NET+Works with GNU Tools Getting Started Guide

Making DEVICE NETWORKING easy™

NET+Works with GNU Tools Getting Started Guide

Operating system/version: 6.3
Part number/version: 90000718_C
Release date: March 2006
www.digi.com

©2006 Digi International Inc.

Printed in the United States of America. All rights reserved.

Digi, Digi International, the Digi logo, the Making Device Networking Easy logo, NetSilicon, a Digi International Company, NET+, NET+OS and NET+Works are trademarks or registered trademarks of Digi International, Inc. in the United States and other countries worldwide. All other trademarks are the property of their respective owners.

Information is this document is subject to change without notice and does not represent a commitment on the part of Digi International.

Digi provides this document “as is,” without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of, fitness or merchantability for a particular purpose. Digi may make improvements and/or changes in this manual or in the product(s) and/or the program(s) described in this manual at any time.

This product could include technical inaccuracies or typographical errors. Changes are made periodically to the information herein; these changes may be incorporated in new editions of the publication.

Contents

Chapter 1: Using NET+Works with GNU Tools and a Hardware Debugger 1

Task 1: Getting ready 2

Gather IP parameters 2

Save files and close applications 2

Verify your access rights 2

Verify hardware requirements 3

MAJIC upgrades 3

Task 2: Installing the NET+Works software 4

Task 3: Setting up the IP address of the MAJIC probe 6

Task 4: Creating an MDI server icon 10

Task 5: Configuring the IP address of the development board 15

Task 6: Building and downloading the template application 17

Building the BSP 17

Building the template application 19

Downloading the template application 21

Task 7: Running and debugging the template application 23

Running the application 23

Debugging the application with a break point 24

Chapter 2: Using NET+Works with GNU Tools and the GDB Software Debugger . 27

Task 1: Getting ready 28

Gather IP parameters 28

TFTP server 28

Save files and close applications 28

Verify your access rights 29

Verify hardware requirements 29

Task 2: Installing the NET+Works software 30

Task 3: Configuring the IP address of the development board 32

Task 4: Setting up the GDB software debugger 34

Updating the BSP 34

Loading the GDB-server-enabled bootloader 35

Adding a second serial connection 36

Updating the template application 36

Launching the TFTP server 37

Executing the GDB software debugger-enabled bootloader 37

Task 5: Debugging the template application 39

Chapter 3: Tips and Suggestions 41

Where should I put my code, and why? 41

What should my next step be? 41

Appendix: Updating the MAJIC Probe's Firmware 43

Using this guide

Review this section for basic information about this guide, as well as for general support contact information.

About this guide

This guide describes NET+OS with GNU Tools and how to use it as part of your development cycle. Part of the NET+Works integrated product family, NET+OS is a network software suite optimized for the NET+ARM.

Software release

This guide supports NET+OS 6.3. By default, this software is installed in the `/cygdrive/c/netos63_gnu` directory. The instructions in this guide are based on the assumption that NET+Works is installed in the default directory.

Who should read this guide

This guide is for software engineers and others who use NET+Works for NET+OS. To complete the tasks described in this guide, you must:

- Be familiar with installing and configuring software.
- Have sufficient user privileges to do these tasks.
- Be familiar with network software and development board systems.

Conventions used in this guide

This table describes the typographic conventions used in this guide:

This convention	Is used for
italic type	Emphasis, new terms, variables, and document titles.
bold, sans serif type	Menu commands, dialog box components, and other screen items.
menu name → option	Menu names and selections selections.
`monospaced type`	Filenames, pathnames, and code examples.

What's in this guide

This table shows where you can find information in this guide:

To read about	See
Installing NET+Works with GNU Tools, configuring IP parameters, and completing a brief exercise using either the MAJIC probe or the Raven debugger	Chapter 1: "Using NET+Works with GNU Tools and a Hardware Debugger"
Installing NET+Works with GNU Tools, configuring IP parameters, and completing a brief exercise using the GDB software debugger	Chapter 2: "Using NET+Works with GNU Tools and the GDB Software Debugger"
Tips for using NET+Works with GNU Tools with your own projects	Chapter 3, "Tips and Suggestions"

Related documentation

- The *Hardware Installation Guide* describes how set up the hardware.
- *NET+Works with GNU Tools BSP Porting Guide* describes how to port the board support package (BSP) to a new hardware application using *GNU Tools*.
- *NET+Works with GNU Tools Programmer's Guide* describes how to use NET+OS to develop programs for your application and hardware.

- NET+Works online help describes the application programmer interfaces (APIs) that are provided with NET+OS. The online help is located in `C:\netos63_gnu\Documentation`.

For information about third-party products and other components, review the documentation CD-ROM that came with your development kit.

For information about the processor you are using, see your NET+Works hardware documentation.

Documentation updates

Digi occasionally provides documentation updates on the Web site.

Be aware that if you see differences between the documentation you received in your NET+Works package and the documentation on the Web site, the Web site content is the latest version.

Customer support

To get help with a question or technical problem with this product, or to make comments and recommendations about our products or documentation, use the contact information listed next:

- United States telephone: 1 877 912-3444
- International telephone: 1 952 912-3444
- email: digi.info@digi.com
- Web site: http://digi.com

Chapter 1: Using NET+Works with GNU Tools and a Hardware Debugger

This chapter provides a series of tasks that will get you up and running with NET+Works with GNU Tools. In these tasks, you will:

- Install NET+Works with GNU Tools.
- Configure IP parameters.
- Complete a brief exercise that demonstrates how to use the tool set, including:
  - Building the board support package (BSP), libraries, and template applications
  - Running and debugging the sample application

You must do all the tasks in this chapter, in the order in which they are presented.

To get started, go on to "Task 1: Getting ready."

Note: If you are using the GDB software debugger, skip to Chapter 2: "Using NET+Works with GNU Tools and the GDB Software Debugger."

Plan to spend approximately two hours completing the tasks.

Task 1: Getting ready

This task describes what you need to do before you get started.

The instructions in this document are based on the assumption that you have already installed your hardware.

