Intel® Internet Exchange Architecture Software Development Kit

Tools Installation Guide

March 2004

Order Number: C53504-003

Intel may make changes to specifications and product descriptions at any time, without notice.

Designers must not rely on the absence or characteristics of any features or instructions marked “reserved” or “undefined.” Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

The Intel® IXA SDK 3.51 may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an ordering number and are referenced in this document, or other Intel literature may be obtained by calling 1-800-548-4725 or by visiting Intel's website at http://www.intel.com.

Contents

Revision History

Date	Revision	Description
March 2004	003	Intel® IXA SDK 3.51
November 2003	002	Intel® IXA SDK 3.5 Pre-Release 2. New Linux* installation procedures.
September 2003	001	Intel® IXA SDK 3.5 Pre-Release 1.

Taking Stock 1

This section provides details about preparing to install the Intel® IXA SDK Tools CD and the optional Intel® IXDP Firmware and Drivers CD, including requirements and decisions to consider beforehand. It includes information examining your development host and, if required, the Intel® IXDP2400 Advanced Development Platform or Intel® IXDP2800 Advanced Development Platform (referred to here as the platform).

1.1 Reviewing Your Development Host

Review your development host before installing the package. The following sub-sections are a checklist of features to take into consideration. The development host is the PC on which you install the Intel® IXA SDK. It is used to develop software and may be connected to either an Intel® IXDP2400 Advanced Development Platform or Intel®IXDP2800 Advanced Development Platform.

1.1.1 PC Basics

Your development host should have the following items and capabilities, grouped here in general categories of Hardware, Operating System Software and File System, Other Software, and Networking.

1.1.1.1 Hardware

Check the hardware for these points:

- Installation of the Intel® IXA SDK Tools CD requires around 300 MB of disk space. A total disk space of 350 MB is recommended for building projects.
- Installation of the Intel® IXA SDK Firmware and Driver CD requires around 55 MB of disk space.

1.1.1.2 Operating System Software and File System

- The following operating systems support the Intel® IXA SDK:
  - Windows 2000* with service pack 4 or greater
  - Windows XP* with service pack 1 or greater
  - Red Hat* Linux 7.3
- For Windows 2000 or Windows XP, the default installation is to *<install drive>:\IXA_SDK_3.5*. The default <install drive>: is C:.
- For Red Hat Linux 7.3, you must install the Intel® IXA SDK in the *opt/ixa_sdk_3.5/* or directory.
- If you have a version 3.0 or 3.1 of the Intel® IXA SDK, you are not required to uninstall it to install the Intel® IXA SDK 3.51. However, if you have Intel® IXA SDK 3.5 installed on your system, you must install it before installing the Intel® IXA SDK 3.51.

1.1.1.3 Other Software

- For creating Foreign Models—Microsoft Visual Studio Version 6.0.
- A terminal emulation package—for example, HyperTerminal for Windows 2000* /Windows XP* or minicom for RedHat* Linux 7.3—is required for working with the platforms (Intel® IXDP2400 Advanced Development Platform or Intel® IXDP2800 Advanced Development Platform).
- WindRiver Tornado* 2.2.1 (Windows 2000* or Windows XP* development host)

  WindRiver Tornado is typically used for Intel XScale® core development. An evaluation copy of the software is shipped with the Intel® IXDP2400 Advanced Development Platform and the Intel® IXDP2800 Advanced Development Platform.

  — WindRiver Tornado includes an FTP server service that should be installed and started on the development host PC. If you plan to download images to the platform over Ethernet, you must install a FTP server on your development host PC.

  — This package is not necessarily required for debugging on microengine hardware.
- MontaVista* Linux Support Package
- VxWorks* Board Support Package

1.1.1.4 Networking

If you plan to download images to the platform over Ethernet, you should have a network hub that connects to your development PC and the platform.

1.2 Reviewing the Contents of the Package

The package or packages you are working with can include the following items:

- Intel® Internet Architecture Software Sofware Development Kit (Intel® IXA SDK)
- Intel® IXDP2400 Advanced Development Platform or Intel® IXDP2800 Advanced Development Platform

1.2.1 Software Development Kit

Read any readme files, readme.txt, and the *Intel® IXA SDK 3.51 Release Notes*, *ReleaseNotesForTools.pdf*, in the Documentation directory of the installed SDK CDs for the latest information on the package and any important notes about installation on your development host PC.

1.2.2 Advanced Development Platform

The Intel® IXA SDK is installed on a development host PC for use with the Developer Workbench the Intel® IXDP2400 Advanced Development Platform or the Intel® IXDP2800 Advanced Development Platform. Procedures detailed in this manual may vary slightly based on which development platform you are using.

The platform's basic package provides cables in addition to the chassis. There are two RJ11 Serial Cables with adaptors—RJ11 to DB9—to make the cables connect to the COM serial ports—for example, COM1 and COM2—of your development PC. These console connections are for working—for example, through HyperTerminal (Windows 2000*/Windows XP*) or minicom (Red Hat Linux*)—with the boot monitors of the ingress (slave) and egress (master) network processors as well as debug access. There is also an RJ45 crossover cable for the Ethernet master and slave management ports.

You must provide cables for making connections to the platform ingress (slave) and egress (master) over Ethernet. The Ethernet connections provide network-speed download of software from the development PC as well as debug access.

1.3 After Taking Stock

To proceed to the installation of the Intel® IXA SDK, refer to Chapter 2, "Windows 2000 or Windows XP Development Hosts" or Chapter 3, "Linux Development Host".

Windows 2000 or Windows XP Development Hosts 2

This chapter details the steps to install the Intel® Internet Exchange Architecture Software Development Kit (Intel® IXA SDK) Tools CD, WindRiver Tornado* software and the Intel® IXDP Firmware and Drivers CD on a Windows 2000 development host or Windows XP development host.

2.1 Contents Summary

This chapter includes the following sections:

2.2 Objectives

When you have finished the Windows 2000* or Windows XP* installation, you will have the Intel® IXA SDK ready for working with the following tools on the following items:

- Microengine Tools
  - Assembler
  - Compiler
  - Linker
  - Loader
  - Transactor
  - Developers Workbench
  - Debug libraries
- Dataplane libraries

2.3 Before You Begin

Perform the tasks in Chapter 1, “Taking Stock”.

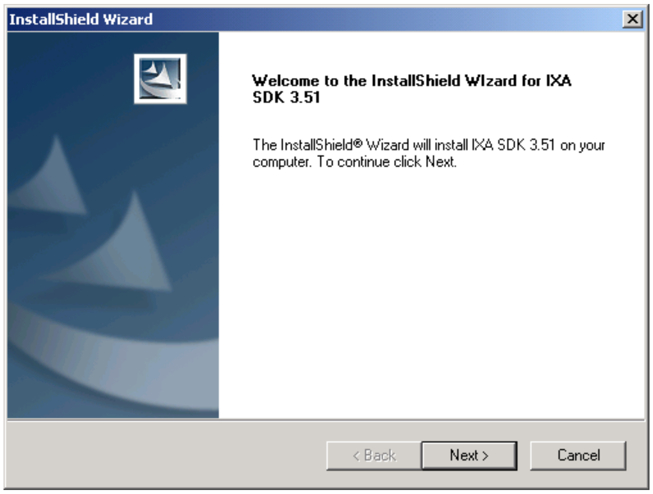
2.4 Procedure: Installing the Tools CD

To install the Intel® IXA SDK Tools CD on your development host, take the following steps:

1. Close all open applications on the desktop, taskbar, and system tray.

Note: If the portmapper is running, the installation won't continue. Please make sure to stop this service if the uninstall did not remove the service. To stop the service: Select *IXP2000 Portmapper service* from *MyComputer\manage\Services and Applications\Services*. Right click on "IXP2000 Portmapper" and select "Stop".

2. Insert the CD into the drive.
3. Choose **Run** from the Start menu.
4. Click **Browse**.
5. Use **Look in** to Browse your computer’s CD drive.
6. **Open** the file *setup.exe*.
7. Click **OK**.
   The InstallShield* program prepares and starts an Install Wizard. The initial screen of the Install Wizard appears. Click **Next>** to proceed with the installation.

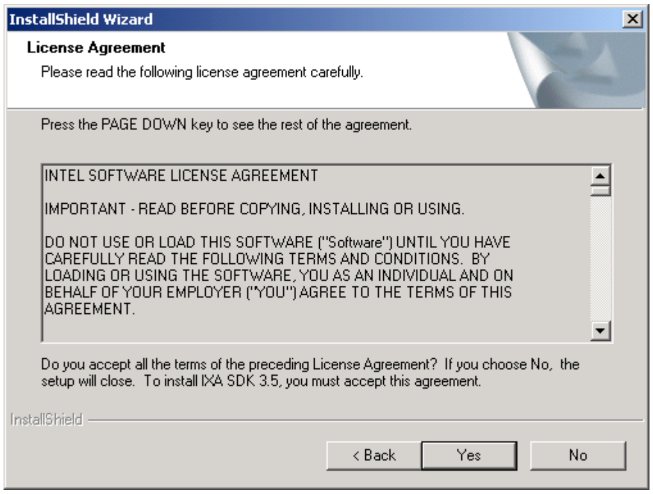


InstallShield Wizard

Welcome to the InstallShield WIzard for IXA SDK 3.51

The InstallShield® Wizard will install IXA SDK 3.51 on your computer. To continue click Next.

8. Read the licensing agreement that appears on the screen. If you accept the terms of the agreement, click **Yes** to continue:



InstallShield Wizard

License Agreement

Please read the following license agreement carefully.

Press the PAGE DOWN key to see the rest of the agreement.

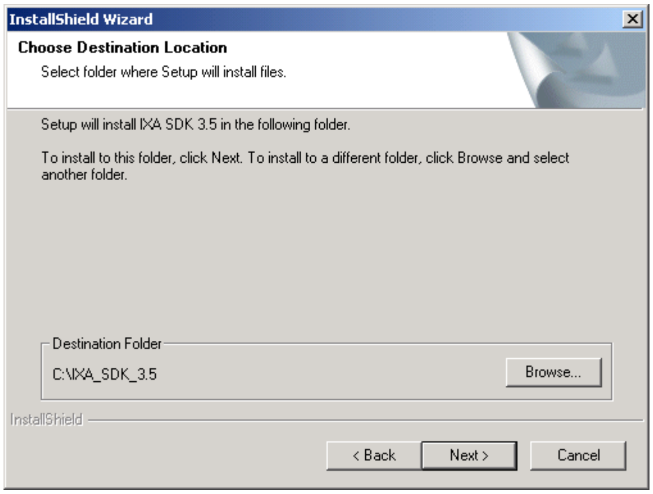
INTEL SOFTWARE LICENSE AGREEMENT

IMPORTANT - READ BEFORE COPYING, INSTALLING OR USING.

DO NOT USE OR LOAD THIS SOFTWARE ("Software") UNTIL YOU HAVE CAREFULLY READ THE FOLLOWING TERMS AND CONDITIONS. BY LOADING OR USING THE SOFTWARE, YOU AS AN INDIVIDUAL AND ON BEHALF OF YOUR EMPLOYER ("YOU") AGREE TO THE TERMS OF THIS AGREEMENT.

Do you accept all the terms of the preceding License Agreement? If you choose No, the setup will close. To install IXA SDK 3.5, you must accept this agreement.

9. The Install Wizard allows you to specify an install directory for the Intel® IXA SDK Tools. You may either change the directory or accept the default directory (recommended). Click **Next>** to continue:



InstallShield Wizard

Choose Destination Location

Select folder where Setup will install files.

Setup will install IXA SDK 3.5 in the following folder.

To install to this folder, click Next. To install to a different folder, click Browse and select another folder.

Destination Folder

C:\IXA_SDK_3.5

10. The Install Wizard begins copying the files to the directory specified in step 9.
11. When all files have been successfully copied, remove the CD and click **Finish**. After installing the Tools, you must restart the computer before you can first use them.

2.4.1 Changes That the Tools Installation Makes

This section describes the changes that the tools installation makes to your system: the files it adds and the registry entries it makes.

2.4.1.1 Files

Files are installed by default in an IXA_SDK_3.5 directory at the root of the C: drive.

2.4.1.2 Registry

The installation adds the following registry hierarchy:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Intel\IXA SDK 3.5
```

2.4.2 Checking the Installation

The default menu item on the Start menu's Programs menu is **IXA SDK 3.5**. In a complete installation, that menu item should contain the following items:

- Documentation folder
- DevWorkbench shortcut
- Uninstall IXA SDK Tools

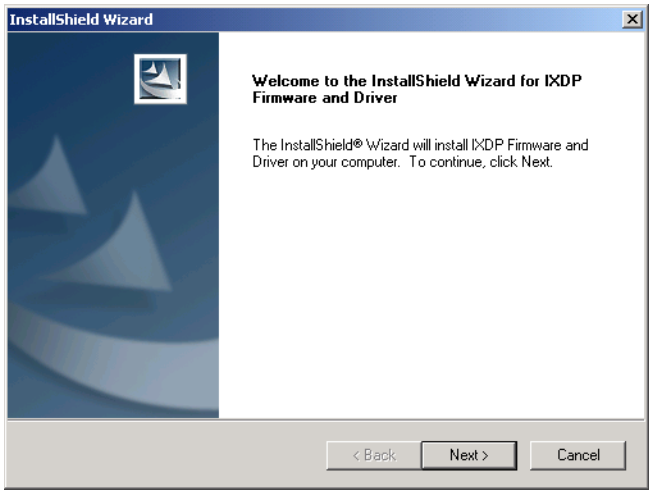
2.5 Procedure: Installing WindRiver Tornado

You can optionally install a copy of WindRiver* Tornado*. A version of Tornado* ships with the Intel® IXDP2400 Advanced Development Platform and the Intel® IXDP2800 Advanced Development Platform. For more information, refer to the WindRiver documentation on the CD or visit www.windriver.com.

2.6 Procedure: Installing the Firmware and Driver CD

To install the Intel® IXDP Firmware and Driver CD on your development PC, take the following steps:

1. Close all open applications on the desktop, taskbar, and system tray.
2. Insert the CD into the drive.
3. Choose **Run** from the Start menu.
4. Click **Browse**.
5. Use **Look in** to Browse your computer's CD drive.
6. **Open** the file *setup.exe*.
7. Click **OK**.
   The InstallShield* program prepares and starts an Install Wizard. The initial screen of the Install Wizard appears. Click **Next>** to proceed with the installation.

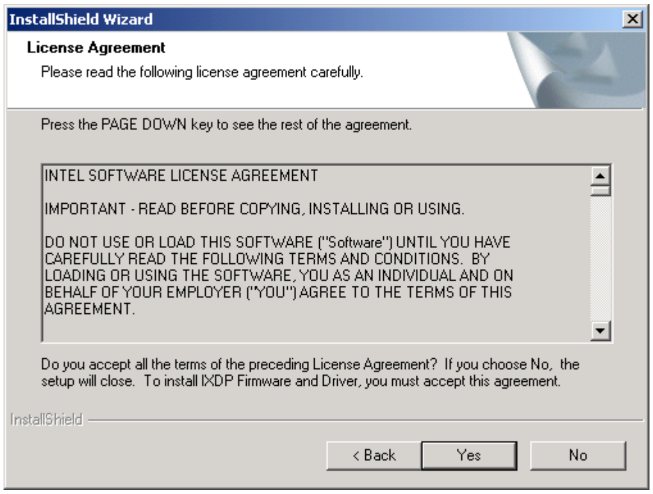


InstallShield Wizard

Welcome to the InstallShield Wizard for IXDP Firmware and Driver

The InstallShield® Wizard will install IXDP Firmware and Driver on your computer. To continue, click Next.

8. Read the licensing agreement that appears on the screen. If you accept the terms of the agreement, click **Yes** to continue.



InstallShield Wizard

License Agreement

Please read the following license agreement carefully.

Press the PAGE DOWN key to see the rest of the agreement.

INTEL SOFTWARE LICENSE AGREEMENT

IMPORTANT - READ BEFORE COPYING, INSTALLING OR USING.

DO NOT USE OR LOAD THIS SOFTWARE ("Software") UNTIL YOU HAVE CAREFULLY READ THE FOLLOWING TERMS AND CONDITIONS. BY LOADING OR USING THE SOFTWARE, YOU AS AN INDIVIDUAL AND ON BEHALF OF YOUR EMPLOYER ("YOU") AGREE TO THE TERMS OF THIS AGREEMENT.

Do you accept all the terms of the preceding License Agreement? If you choose No, the setup will close. To install IXDP Firmware and Driver, you must accept this agreement.

9. The Install Wizard begins copying the files to the same directory as the Intel® IXA SDK Tools.
10. When all files have been successfully copied, remove the CD and click **Finish**. After installing the firmware and drivers, you must restart the computer before you can first use them.

2.6.1 Changes That the Firmware and Drivers Installation Makes

This section describes the changes that the firmware and drivers installation makes to your system.

2.6.1.1 Files

Firmware and driver files are installed in the same directory as the Intel® IXA SDK Tools files. The default directory is IXA_SDK_3.5 at the root of the C: drive.

2.6.2 Checking the Installation

The default menu item on the Start menu's Programs menu is **IXA SDK 3.5**. In a complete firmware and driver installation, that menu item should now contain an **Uninstall Firmware and Driver** option in addition to the options listed in Section 2.4.2.

2.7 Uninstall Procedures

To uninstall the Intel® IXA SDK Tools and the Intel® IXDP Firmware and Drivers, uninstall the components in the following order:

1. Intel® IXDP Firmware and Drivers
2. WindRiver* Tornado* (if applicable). Refer to the WindRiver Tornado documentation for information about uninstalling the software.
3. Intel® IXA SDK Tools

2.7.1 Procedure: Uninstalling the Firmware and Drivers

Take the following steps:

1. Close the Developer's Workbench, and all open applications on the desktop, taskbar, and system tray.
2. Choose **Programs** > **IXA SDK 3.5** > **Uninstall Firmware and Driver** from the Start menu. This launches the Uninstaller.
3. When the uninstaller has removed all Firmware and Driver files, click **Finish**.
4. To free resources and avoid conflicts, restart your computer.

2.7.2 Procedure: Uninstalling the Tools

Take the following steps:

1. Close the Developer's Workbench, close all open applications on the desktop, taskbar, and system tray.
2. Choose **Programs** > **IXA SDK 3.5** > **Uninstall IXA SDK Tools** from the Start menu. This launches the Uninstaller.
3. Click **Finish** when the uninstaller is complete.
4. To free resources and avoid conflicts, restart your computer.

Linux Development Host 3

This chapter details the steps to install the Intel® Internet Exchange Architecture Software Development Kit (Intel® IXA SDK) the Tools CD and the Intel® IXDP Firmware and Drivers CD on a RedHat* Linux 7.3 development host.

3.1 Contents Summary

This chapter includes the following sections:

3.2 Objectives

When you have finished the Red Hat Linux installation, you will have the Intel® IXA SDK Tools ready for working with the following items:

- Microengine Tools
- Dataplane libraries

3.3 Before You Begin

Perform the tasks in Chapter 1, “Taking Stock”.

3.4 Installation Overview

Before using/building the Linux kernel for the development platform, you need to first install the MontaVista* Linux (MVL) distribution so you have a cross-compiler environment and a target NFS file system. The MVL environment contains all the tools and target binaries that you need for your development.

This guide also shows you how to cross compile a simple Hello World C program and execute the program from an Intel® IXDP2400 Advanced Development Platform or Intel® IXDP2800 Advanced Development Platform.

The following installs have to be done in order to set up the Intel® IXA SDK in a Red Hat Linux 7.3 environment:

1. Install the Intel® IXA SDK Tools on a Red Hat Linux 7.3 development host
2. Install the following MontaVista* Linux components on a Red Hat Linux 7.3 development host.

Note: Instructions in this section are provided for your convenience. Please review the latest MontaVista* documentation for the instruction on installing the MontaVista* distribution.

— Host binaries for Red Hat* Linux 7.3 (host-mvl3.0.0.iso on the MontaVista* Distribution CD).

— Target binaries for Red Hat* Linux 7.3 (arm-xscale-be-mvl3.0.0.iso on the MontaVista* Distribution CD).

— ISO image for Linux kernel sources (ixdp2400_arm_xscale_be-sb030227.img). Compilation of this source generates zImage

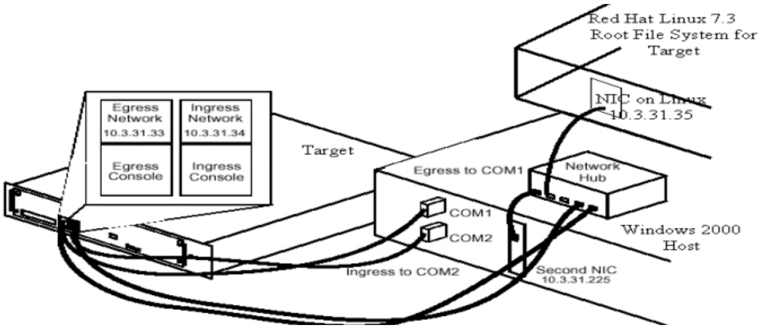
or

— Pre-compiled kernel image for Intel® IXDP2400 (vmlinuz-intel-ixdp2400) or Intel® IXDP2800 (vmlinux_intel_ixdp2800). This is the embedded linux kernel. It should have been burnt into the flash.

3. Install the Intel® IXA SDK Tools on a Windows 2000* or Windows XP* development host for Developer Workbench and transactor.

The following steps assume that boot loader is used to install Linux. Bootmonitor (Red Hat* boot loader) is employed.

Figure 1. Development Platform Configuration with a Linux Host.



3.5 Procedure: Installing the Tools CD

To install the Intel® IXA SDK Tools on a Red Hat* Linux 7.3 development host, perform the following steps:

1. Log in to the Linux system as root.
2. Change directory to */opt*. (Create an /opt directory if it does not exist.)
3. Insert the Intel® IXA SDK Tools 3.51 CD in the CD-ROM drive. Then, if your system is not set up to automount the CD-ROM drive when a CD is inserted, enter the following command to mount the CD to /mnt/cdrom:

   ```
   mount /dev/cdrom /mnt/cdrom
   ```

   For further information about mounting a directory in Linux, see the Red Hat* Linux* documentation.
4. Enter the following command to install the Intel® IXA SDK Tools in the directory specified in step 2 above:

   ```
   tar -xvzf /mnt/cdrom/ixa_sdk_3.51.tgz
   ```
5. The Intel® IXA SDK 3.51 tools should now be installed in */opt/ixa_sdk_3.5*

3.6 Procedure: Installing the Firmware and Driver CD

To install the Intel® IXDP Firmware and Driver CD on a Red Hat Linux 7.3* development host, perform the following steps:

1. Log in to the Linux system as root.
2. Change directory to where you installed the Intel® IXA SDK Tools.
3. Insert the Intel® IXDP Firmware and Driver CD in the CD-ROM drive. Then, if your system is not set up to automount the CD-ROM drive when a CD is inserted, enter the following command to mount the CD to /mnt/cdrom:

   ```
   mount /dev/cdrom /mnt/cdrom
   ```

   For further information about mounting a directory in Linux, see the Red Hat* Linux documentation.
4. Enter the following command to install the Intel® IXDP Firmware and Driver in the directory specified in step 2 above:

   ```
   tar -xvzf /mnt/cdrom/ixa_sdk_firmware.tgz
   ```

3.7 Installing MontaVista* 3.0 Host Binaries and Target Binaries

Please review the current MontaVista* documentation for the latest instruction on installing the MontaVista* distribution.

1. Insert HOST Binaries CD (host-mvl3.0.0.iso) into CD ROM drive. Mount the CD-ROM drive then runs the following command to install the packages:

   ```
   bash# /mnt/cdrom/bin/hhl-host-install
   ```

   then type “?” for a list of all packages available for installation.

Choose ADI –BRH-ARM_XSCALE_BE

2. Once the Host binary CD is done installing the host binaries, you will be prompted for the “Target Binaries for xscale_be or MontaVista* Linux arm_xscale_be“ CD (arm-xscale-be-mvl3.0.0.iso).

   Insert the CD. And press enter

   Once this installation is done, you will have all the tools available under the

   /opt/hardhat/devkit/arm/xscale_be/bin directory.

3. Once these 2 CDs (host and target binaries) are completely installed, you will be prompted with a message:

