KiraTool v1.5.14
User Manual

Contents

Figures

Tables

Preface

Copyright

Raritan Deutschland GmbH

Scheringerstr. 1

08056 Zwickau

Germany

Document Version and Date

Version: 1.5.14

Date: Tuesday, May 13th, 2008

Trademarks

Raritan acknowledges the following trademarks:

- Intel is a registered trademark of Intel Corporation.
- Windows 98, Microsoft Windows, Windows NT, Windows 2000, Windows XP and Windows Vista are trademarks of Microsoft Corporation.
- IBM, AT, VGA, PS/2, and OS/2 are registered trademarks and XT and CGA are trademarks of International Business Machines Corporation.
- Raritan is a registered trademark of Raritan Incorporated.

Other trademarks and trade names may be used in this document to refer to either the entities claiming the marks and names or their products. Raritan disclaims any proprietary interest in trademarks and trade names other than its own.

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/).

Authors: Raritan Team

About the KiraTool

The KiraTool software provides a command line based tool to probe, manage and configure Raritan's Remote Management Modules (modules). As the tool is text based it is ideal to be used in batch mode command scripts. This makes it a lot easier to configure and control several modules at once.

Limited Warranty

The buyer agrees that if this product proves to be defective, Raritan is only obligated to repair or replace this product at Raritan's discretion according to the terms and conditions of Raritan's general trading conditions.

Raritan shall not be held liable for any loss, expenses or damage, directly, incidentally or consequentially resulting from the use of this product. Please see the Warranty Information shipped with this product for full warranty details.

Limitations of Liability

Raritan shall in no event be held liable for any loss, expenses or damages of any kind whatsoever, whether direct, indirect, incidental, or consequential (whether arising from the design or use of this product or the support materials provided with the product). No action or proceeding against Raritan may be commenced more than two years after the delivery of the product to the buyer.

The licensee agrees to defend and indemnify Raritan from any and all claims, suits, and liabilities (including attorney's fees) arising out of or resulting from any actual or alleged act or omission on the part of Licensee, its authorized third parties, employees, or agents, in connection with the distribution of Licensed Software to end-users, including, without limitation, claims, suits, and liability for bodily or other injuries to end-users resulting from use of Licensee's product not caused solely by faults in Licensed Software as provided by Raritan to Licensee.

Technical Support

If you need help installing, configuring, or running the module, call your Raritan Technical Support representative.

We invite you to access Raritan's Web site (www.raritan.com) where you shall find all modifications made after the editorial deadline. You may also contact us via e-mail to support@raritan.com

Chapter 1: KiraTool Quick Start Guide

About the KiraTool Software

The Raritan KiraTool is a program which allows you to manage the Raritan Remote Management Modules (modules). The KiraTool (KT) does not use a graphical user interface, but a text based one. It is very useful if you cannot use a web browser to get access to your module or if the module needs to be configured quickly.

Therefore you can easily use KT in your command scripts and batch files to manage several modules using automated methods.

KT is available for Windows and DOS, EFI (Extensible Firmware Interface), as well as Linux.

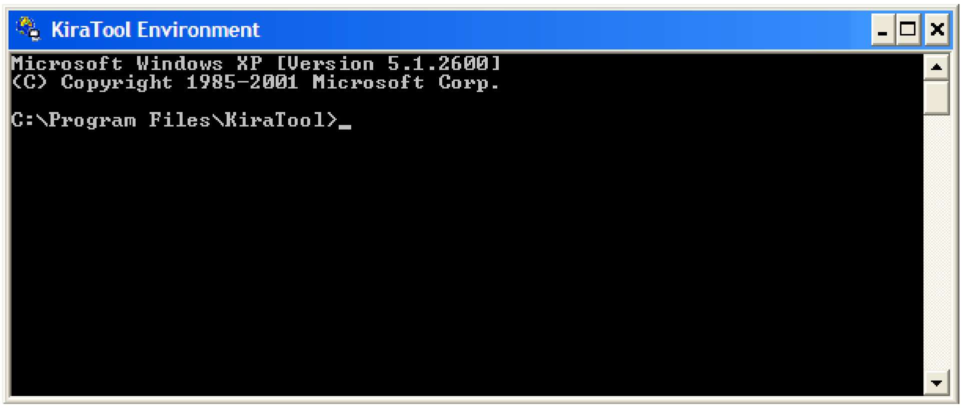
How to Install the KiraTool

Installing the KT is fairly easy and intuitive.

- In the case of the DOS version you will simply receive an executable file. Just copy it to your hard disk and invoke it from there. This DOS-version is provided by Raritan free of charge. The other versions are only available for a licensing fee.
- In case of the EFI version you will also receive a binary file.
- The Windows version (for Windows 2000, 2003, XP or newer) is a normal Windows executable file which runs a traditional Windows installer. Simply follow the installation screens suggestions and the KT is installed.
- The Linux version comes as a binary executable file. In order to install the KT, copy the executable into a binary directory in your $PATH. Usually this is something like */usr/local/bin*. Make sure that the KT binary is executable for everyone. You will usually need to be *root* in order to copy the file into this system directory.