Gather IP parameters

See your network administrator for IP parameter information:

- If you are using a Raven debugger, you need IP parameters for the development board *only*.
- If you are using a MAJIC probe, you need IP parameters for both the development board and the MAJIC.

You may find it convenient to write the information in the spaces provided.

Development board

- IP address:

  _____

- Subnet mask:

  _____

- Default gateway:

  _____

MAJIC probe

- IP address:

  _____

- Subnet mask:

  _____

- Default gateway:

  _____

Save files and close applications

Save all open files and close any open applications, because after you install the software, you will reboot your PC.

Verify your access rights

Make sure you have administrative rights on the PC on which you are installing NET+Works.

Verify hardware requirements

- Your PC must be running either Microsoft Windows 2000 or XP.

  Be aware that Windows 95/98/ME, Windows NT 4.0, and Windows Server 2003 are *not* supported.
- If you are using a MAJIC probe and need to configure it, it's most convenient to have two COM ports on your PC.

MAJIC upgrades

If you are using a MAJIC probe with firmware prior to version 3.6.4, you must upgrade the MAJIC firmware.

For instructions, see the Appendix.

What's next?

Go on to the next task to install the NET+Works software.

Task 2: Installing the NET+Works software

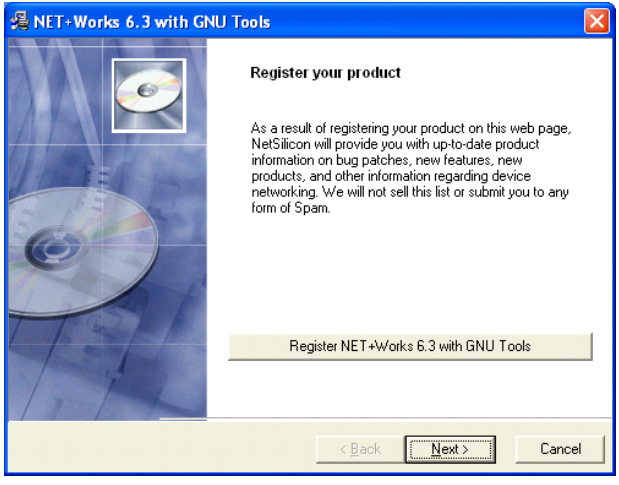
In this task, you will install the NET+Works software on your system, using a wizard that guides you through the process.

During the installation, if you register NET+Works, you are prompted for a serial number. Use the serial number that's located on your development board. If the development board is not available, use `Vnnnnnnnn`.

▶ **To install NET+Works:**

1 Place the installation CD in your CD drive, and follow the wizard prompts.

After about 40 minutes, the **NET+Works 6.3 with GNU Tools** dialog box prompts you to register your product:



Digi recommends that you register your software so you can receive information about software updates.

2 If you want to register your software, click **Register NET+Works 6.3 with GNU Tools**, and follow the prompts.

Digi recommends that you register your software so you can receive information about software updates.

3 To complete the installation, in the **NET+Works 6.3 with GNU Tools** dialog box, click **Next**.

After the NET+Works software is installed, you see a prompt about whether to restart your PC.

4 You *must* click **OK**.

What's next?

If you are using a MAJIC probe, go to the next task to set up the MAJIC's IP address.

If you are using a Raven debugger, skip to Task 5, "Configuring the IP address of the development board."

Task 3: Setting up the IP address of the MAJIC probe

This task describes how to set up the IP address of the MAJIC probe. In this procedure, you'll use the EPI MAJIC Setup Wizard and provide the information you gathered in Task 1.

You must follow the instructions in this task if:

- You are a new user.
- You are upgrading and do not know the IP address of your MAJIC probe.

Otherwise, skip to the next task.

▶ **To set up the MAJIC's IP address:**

1 Using the serial cross cable that ships with the MAJIC, connect the MAJIC to an available COM port on your PC.

If you don't have an available connection on your PC, reuse the one that's already connected to the development board. In this case, close all applications associated with the COM port, such as HyperTerminal.

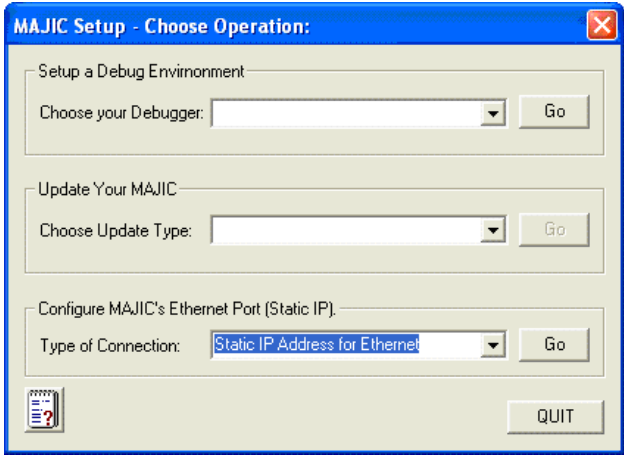
2 Do one of these steps:

- **Windows XP systems.** Select **Start → All Programs → EPI Tools-EDTA 2.2a → MAJIC Setup Wizard.**
- **Windows 2000 systems.** Select **Start → Programs → EPI Tools-EDTA 2.2a → MAJIC Setup Wizard.**

The **EPI MAJIC Setup Wizard Introduction** window opens:

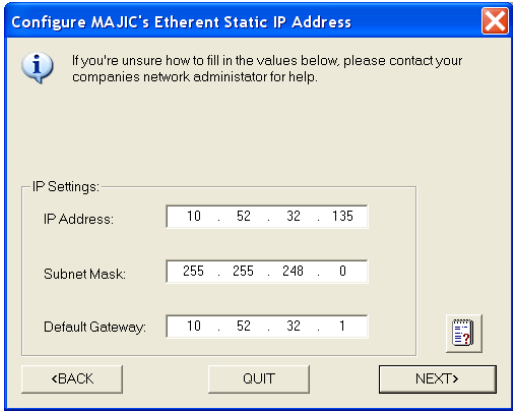
3 Click **NEXT.**

The **Choose Operations** window opens:



4 From the **Type of Connection** pulldown menu, select **Static IP Address for Internet**, and then click **Go**.

The **Configure MAJIC's Ethernet Static IP Address** window opens; this example has the information filled in.