```
============================================

MontaVista* Linux 3.0 Installation Complete

============================================
```

3.8 Install Kernel Sources from ISO images

Instructions in this section are provided for your convenience. It is strongly recommended that you review the current MontaVista* documentation for the latest instruction on installing the MontaVista* Distribution.

Precompiled image from MontaVista* is called vmlinuz-intel-ixdp2400 which is same as zImage. MontaVista* has renamed their binary from zImage to vmlinuz-intel-ixdp2400. Both names can be used interchangeably in this document.

Once both the MontaVista* tools are installed, you install the ISO image.

1. Copy the ISO image to the /root

   For the IXDP2400:

   ```
   cp /mnt/cdrom/2400_arm_xscale_be-sb030227.img  /root
   ```

   For the IXDP2800:

   ```
   cp /mnt/cdrom/2800_arm_xscale_be-sb030513.img  /root
   ```

2. #mount /mnt/cdrom

   For the IXDP2400:

   ```
   #mount -t iso9660 -o loop ixdp2400_arm_xscale_be-sb030227.img
   /mnt/cdrom
   ```

   For the IXDP2800:

   ```
   #mount -t iso9660 -o loop ixdp2800_arm_xscale_be-sb030513.img
   /mnt/cdrom
   ```

3. Enter the command:

   ```
   #/mnt/cdrom/bin/hhl-host-install
   ```

The following will appear on the screen:

```
====================================================
MontaVista* Linux 3.0 Professional Edition Installer
=====================================================
```

type “?” to see a list of the available Linux Support Packages:

For the IXDP2400:
Choose “intel-ixdp2400-arm_xscale_be” and press enter.

For the IXDP2800:
Choose “intel-ixdp2800-arm_xscale_be” and press enter.

4. Once the package is completely installed, you will get this message

```
===================================================
MontaVista* Linux3.0 Installation complete
===================================================
```

The Sources are installed under:

For the IXDP2400:

```
/opt/hardhat/devkit/lsp/intel-ixdp2400-arm_xscale_be/linux-
2.4.18_mvl30
```

For the IXDP2400:

```
/opt/hardhat/devkit/lsp/intel-ixdp2800-arm_xscale_be/linux-
2.4.18_mvl30
```

5. Compiling the Linux kernel (zimage):

For the IXDP2400:

```
bash# cd /opt/hardhat/devkit/lsp/intel-ixdp2400-arm_xscale_be/
linux-2.4.18_mvl30
```

Export PATH for armtoolchain as:

Note: This will take a few minutes.

```
bash # export PATH=$PATH:/opt/hardhat/devkit/arm/xscale_be/bin
bash# make ixdp2400_config; make oldconfig; make dep; make
zImage
```

Once the compilation is done, a compressed zImage should reside in:

```
/opt/hardhat/devkit/lsp/intel-ixdp2400-arm_xscale_be/linux-
2.4.18_mvl30/arch/arm/boot/zImage
```

For the IXDP2800:

```
bash# cd /opt/hardhat/devkit/lsp/intel-ixdp2800-arm_xscale_be/
linux-2.4.18_mvl30
```

Export PATH for armtoolchain as:

Note: This will take a few minutes.

```
bash # export PATH=$PATH:/opt/hardhat/devkit/arm/xscale_be/bin
```

```
bash# make ixdp2800_config; make oldconfig; make dep; make
zImage
```

Once the compilation is done, a compressed zImage should reside in:

```
/opt/hardhat/devkit/lsp/intel-ixdp2800-arm_xscale_be/linux-
2.4.18_mvl30/arch/arm/boot/zImage
```

3.9 Setting Linux Host Network Services

Please refer to the Red Hat* documentation for additional information.

3.9.1 Establish A Red Hat Linux Remote Terminal

To communicate with the network processor, you must establish a remote terminal on your development workstation.

From a Linux shell, you can use the Linux minicom tool to connect over the serial port to the network processor.

The minicom tool requires a serial connection between your development platform and the network processor.

For example, to use minicom from a Linux development host:

1. Log in as root in a Linux command shell on the development host.
2. Delete the modem device:
   ```
   $ rm -f /dev/modem
   ```
3. Create a soft link between the modem device and the serial port terminal (ttyS0 for com1, or ttyS1 for com2):
   ```
   $ ln -s /dev/ttyS0 /dev/modem
   ```
4. Create a directory for the minicom scripts:
   ```
   $ mkdir /minicom
   ```
5. Run minicom:
   ```
   $ cd /minicom
   $ minicom
   ```
   If the serial port is locked, an error message appears and the minicom shell does
   not start. In this case, go to /var/lock and remove the lock file for the port.
6. Type CTRL A O to get a minicom configuration menu.
7. Choose serial port setup and set up for 57600 8N1 and no flow control (neither hardware nor software).
8. In the minicom menu, choose save setup as and save it as ixp2400.
9. Choose exit.
10. Exit minicom with CTRL A X.
11. To select the previously-saved configuration, start minicom by typing:
    ```
    $ minicom ixp2400
    ```

3.9.2 Setting Up DHCP and NFS Root File System

This section assumes that you have installed the MontaVista* Distribution, which provides Target Root Filesystem under */opt/hardhat/devkit/arm/xscale_be/target*.

In order to use DHCP and the NFS Root Filesystem, you need to determine the following:

1. MAC addresses of your target Ethernet interfaces. Issue the following command on your target Master console to determine the MAC addresses:

   ```
   Master-RedBoot> cfg read -n 1
   ```

2. You need to determine IP addresses for both Master and Slave Ethernet interfaces.
3. Make sure that Host PC and the Target interfaces are in same subnet.

3.9.2.1 Enabling Red Hat Linux DHCP service

You need to install the DHCP server package (dhcp-2.0pl5-8.i386.rpm for Red Hat 7.3) on your Linux host if not already installed. To see if the package is already installed, issue the following command:

```
hostpc# rpm -qa |grep dhcp
```

If you don't see the above-mentioned package, then (using Red Hat 7.3 CD# 2) reinstall it as:

```
hostpc# rpm -i /mnt/cdrom/redhat/RPMS/dhcp-2.0p15-8.i386.rpm
```

Then edit /etc/dhcpd.conf, add the following lines, and change the MAC addresses of your interfaces and the IP that need to be assigned for each interface. For example:

```
subnet 10.3.31.0 netmask 255.255.255.0 {

            host master-eth0.intel.com {

            hardware ethernet 00:02:B3:3C:14:0C;

            fixed-address 10.3.31.33;

            option root-path

            "/opt/hardhat/devkit/arm/xscale_be/target";

      }

            host slave-eth0.intel.com {

                  hardware ethernet 00:02:B3:3C:14:0D;

                  fixed-address 10.3.31.34;

                  option root-path

                  "/opt/hardhat/devkit/arm/xscale_be/target";

            }

      }
```

Start the dhcpd daemon by issuing the following command:

```
hostpc# /etc/rc.d/init.d/dhcpd start
```

3.9.2.2 Enabling Red Hat Linux TFTP service

Make sure the tftp server package (tftp-server-0.28-2.i386.rpm) is installed. Issue the following command to see if it is there:

```
hostpc# rpm -qa |grep tftp
```

If you don't see the above mentioned package then reinstall it as (use Red Hat* 7.3 CD# 1):

```
hostpc# rpm -i /mnt/cdrom/redhat/RPMS/tftp-server-0.28-
2.i386.rpm
```

create a /tftpboot directory by issuing the following command:

```
hostpc# mkdir /tftpboot
```

Copy the kernel image (zImage) to /tftpboot using the following command:

```
hostpc# cp /opt/hardhat/devkit/lsp/intel-ixdp2400-
arm_xscale_be/linux-2.4.18_mvl30/arch/arm/boot/zImage tftpboot
```

Make sure that the tftp protocol is enabled in the /etc/xinetd.d/tftp file, and change the line from disable = yes to disable = no

3.9.2.3 Enabling Red Hat Linux NFS Service

To export the root filesystem, edit /etc/exports and add this line:

```
/opt/hardhat/devkit/arm/xscale_be/target *(rw,no_root_squash,no_all_squash)
```

Verify the export is there by issuing the following command:

```
hostpc# exportfs
/opt/hardhat/devkit/arm/xscale_be/target   *
```

Restart DHCP as:

```
hostpc# /etc/rc.d/init.d/dhcp  restart
```

Restart NFS as:

```
hostpc# /etc/rc.d/init.d/nfs  restart
```

Restart the xinetd daemon by issuing the following command:

```
hostpc# /etc/rc.d/init.d/xinetd  restart
```

3.10 Uninstall Procedure

Use the Linux remove file (`rm`) command to uninstall the Intel® IXA SDK files. For example, if you installed the Intel® IXA SDK in the *\opt\ixa_sdk_3.5* directory, you would enter the following command:

```
rm -rf /opt/ixa_sdk_3.5
```

Note: The remove file command will delete all Intel® IXA SDK files (Intel® IXA SDK Tools and Intel® IXDP Firmware and Drivers) from the specified directory.

Loading Application Code 4

This chapter gives procedures for working with the Intel® IXDP2400 Advanced Development Platform and the Intel® IXDP2800 Advanced Development Platform. Some procedures assume a start from scratch, so many facilities you may already have on your PC are explained here, and there is some duplication.

4.1 Contents Summary

This chapter includes the following sections that deal with loading components from the development host on the platform's microengines as well as an application, depending on the capabilities of your development platform. You can download components as follows:

- From Developer Workbench to a VxWorks* or Monta Vista* Linux target via either of the following:
  - Serial Connection

    For VxWorks targets, you can load Developer Workbench components through a serial connection if you have WindRiver* Tornado*. Refer to Section 4.2, "Loading Workbench Components Through a Serial Connection" .

    For Monta Vista Linux Targets, you can use a Windows 2000 or Windows XP development host to download components over a serial connection. Refer to
  - Ethernet Connection

    You can load Developers Workbench components through an Ethernet connection with or without WindRiver* Tornado*.

    If you do not have WindRiver Tornado, you can use a procedure to establish the connection between the platform and Developer's Workbench. Refer to Section 4.3, "Loading Workbench Components Through an Ethernet Connection" .

    If you are using WindRiver Tornado, refer to Section 4.4, "Loading Workbench Components Through an Ethernet Connection Using Tornado" .

This chapter also includes procedural information for loading embedded Linux* images onto both your development host and your development platform. This allows you to and boot your development platform with embedded Linux. Refer to Section 4.5, "Loading and Booting Linux on the Intel® IXDP2400" .

4.2 Loading Workbench Components Through a Serial Connection

4.2.1 VxWorks Target and Windows Development Host

This procedure details the task of loading workbench components on target hardware through the console management ports of the platform.

4.2.1.1 Installing with Tornado

You must install Tornado 2.2.1 in order to load Developer Workbench components through a serial connection. Refer to Chapter 2, “Procedure: Installing WindRiver Tornado” for information about installing Tornado.

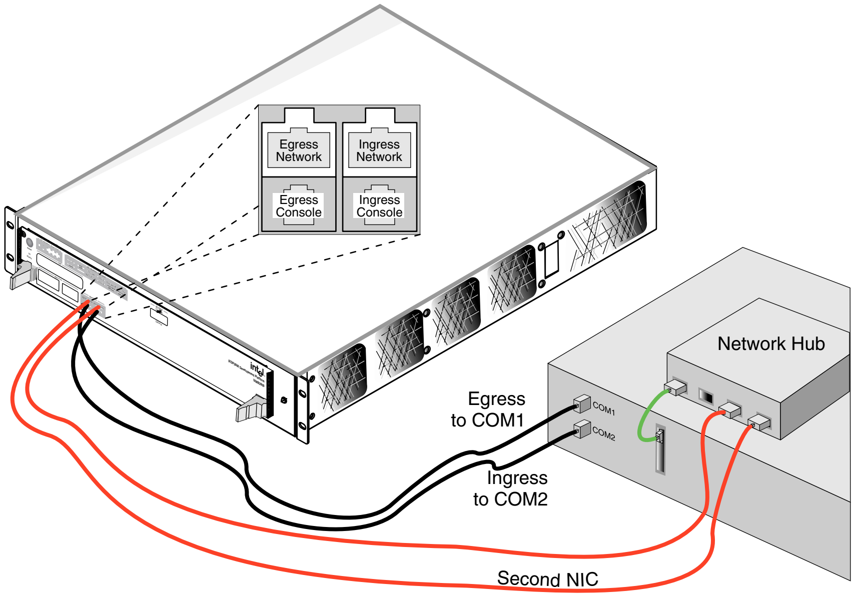
4.2.1.2 Procedure

Although this procedure shows initial setting up for both NPUs, the procedure, for illustration, uses the master-egress connection for loading Workbench components. You decide, starting with step 6, if you want to load one NPU or both NPUs. You can have two server sessions at once, provided they use separate ports.

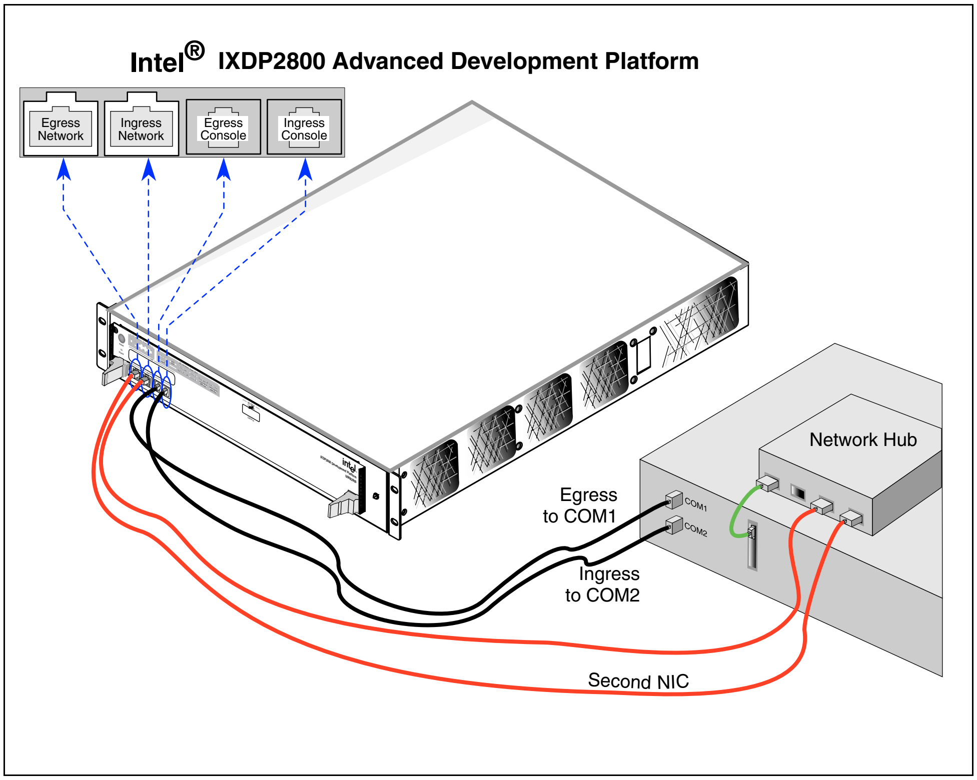
Note: The ports COM1 and COM2 are mentioned in this procedure for simplicity; you can use any COM port available on your machine.

1. Connect the Console management ports of the Intel® IXDP2400 Advanced Development Platform or Intel® IXDP2800 Advanced Development Platform for the Egress-Master NPU and the Ingress-Slave NPU to the development host serial ports COM1 and COM2 as needed.

   For illustration, the following figures show the Egress-Master Console connection going to COM1 and the Ingress-Slave Console connection going to COM2 for both the IXDP2400 and the IXDP2800. On the faceplate of the platform, the ports are marked Console in the diagram next to the management ports.



Intel® IXDP2400 Advanced Development Platform



2. Ensure that the Tornado 2.x system variable is set correctly. This is done as follows:

   a. Access the **Control Panel** from the Windows* Start menu.

   b. Click on the **System** icon.

   c. Select the **Advanced** tab.

   d. Click the **Environment Variables...** button.

   a. In the **System Variables** window, ensure that the path for Tornado 2.x points to the following:
   <tornado home directory>\host\x86-win32\bin
   The default *<tornado home directory>* is *C:\Tornado2.x*.

3. Start HyperTerminal twice to create connections—for example, COM1EgressMaster and COM2IngressSlave

   For the IXDP2400 the required settings for both the Ingress and Egress (shown in the figures below) are:

   — No flow control

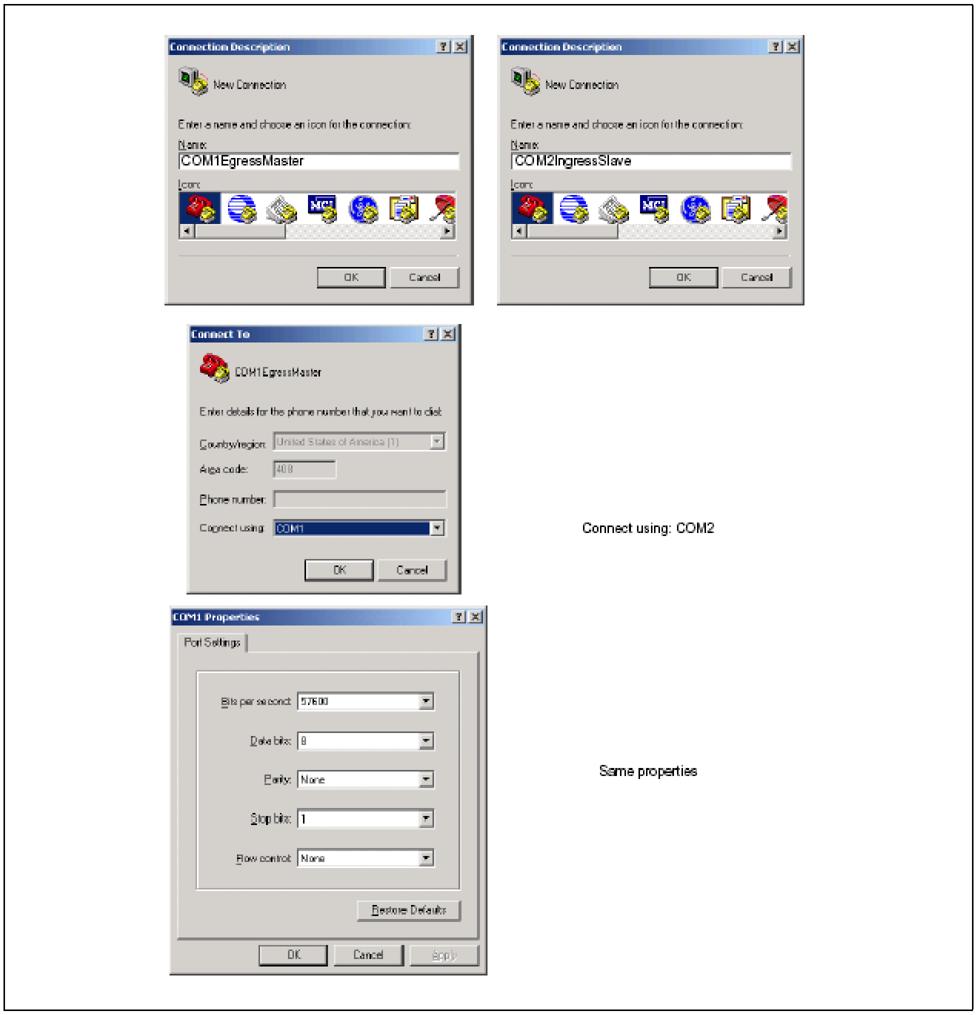
   — No parity bits

   — 8 data bits

   — 1 stop bit

   — baud rate of 57600

   For the IXDP2800 the required settings are the same, but the baud rate is 9600.



Connection Description

New Connection

Enter a name and choose an icon for the connection:

Name: COM1EgressMaster

Connection Description

New Connection

Enter a name and choose an icon for the connection:

Name: COM2IngressSlave

Connect To

COM1EgressMaster

Enter details for the phone number that you want to dial:

Country/region: United States of America (1)

Area code: 408

Phone number:

Connect using: COM1

Connect using: COM2

COM1 Properties

Port Settings

Bits per second: 57600

Data bits: 8

Parity: None

Stop bits: 1

Flow control: None

Same properties

If needed, you can save these HyperTerminal connections, and they will be available in the Start menu path **Programs > Accessories > Communications > HyperTerminal**. HyperTerminal displays a window for each connection.

4. Press I on the rocker switch at the rear of the platform to power up the system, or if the platform is already on, press the reset button in the faceplate.

5. Wait for the power up and reset cycle to complete.

   When the NPUs are ready, the HyperTerminal windows, Egress-Master and Ingress-Slave, display the following prompts,

   For the IXDP2400

   ```
   Master-Redboot>
   ```

   ```
   Slave-Redboot>
   ```

   For the IXDP2800

   ```
   Ingress>
   ```

   ```
   Egress>
   ```

6. For illustration, the rest of this procedure uses the Egress connection for loading and booting VxWorks. At the HyperTerminal window for the Egress connection, load and launch VxWorks with the following commands:

   For the IXDP2400

   ```
   Master-Redboot> fis load -w vxworks_serial
   Master-Redboot> go
   ```

   For the IXDP2800

   ```
   Egress> launch C4808000
   ```

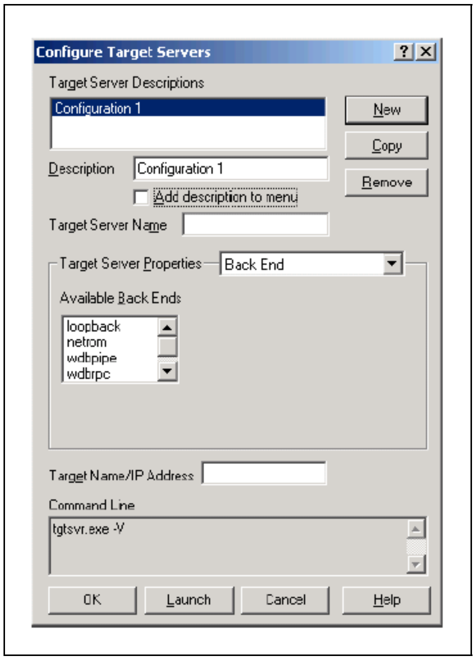
7. Wait for the message `WDB READY`.

8. Choose **Call** > **Disconnect**.

   HyperTerminal releases the COM port for that connection.

9. Launch Tornado—the default menu path is **Start** > **Programs** > **Tornado2.x** > **Tornado**—and click **Cancel** in Create Project in New/Existing Workspace if it appears.

10. Select **Tools** > **TargetServer** > **Configure**. Tornado displays a target server configuration window:



Target Server Descriptions: Configuration 1

Description: Configuration 1

□ Add description to menu

Target Server Name:

Target Server Properties: Back End

Available Back Ends: loopback, netrom, wdbpipe, wdbrpc

Target Name/IP Address:

Command Line:

```
tgtsvr.exe -V
```

11. Assuming **Target Server Descriptions** has no configurations in it, click New and the name Configuration1 should appear in this box.

    You can re-use target servers.

12. Type a name—for example, MasterEgressCOM1 or SlaveIngressCOM2—in **Target Server Name**. Record the name; you will need it for a later step.

13. Type a name—for example, Egress2400 or Ingress2400 (or Egress2800 and Ingress2800)—in **Target Name/IP Address**.

    In the case of a target server for a serial connection, this name serves as a placeholder only.

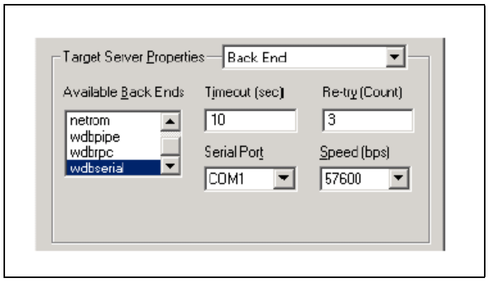
There are three groups of properties called Target Server Properties to set. You set each group by first selecting its name in **Target Server Properties**. The first group is Back End which is already displayed.

14. The following table shows the three target server property group names and, presented in order, the settings within those properties that you make. If a property setting is not shown in the table, leave it at its default setting

Target Server Property Group Name	IXDP2400 Property Settings	IXDP2800 Property Settings
Back End	**Available Back Ends**: wdbserial **Serial Port**: COM1 **Speed (bps)**: 57600 **Timeout (sec)**: 10 **Re-try (Count)**: 3	**Available Back Ends**: wdbserial **Serial Port**: COM1 **Speed (bps)**: 9600 **Timeout (sec)**: 10 **Re-try (Count)**: 3
Core File and Symbols	**File**: path\vxWorks_rom_ser[a] **All Symbols**: (chosen)	**File**: path\vxWorks_rom_ser[a] **All Symbols**: (chosen)
Console and Redirection	**Redirect Target IO**: (checked)	**Redirect Target IO**: (checked)

a. For example, `vxWorks_rom_ser` provided on the Firmware and Driver CD.

The following figures show the settings for the three target server properties listed in the table for the IXDP2400 Property Settings:



Target Server Properties: Back End

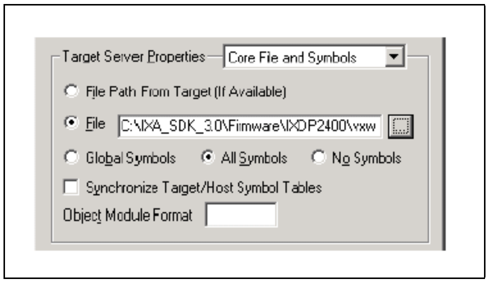
Available Back Ends: netrom, wdbpipe, wdbrpc, wdbserial (selected)

Timeout (sec): 10

Re-try (Count): 3

Serial Port: COM1

Speed (bps): 57600



Target Server Properties: Core File and Symbols

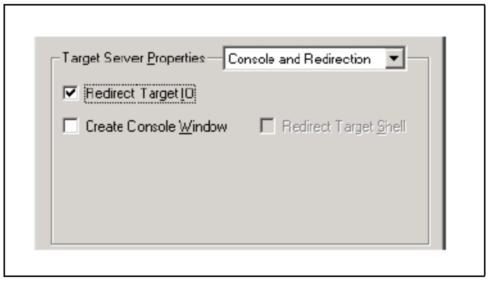
○ File Path From Target (If Available)

● File: C:\IXA_SDK_3.0\Firmware\IXDP2400\vxw

○ Global Symbols ● All Symbols ○ No Symbols

□ Synchronize Target/Host Symbol Tables

Object Module Format:



Target Server Properties: Console and Redirection

☑ Redirect Target IO

□ Create Console Window □ Redirect Target Shell

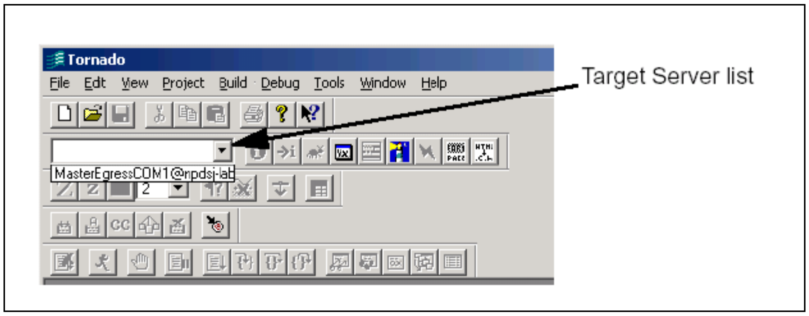
15. If you are unsure of the status of previous target server sessions, bring up Task Manager (press Ctrl+Alt+Delete and click **Task Manager** in Windows Security) to make sure there is no other target server running by looking in **Processes** for `tgtsvr.exe`. If it is in that list, then click `tgtsvr.exe` and click **End Process**. Exit Task Manager when the process finishes. If you are familiar with Tornado* Registry, you can manage target servers with that tool.

16. Click **Launch** in the Tornado window **Configure Target Servers**.

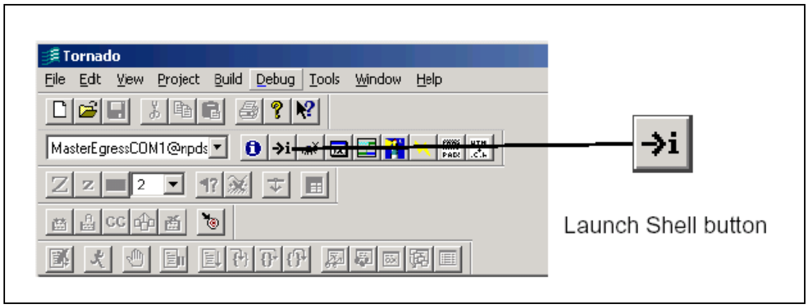
    Tornado creates in the system tray a Log Console for the target server. When you right-click the target icon and choose Show, the log console opens to display three success log messages followed by two warning log messages. Ignore the two warning messages. The target also appears in the Tornado Registry which is also in the system tray.

    In place of steps 16 and 17, you can also choose **Tools > Shell**, select the target server, and click **OK**.

17. Click the name of the target server—for example, **MasterEgressCOM1@*machinename***—in the target server list of the Launch toolbar as shown in the following figure:

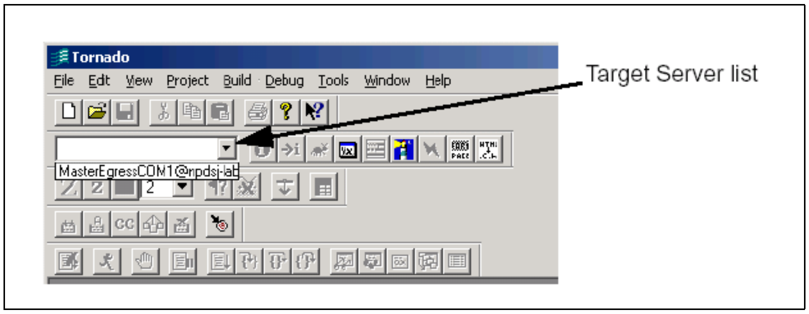

Tornado
File Edit View Project Build Debug Tools Window Help
MasterEgressCOM1@npdsj-lab
2
Target Server list

18. Click the Launch Shell button on the Launch toolbar.

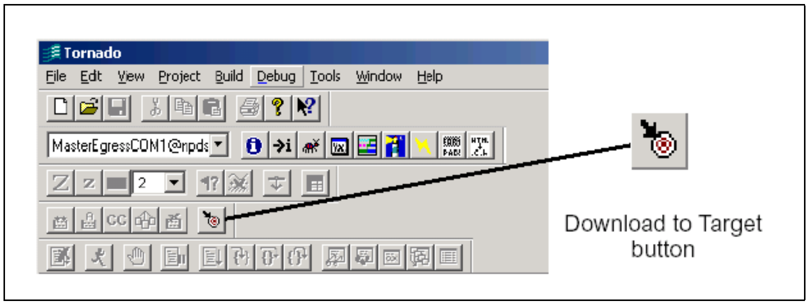


    Tornado displays a window for the shell—for example, **Shell MasterEgressCOM1@*machinename***.

19. Click the name of the target server—for example, **MasterEgressCOM1@*machinename***—in the target server list of the Launch toolbar as shown in the following figure:


Tornado
File Edit View Project Build Debug Tools Window Help
MasterEgressCOM1@npdsj-lab
Target Server list

20. Click the Download to Target button, on the Launch toolbar.


Tornado
File Edit View Project Build Debug Tools Window Help
MasterEgressCOM1@npds
Download to Target button

A browse dialog box, **Download objects**, appears.

21. Browse to the Workbench Backend—for example, `C:\IXA_SDK_3.5\me_tools\bin_vxw_be\WBSrvr.o`.

22. Click **Download**.

    `WBSrvr.o` downloads to the target for the target server.

Note: The `SHELL` commands `CD` and `LD` may be used to do the above, but the interface shown here is more simple.

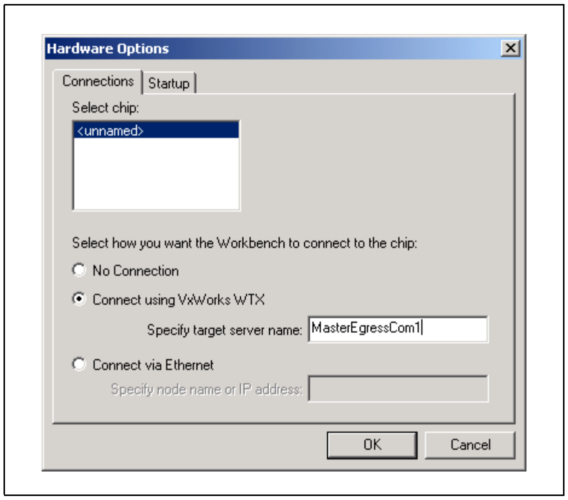
The Workbench backend is linked to the following libraries: `Rs_udebug.a`, `dbgme.a`, `halmev2.a`, `utils.a`, `osApi.a`, `rs_cntl.a`, and `uclo.a`.

23. Start Developer's Workbench (**Programs** > **IXA SDK 3.5** > **DevWorkbench**) and open a microcode project (**File** > **Open Project**).

    Choose **Debug** > **Hardware**.

24. Choose **Hardware** > **Options** > **Startup** and select **Reset the microengines and load the microcode.**

25. Click **Connections**.



Select chip: <unnamed>

Select how you want the Workbench to connect to the chip:

○ No Connection

● Connect using VxWorks WTX

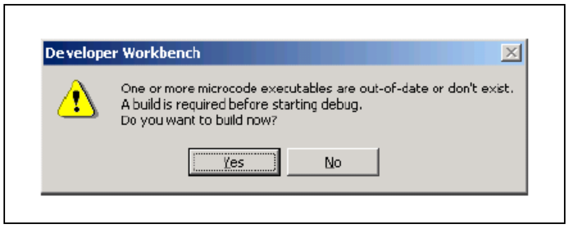
Specify target server name: MasterEgressCom1

○ Connect via Ethernet

Specify node name or IP address:

26. Click **Connect using VxWorks WTX**.
27. Type the target server name you typed in step 12.
28. Choose **Debug > Start Debugging**.

    Developer's Workbench shows you the following message in the following case:



One or more microcode executables are out-of-date or don't exist.
A build is required before starting debug.
Do you want to build now?

If you click **Yes**, Developer's Workbench downloads the microcode and enables Breakpoints and Watch functionality.

29. Set Breakpoints in your source, run the microcode, and monitor the data.

4.2.2 Monta Vista Linux Target and Windows Development Host

In standard Intel® IXDP2400 Advanced Development Platform and Intel® IXDP2800 Advanced Development Platform configurations, the Intel® IXP2400/Intel® IXP2800 network processor has access to an Ethernet port with which it can communicate to the outside world (for example to the Developer's Workbench).

However, in some other systems that use the Intel® IXP2400 or Intel® IXP2800 network processor, there may not be an Ethernet adapter available. If the NPU is running VxWorks*, then the Developer's Workbench can connect (over the serial line) via WTX. However this will not work if the NPU is running Monta Vista* Linux.

This section describes how to set up a TCP/IP link between a Windows 2000* or Windows XP* development host and the NPU running Monta Vista Linux over a serial line. The link is established using PPP (point-to-point protocol).

Once this link is established, the Developer’s Workbench can connect using the appropriate IP address, as if the two machines were connected via Ethernet.

All of the necessary configuration occurs at the operating system level; no configuration is needed at the Intel® IXA SDK level.

This process can be divided into the following steps:

- creating a "serial port only" version of Linux
- configuring PPP on the Linux side
- configuring the network connection on the Windows development host side
- starting the system

Note: Based on the version of Monta Vista* Linux that you are using, some of the system files may be located in different directories than as shown in this section.

4.2.2.1 Creating a "Serial Port Only" Version of Linux

You need to build an appropriate kernel and ram disk image that is subject to the configuration issues described in the Section 4.2.2.2, “Monta Vista Linux Target Configuration” . The details of doing this are beyond the scope of this document. Contact Monta Vista* or refer to Monta Vista documentation for details.

4.2.2.2 Monta Vista Linux Target Configuration

The Linux kernel must be configured with networking support and point-to-point protocol. An example of the point-to-point protocol section of the kernel *.config* file is shown below:

```
CONFIG_PPP=y                                Needed
# CONFIG_PPP_MULTILINK is not set           Not needed
CONFIG_PPP_ASYNC=y                          Needed
# CONFIG_PPP_SYNC_TTY is not set            Not needed
CONFIG_PPP_DEFLATE_=y                       Not needed but recommended
CONFIG_PPP_BSDCOMP=y                        Not needed but recommended
# CONFIG_PPOE is not set                    Not needed
```

A number of PPP configuration files need to be modified/created and included in the ram disk. These files are detailed in the following sections:

4.2.2.2.1 /etc/ppp/options

Ensure that hardware flow control is not enabled:

```
# Use hardware flow control (i.e. RTS/CTS) to control the flow of data
# on the serial port
# crtscts
```

The default values of the remaining settings in this file are acceptable.

4.2.2.2.2 /etc/ppp/options.ttyS0

This file should contain the following line:

```
connect 'chat -v -f /etc/ppp/winclient.chat'
```

This line is required because the Windows PPP connection requires a non-standard handshake. Note that the name of the chat file is arbitrary, but it must contain the text described in Section 4.2.2.2.3, “/etc/ppp/winclient.chat” .

4.2.2.2.3 /etc/ppp/winclient.chat

This is a new file that needs to be created. It should contain the following line:

```
CLIENT CLIENTSERVER\c
```

This instructs the chat program to wait until it sees "CLIENT" and then to respond with "CLIENTSERVER" (with no new line).

4.2.2.2.4 /dev/ppp

Ensure that the target file system has a `/dev/ppp` in it. If not, create it with a command similar to the following:

```
mknod /dev/ppp c 108 0
```

4.2.2.2.5 Other Files/Configuration Issues

The portmapper (`/sbin/portmap`) and inet daemons (`/usr/sbin/inetd`) should be included and started.

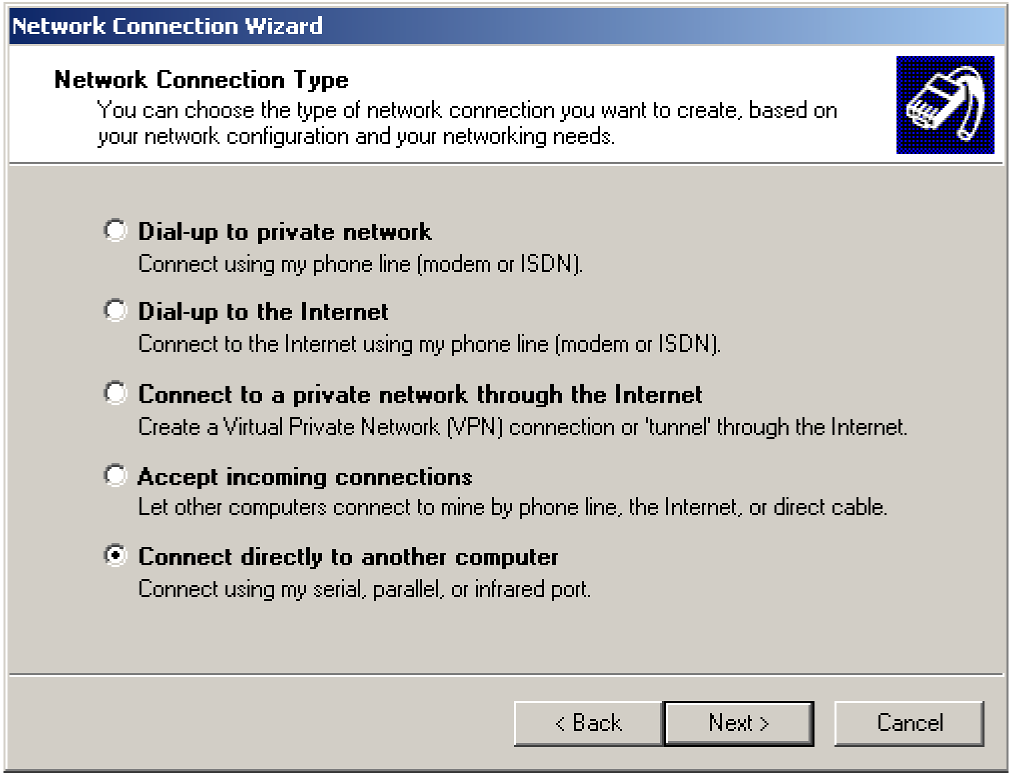
The PPP configuration files (`/etc/ppp/*`) should be included.

The PPP-related applications (`/usr/sbin/pppd` and `/usr/sbin/chat`) should be included.

For Developer’s Workbench use, the Intel® IXA SDK files (`halMeDrv.o` and `WBSrvr`) need to be included.

4.2.2.3 Windows 2000 Development Host Configuration

1. To create a connection under Windows 2000*, go to **Start>Settings>Network and Dial-up Connections>Make New Connection.**
2. This opens up the **Network Connection Wizard**. Click **Next** and specify the type as **Connect directly to another computer** as shown below:



Network Connection Wizard

Network Connection Type

You can choose the type of network connection you want to create, based on your network configuration and your networking needs.

○ **Dial-up to private network**
Connect using my phone line (modem or ISDN).

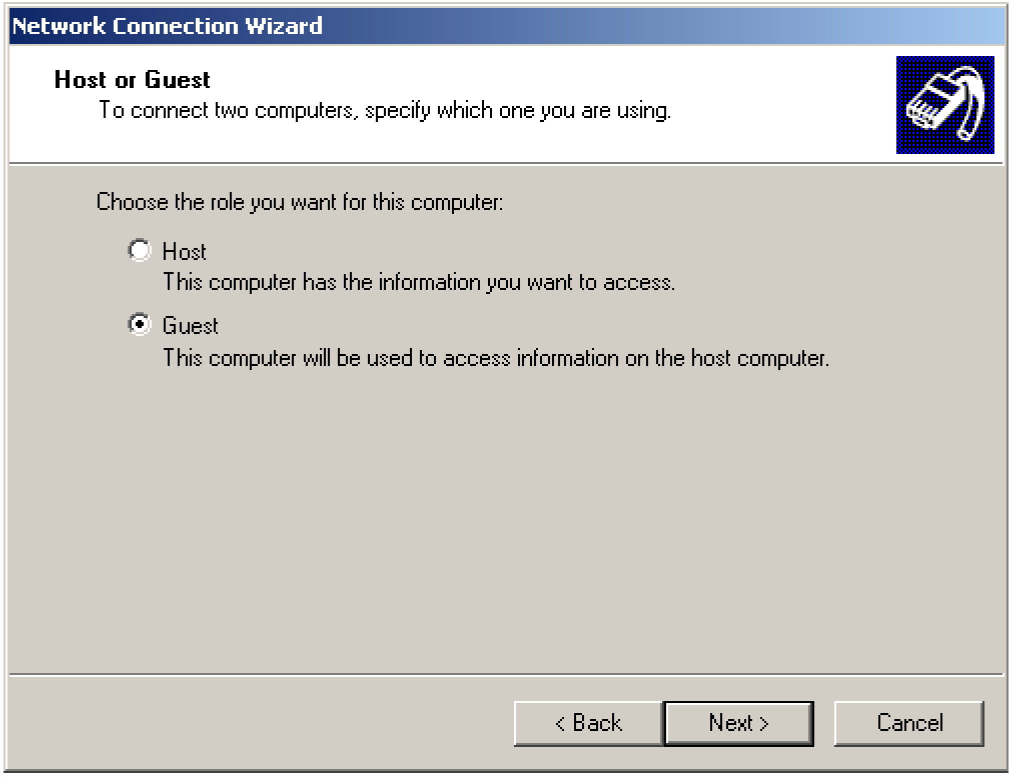
○ **Dial-up to the Internet**
Connect to the Internet using my phone line (modem or ISDN).

○ **Connect to a private network through the Internet**
Create a Virtual Private Network (VPN) connection or 'tunnel' through the Internet.

○ **Accept incoming connections**
Let other computers connect to mine by phone line, the Internet, or direct cable.

● **Connect directly to another computer**
Connect using my serial, parallel, or infrared port.

3. Click **Next** and select **Guest**:



Network Connection Wizard

Host or Guest

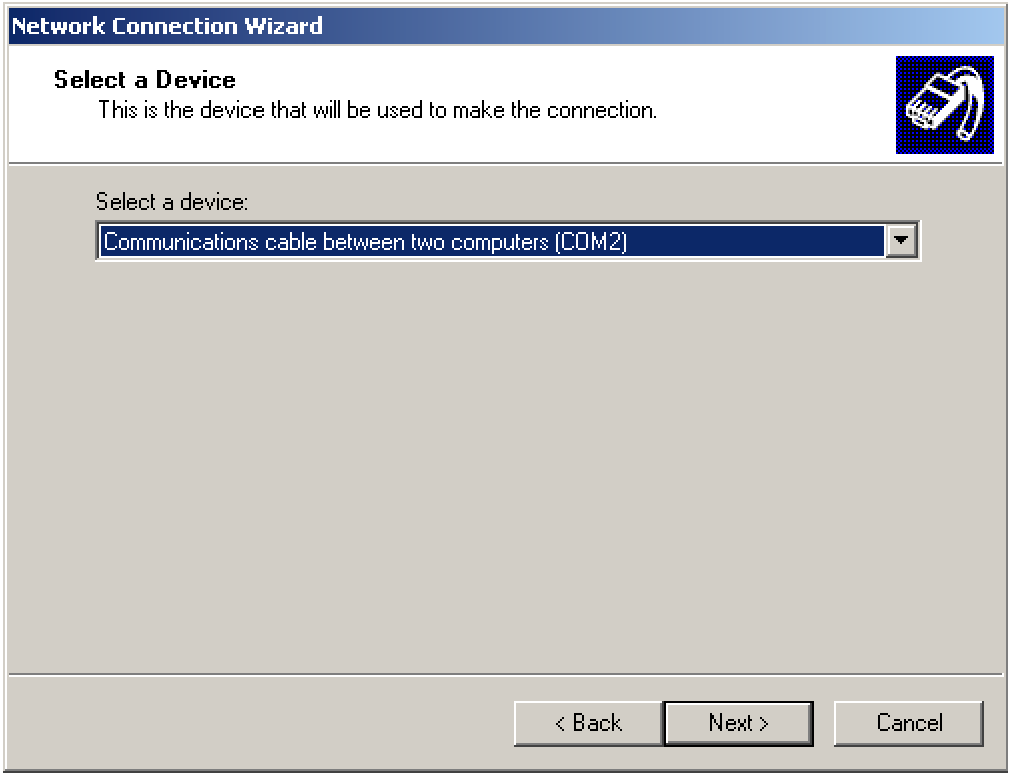
To connect two computers, specify which one you are using.

Choose the role you want for this computer:

○ Host
This computer has the information you want to access.

● Guest
This computer will be used to access information on the host computer.

4. Click **Next** and select the appropriate serial port (COM1 or COM2):



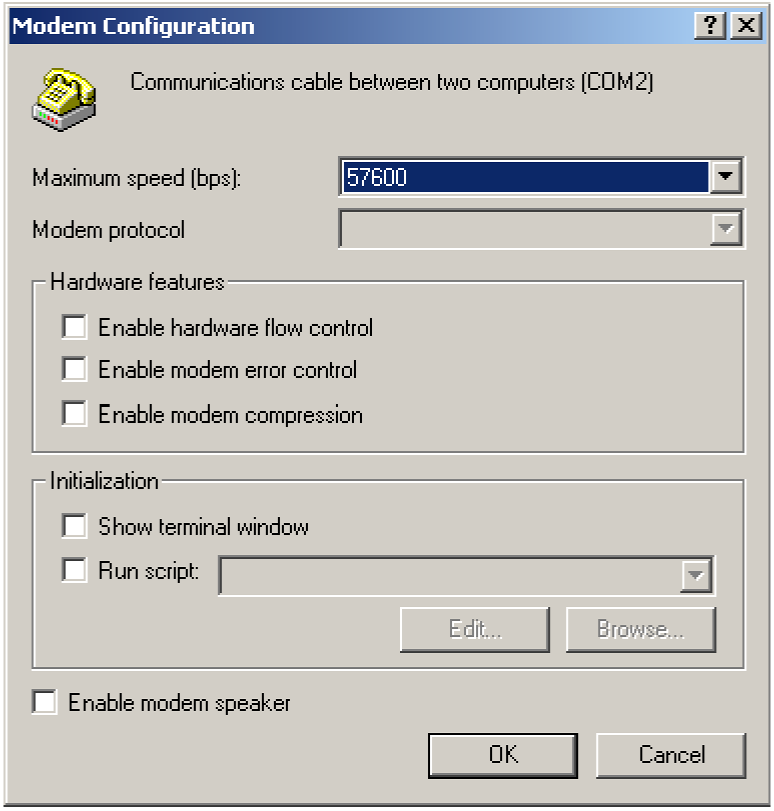
Network Connection Wizard

Select a Device

This is the device that will be used to make the connection.

Select a device: Communications cable between two computers (COM2)

5. Click **Next**. Select whether you want the connection to be for all users (recommended) or only for a single user. Then click **Next**, give it an identifiable name and click **Finish**.
6. Then go to **Start>Settings>Network and Dial-up Connections/Direct Connection 1** (replacing **Direct Connection 1** with whatever name you set in step 5).
7. Right click and select **Properties**. On the **General** tab, select **Configure…**



Modem Configuration

Communications cable between two computers (COM2)

Maximum speed (bps): 57600

Modem protocol:

Hardware features

☐ Enable hardware flow control

☐ Enable modem error control

☐ Enable modem compression

Initialization

☐ Show terminal window

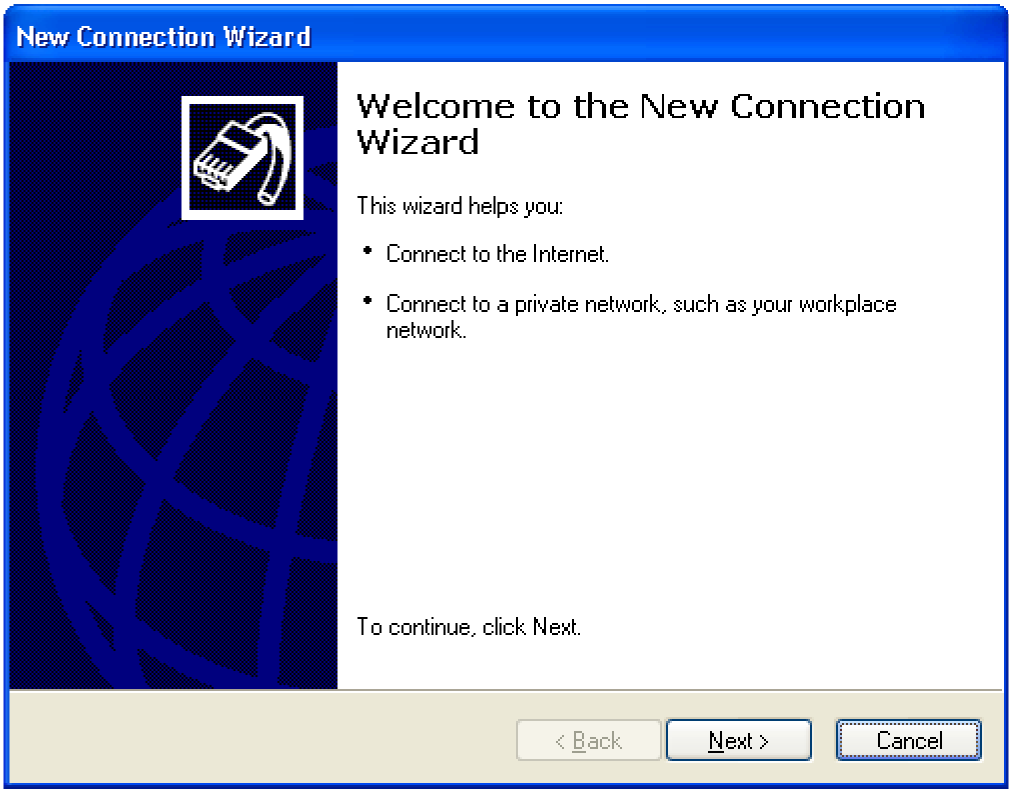
☐ Run script:

☐ Enable modem speaker

8. Select a baud rate of 57600 Ensure the **Enable hardware flow control box** is not checked.
9. Click **OK**.

4.2.2.4 Windows XP Development Host Configuration

1. To create a connection under Windows XP*, go to **Start>Settings>Network Connections>New Connection Wizard**. This starts the **New Connection Wizard**:


New Connection Wizard

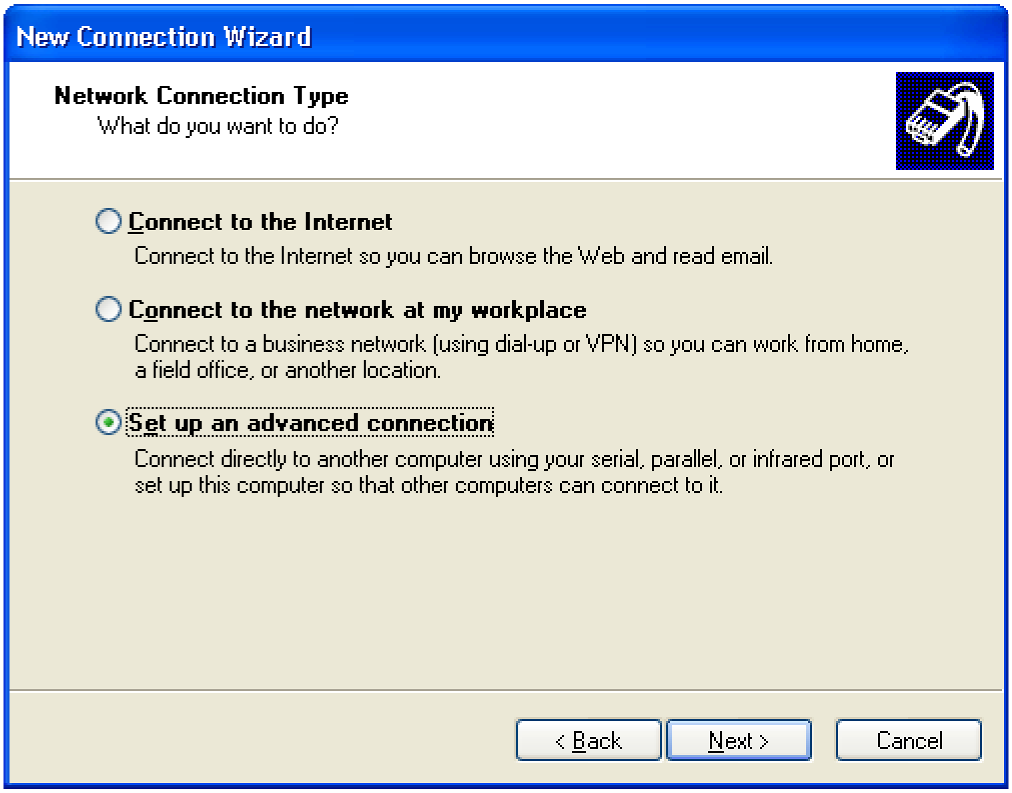
Welcome to the New Connection Wizard

This wizard helps you:

- Connect to the Internet.
- Connect to a private network, such as your workplace network.

To continue, click Next.

2. Click **Next** and specify an **advanced connection**:



New Connection Wizard

Network Connection Type

What do you want to do?

○ **Connect to the Internet**

Connect to the Internet so you can browse the Web and read email.

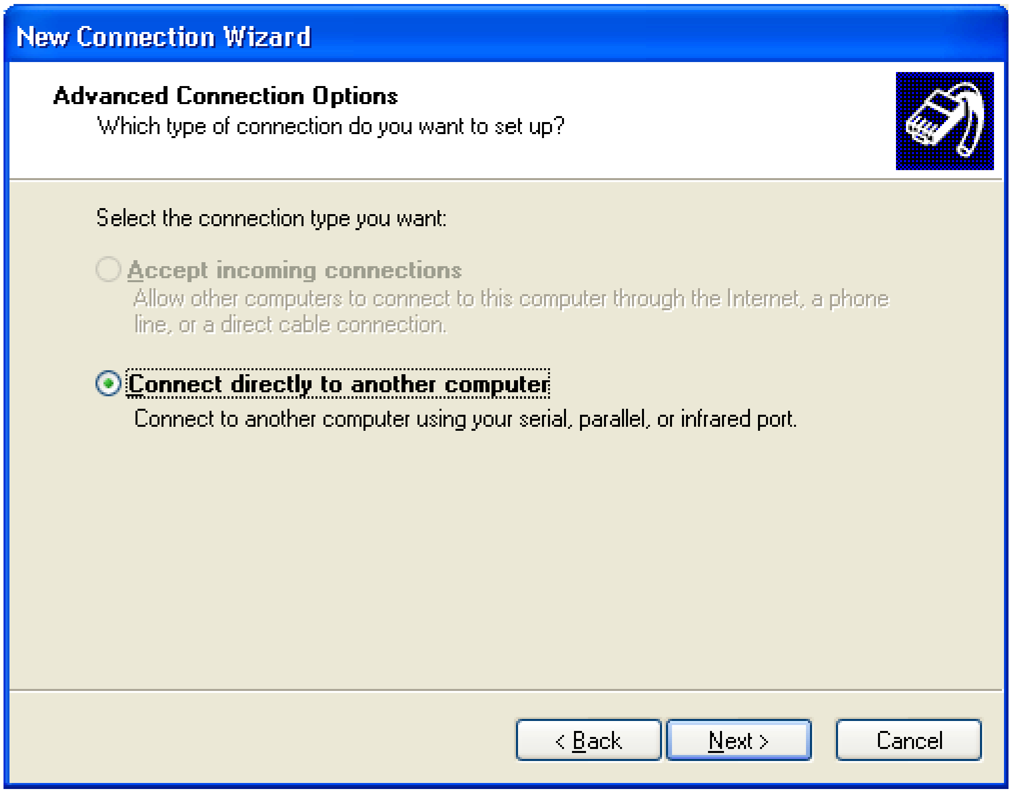
○ **Connect to the network at my workplace**

Connect to a business network (using dial-up or VPN) so you can work from home, a field office, or another location.

● **Set up an advanced connection**

Connect directly to another computer using your serial, parallel, or infrared port, or set up this computer so that other computers can connect to it.

3. Click **Next**. Select **Connect directly to another computer**:



Advanced Connection Options

Which type of connection do you want to set up?

Select the connection type you want:

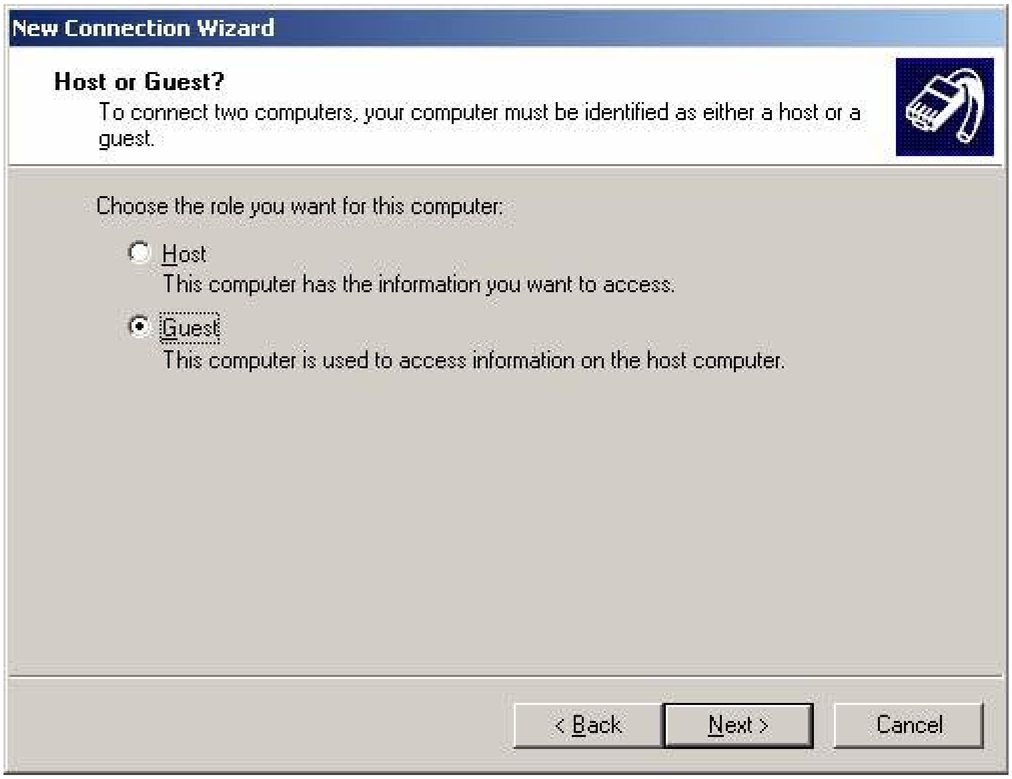
○ Accept incoming connections

Allow other computers to connect to this computer through the Internet, a phone line, or a direct cable connection.

