

1) Edit the makefile to NOT use the STANDALONE_BOOT addresses for ROM:

a) uncomment the following as shown in makefile:

```
# Use the following for Boot Manager launchable build
ROM_BASE_ADRS = c4800000 # base address of EPROM
ROM_TEXT_ADRS = c4808000 # code start addr in EPROM
ROM_SIZE      = 00800000 # 8MB size of EPROM
```

b) comment OUT the following as shown in makefile:

```
# Use the following for STANDALONE_BOOT vxWorks build
#ROM_BASE_ADRS = c4000000 # base address of EPROM
#ROM_TEXT_ADRS = c4008000 # code start addr in EPROM
#ROM_SIZE      = 01000000 # 16MB size of EPROM
```

2) Add the following line to config.h:

```
#undef STANDALONE_BOOT
#define SERIAL_DEBUG
```

3) Change Master and Slave Default bootlines with the IP addresses (in config.h) for your environment.

4) Build the vxWorks_rom_ser.BIN

5) Copy the vxWorks_rom_ser.bin from the TFTP host. At the Boot Manager Prompt, substituting the appropriate parameters for you environment, type:

```
"copy \\17.17.17.81\c:\ixdp2800_be\vxWorks_rom_ser.bin vxWorks_rom_ser.bm"
```

6) Execute a "dir" command at the boot manager prompt to confirm the file has been copied to the flash file system of the IXDP2800.

7) Load the file into memory. At the Boot Manager Prompt, type:

```
"load vxWorks_rom_ser.bm 100000"
```

8) Copy the vxWorks_rom_ser.bm image from memory to the free flash area.

```
At the Boot Manager Prompt, type "copy2flash <sourceaddr>
<destaddr> <length>": "copy2flash 100000 c4800000 <length>"
```

9) Launch execution of the vxWorks_rom_ser (slave(ingress) FIRST, then master(egress)).

```
At the Boot Manager Prompt, type:
"launch c4808000"
```

10) Once vxworks is booted on both the NPUs, use 'vxworks_rom_ser', as the corefile in target server configuration and connect to target server.

Connecting to Tornado using the vxWorks_rom_ser image in standalone mode:

1) Apply the procedure for Booting standalone BOOTROM (using 'BOOTROM.bin') above, and rebuild the vxWorks_rom_ser.bin image.

```
*****
*-----*
*                DEVICE DRIVERS                *
*-----*
*****
```

I XD28192 vxWorks Device Driver:

Changes made in this release:

- 1. Updated Licensing information in the following files:

ixf18100_pos_b.h
 ixf18100_spi4_b.h
 movealobjs.bat

- 2. The following file was removed from this release:

envset.bat
 test_app\prjObjs.lst
 readme.angel

Known Issues:

There are no known issues.

#####

The following images are included in this release:

ixd28192.o (vxWorks downloadable image)

To compile vxWorks ixd28192 driver:

- 1) Install Tornado 2.2.1.
- 2) Run c:\WINBASE\host\x86-win32\bin\torvars.bat
- 3) Go to the ixd28192 directory on the PC.
- 4) Type: "make driver"

To begin using the vxWorks ixd28192 driver:

- 1) Download the vxWorks ixd28192 driver to the vxWorks operating system using the desired method (ie: Tornado target shell).
- 2) The vxWorks ixd28192 driver API is now ready to be called. Refer to the Intel I XDP2400/I XDP2800 Advanced Development Platform I/O Card Driver API Developer's Manual for the ixd28192 driver API details.

I XD2810 vxWorks Device Driver:

Changes made in this release:

Updated Licensing information in the two makefiles

Known Issues:

There are no known issues.

#####

The following images are included in this release:

ixd2810_tor22.out (vxWorks downloadable image)

To compile vxWorks ixd2810 driver:

- 1) Install Tornado 2.2.1.
- 2) Run c:\WINDOWS\host\x86-win32\bin\torvars.bat
- 3) Go to the ixd2810 directory on the PC.
- 4) Type: "make" (the Tornado project file is also supplied should you prefer to build in the Tornado gui).

To begin using the vxWorks ixd2810 driver:

- 1) Download the vxWorks ixd2810 driver to the vxWorks operating system using the desired method (ie: Tornado target shell).
- 2) The vxWorks ixd2810 driver API is now ready to be called. Refer to the Intel IXP2400/IXDP2800 Advanced Development Platform I/O Card Driver API Developer's Manual for the ixd2810 driver API details.

IXD28192 Linux Device Driver:

Changes made in this release:

1. Updated licensing information in the following files:

- ixf18100_pos_b.h
- ixf18100_spi4_b.h
- linuxkern.h
- linux_int.c
- movedrvobj.s.bat
- test18101.c

2. Removed the following files from the release:

- moveallobj.s.bat
- prjobjs.lst
- makefile
- envset.bat

3. Added the file makefile.linux

Known Issues:

There are no known issues.

#####

The following images are included in this release:

i xd28192.o (Linux loadable module)

To begin using the vxWorks ixd2810 driver:

For ixd28192 Linux kernel mode:

To build driver, in toplevel directory:

\$ make -f Makefile.linux CROSS_COMPILE=xscal_e_be- ixdp28192.o

To load driver:

copy to target filesystem,

\$ insmod ixdp28192.o

To build test driver, in toplevel directory:

\$ make -f Makefile.linux CROSS_COMPILE=xscal_e_be- ixf/test18101.o

To load:

copy to target filesystem,

\$ insmod test18101.o

Current features not tested:

IxfApi SetCfg().

IxfApi GetCfg().

IxfApi GetStatus().

IxfApi GetOpMode().

IxfApi SetOpMode().

IxfApi GetCounters().

IxfApi SetChipOnline().

IxfApi SetChipOffline().

IxfApi CfgTest().

IxfApi SetAlarmCfg().

IXD2810 Linux Device Driver:

Changes made in this release:

1. Updated licensing information in the following files:

/ixf1110/bld1110.h
/testApp/test2810.c

2. Removed the following files:

moveallobjs.bat
movedrvobjs.bat

Known Issues:

There are no known issues.

#####

The following images are included in this release:

ixd2810ixf.o (Linux downloadable module)
ixd2810IntMod.o (Linux downloadable module)
test2810.o (Linux downloadable module)

To compile Linux ixd2810 driver:

-
- 1) Install the driver source onto the Linux host system
 - 2) Install the MontaVista 3.0 tool chain onto the Linux host
 - 3) Go to the ixd2810 directory
 - 4) Type: "make all" at the prompt to make the three modules listed above

To begin using the Linux ixd2810 driver:

After copying the products of the above make procedure to the filesystem mounted by the target type the following at the target console prompt:

```
insmod ixd2810ixf.o  
insmod ixd2810IntMod.o
```

The driver is now ready to be called by an application. In addition, if it is desired to run the supplied test application, type the following:

```
insmod test2810.o
```

The test application results will be logged to the target console.