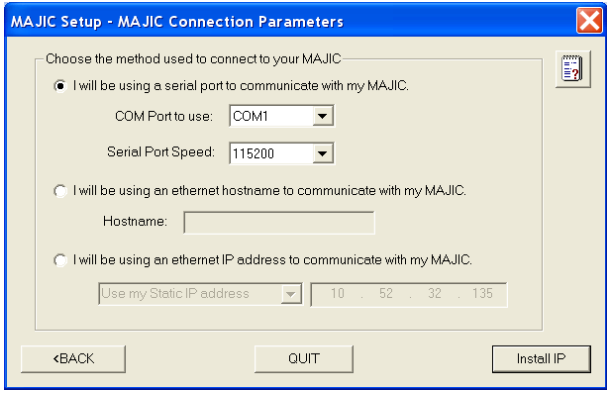


5 Enter the information you got from your network administrator, using the format shown in the example:

- **IP Address**
- **Subnet Mask**
- **Default Gateway**

and then click **NEXT**.

The **MAJIC Connection Parameters** window opens:



6 Do these steps:

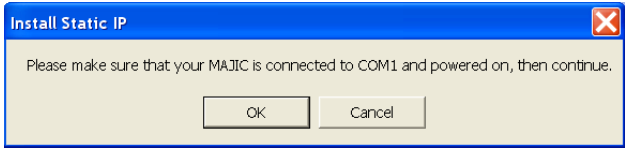
a Click **I will be using a serial port to communicate with my MAJIC.**

b From the **COM port to use** pulldown menu, select the serial port number.

Make sure no other programs are using the COM port you select.

c Click **Install IP.**

The **Install Static IP** dialog box opens:



7 In the **Install Static IP** dialog box, click **OK.**

A dialog box and a DOS window open.

8 In the **Check Your Installation Result** dialog box, confirm that the IP address information in the DOS window is correct by clicking **OK.**

If there is a problem, correct it, and go back to step 1 of this task.

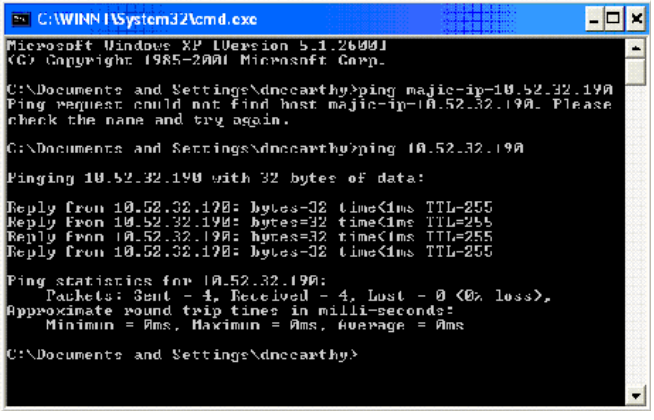
9 After the MAJIC configuration completes, power-cycle the MAJIC.

10 In the DOS window, which opened as a result of step 7, `ping` the MAJIC's IP address; for example:

`ping` *`10.52.32.135`*

where *`10.52.32.135`* is the IP address of your MAJIC, which you configured in step 5.

This is what you see in the DOS window:



If the `ping` receives replies, the IP address is installed.

If the ping times out, do these steps:

- Check the Ethernet connection to the MAJIC.
- Confirm that the IP parameters are legal.
- Verify that you power-cycled your MAJIC.

Otherwise, repeat this task.

11 Close the DOS window.

What's next?

Now that you've set up your MAJIC's IP address, you'll create an MDI-server icon that you can use to debug with gdbtk. Go on to the next task.

Task 4: Creating an MDI server icon

The MDI server icon is the interface between the MAJIC and GDB, which is the debug environment. In this task, you'll make software configuration settings for the MDI server icon.

The EPI MAJIC Setup Wizard guides you through this process, in which you'll make selections and provide information.

▶ **To create the MDI server icon:**

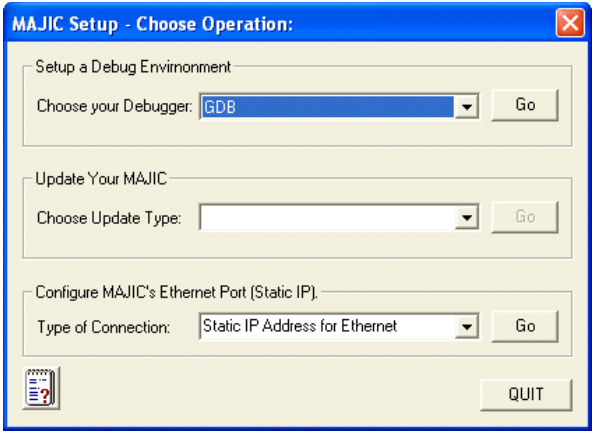
1 Start the EPI MAJIC Setup Wizard by doing one of these steps:

- **Windows XP systems.** Select **Start → All Programs → EPI Tools-EDTA 2.2a → MAJIC Setup Wizard.**
- **Windows 2000 systems.** Select **Start → Programs → EPI Tools-EDTA 2.2a → MAJIC Setup Wizard.**

The **EPI MAJIC Setup Wizard Introduction** window opens:

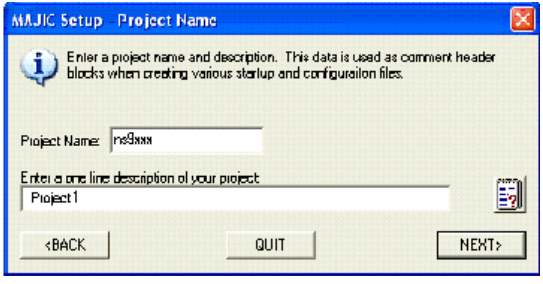
2 After you review the introduction, click **NEXT.**

The **Choose Operation** window opens:



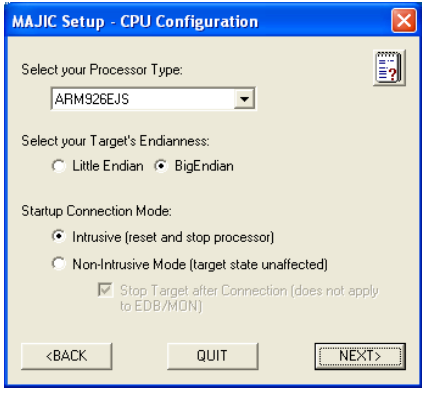
3 From the **Choose Your Debugger** pulldown menu, select **GDB**, and then click **Go**.