● Connect directly to another computer

Connect to another computer using your serial, parallel, or infrared port.

4. Click **Next** and select a **Guest** connection:



New Connection Wizard

Host or Guest?

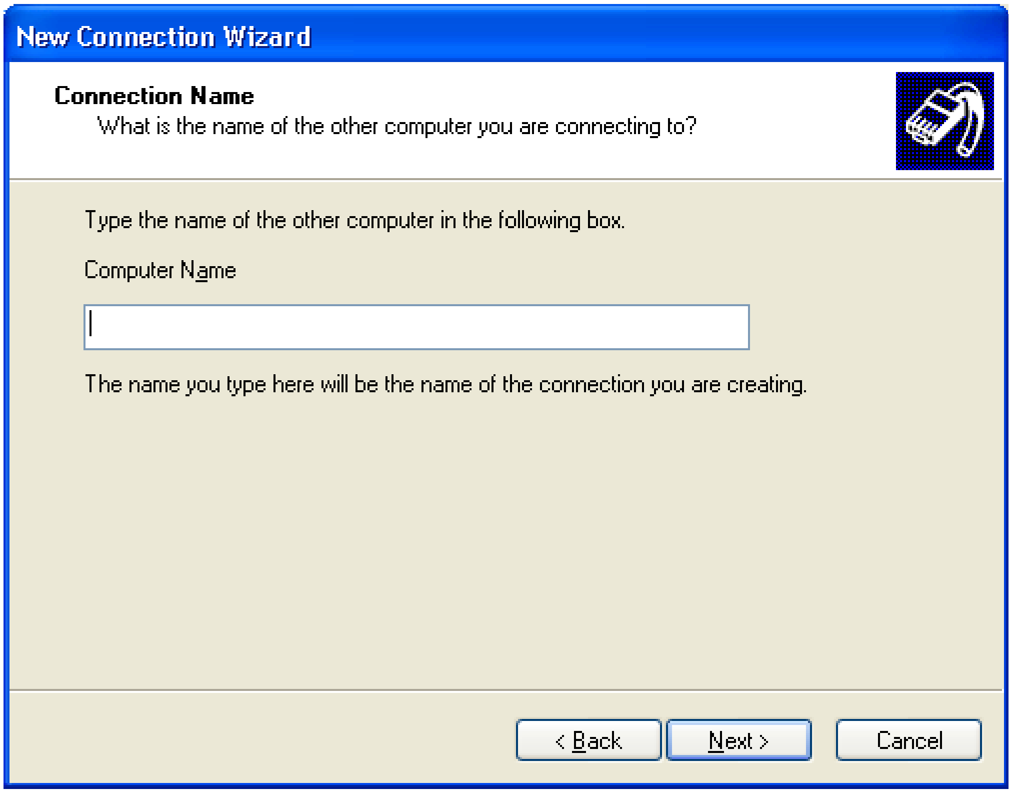
To connect two computers, your computer must be identified as either a host or a guest.

Choose the role you want for this computer:

○ Host
This computer has the information you want to access.

● Guest
This computer is used to access information on the host computer.

5. Click **Next** and enter an identifiable name for this connection:


New Connection Wizard

Connection Name

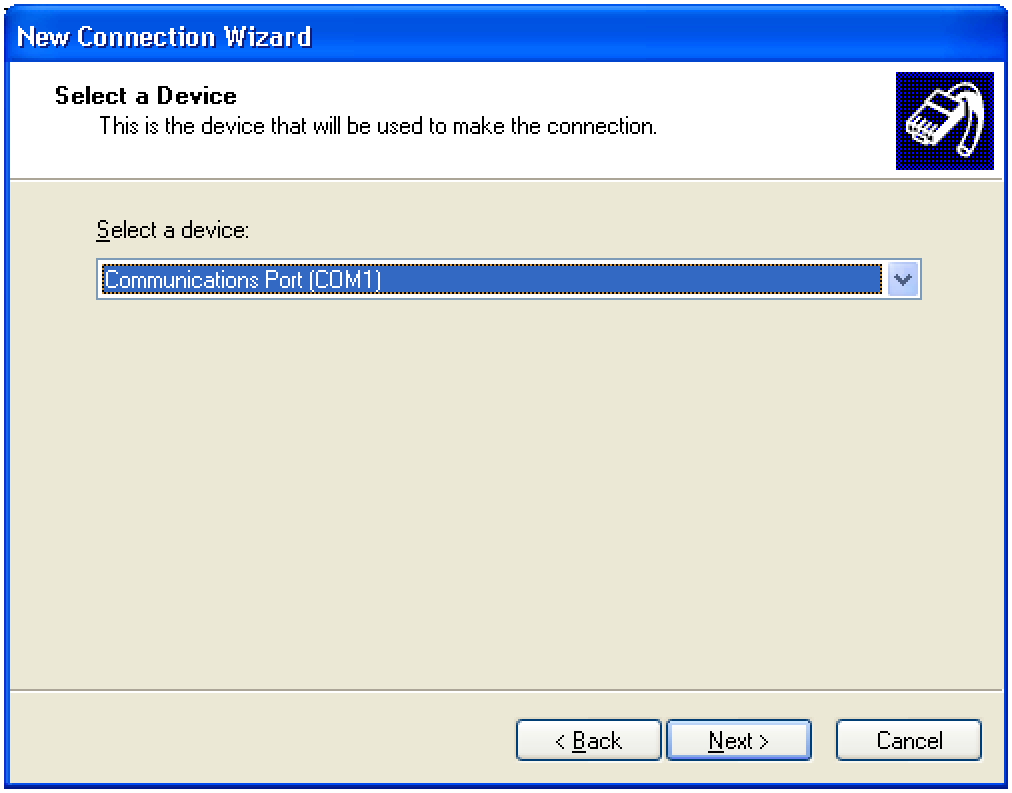
What is the name of the other computer you are connecting to?

Type the name of the other computer in the following box.

Computer Name

The name you type here will be the name of the connection you are creating.

6. Click **Next** and select the appropriate device (e.g. COM1 or COM2):



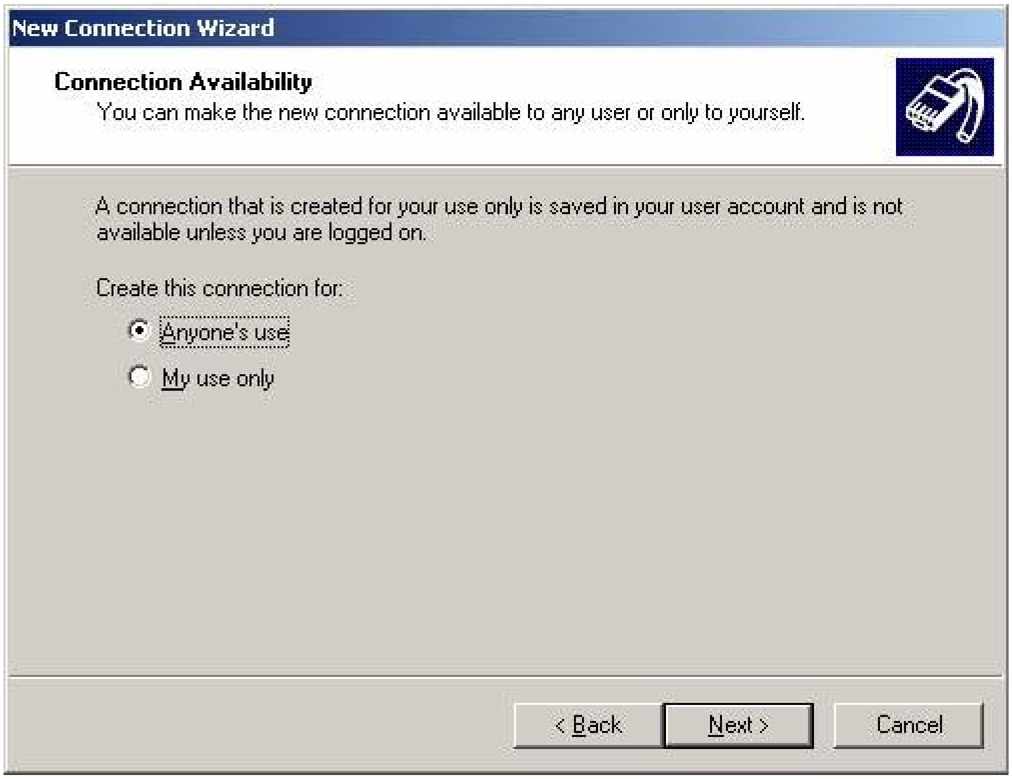
Select a Device

This is the device that will be used to make the connection.

Select a device:

Communications Port (COM1)

7. Click **Next** and specify whether the connection is for anyone's use (recommended) or only your own use:



New Connection Wizard

Connection Availability

You can make the new connection available to any user or only to yourself.

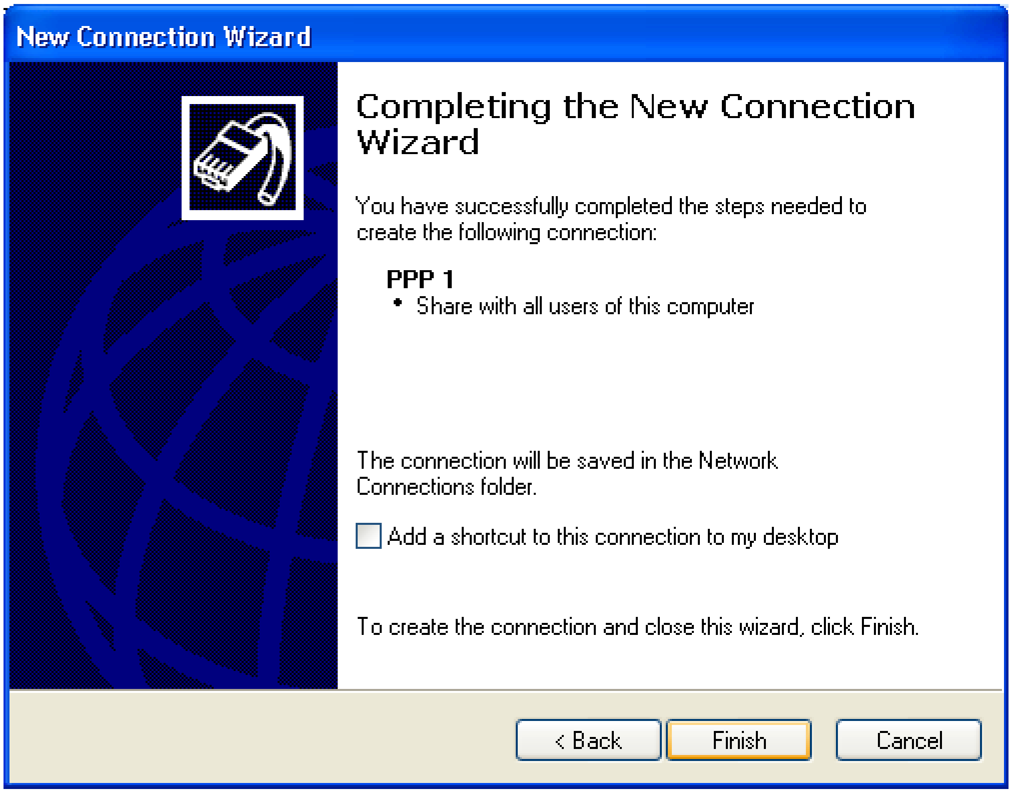
A connection that is created for your use only is saved in your user account and is not available unless you are logged on.

Create this connection for:

● Anyone's use

○ My use only

8. Click **Next**. Optionally add a shortcut to your desktop:



Completing the New Connection Wizard

You have successfully completed the steps needed to create the following connection:

PPP 1

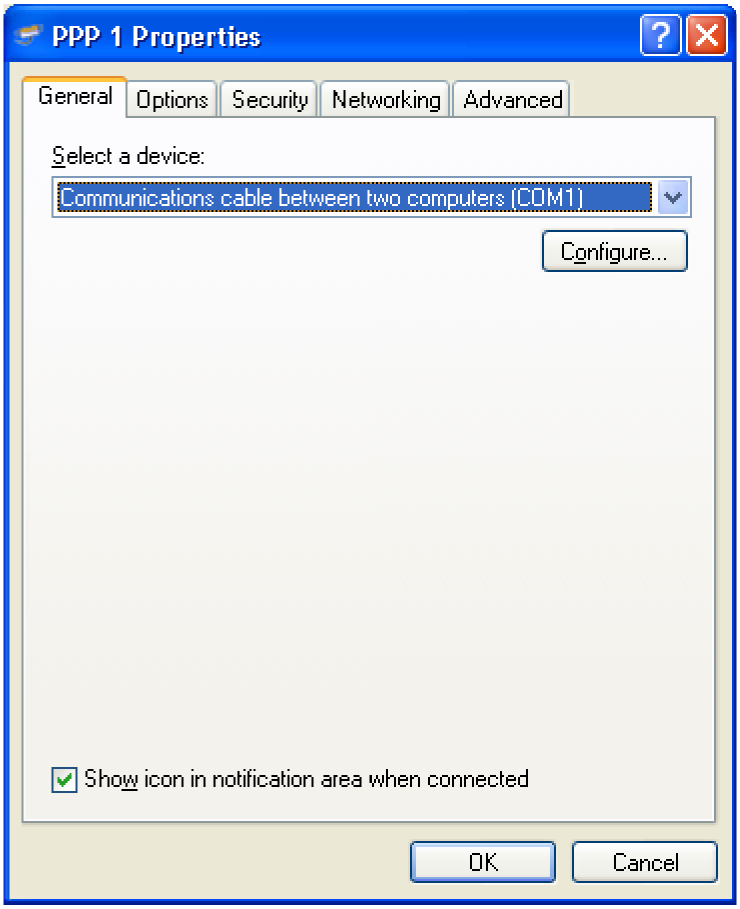
- Share with all users of this computer

The connection will be saved in the Network Connections folder.

□ Add a shortcut to this connection to my desktop

To create the connection and close this wizard, click Finish.

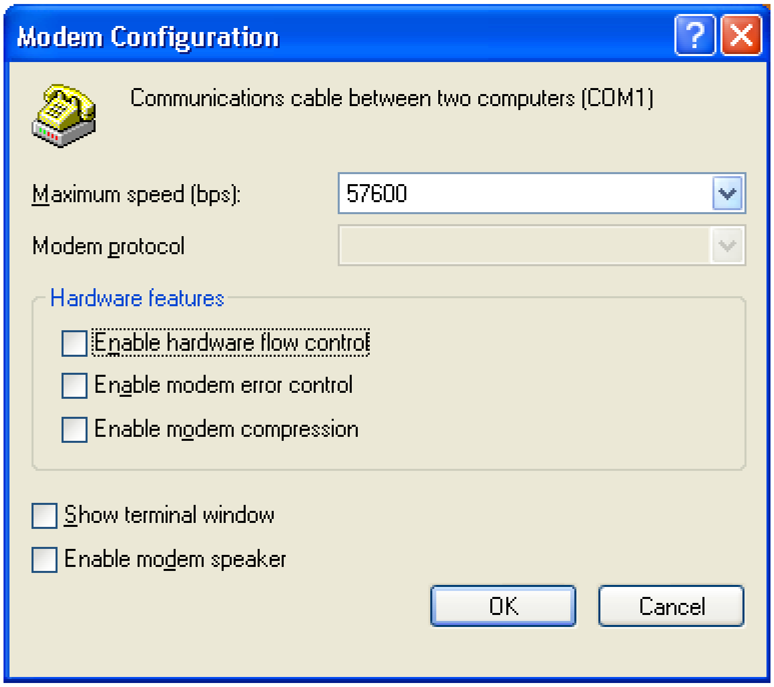
9. Click **Finish** to end the New Connection Wizard.
10. Go to **Start>Settings>Network Connections** and select the connection you created (using the name you specified in step 5). Select **Properties**.



Select a device: Communications cable between two computers (COM1)

☑ Show icon in notification area when connected

11. Click **Configure…**. Select a baud rate of 57600. Ensure the **Enable hardware flow control box** is not checked**.**



Modem Configuration

Communications cable between two computers (COM1)

Maximum speed (bps): 57600

Modem protocol

Hardware features

☐ Enable hardware flow control

☐ Enable modem error control

☐ Enable modem compression

☐ Show terminal window

☐ Enable modem speaker

12. Click **OK**.

4.2.2.5 Starting the System

1. To boot the NPU, both the kernel and the ramdisk image must be loaded into main memory. The text below describes how this might be done on an Intel® IXDP2400 Advanced Development Platform with an Ethernet connection.

   ```
   load -r -b 0x1c208000 -m tftp zImage
   load -r -b 0x1c600000 -m tftp ramdisk_ixp.gz
   go 0x1c208000
   ```

   For a system with no Ethernet connection, the images can either be burned in ROM or downloaded over the serial connection. The serial connection method is slower.

2. After the system boots, log in as **root**.

3. The **lo** interface needs to be running. This can by checked by running the **ifconfig** command. If the **lo** interface is not running, enter the following command:

   ```
   ifconfig lo 127.0.0.1 up
   ```

   Note: You can also do a **ps x** to see if **portmap** and **inetd** are running.

4. Start the PPP connection be entering the following command (all on one line):

   ```
   pppd /dev/ttyS0 57600 -detach 10.3.31.41:10.3.31.42 local noauth
   passive
   ```

   Note that **57600** should be replaced by whatever baud rate you want to use. **10.3.31.41** is the IP address of the NPU, and **10.3.31.42** is the IP address of the Windows development host. These values should be changed to appropriate values. Ensure that the addresses used do not conflict with existing IP addresses elsewhere on the network.

5. Disconnect the HyperTerminal, and start the network:

Start/Settings/Network and Dial-up Connections/Direct Connection 1

6. After the network link is established, telnet into the NPU, e.g. **telnet 10.3.31.41**, log in, and continue as you normally would; e.g.:

```
insmod halMeDrv.o
WBSrvr &
```

7. The same thing can be done with the other NPU on the board (i.e. Ingress vs. Egress), although in this case, two different IP addresses should be used:

```
pppd /dev/ttyS0 57600 -detach 10.3.31.51:10.3.31.52 local noauth
passive
```

After issuing the **pppd** command, if the Windows development host side does not connect within some amount of time, **pppd** will abort and return the serial port to the shell prompt

4.2.2.6 Windows Development Host Issues

The procedures outline above seem to make the NPU the default gateway for Windows. This is not what you want if the Windows host is connected to a LAN. To fix this problem, after the PPP connection is established, issue the following command from a DOS window:

```
route delete 0.0.0.0 mask 0.0.0.0 10.3.31.42
```

where **10.3.31.42** is the address for the Windows development host side of the connection.

Note that if you establish two connections (one for the Egress and one for the Ingress), then you need to issue two route commands (in either order):

```
route delete 0.0.0.0 mask 0.0.0.0 10.3.31.42
route delete 0.0.0.0 mask 0.0.0.0 10.3.31.52
```

4.2.2.7 Developer’s Workbench Issues

When selecting the hardware connection method in the Developer’s Workbench, there is an item labeled **Connect via Ethernet**. This is the method you should select, even though you are connected over the serial port. You would use the IP address for the NPU as specified earlier (e.g. 10.3.31.41 or 10.3.31.51).

4.2.2.8 Shutting Down the Connection

The connection can be shut down from the Windows development host side by selecting the connection and selecting **disconnect**. After this is complete, the Hyperterminal can be reconnected. You can then type commands to the shell, however, the shell will not have echo enabled. The easiest way to fix this is to issue the **reset** command in the Hyperterminal/shell window. Alternatively, you can reboot Linux.

4.3 Loading Workbench Components Through an Ethernet Connection

If you are performing this procedure for a VxWorks target, you can use one of a number of available FTP programs—for example, WFTPD Pro*—to establish an FTP server on your development PC through which Developer Workbench can connect to the target NPU or NPUs for downloading components to the microengines.The procedures require a number of predetermined settings; refer to Table 1, "Field Names and Sample Settings for Ethernet Connections Without Tornado" on page 59.

If you have WindRiver* Tornado*, you can use its FTP server capability. For details, refer to Section 4.4, "Loading Workbench Components Through an Ethernet Connection Using Tornado" .

Note: This procedure requires files provided on the Intel® IXDP Firmware and Driver CD.

4.3.1 Setting Up the FTP Server Default Folder for a VxWorks Target

For convenience, the following steps involve files that the download process requires, copied to a folder the FTP server will use as a default.

1. Create the directory C:\IXP_SDK_3.5\shared.
2. Copy to C:\IXP_SDK_3.5\shared\ the following files from the following sources:

   a. `VxWorks` and `VxWorks.sym` from the Firmware directory.

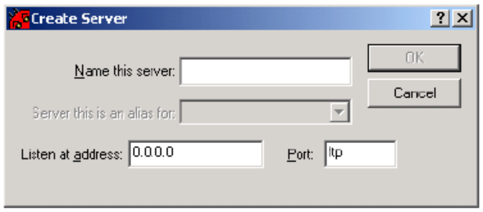
   b. `WBSrvr.o` from the IXP SDK 3.5 directory `me_tools\bin_vxw_be`.

4.3.2 Setting Up the FTP Server for a VxWorks Target

This procedure shows the use of WFTPD Pro*, available at http://www.wftpd.com; one of any number of FTP programs could be used.

After downloading and installing WFTPD Pro*, you create a server, taking the following steps:

1. Choose **Settings>Control Panel** from the Start menu.
2. From Control Panel, click WFTPD FTP Server.
   WFTPD Pro* Main Control opens.
3. Click the new server icon .
   Create Server opens.



Listen at address: 0.0.0.0

Port: ftp

   a. Type the FTP Server Name adpftp in **Name this server**.

   b. Type the Listening Address 10.3.31.225 in **Listen at address**.

   c. Leave ftp in **Port**.

   d. Click **OK**.
   In the left panel of the Main Control, the server is listed with a red STOP sign next to the name.

4. Right-click the server name adpftp and choose **Users for Server**.

   User Configuration Dialog opens.

5. Click the new user icon 👤.

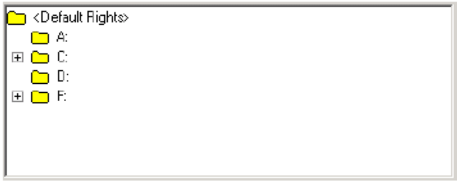
   Create New User opens.

   a. Type the FTP User name in all lowercase letters—for the default settings of the platform, use target—in **User name**, and click **OK**.

   Set user password opens.

   b. Type the FTP Password—for the default settings of the platform, use target—in **Enter Password** and **Verify Password**, and click **OK**.

   User Configuration Dialog refreshes with the user you have just established and displays a list of available drives as folders at the panel to the right:



```
<Default Rights>
    A:
⊞   C:
    D:
⊞   F:
```

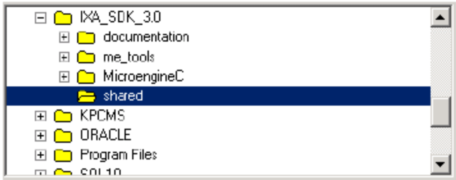
   c. Click the plus to expand the folder list for the C: drive.

   The Home Directory button, 🏠, activates.

   d. Scroll to the folder IXA_SDK_3.5.

   e. Click the plus to expand the folder list for the folder IXA_SDK_3.5.

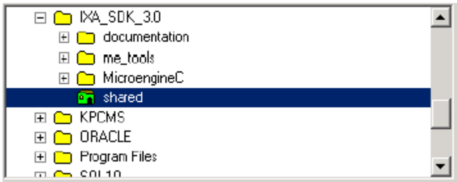
   f. Select the folder shared within the folder IXA_SDK_3.5.



```
⊟ IXA_SDK_3.0
    ⊞ documentation
    ⊞ me_tools
    ⊞ MicroengineC
      shared
⊞ KPCMS
⊞ ORACLE
⊞ Program Files
⊞ SQL10
```

   g. Click the home directory button, 🏠, with C:\IXA_SDK_3.5\shared selected.

   C:\IXA_SDK_3.5\shared is set as the home directory for the user target, and User Configuration Dialog refreshes with the setting.



```
⊟ IXA_SDK_3.0
    ⊞ documentation
    ⊞ me_tools
    ⊞ MicroengineC
      shared
⊞ KPCMS
⊞ ORACLE
⊞ Program Files
⊞ SQL10
```

h. Click **OK**.

User Configuration Dialog closes, and Main Control is displayed.

6. Right-click the server name adpftp and choose **Configure Server**.

   Configure Server adpftp opens. In the following sub-steps, you specify a folder for the server to use as a default.

   a. Click Securty.

   b. Type an established path in **Restrict Server to Path** or select the path with **Browse**—in line with the previous steps, you would use `C:\IXA_SDK_3.5\shared\`.

   c. Click OK.

   Configure Server adpftp closes, and Main Control is displayed.

7. Click **Close (Esc)**. or start again at with the ingress parameters.

4.3.3 Starting the FTP Server for a VxWorks Target

The FTP server must be started before VxWorks can download in the procedure under Section 4.3.4. To start the FTP server, take the following steps:

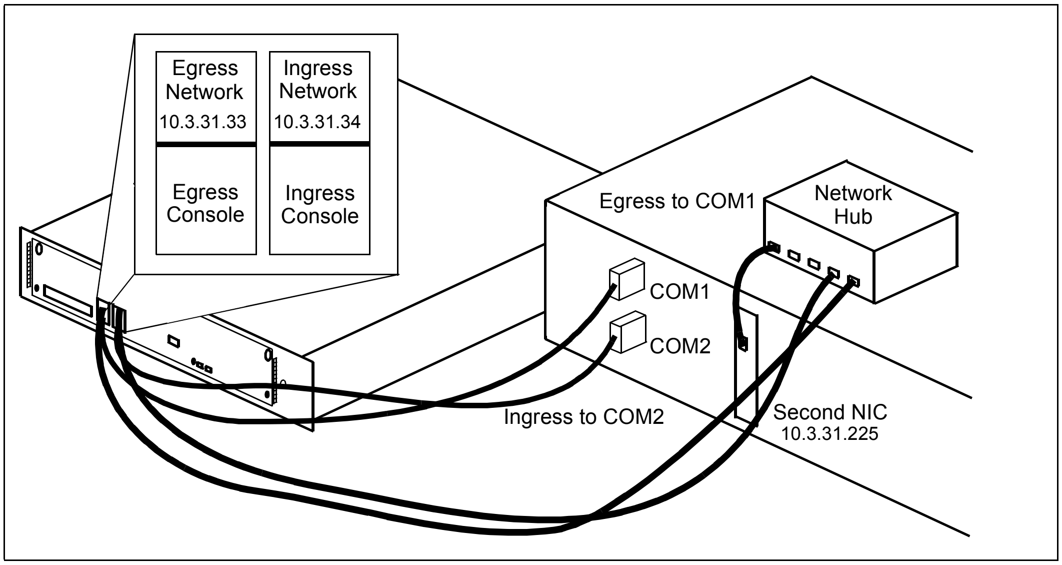
1. Choose **Settings>Control Panel** from the Start menu.
2. From Control Panel, click **WFTPD FTP Server**.
   WFTPD Pro* Main Control opens.
3. Right-click the server name adpftp and choose **Start Server**.
   The red STOP changes to a green GO. The FTP server is ready for use.

4.3.4 Loading Components for a VxWorks Target

Although this procedure shows initial setting up for both NPUs, the procedure, for illustration, uses the master-egress connection for loading Workbench components. You decide, starting with step 6, if you want to load one NPU or both NPUs; it depends on the application. You can have two server sessions at once, provided they use separate IP addresses.

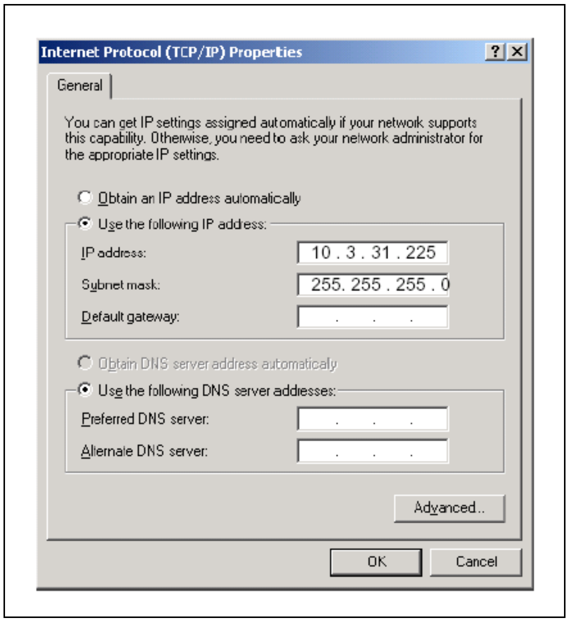
Note: The ports COM1 and COM2 are mentioned in this procedure for simplicity; you can use any COM port available on your machine. The IP addresses shown in Figure 2 are the default values for both the IXDP2400 and IXDP2800 platforms; you can change these values.

Figure 2. Loading Components for a VxWorks Target on an IXDP2400



1. Connect the Console management ports of the development platform for the Egress-Master NPU and the Ingress-Slave NPU to the development host serial ports COM1 and COM2 as needed. Use the RJ11 cables with the RJ11 to DB-9 adaptors.
2. Connect the Network management ports of the platform to a network hub with standard Ethernet patches.
3. Connect the network hub to a first or, if necessary, second NIC in the development PC with a standard Ethernet patch.
4. Adjust the address of the appropriate NIC by choosing **Start** > **Settings** > **Network and Dial-Up Connections**.
5. Right-click the Local Area Connection for the NIC you connected the network hub to in step 3, and choose **Properties**.
6. Click **Internet Protocol (TCP/IP)** in the components list and click **Properties**.

Make adjustments to the addresses like those shown in the following figure:



Internet Protocol (TCP/IP) Properties

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically

● Use the following IP address:

IP address: 10 . 3 . 31 . 225

Subnet mask: 255. 255 . 255 . 0

Default gateway: . . .

○ Obtain DNS server address automatically

● Use the following DNS server addresses:

Preferred DNS server: . . .

Alternate DNS server: . . .

7. Click **OK**.
8. Start HyperTerminal twice to create connections—for example, COM1EgressMaster and COM2IngressSlave.

   For the IXDP2400 the required settings for both Ingress and Egress (shown in the figures below) are:

   — No flow control

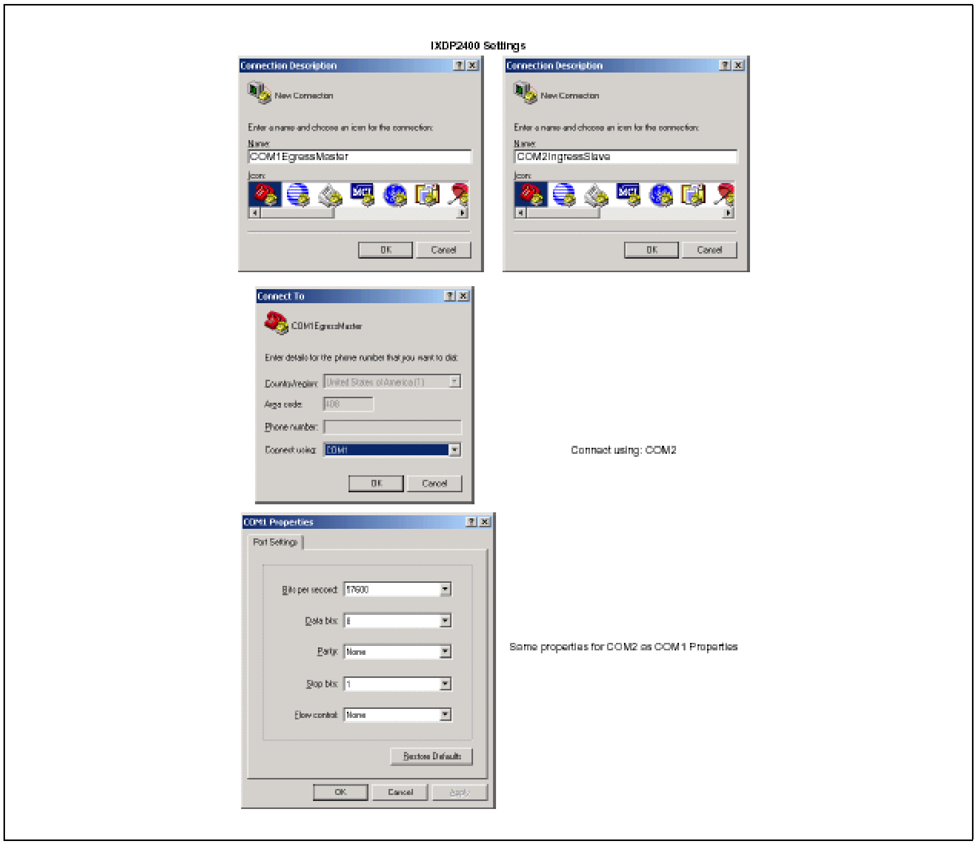
   — No parity bits

   — 8 data bits

   — 1 stop bit

   — baud rate of 57600

Note: For the IXDP2800 the required settings are the same, but the baud rate is 9600.


IXDP2400 Settings

Connection Description

New Connection

Enter a name and choose an icon for the connection:

Name: COM1EgressMaster

Connection Description

New Connection

Enter a name and choose an icon for the connection:

Name: COM2IngressSlave

Connect To

COM1EgressMaster

Enter details for the phone number that you want to dial:

Country/region: United States of America (1)

Area code: 408

Phone number:

Connect using: COM1

Connect using: COM2

COM1 Properties

Port Settings

Bits per second: 57600

Data bits: 8

Parity: None

Stop bits: 1

Flow control: None

Same properties for COM2 as COM1 Properties

If needed, you can save these HyperTerminal connections, and they will be available in the Start menu path **Programs > Accessories > Communications > HyperTerminal**. HyperTerminal displays a window for each connection.

9. Press I on the rocker switch at the rear of the platform to power up the system.

10. Press the reset button in the platform's faceplate.

    For the IXDP2400 platform perform Step 11 and Step 12. For the IXDP2800, go to step13.

11. Wait for the power up and reset cycle to complete.

    When the NPUs are ready, the HyperTerminal windows, Egress-Master and Ingress-Slave, display the following prompts.

    ```
    Master-Redboot>
    ```

    ```
    Slave-Redboot>
    ```

12. At the HyperTerminal window for the Master Egress connection, type the following commands:

    a. `fis list` to check that `bootrom` is available.

    b. `go 0xc4048000` to begin bootrom boot.

       The boot process displays a countdown and the message `Press any key to stop auto-boot.`

    c. Go to step Step 17 to continue.

13. For the IXDP2800 platform perform Step 14 through Step 16.

14. Wait for the power up and reset cycle to complete.

    When the NPUs are ready, the HyperTerminal windows, Egress and Ingress, display the following:

    ```
    Type "shell" to set IP and/or MAC addresses
    Type "help" to see available commands
    >Ingress>

    Type "shell" to set IP and/or MAC addresses
    Type "help" to see available commands
    Egress>.
    ```

15. Enter the shell command at the prompt to set the IP addresses. Each field will appear on a new line. Enter the value and press Enter to continue, or just hit Enter to leave the entry unchanged.

    ```
    >Ingress>shell
                 Command Channel = 0: serial
            Boot Delay (seconds) = 3
                     Boot Script =
                   Prompt String = Ingress
                     MAC Address =
                      IP Address = 10.3.31.34
                     Subnet Mask = 255.255.255.0
                 Default Gateway = 0.0.0.0
            Remote System 1 Name = TestTarget2
              Remote System 1 IP = 0.0.0.0
            Remote System 2 Name =
            Remote System 3 Name =
            Remote System 4 Name =
    ```

    Repeat this for the Egress.

16. At the prompt for the Ingress, type the following to launch VxWorks:

    `>Ingress>launch C4808000`

    The boot process displays a countdown and the message `Press any key to stop auto-boot.`

Note: You must launch VxWorks for Ingress first then Egress.

At the prompt for the Egress, type the following to launch VxWorks:

`>Egresss>launch C4808000`

In each HyperTerminal window, the boot process displays a countdown and the message `Press any key to stop auto-boot.`

17. Press the space bar to stop the countdown.

    The boot process displays the prompt `[VxWorks Boot]:.`

18. Type `p`, the print command, and examine the boot parameters.

    VxWorks Boot displays, among other parameters, the inet on ethernet address of the NPU and the host inet. For egress, the default is 10.3.31.33; for ingress the default is 10.3.31.34. The default for host inet is 10.3.31.225. The defaults for username and password are target and target.

19. Type `c`, the change command, if you need to adjust the boot parameters.

VxWorks Boot prompts you with each boot parameter. Press the Enter key if you accept the parameter, or type the new parameter and press the Enter key to change the parameter. The following steps are provided as examples.

— To change the value of inet on ethernet, type the appropriate IP address when VxWorks Boot displays the inet on ethernet parameter.

— To change the value of the host inet, type the appropriate IP address when VxWorks Boot displays the host inet parameter.

— To change the value of the user name, type an appropriate user name when VxWorks Boot displays the user name (u) parameter.

— To change the value of the user name's password, type an appropriate password when VxWorks Boot displays the ftp password (pw) parameter.

— Specify the 'file name' as the complete path and name of the VxWorks BSP

For more information on these parameters, refer to the Intel® IXDP2400 Advanced Development Platform System User's Guide. Refer also to any readme files included with the Board Support Package readme files for the latest information on connections and boot procedures.

20. Press Control-D when you are done examining the defaults or you have no more parameters to change.

    VxWorks Boot stores the parameters.

21. Wait for the prompt `[VxWorks Boot]:`.

22. Type @ and press the Enter key.

    VxWorks reloads with the newly established boot parameters.

23. Type `ld < WBSrvr.o`.

    The Developer's Workbench Backend WBSrvr.o loads on the NPU.

24. Type `WbSvr_init`.

    The Developer's Workbench Backend starts, and the window shows `value message = 0 = 0x0`.

25. Open an egress project—for example, oc48_pos_egress.dwp for the Egress IXP2400 on the IXDP2400 development platform.

26. Choose **Debug > Hardware** in Developer's Workbench.

    The project is set for hardware debugging.

27. Choose **Hardware > Options**.

    Hardware Options opens.

    a. Click **Connections**.
    There are projects where the Connections tab shows chip names; many projects do not involve chip names.

    b. Click on a chip name if shown—for example, egress.

    c. Click **Connect via Ethernet**.

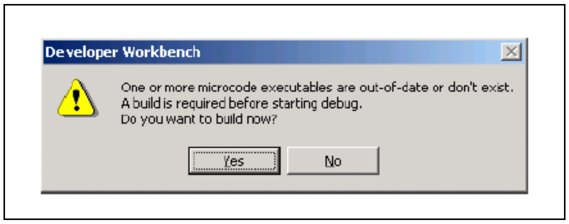
    Specify node name or IP address clears.

    d. Type in **Specify node name or IP address** the IP address—the default 10.3.31.33 or the IP address you established for the Egress Master NPU.

28. Choose **File > Save All** to save the project settings.

29. Choose **Debug** > **Start Debugging**.

Developer's Workbench shows you the following message in the following case:



One or more microcode executables are out-of-date or don't exist.
A build is required before starting debug.
Do you want to build now?

If you click **Yes**, Developer's Workbench performs the build.

30. Set Breakpoints in your source, run the microcode, and monitor the data.

Note: Some projects involve both egress and ingress NPUs. In this case, you would repeat this procedure, substituting the values you have determined for the ingress NPU and choosing the name of the ingress chip in the Connections tab of Hardware Options. Table 1, "Field Names and Sample Settings for Ethernet Connections Without Tornado" on page 59 provides a sample list of values.

4.3.5 Summary of Settings

The previous procedures required some planned settings. The following table shows a summary list of the settings for your convenience:

Table 1. Field Names and Sample Settings for Ethernet Connections Without Tornado

WFTPD Pro* Field Name	Internet Protocol (TCP/IP) Properties Field Name	Bootrom Field Name	Description	Setting
Name this server	n.a	n.a	Name of FTP Server to start for download platform	adpftp
listening address	IP Address	host inet	IP address of nic	10.3.31.225
n.a	Subnet Mask	n.a	For IP address of NIC	255.255.255.0
Port	n.a	n.a	Port at which WFTPD Pro* will listen	ftp
User name	n.a	n.a	user name (u)	target
Enter password Verify password	n.a	ftp password (pw)	FTP password	target
Restrict Server to Path	n.a	n.a	n.a	C:\IXP_SDK_3.5\shared
n.a	n.a	inet on ethernet	IP addresses of egress and ingress NPUs	egress: 10.3.31.33 ingress: 10.3.31.34

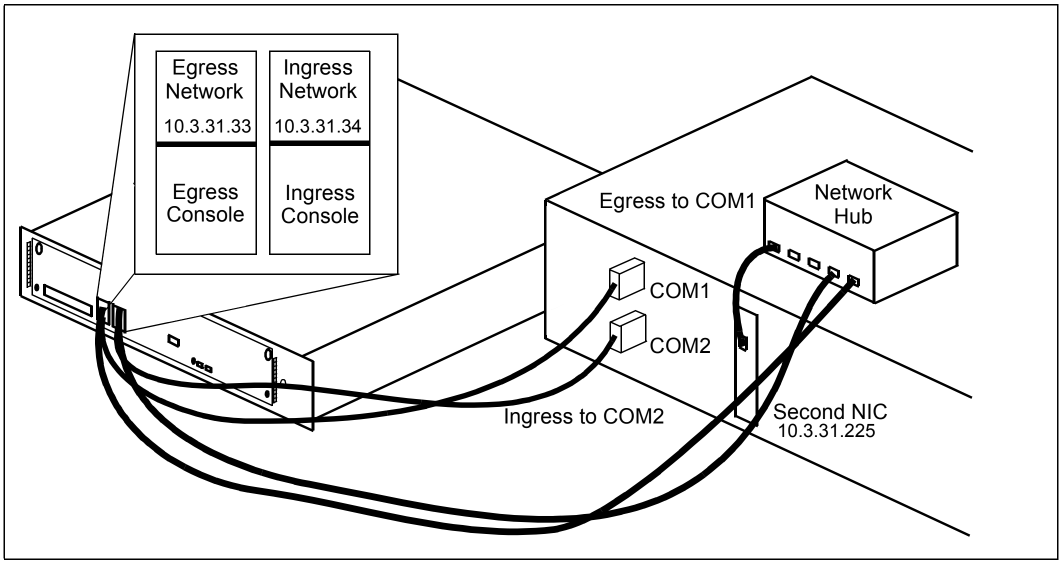
4.4 Loading Workbench Components Through an Ethernet Connection Using Tornado

This procedure details the task of loading workbench components on target hardware through the network management ports of the platform.

Although this procedure shows initial setting up for both NPUs, the procedure, for illustration, uses the master-egress connection for loading Workbench components. You decide, starting with Step 8, if you want to load one NPU or both NPUs. You can have two server sessions at once, provided they use separate IP addresses.

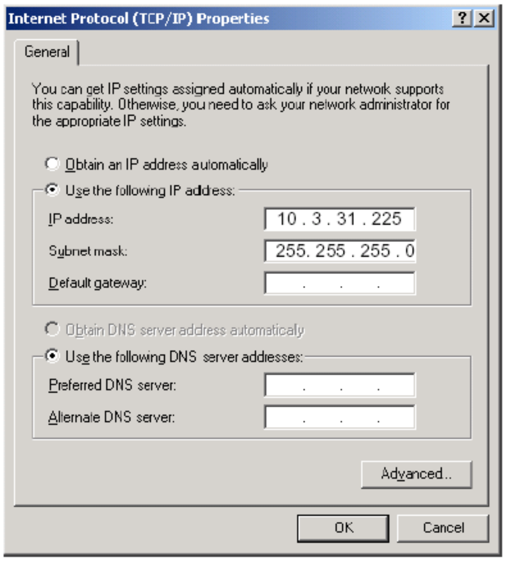
Note: The ports COM1 and COM2 are mentioned in this procedure for simplicity; you can use any COM port available on your machine. The IP addresses shown in Figure 3 are the default values for the IXDP2400 and IXDP2800; you can change these values.

Figure 3. Loading Workbench Components Through Ethernet on an IXDP2400



1. Connect the Console management ports of the development platform for the Egress-Master NPU and the Ingress-Slave NPU to the development host serial ports COM1 and COM2 as needed. Use the RJ11 cables with the RJ11 to DB-9 adaptors.
2. Connect the Network management ports of the platform to a network hub with standard Ethernet patches.
3. Connect the network hub to a first or, if necessary, second NIC in the development PC with a standard Ethernet patch.
4. Adjust the address of the second NIC by choosing **Start > Settings > Network and Dial-Up Connections**.
5. Right-click the Local Area Connection for the NIC you connected the network hub to in step 3, and choose **Properties**.
6. Click **Internet Protocol (TCP/IP)** in the components list and click **Properties**.

   Make adjustments to the addresses like those shown in the following figure:



You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically

● Use the following IP address:

IP address: 10 . 3 . 31 . 225

Subnet mask: 255. 255 . 255 . 0

Default gateway: . . .

○ Obtain DNS server address automatically

● Use the following DNS server addresses:

Preferred DNS server: . . .

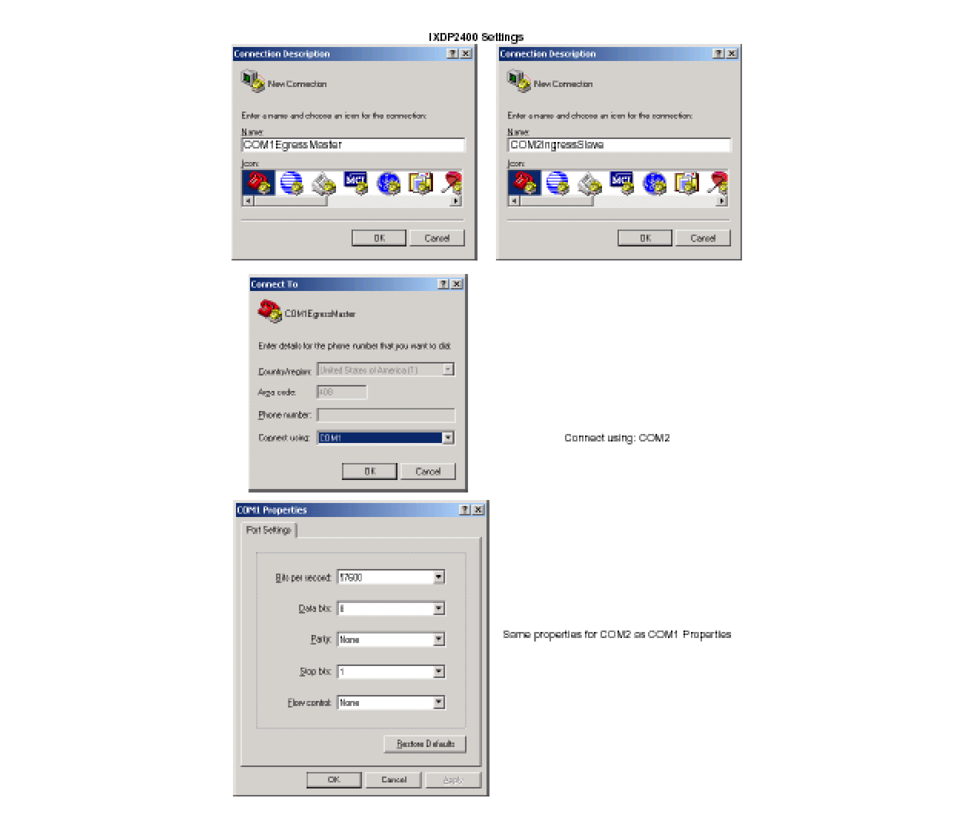
Alternate DNS server: . . .

7. Click **OK**.
8. Start HyperTerminal twice to create connections—for example, COM1EgressMaster and COM2IngressSlave

For the IXDP2400 the required settings for both Ingress and Egress (shown in the figures below) are:

- — No flow control
- — No parity bits
- — 8 data bits
- — 1 stop bit
- — baud rate of 57600

Note: For the IXDP2800 the required settings are the same, but the baud rate is 9600.


IXDP2400 Settings

Connection Description

New Connection

Enter a name and choose an icon for the connection:

Name: COM1EgressMaster

Connection Description

New Connection

Enter a name and choose an icon for the connection:

Name: COM2IngressSlave

Connect To

COM1EgressMaster

Enter details for the phone number that you want to dial:

Country/region: United States of America (1)

Area code: 408

Phone number:

Connect using: COM1

Connect using: COM2

COM1 Properties

Port Settings

Bits per second: 57600

Data bits: 8

Parity: None

Stop bits: 1

Flow control: None

Same properties for COM2 as COM1 Properties

If needed, you can save these HyperTerminal connections, and they will be available in the Start menu path **Programs > Accessories > Communications > HyperTerminal**. HyperTerminal displays a window for each connection.

9. Press I on the rocker switch at the rear of the platform to power up the system.
10. Press the reset button in the faceplate.
11. For the IXDP2400 platform perform Step 12 and Step 13.
12. Wait for the power up and reset cycle to complete.

    When the NPUs are ready, the HyperTerminal windows, Egress-Master and Ingress-Slave, display the following prompts, respectively: `Master-Redboot>` and `Slave-Redboot>`.

Note: For illustration, the rest of this procedure uses the master-egress connection for loading and booting VxWorks.

13. At the HyperTerminal window for the Master Egress connection, type the following commands:

    a. `fis list` to check that `bootrom` is available.

    b. `go 0xc4048000` to begin bootrom boot.

    The boot process displays a countdown and the message `Press any key to stop auto-boot.`

    c. Go to step Step 18 to continue.

14. For the IXDP2800 platform perform Step 15 through Step 17.
15. Wait for the power up and reset cycle to complete.

    When the NPUs are ready, the HyperTerminal windows, Egress and Ingress, display the following:

```
Type "shell" to set IP and/or MAC addresses
Type "help" to see available commands
>Ingress>

Type "shell" to set IP and/or MAC addresses
Type "help" to see available commands
Egress>
```

16. Enter the shell command at the prompt to set the IP addresses. Each field will appear on a new line. Enter the value and press Enter to continue, or hit Enter to leave the entry unchanged.

```
>Ingress>shell
             Command Channel = 0: serial
        Boot Delay (seconds) = 3
                 Boot Script =
               Prompt String = Ingress
                 MAC Address = 11-22-33-44-55-66
                  IP Address = 10.3.31.34
                 Subnet Mask = 255.255.255.0
             Default Gateway = 0.0.0.0
        Remote System 1 Name = TestTarget2
          Remote System 1 IP = 0.0.0.0
        Remote System 2 Name =
        Remote System 3 Name =
        Remote System 4 Name =
```

    Repeat this for the Egress.

17. At the prompt, type the following to launch VxWorks:

    `>Ingress>launch C4808000`

    The boot process displays a countdown and the message `Press any key to stop auto-boot.`

Note: You must launch VxWorks for Ingress first then Egress.

At the prompt for the Egress, type the following to launch VxWorks:

`>Egress>launch C4808000`

The boot process displays a countdown and the message `Press any key to stop auto-boot.`

18. Press the space bar to stop the countdown.

    The boot process displays the prompt `[VxWorks Boot]:`.
19. Type the print command `p`.

    VxWorks Boot displays, among other parameters, the inet on ethernet address of the NPU and the host inet. For egress, the default is 10.3.31.33; for ingress the default is 10.3.31.34. The default for host inet is 10.3.31.225. The defaults for username and password are target and target.

20. To adjust the addresses, type `c`.

    VxWorks Boot prompts you with each boot parameter. Press the Enter key if you accept the parameter, or type the new parameter and press the Enter key to change the parameter.

    The following steps are provided as examples.

    — To change the value of inet on ethernet, type the appropriate IP address when VxWorks Boot displays the inet on ethernet parameter.

    — To change the value of the host inet, type the appropriate IP address when VxWorks Boot displays the host inet parameter.

    — To change the value of the user name, type an appropriate user name when VxWorks Boot displays the user name (u) parameter.

    — To change the value of the password, type an appropriate password when the VxWorks Boot displays the ftp password (pw) parameter.

    For more information on these parameters, refer to the *Intel® IXDP2400 Advanced Development Platform System User's Guide* or *Intel® IXDP2800 Advanced Development Platform System User's Guide*. Refer also to Board Support Package readme files for the latest information on connections and boot procedures.

    Once you are finishing accepting or changing parameters, VxWorks Boot stores the parameters and displays its prompt.

21. Press Control-D when you are done examining the defaults or you have no more parameters to change.

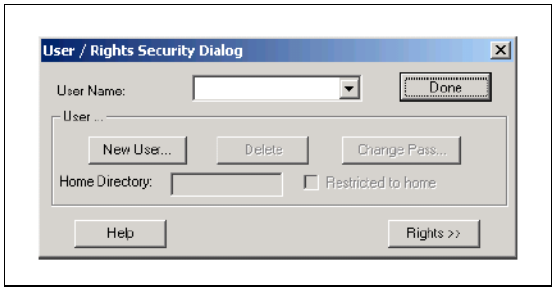
    VxWorks Boot stores the parameters.

22. Wait for the prompt `[VxWorks Boot]:`.

23. Choose **Start > Tornado2.2 > FTP Server**.

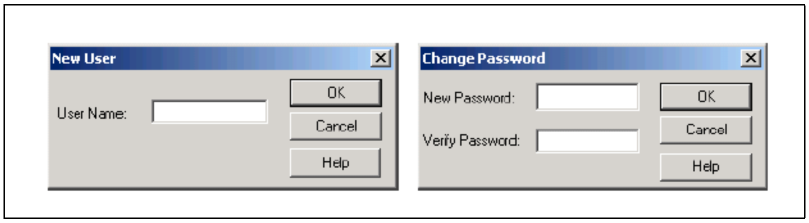
24. Choose **Security > User/Rights**.

    FTP Server displays the following window:



25. Click **New User**.

    FTP Server displays windows for user information:



26. Enter the user name and password you provided for the boot parameters in step Step 20.

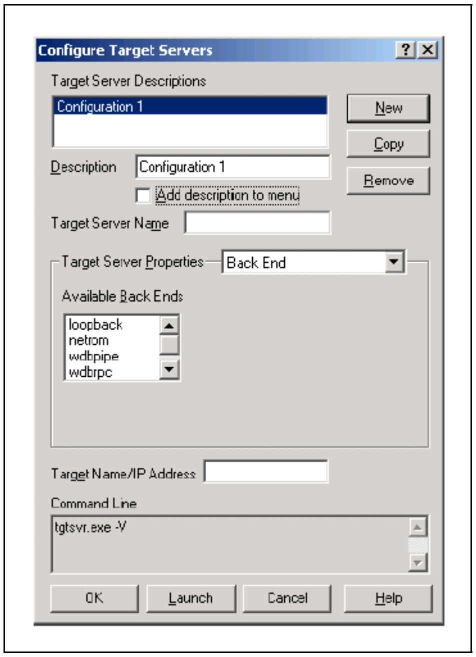
27. In **User/Rights Security Dialog**, enter in **Home Directory** the new user's path where a loadable vxWorks image is—for example, `\VxWorks\bin in` the Firmware directory, `C:\Tornado` or an appropriate WindRiver Tornado directory.
28. Click **Done**.
29. At the HyperTerminal window for the Master Egress connection, type @.

    VxWorks boots with the new parameters.

Note: To make adjustments to the ingress slave NPU, you would for example use a connection like COM2IngressSlave and work with the default value 10.3.31.34 for ingress inet on ethernet and the default value 10.3.31.225 for host inet or change them appropriately.

30. Wait for the VxWorks banner that ends with `WDB: Ready`.
31. In the HyperTerminal window, choose **Call** > **Disconnect**.

    HyperTerminal releases the COM port for that connection.
32. Launch Tornado—the default menu path is **Start** > **Programs** > **Tornado2.2** > **Tornado**—and click **Cancel** in Create Project in New/Existing Workspace if it appears.
33. Select **Tools** > **TargetServer** > **Configure**. Tornado displays a target server configuration window:



Configure Target Servers

Target Server Descriptions: Configuration 1

Description: Configuration 1

☐ Add description to menu

Target Server Name:

Target Server Properties: Back End

Available Back Ends: loopback, netrom, wdbpipe, wdbrpc

Target Name/IP Address:

Command Line: tgtsvr.exe -V

34. Assuming **Target Server Descriptions** has no configurations in it, click New and the name Configuration1 should appear in this box.

    You can re-use target servers.
35. Type a name—for example, MasterEgressNIC2 or SlaveIngressNIC2—in **Target Server Name**. Record the name; you will need it for a later step.
36. Type the IP address you set for the target in step 15 in **Target Name/IP Address**—the defaults are 10.3.31.33 for MasterEgressNIC2 or 10.3.31.34 for SlaveIngressNIC2.

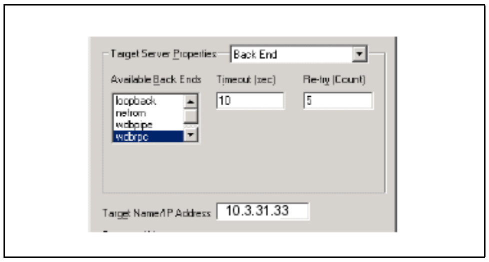
There are three groups of properties called Target Server Properties to set. You set each group by first selecting its name in **Target Server Properties**. The first is Back End which is already displayed.

37. The following table shows the three target server property group names and, presented in order, the settings within those properties that you make. If a property setting is not shown in the table, leave it at its default setting

Target Server Property Group Name	IXDP2400 Property Settings	IXDP2800 Property Settings
Back End	**Available Back Ends**: wdbrpc **Timeout (sec)**: 10 **Re-try (Count)**: 5 **Target Name/IP Address**: (egress) 10.3.31.33 (ingress) 10.3.31.34	**Available Back Ends**: wdbrpc **Timeout (sec)**: 10 **Re-try (Count)**: 5 **Target Name/IP Address**: (egress) 10.3.31.33 (ingress) 10.3.31.34
Core File and Symbols	**File**: path\vxWorks[a] **All Symbols**: (chosen)	**File**: path\vxWorks[a] **All Symbols**: (chosen)
Console and Redirection	**Redirect Target IO**: (checked, optional)	**Redirect Target IO**: (checked, optional)

a. For example, `VxWorks` provided on the Firmware and Driver CD.

Settings for the three target server properties for the IXDP2400 listed in the table are shown below.



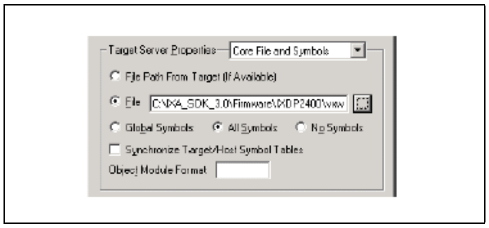
Target Server Properties: Back End

Available Back Ends: loopback, netrom, wdbpipe, wdbrpc (selected: wdbrpc)

Timeout (sec): 10

Re-try (Count): 5

Target Name/IP Address: 10.3.31.33



Target Server Properties: Core File and Symbols

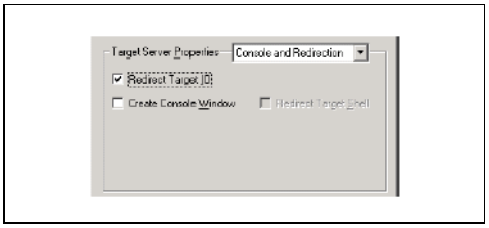
○ File Path From Target (If Available)

● File: C:\IXA_SDK_3.0\Firmware\IXDP2400\vxw

○ Global Symbols ● All Symbols ○ No Symbols

□ Synchronize Target/Host Symbol Tables

Object Module Format:



Target Server Properties: Console and Redirection

☑ Redirect Target IO

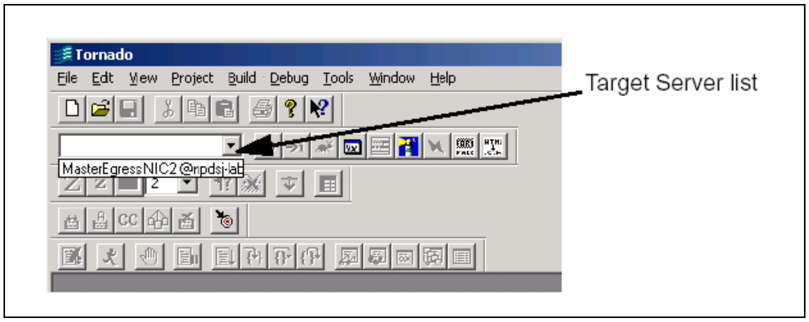
□ Create Console Window □ Redirect Target Shell

38. Click **Launch** in the Tornado window **Configure Target Servers**.

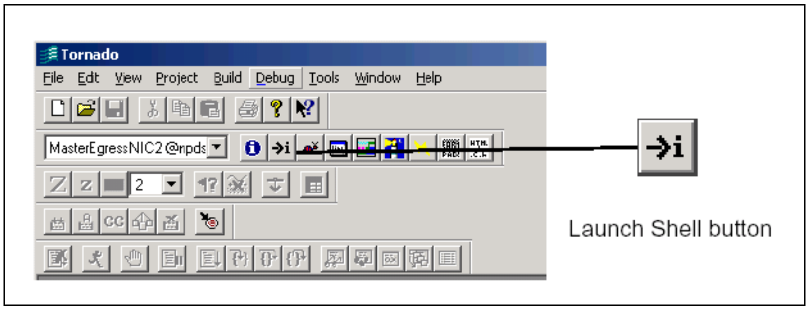
    Tornado creates in the system tray a Log Console for the target server. When you right-click the target icon and choose Show, the log console opens to display three success log messages followed by two warning log messages. Ignore the two warning messages. The target also appears in the Tornado Registry which is also in the system tray.

    You can also choose **Tools > Shell**, select the target server, and click **OK**.

39. Click the name of the target server—for example, **MasterEgressNIC2@*machinename***—in the target server list of the Launch toolbar as shown in the following figure:

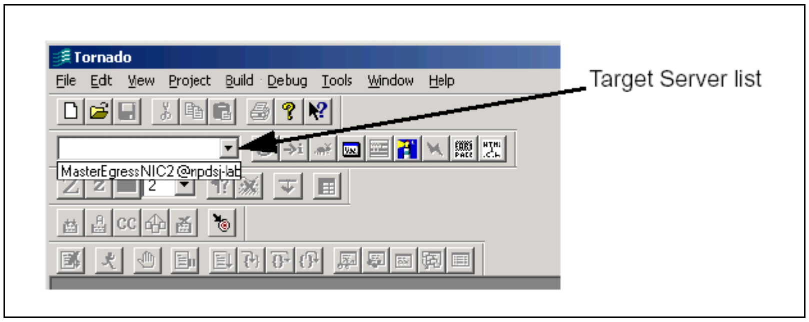

Tornado
File Edit View Project Build Debug Tools Window Help
MasterEgressNIC2@npdsj-lab
Target Server list

40. Click the Launch Shell button on the Launch toolbar.

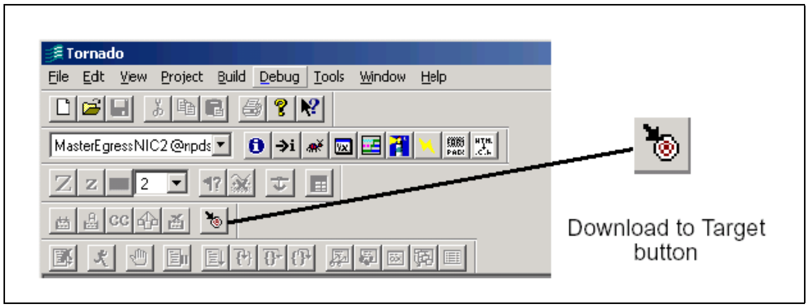


Tornado displays a window for the shell—for example, **Shell MasterEgressNIC2@*machinename***.

41. Click the name of the target server—for example, **MasterEgressCOM1@*machinename***—in the target server list of the Launch toolbar as shown in the following figure:


Tornado
File Edit View Project Build Debug Tools Window Help
MasterEgressNIC2@npdsj-lab
Target Server list

42. Click the Download to Target button on the Launch toolbar.


Tornado
File Edit View Project Build Debug Tools Window Help
MasterEgressNIC2@npds
2
Download to Target button

A browse dialog box, **Download objects**, appears.

43. Browse to the Workbench Backend—for example, `C:\IXA_SDK_3.5\me_tools\bin_vxw_be\WBSrvr.o`.

44. Click **Download**.

    `WBSrvr.o` downloads to the target for the target server.

Note: The `SHELL` commands `CD` and `LD` may be used to do the above, but the interface shown here is more simple.

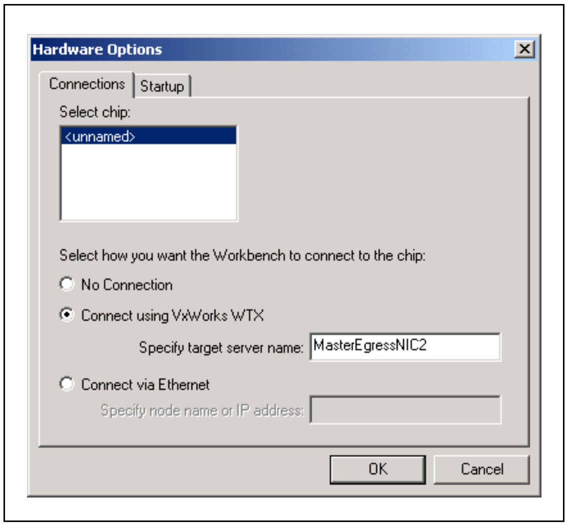
The Workbench backend is linked to the following libraries: `Rs_udebug.a`, `dbgme.a`, `halmev2.a`, `utils.a`, `osApi.a`, `rs_cntl.a`, and `uclo.a`.

45. Start Developer's Workbench (**Programs** > **IXA SDK 3.5** > **DevWorkbench**) and open a microcode project (**File** > **Open Project**).

    a. Choose **Debug** > **Hardware**.

46. Choose **Hardware** > **Options** > **Setup** and select **Reset the microengines and load the microcode.**

47. Click **Connections**.



Hardware Options

Connections | Startup

Select chip:

<unnamed>

Select how you want the Workbench to connect to the chip:

○ No Connection

● Connect using VxWorks WTX

Specify target server name: MasterEgressNIC2

○ Connect via Ethernet

Specify node name or IP address:

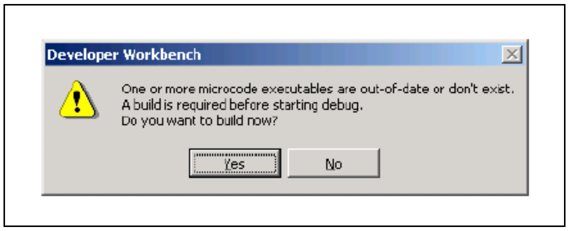
OK Cancel

48. Click **Connect using VxWorks WTX**.

49. Type the target server name you typed in Step 48.

50. Choose **Debug > Start Debugging**.

    Developer's Workbench shows you the following message in the following case:

    One or more microcode executables are out-of-date or don't exist.
    A build is required before starting debug.
    Do you want to build now?

    If you click **Yes**, Developer's Workbench downloads the microcode and enables Breakpoints and Watch functionality.

51. Set Breakpoints in your source, run the microcode, and monitor the data.

4.5 Loading and Booting Linux on the Intel® IXDP2400

The following sections detail various ways to boot and load Linux on the Intel® IXDP2400 Advanced Development Platform. For additional information, review the BSP readme file that accompanies the BSP provided on the Firmware and Driver CD.

4.5.1 Booting the Linux zImage from Flash

The Linux kernel binary (zImage) can be saved in Flash so that you don't have to install via the network each time you boot up. The following describes how to save the zImage onto flash:

First load the zImage into SDRAM at an address specified by -b option of the RedBoot "load" command, then use RedBoot "fis" command to burn the image into the flash. In the following command example, we assume that you will be loading image from the network, but you could also load the image over a serial port.