The **Project Name** window opens:



4 Create a new project by entering a project name and a brief description. Then click **NEXT**.

The **CPU Configuration** window opens:



5 Do these steps:

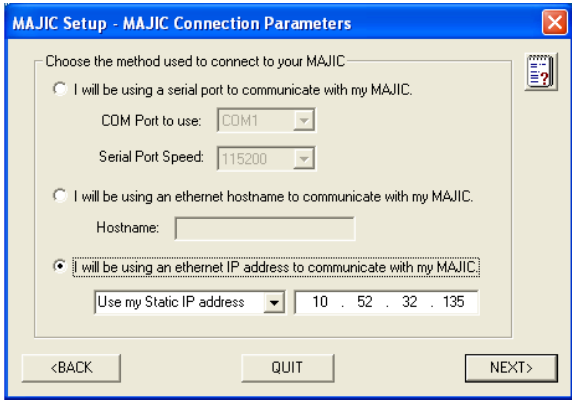
a From the **Select Your Processor Type** pulldown menu, select **ARM926EJS**.

b Under **Select your Target's Endianness**, click **BigEndian**.

c Under **Startup Connection Mode**, click **Intrusive Mode (reset and stop processor)**.

Then click **NEXT**.

The **MAJIC Connection Parameters** window opens:



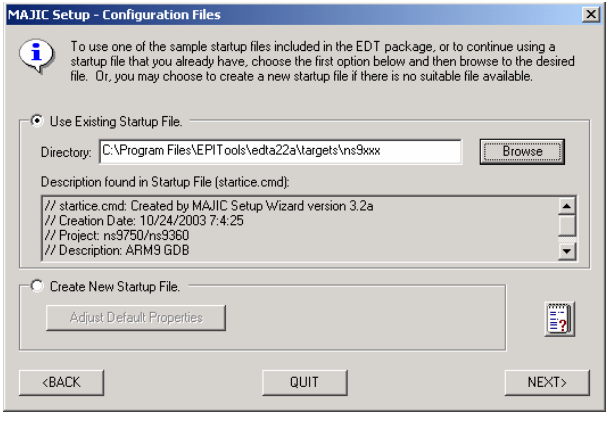
6 Do these steps:

a Click **I will be using an Ethernet IP address to communicate with my MAJIC**, and verify the IP address for the MAJIC. The address should match the one you installed in Task 3

b Make sure **Use My Static IP address** is selected.

Then click **NEXT**.

The **Configuration Files** window opens:



7 Click **Use Existing Startup File.**

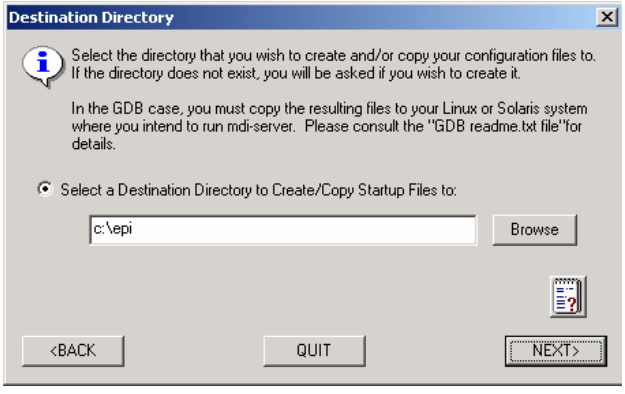
Then do these steps:

a Click **Browse.**

b Navigate to the **Program Files** → **EPITools** → **edta22a** → **targets** → **ns9xxx** directory, and click **Open.**

Then click **NEXT.**

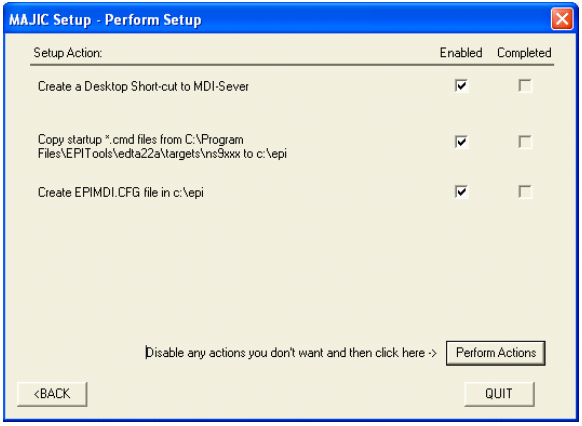
The **Destination Directory** window opens:



8 Click **Select a Destination Directory to Create/Copy Startup Files to.**

Then either browse to or create the directory you want to use for files that are created or copied during the MAJIC setup, and click **NEXT.**

The **Perform Setup** window opens with a summary of your selections:



9 Check **Enabled** for each item, and then click **Perform Actions**.

This step creates MAJIC setup files in the directory you specified in step 8.

If the directory does not exist, the MAJIC Setup Wizard prompts you with a popup warning to create one. To create the directory, click **Yes**.

The wizard also creates a shortcut to the MDI server on your desktop.

10 To exit from the wizard, click **Done**.

What's next?

Go to the next task to configure the IP address of your development board.

Task 5: Configuring the IP address of the development board

In this task, you'll configure the IP address of your development board, using the information you gathered in Task 1.

During this procedure, be prepared to move quickly to your HyperTerminal window at step 3, because you have only five seconds to respond to the prompt.

If you re-used serial cables to communicate with your MAJIC, reconnect them to your development board now, following the instructions in the *Hardware Installation Guide*.

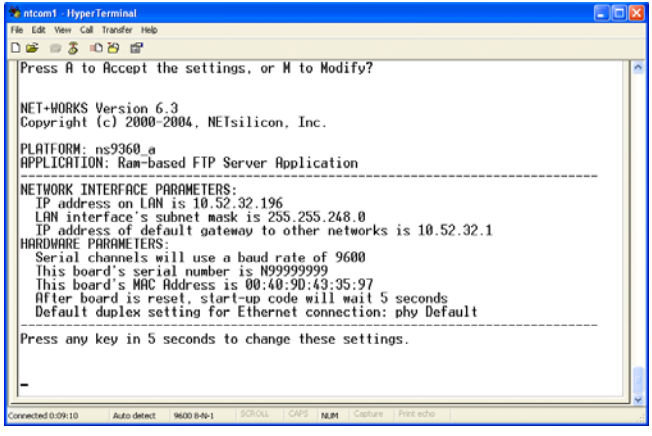
▶ **To configure the IP address of your development board:**

1 Open a **HyperTerminal** window by doing one of these steps:

- **Windows XP systems.** Select **Start** → **All Programs** → **NET+OS 6.3 GNU Tools** → **HyperTerminal Connection**, and then select the COM port you connected your development board to (either **COM1** or **COM2**).
- **Windows 2000 systems.** Select **Start** → **Programs** → **NET+OS 6.3 GNU Tools** → **HyperTerminal Connection**, and then select COM port you connected your development board to (either **COM1** or **COM2**).

2 Reboot the development board by disconnecting it from the power source and reconnecting it.

In the **HyperTerminal** window, you see information similar to this:



3 *Within five seconds*, press a key.

At the bottom of the window, you see a prompt:

```
Press A to Accept the settings, or M to Modify?
```

4 To change the configuration, press M, and then press Enter.

You are prompted for a root password.

5 Enter the default root password — `Netsilicon` —and press Enter.

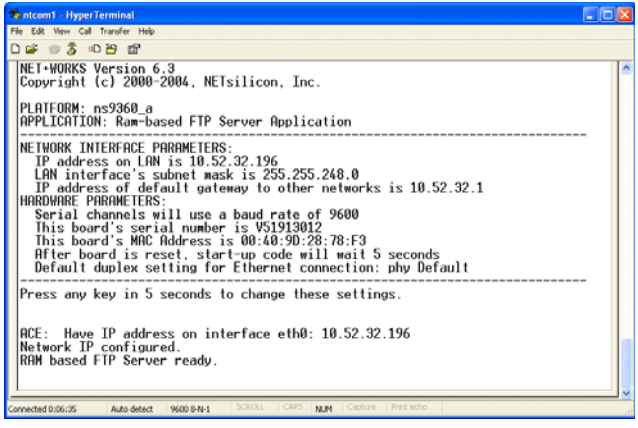
The first of a series of configuration prompts appears.

6 At each prompt, do one of these steps:

- To accept the current or default value, press Enter.
- To change a setting, enter a value and press Enter.

After you finish entering the settings, a prompt indicates that you must press a key within five seconds if you want to change additional settings.

After you make your selections, the **HyperTerminal** window looks similar to this one:



What's next?

You are ready to use the development board. You can now build, download, and debug the template application.

Task 6: Building and downloading the template application

In this task, you will build the BSP and Hello World, one of the sample applications provided with NET+Works. This simple application writes `Hello World` to the HyperTerminal window (which you opened in Task 5, step 1). After you build, you'll download the application.

This task uses the NS9360 development board as the example platform.

If you are a first-time Cygwin user, be aware that Cygwin, unlike Windows, is case-sensitive. As you do the steps in this procedure, be sure to follow the case-sensitive instructions.

Building the BSP

In typical use, you don't need to build the BSP every time you build an application. In your first use of the software, however, you *must* first build the BSP, and then build your application.

▶ **To build the BSP:**

1. Double-click the **NET+Works 6.3 Build Environment** icon on your desktop.

   An X-Tools shell opens.

2. In the X-Tools shell, change the working directory to the top-level BSP directory by typing:

   ```
   cd /cygdrive/c/netos63_gnu/src/bsp
   ```

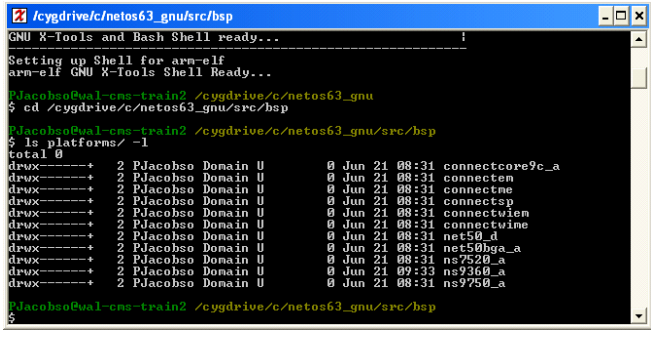
   and then press Enter.

3. To see the supported platforms, list the contents of the `platforms` directory by entering this command:

   ```
   ls platforms/ -l
   ```

   and press Enter.

You see a list of platforms:



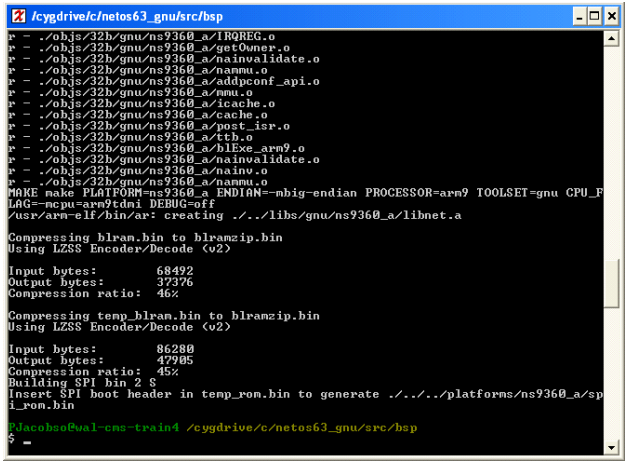
4 In the X-Tools shell, to build the NS9360 development board, enter this command:

```
make PLATFORM=ns9360_a clean all
```

and press Enter.