```
Master-RedBoot> load -v -r -b 0x1c208000 -m tftp -h <server-ip>
zImage

Master-RedBoot> fis list

Name              FLASH addr  Mem addr    Length      Entry point

RedBoot           0xC4000000  0x00000000  0x00040000  0x00000000

System Log        0xC4FA0000  0xC4FA0000  0x00020000  0x00000000

RedBoot config    0xC4FDF000  0xC4FDF000  0x00001000  0x00000000

FIS directory     0xC4FE0000  0xC4FE0000  0x00020000  0x00000000
```

To create a new partition for Linux image in flash, first load the images to RAM address (as explained in the previous step) specified by -b option, then issue this command:

```
Master-RedBoot> fis create -b 0x1c208000 -l 0x100000 -e 0x1c208000
linux
```

```
... Erase from 0xc4420000-0xc4520000: ........

... Program from 0x1c208000-0x1c308000 at 0xc4420000: ........

... Unlock from 0xc4fe0000-0xc5000000: .

... Erase from 0xc4fe0000-0xc5000000: .

... Program from 0x1ffdf000-0x1ffff000 at 0xc4fe0000: .

... Lock from 0xc4fe0000-0xc5000000: .

Master-RedBoot>
```

To re-write the burnt linux image in flash:

```
Master-RedBoot> fis write -b 0x1c208000 -l 0x100000 -b 0x1c208000
```

To boot up the zImage burnt into the flash:

```
Master-RedBoot> fis load linux
Master-RedBoot> go
```

If running in Dual Flash mode, repeat this process on the Slave NPU console.

4.5.2 Loading and Booting the zImage over Serial Port

To boot using a serial console, follow these steps:

1. Set your serial port configuration for the Intel® IXDP2400 as follows:
   57600 baud
   8 bit data
   stop 1
   parity None
   no flow control

Note: The settings for the Intel® IXDP2800 are the same except that the baud rate is 9600.

2. Issue the following command:
   ```
   Master-RedBoot> load -r -b 0x1c208000 -m y
   ```
3. Now, using Hyperterminal or Minicom, begin transferring the zImage file using ymodem protocol.
4. To boot up Linux, issue the following command:
   ```
   Master-RedBoot> go 0x1c208000
   ```
5. You can also burn the image to the flash and boot from flash thereafter. (skip step 4 above)

To boot up the zImage burnt into the flash, enter:

```
Master-RedBoot> fis load linux
Master-RedBoot> go
```

4.5.3 Linux Target Setup - Get File From a Linux Host

Once Linux is launched on the target (Intel® IXDP2400), perform the following:

1. At the login prompt, log in as root without any password. This gets you to the Linux bash shell.
2. On the Linux host system with the Intel® IXA SDK 3.51 Tools installed (see Section 3), you must copy the Workbench server, WBSrvr, and MicroEngine driver, halmedrv.o, from the development host to the shared target file system for the egress and ingress processors. Enter the following commands:

```
cp <path to IXA SDK 3.51 installation>/me_tools/bin_linux_be/
halMeDrv.o /opt/hardhat/devkit/arm/xscale_be/target/root/.
cp <path to IXA SDK 3.51 installation>/me_tools/bin_linux_be/
WBSrvr /opt/hardhat/devkit/arm/xscale_be/target/root/.
chmod 777 /opt/hardhat/devkit/arm/xscale_be/target/root/WBSrvr
```

3. You must now install a special character device for the WorkBench server to work correctly. Enter the following commands on either the egress and ingress console:

```
rm -rf /dev/MeDrv0
mknod /dev/MeDrv0 c 10 244
```

4. You are now ready to launch the WorkBench server and start debugging from the Developer's WorkBench. Enter the following commands on both the egress and ingress consoles from root's home directory (cd ~):

```
insmod -f medrv.o
./WBSrvr &
```

Note: For complete details on running the Developer's Workbench in a Win32 environment, refer to Section 4.4, Step 14 and perform the procedures listed there.

4.5.4 Running the Hello World Program from the Intel® IXDP2400

Once Linux boots up and display the Linux login prompt. Login as `root', without any password.

You are nfs mounted to /opt/hardhat/devkit/arm/xscale_be/target/root directory at Linux host PC.

From the IXDP2400:

```
Ixdp2400 bash# pwd
/root
```

At the Linux host PC:

Write a simple Hello World program under the mounted directory, /opt/hardhat/devkit/arm/xscale_be/target/root .

```
HostPC bash# vi hello.c
}#include <stdio.h>
void main ( void)
{
         printf("Hello World\n");
 return;
```

```
    }
```

Compile the program with gcc compiler situated at /opt/hardhat/devkit/arm/xscale_be/bin

```
HostPC bash# ./xscale_be-gcc -mbig-endian -o ../target/root/
hello.o ../target/root/hello.c
```

At IXDP2400 console:

```
Ixdp2400 bash# ./hello.o
       Hello World
```

4.5.5 Compiling Linux Sources

Once you install the MontaVista* distribution and you want to simply use the pre-compiled binary, then use this binary
/opt/hardhat/devkit/arm/xscale_be/target/boot/vmlinuz-intel-ixdp2400.

But if you want to use the kernel image with latest changes, then you need to recompile the source. The installation installs sources for Linux BSP at the following directory:
/opt/hardhat/devkit/lsp/intel-ixdp2400/linux-2.4.18_mvl30

4.6 Loading and Booting Linux on the Intel® IXDP2800

Complete details on loading and launching Linux on the Intel® XDP2800 Advanced Development Platform are contained in the *Intel ® IXDP2800 Advanced Development Platform System User's Manual*, Chapter 2 - System Setup and Configuration. The System User's manual ships with the development platform.

4.6.1 Linux Target Setup - Get File from a Win32 Host

Once Linux is launched on the target (IXDP2800), perform the following:

1. At the login prompt, log in as root without any password. This gets you to the Linux bash shell.
2. Use the FTP client to download the test application (WBSrvr) and driver (halmedrv.o) from the development host. Enter the following command:
   ```
   ftp 10.3.31.225
   ```
3. Enter login user name and password. Then transfer the files from \IXA_SDK_3.5\me_tools\bin_linux_be by entering the following commands:
   ```
   binary
   get halmedrv.o
   get WBSrvr
   bye
   ```
4. You will be returned to the Linux bash shell. Enter the following commands:
   ```
   chmod 777 WBSrvr
   insmod -f halmedrv.o
   ./WBSrvr &
   ```

Note: For complete details on running the Developer's Workbench in a Win32 environment, refer to Section 4.4, Step 13 and perform the procedures listed there.

Both the master and slave NPUs should now be able to get the IP addresses and root file system from the development host.

4.6.2 Linux Target Setup - Get File from a Linux Host

Once Linux is launched on the target (IXDP2800), perform the following:

1. At the login prompt, log in as root without any password. This gets you to the Linux bash shell.
2. On the Linux host system with the Intel® IXA SDK 3.51 Tools installed (see Section 3), you must copy the Workbench server, WBSrvr, and MicroEngine driver, halmedrv.o, from the development host to the shared target file system for the egress and ingress processors. Enter the following commands:

   ```
   cp <path to IXA SDK 3.51 installation>/me_tools/bin_linux_be/
   halMeDrv.o /opt/hardhat/devkit/arm/xscale_be/target/root/.
   cp <path to IXA SDK 3.51 installation>/me_tools/bin_linux_be/
   WBSrvr /opt/hardhat/devkit/arm/xscale_be/target/root/.
   chmod 777 /opt/hardhat/devkit/arm/xscale_be/target/root/WBSrvr
   ```

3. You must now install a special character device for the WorkBench server to work correctly. Enter the following commands on either the egress and ingress console:

   ```
   rm -rf /dev/MeDrv0
   mknod /dev/MeDrv0 c 10 244
   ```

4. You are now ready to launch the WorkBench server and start debugging from the Developer's WorkBench. Enter the following commands on both the egress and ingress consoles from root's home directory (cd ~):

   ```
   insmod -f medrv.o
   ./WBSrvr &
   ```

Note: For complete details on running the Developer's Workbench in a Win32 environment, refer to Section 4.4, Step 14 and perform the procedures listed there.

Rebuilding on VxWorks A

This appendix contains procedures for rebuilding the Intel XScale® Core Libraries and Workbench RPC Server on Windows 2000*/Windows XP*.

A.1 Files and Directories

Intel XScale® Core Libraries:

Pre-built Intel XScale® core libraries are located under .\IXA_SDK_3.5\me_tools\lib_linux_be. Source files for building these libraries are located at following locations:

Library	Path
utils.a	.\IXA_SDK_3.5\me_tools\utils
osApi.a	.\IXA_SDK_3.5\me_tools\osApi
dbgme.a	.\IXA_SDK_3.5\me_tools\XSC_CoreLibs\dbgme
halMev2.a	.\IXA_SDK_3.5\me_tools\XSC_CoreLibs\halMev2
rs_cntl.a	.\IXA_SDK_3.5\me_tools\XSC_CoreLibs\rs_cntl
rs_udebug.a	.\IXA_SDK_3.5\me_tools\share\rs_udebug
uclo.a	.\IXA_SDK_3.5\me_tools\XSC_CoreLibs\uclo
RdDriver.a	.\IXA_SDK_3.5\me_tools\PMU\RemoteDriver\
libossl.a	.\IXA_SDK_3.5\me_tools\ossl\source

Workbench RPC Server:

Pre-built Workbench RPC server (WBSrvr) is located under .\IXA_SDK_3.5\me_tools\bin_Linux_be.

Source files for building these binaries are located in following locations:

Binary	Path
WBSrvr	.\IXA_SDK_3.5\me_tools\XSC_CoreLibs\rs_cntl

A.2 Setting the System Paths and Variables

Before you begin, ensure that Tornado is installed and the system variables and path are set correctly as follows:

1. Set the variable WIND_BASE to your Tornado* install (For example: WIND_BASE=C:\tornado2.x)
2. Set the variable WIND_HOST_TYPE=x86-win32
3. Add Tornado to the path (For example C:\Tornado2.x\host\x86-win32\bin)
4. Run torvars from Command Prompt to confirm the setting.
   >torvars

A.3 Rebuilding the Intel XScale® Core Libraries and Workbench RPC Sever

To rebuild all the release version of Intel XScale® core libraries and Workbench RPC server enter the following steps:

```
>cd IXA_SDK_3.5\me-tools\XSC_CoreLibs\makDefs
>make SYS=VXWORKS BE=1 -f coreLibs.mak clean
>make SYS=VXWORKS BE=1 -f coreLibs.mak
```

A.4 Rebuilding Debug Versions

To rebuild all the debug version of libraries and Workbench RPC server perform the following steps:

```
>cd IXA_SDK_3.5\me-tools\XSC_CoreLibs\makDefs
>make SYS=VXWORKS CCMODE=DEBUG BE=1 -f coreLibs.mak clean
>make SYS=VXWORKS CCMODE=DEBUG BE=1 -f coreLibs.mak
```

Rebuilding on Linux B

This appendix contains procedures for rebuilding the Intel XScale® Core Libraries, Workbench RPC Server and the Linux driver on Red Hat*Linux 7.3.

B.1 Files and Directories

Intel XScale® Core Libraries

Prebuilt XScale core libraries are located under IXA_SDK_3.5/me_tools/lib_linux_be. Source files for building these libraries are located at following locations:

Library	Path
utils.a	./IXA_SDK_3.5/me_tools/utils
osApi.a	./IXA_SDK_3.5/me_tools/osApi
dbgme.a	./IXA_SDK_3.5/me_tools/XSC_CoreLibs/dbgme
halMev2.a	./IXA_SDK_3.5/me_tools/XSC_CoreLibs/halMev2
rs_cntl.a	./IXA_SDK_3.5/me_tools/XSC_CoreLibs/rs_cntl
rs_udebug.a	./IXA_SDK_3.5/me_tools/share/rs_udebug
uclo.a	./IXA_SDK_3.5/me_tools/XSC_CoreLibs/uclo
RdDriver.a .	./IXA_SDK_3.5/me_tools/PMU/RemoteDriver\
libossl.a	./IXA_SDK_3.5/me_tools/ossl/source

Prebuilt Workbench RPC server (WBSrvr) and Linux driver (halMeDrv.o) are located under IXA SDK 3.5/ me_tools/bin_Linux_be. Source files for building these binaries are located in following locations:

Workbench RPC Server

Binary	Path
halMeDrv.o	/IXA_SDK_3.5/me_tools/XSC_CoreLibs/halMev2
WBSrvr	/IXA_SDK_3.5/me_tools/XSC_CoreLibs/rs_cntl

B.2 Rebuilding the Intel XScale® Core Libraries and Workbench RPC Sever

To rebuild all the release version of Intel XScale® core libraries, WB Server and Linux driver enter the following commands:

```
>cd IXA_SDK_3.5/me-tools/XSC_CoreLibs/makDefs
>make SYS=LINUX BE=1 -f coreLibs.mak clean
>make SYS=LINUX BE=1 -f coreLibs.mak
```

B.3 Rebuilding Debug Versions

To rebuild all the debug version of Intel XScale® core libraries, WB Server and Linux driver perform the following steps:

```
>cd IXA_SDK_3.5/me-tools/XSC_CoreLibs/makDefs
>make SYS=LINUX CCMODE=DEBUG BE=1 -f coreLibs.mak clean
>make SYS=LINUX CCMODE=DEBUG BE=1 -f coreLibs.mak
```