The build takes several minutes to complete; during this time, you don't see any activity on your screen.

This is an example of a successful BSP build:



Building the template application

▶ **To build the template application:**

1 In the X-Tools shell, change to the build directory of the template sample:

```
cd /cygdrive/c/netos63_gnu/src/apps/template/32b
```

and press Enter.

2 To create a `.gdbinit` file, do one of these steps:

– If you are using a MAJIC, enter this command:

```
make PLATFORM=ns9360_a gdbinit
```

and press Enter.

This step creates an ARM9 MAJIC `.gdbinit` file.

– If you are using a Raven, do either of these steps:

ARM9-based platforms. Copy `/cygdrive/c/netos63_gnu/debugger_files/gdbns9xxx.raven` to the `32b` folder for your application, and rename it `.gdbinit` by entering:

```
cp ./../../../../debugger_files/gdbns9xxx.raven ./.gdbinit
```

and press Enter.

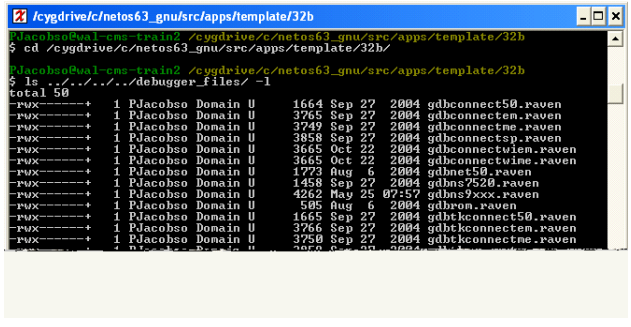
ARM7-based platforms. Copy `/cygdrive/c/netos63_gnu/debugger_files/gdb`*`myplatform`*`.raven` to the `32b` folder for your application, and rename it `.gdbinit`

where *`gdbmyplatform`* corresponds to the platform you are using. For example, for the ConnectSP, you would enter:

```
cp ./../../../../debugger_files/gdbconnectsp.raven ./.gdbinit
```

and press Enter.

You see a list of debugger scripts in the X-Tools shell:



Do not use the `gdbtk*` scripts.

5 Enter the default root password — `Netsilicon` — and then press Enter.

The first of a series of configuration prompts appears.

6 At each prompt, do one of these steps:

- To accept the current setting, press Enter.
- To change a setting, enter a value, and press Enter.

After you scroll through the settings, a prompt indicates that you must press a key within five seconds if you want to change additional settings.

If you don't press a key, after the five-second timeout, you see your selections in the **HyperTerminal** window.

What's next?

Go on to the next task to set up the GDB software debugger.

Task 4: Setting up the GDB software debugger

This task demonstrates how you can enable the GDB software debugger on your platform. The examples use the ConnectCore 9C, using serial port B ("/com/1") to transport the GDB protocol.

In this task, you'll:

- Update the BSP.
- Build the BSP.
- Download the GDB software debugger-enabled bootloader to the development board
- Update the application.
- Launch a TFTP server that contains the debug image.
- Execute the GDB-enabled bootloader.

Updating the BSP

To create a GDB-server-enabled bootloader and application, you need to update the platform BSP.

▶ **To update the BSP:**

1. Edit the `/cygdrive/c/netos63_gnu/src/bsp/platforms/connectcore9c_a/bsp.h` file by setting `BSP_GDB_SERVER_ENABLE` to `TRUE`.
2. Edit the `/cygdrive/c/netos63_gnu/src/bsp/platforms/connectcore9c_a/gdb.h` file by setting `NA_GDB_TFTP_SERVER` to the TFTP server address you got in Task 1.

▶ **To build the BSP:**

1. Double-click the **NET+Works 6.3 Build Environment** icon on your desktop.

   An X-Tools shell opens.
2. In the X-Tools shell, change the working directory to the top-level BSP directory by typing:

   ```
   cd /cygdrive/c/netos63_gnu/src/bsp
   ```

   and then press Enter.

3 In the X-Tools shell, build the ConnectCore 9C development board by entering this command:

```
make PLATFORM=connectcore9c_a clean all
```

and press Enter.

The build takes several minutes to complete; during this time, you don't see any activity on your screen.

Loading the GDB-server-enabled bootloader

The development board ships with a default application that contains an FTP server. You will use a remote FTP client to update the development board's original bootloader (`rom.bin`), using the development board's FTP server. The FTP server login is:

username: `root`

password: `Netsilicon`

▶ To load the GDB-server-enabled bootloader:

1 Power-cycle the development board.

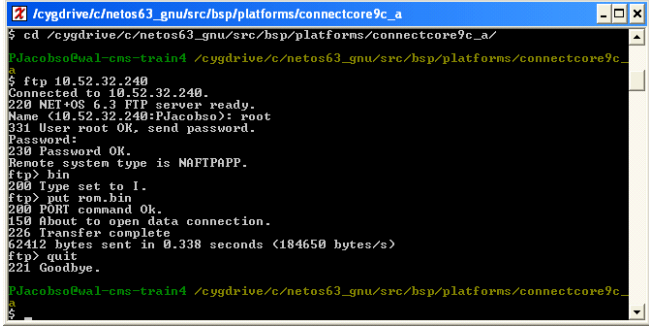
2 From an X-Tools shell, set the working directory to the BSP platform directory; for example:

```
cd /cygdrive/c/netos63_gnu/src/bsp/platforms/
connectcore9c_a
```

and then press Enter.

3 Using the X-Tools FTP client, download the platform's `rom.bin` bootloader file, in binary mode, to the development board.

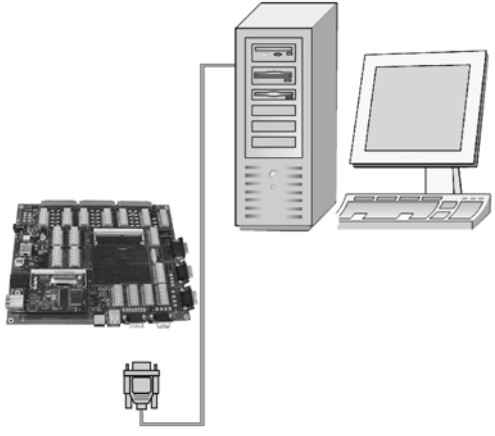
This is what you see during the download:



The `rom.bin` bootloader file contains the GDB software debugger.

Adding a second serial connection

Connect a second crossover cable from COM2 on your PC to the development board's serial port B, as shown here:



Updating the template application

Make these changes so the application can use the GDB software debugger. The template application is located in:

`/cygdrive/c/netos63_gnu/src/apps/template`

▶ **To update the template application:**

1. In the template application's `root.c` file, make these changes:
   - Add `naGdbStart("/com/1");` to the first call in the `applicationStart` function.
   - Add `#include "armgdb.h"` to the top of the file.
2. In the application's `Makefile`, add `libgdb.a` to the `APP_LIBS definition`.

3 In the X-Tools shell, change to the template build directory:

```
cd /cygdrive/c/netos63_gnu/src/apps/template/32b
```

and press Enter.

4 In the X-Tools shell, build the application:

```
make PLATFORM=connectcore9c_a clean all
```

and press Enter.

Launching the TFTP server

Start the TFTP server, and point it to the template application `image.bin` file located in this directory:

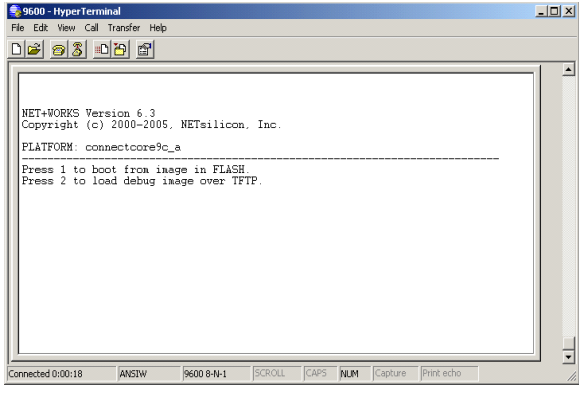
```
/cygdrive/c/netos63_gnu/src/apps/template/32b
```

Executing the GDB software debugger-enabled bootloader

▶ **To execute the GDB-server-enabled bootloader:**

1 Power-cycle the development board.

This menu appears on the standard serial out port:



This menu has a five-second timeout, after which it defaults to a boot from flash.

The development board contains a default FTP server image that it runs if you don't intervene within five seconds. Allow this timeout; the standard NET+Works dialog scrolls after a pause.

2 Power-cycle the development board again.

The menu from step 1 opens again.

3 Enter 2.

The menu continues.

4 At the `Enter TFTP server IP address` prompt, press Enter.

5 At the `Enter debug image filename` prompt, press Enter.

The GDB-server-enabled bootloader downloads the debug `image.bin` application and begins execution. You see the template application dialog in your **HyperTerminal** window.

The development board is now waiting for the GDB software debugger, which is at a break point at `naGdbStart` in the `root.c` file.

What's next?

Go to the next task to debug with the GDB software debugger.

Task 5: Debugging the template application

▶ **To debug the template application:**

1. Create the application's `.gdbinit` file in the application's build directory; for example:

   ```
   make PLATFORM=connectcore9c_a gdbdebuginit
   ```

   and then press Enter.

   The `gdbinit` file sets up the PC to communicate with the GDB software debugger on `/dev/com1` at 115200 baud rate.

2. In the `.gdbinit` file, change the last line from `/dev/com1` to `/dev/com2`.

3. To start the debugger, enter:

   ```
   gdbtk -se image.elf
   ```

   and press Enter.

   As soon as you start GDB, it is idling at the `naGdbStart` break point.

4. To continue to run the template application in the console window, enter `c`.

What's next?

You have now completed all the tasks in this chapter.

You may, however, want to see Chapter 3 for some tips and suggestions.

Chapter 3: Tips and Suggestions

Now that you've completed the exercise, here are some tips for when you start using NET+Works with GNU Tools with your own projects.

Where should I put my code, and why?

Add your code as a subdirectory of the `netos63_gnu/src/` examples directory. The software calls the `applicationStart` function in the `root.c` file.

Start by duplicating some other example and modify the `Makefiles`. (For information about modifying `Makefiles`, see the *NET+OS with GNU Tools BSP Porting Guide*).

A good choice is `naftpapp`, the FTP server example. Add your application to this example, which allows you reload new code after it's running in flash. Without the FTP server in your application, you can't re-flash the system.

Modify the `root.c` file to load your specific application, and modify the `appconf.h` file for your board settings.

What should my next step be?

The next step should be running your application from flash. The flash code is broken up into two parts: the bootloader and your application.

The bootloader (`rom.bin`) is located in the `netos63_gnu/src/bsp/platforms/`*`your platform`* directory.

Your application (`image.bin`) is located in the `netos63_gnu/src/examples/`*`your example`*`/32b` directory.

Appendix: Updating the MAJIC Probe's Firmware

This appendix describes how to update the firmware for your MAJIC probe using the MAJIC Setup Wizard.

You must do this procedure if you have a previous version of the MAJIC firmware.

If you need to set up the IP address of the MAJIC probe, see Task 3.

▶ **To update the MAJIC's firmware:**

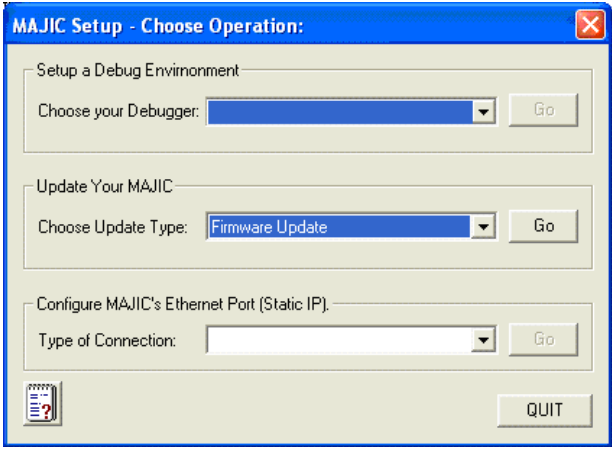
1 Start the MAJIC Setup Wizard by doing one of these steps:

- **Windows XP systems**. Select **Start** → **All Programs** → **EPI Tools-EDTA 2.2a** → **MAJIC Setup Wizard**.
- **Windows 2000 systems**. Select **Start** → **Programs** → **EPI Tools-EDTA 2.2a** → **MAJIC Setup Wizard**.

The **EPI MAJIC Setup Wizard Introduction** window opens:

2 After you review the introduction, click **NEXT**.

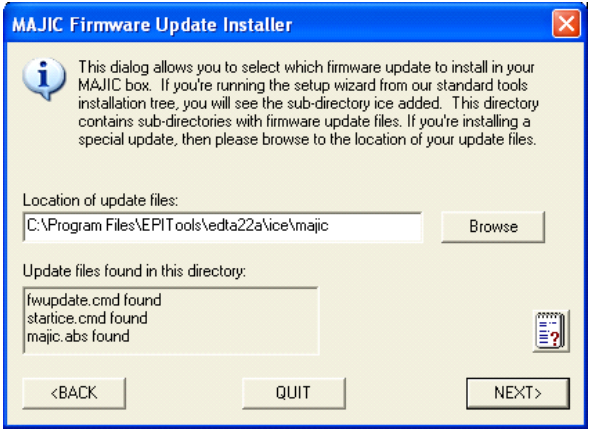
The **Choose Operation** window opens:



3 From the **Choose Update Type** pulldown menu, select **Firmware update.**

Then click **Go.**

The **MAJIC Firmware Update Installer** dialog box opens:

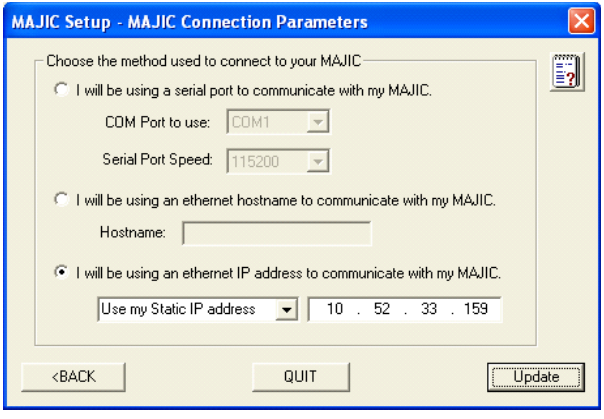


4 If the directory name is not already in the **Location of update files** input box, browse to this directory:

`C:\Program Files\EPITools\edta22A\ice\majic`

and click **NEXT.**

The **MAJIC Connection Parameters** dialog box opens:



5 Do these steps:

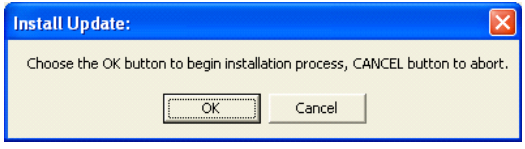
a Click **I will be using an Ethernet IP address to communicate with my MAJIC.**

b Enter the IP address for the MAJIC.

c Make sure **Use My Static IP address** is selected.

Then click **Update.**

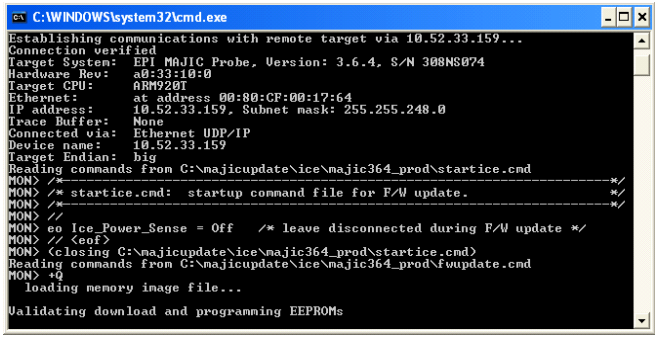
The **Install Update** dialog box opens:



6 To begin the update, click **OK**.

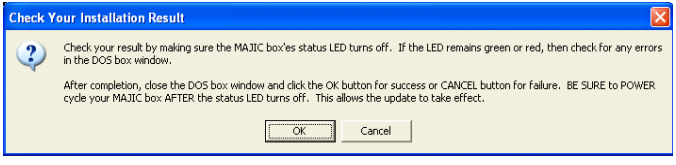
A DOS window and the **Check Your Installation Result** dialog box open.

The firmware download takes place in the DOS window:



7 When the download finishes, do one of these steps:

- If the download was successful, click **OK** in the **Check Your Installation Result** dialog box.
- If the download was not successful, click **Cancel** in the **Check Your Installation Result** dialog box, and repeat this procedure.



Index

A

access rights, verifying 2, 29
appconf.h file 41
applicationStart function 41
ARM MAJIC .gdbinit file 19

B

bootloader 41
BSP, building 17, 34
building the BSP 17, 34
building the sample application 17
building the template application 19

C

configuring
    the development board's IP address 32
Creating an MDI server icon 10

D

debugging the template application 24
default gateway 2
default installation directory ix
default root password 16
development board IP parameters 2
development board, configuring IP address of 15, 32
downloading the template application 21

E

EPI MAJIC Setup Wizard 6, 10, 43

executing GDB software debugger-enabled bootloader 37

G

gateway, default 2

GDB software debugger

setting up 34

GDB, defined 10

H

hardware requirements 3

hardware requirements, verifying 29

HyperTerminal 23, 38

I

image.elf file 20

installation directory, default ix

installing NET+Works 4

IP addresses

development board 15

MAJIC probe 6

IP parameters for development board and MAJIC 2, 28

L

launching the TFTP server 37

M

MAJIC probe IP parameters 2

MAJIC Setup Wizard 43

Makefiles 41

MDI server icon, creating 10

N

naftpapp application 41

NET+Works, installling 4

P

ping command 9

R
rom.bin file 20
root password 16
root.c file 38, 41
running your application from flash 41

S
second serial connection, adding 36
subnet mask 2

T
template application
    debugging 24, 39
    downloading 21
template application, building 19
template application, updating 36
TFTP server, launching 37

Digi®