Once you have installed the KT properly, you will find (in the Windows version) an icon on your desktop called “KiraTool Environment”. Double-clicking this icon will open a Windows dialogue box in the proper KT directory.

Figure 1 - 1. Windows KiraTool Environment



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Program Files\KiraTool>_
```

Using the KiraTool

The KT has plenty of options and commands. Most of these are quite intuitive and are listed in **Chapter 3. KiraTool Commands in Detail**.

The general syntax for the KT is as follows:

```
kiratool [option] [command]
```

__Note:__ single-letter options are preceded with a dash or hyphen such: __-s__, commands are several characters long and do not have a preceding hyphen, for example: __reset__.

For example you can ask KT to give you its version number:

```
win> kiratool ver
        KiraTool 1.5.14
Copyright (C) 2008, Raritan Inc.
```

If you want to get a quick overview of the KT functions simply consult the listing in **Appendix A. KiraTool Commands**.

Accessing the KiraTool

Depending on which OS will be used the module provides different ways of accessing:

Via LAN:
will be available under the Windows and Linux version of the KiraTool

Via SCSI (over USB)
will be available under the Windows, Linux and EFI version

Via SMI (System Management Interface)
the DOS and the Linux version allow direct access.

> *__Note:__ The execeptional case is the usage of the KiraTool for ASMI under Linux. Therefore SCSI (over USB) is used by default and no SMI available.*

Chapter 2. Using the KiraTool

Basic Usage and Background

In order to properly use the KT you have to know several things:

- How do you wish to connect to the module? Via SCSI/USB, SMI or LAN?
- Administrator login name and password
- What you wish to achieve

KT Options and Commands

KT's behavior is governed by Options and Commands. Options allow you to change the basic parameters on which KT operates, they steer the behavior of KT itself. Commands are based on the options and are directed at the module.

Needless to say there are some exceptions to the rule, just enough to confuse you. First we discuss the options, then the commands. The commands are then described in more detail in the following **Chapter 3. KiraTool Commands in Detail**.

Options for the Connection Type

This option sets the type of physical connection you wish to use in order to talk to the module. You either want to use a direct hardware connection using the USB bus – you then can access the module directly. The other method is via LAN using Ethernet. However the LAN connection usually only works, when the modules have received an IP address.

KT allows you to specify the following connection modes:

- ***-l <IP Address>*** – use the specified IP address to talk to the module over LAN.
- ***-s*** – use IPMI over SCSI/USB – this can be augmented with the following
- ***-d device*** option: here you can specify the drive identification for SCSI acces of the module. For the Windows version: the drive letter, e.g. **`F:`** and for the Linux version **`/dev/sg1.`** If you omit this option KT will attempt to autodetect the module by probing the SCSI drive identification.
- The EFI version of the KT allows direct module access using SCSI.

Note: *If you do not specify a connection type, the default is used. In the DOS version that is the SMI, in EFI and Windows it is the SCSI (over USB). Under Linux version it depends on which module is used, if you use a ASMI module the default will be SCSI (over USB) and otherwise SMI.*

Authentication Options

In order to execute administrative functions on the module you need to authenticate yourself. All Raritan products come with a pre-configured administrator login with a factory default password. This authentication set varies from product to product – it also is different if you are using an OEM version of our products.

Note: *you can glean the factory default authentication settings for your module from its own User Manual.*

For most KT commands you have to specify the administrative login and password to the module using the following options:

- **-u** – the admin user
- **-p** – the admin password

As in

```
linux# kiratool –u admin –p password
```

Important: if you use KT from a batch or script file, you will almost certainly enter these passwords in clear text in the file! This is a potential security problem: anyone who can read your command file can attain administrative access to your modules and is able to reconfigure them or take them down. This can have a serious impact on your servers or your network. MAKE SURE YOU ADEQUATELY PROTECT THESE FILES FROM UNAUTHORISED ACCESS!

In order to "somewhat" reduce the risk of the clear text passwords in such files, KT offers an option:

- **-P** – prompt for admin password

You would then invoke KT as follows:

```
linux# kiratool –u admin –P
Password:
```

Note: *when you type the password, your characters will not be echoed: they do not appear as you type.*

Those amongst the readers of this manual who administer MySQL databases know this behavior from the *mysql* or *mysqladmin* programs.

Other Options

There are still some other options, which do not exactly fit the normal option category. Basically the cause KT to take an action and are dealt with by KT and not the module. This is the confusion mentioned above.

Other options are:

- **`-f`** – force. This will cause a command to the module to be executed WITHOUT any user confirmation.
- **`-a`** – use ASMI mode – needed if you want to access an ASMI module
- **`-v`** – verbose. This causes KT to be more informative about the actions taken. The output (like all output of KT) will go the *stdout*. You can use this option more than once, each use increases the level of verbosity.
- **`-c`** – calm. This option is like the –q (quiet) option of other programs: KT will not generate any output.
- **`-h/-?`** – This (like failure) is not an option, but rather it causes KT to print out online help. Strictly speaking it is not an option but a command.

KiraTool Actions

The commands you can give KT as a parameter allow you to extensively probe and configure your module. They fall into several categories:

`ver`	Show program version and information
`info`	Show information about the BMC
`serial`	Serial number operations
`reset`	Reset the device
`defaults`	Reset device to factory settings
`cfg`	Backup or restore device configuration
`raw`	Execute raw commands
`admin`	Show or set admin name and password
`mac`	Read or set MAC address
`ip`	Read or set IP address
`netmask`	Read or set subnet mask
`gw`	Read or set default gateway address
`ipsrc`	Get or specify configuration for the IP address
`fw`	Firmware operations
`fni`	IPMI over FML forwarding commands
`test`	Execute some self tests

Chapter 3. KiraTool Commands in Detail

This chapter gives you a more detailed description of the KT commands and their output.

Note: *the examples given are for an OPMA device made by Raritan. The output may differ if you access an ASMI device OR if you are using an OEM version of our products.*

General Commands

ver(sion)

The **`ver`** command shows the version of the KT itself:

```
C:\Program Files\KiraTool>kiratool ver
        KiraTool 1.5.14
Copyright (C) 2008, Raritan Inc.
```

info(rmation)

The **`info`** command shows basic information (manufacturer identification and product ID) of the module. The example given also shows the use of the `–l`, `–u` and `–p` options:

```
C:\Program Files\KiraTool>kiratool -l 192.168.2.6 -u super -p !beagle info
Manufacturer ID: 10437 (0x28c5)
Product ID:      0 (0x0)
```

serial [show]

The **`serial`** command displays the serial number of the module. Serial numbers are not only strictly numbers but alpha-numeric strings.

```
C:\Program Files\KiraTool>kiratool -l 192.168.2.5 -u snoopy -p !beagle serial
Serial number: 007-BOND
```

reset

The **`reset`** command resets the device. This means for example that the module may get a new IP address in case you have configured it to get an IP address via DHCP. In this case you have to find out the new IP address.

```
C:\Program Files\KiraTool>kiratool -l 192.168.2.6 -u super -p !beagle reset
Resetting device. The device might not respond for about one minute.
Successfully reset the device.
```

defaults

The **`defaults`** command will reset the device to factory defaults.

Note: *this will also reset the administrative password and any other commands may fail, once you have reset to defaults unless you change the login back to what you and your scripts expect!*

```
C:\Program Files\KiraTool>kiratool –l 192.168.2.6 -u super –p !beagle defaults
```

```
Successfully reset device to factory settings.
```

cfg backup <filename>

The **`cfg backup`** command will backup the device's configuration to a file.

cfg restore <filename>

The **`cfg restore`** command will restore the device's configuration from a file.

raw

The **`raw`** command allows you to execute very basic commands on the module.

These command codes are specific to your module and depend heavily on the version, an OEM product etc. Hence this example here is only an academic example. Normal users of the KT will not need raw commands.

Important: They are intended for advanced development and debugging use only.

```
C:\Program Files\KiraTool>kiratool -l 192.168.2.6 -u snoopy -p !beagle raw 06 01
```

```
Executed raw command.
    Return code:     0x00
    Returned bytes:  0x20 0x01 0x04 0x02 0x02 0x8f 0xc5 0x28
                     0x00 0x02 0x00 0x00 0x00 0x53 0x59
```

User Administration

The following commands allow you to manage the administrative login for the module.

admin [show]

The **`admin`** command shows the current setting of the admin account. The `show` verb is optional. This is kind of redundant, as you have to know the admin login in order to enquire it.

```
C:\Program   Files\KiraTool>kiratool   -l   192.168.2.6   -u   super
-p !beagle admin
```

```
Administrator username: super
```

admin name

Set the admin users name:

```
C:\Program Files\KiraTool>kiratool -l 192.168.2.6 -u super -p pass
admin name snoopy
```

```
Successfully set administrator username to snoopy
```

admin password

Set the password for the new and improved admin password:

```
C:\Program Files\KiraTool>kiratool -l 192.168.2.6 -u snoopy -p pass
admin passwd beagle
```

```
Successfully set administrator password.
```

Network Interface Commands

The following commands allow you to set the parameters for the modules LAN interface. They are quite straightforward. You can basically inspect and manipulate any parameter: MAC address, IP address, gateway address, netmask as well as the network interface configuration method.

Important: when you change these parameters you can VERY EASILY make the module unreachable on the network. Especially changing the MAC or IP address will cause problems with your ARP caching and the DHCP server access information. Normally you should not encounter a real need to change these addresses.

mac [show [-c]]

This command shows the module’s Ethernet or MAC address:

```
C:\Program   Files\KiraTool>kiratool   -l   192.168.2.6   -u   snoopy
-p beagle mac
```

```
MAC address: fe:00:00:51:00:38
```

The optional –c option to this command displays the MAC address in a compact format.

```
C:\Program   Files\KiraTool>kiratool   -l   192.168.2.6   -u   snoopy
-p beagle mac show -c
```

```
MAC address: FE0000510038
```

mac set <mac addres>

This command allows you to set the MAC address of the module. Of course you can also use the below expanded ":" notation for the MAC address.

```
C:\Program  Files\KiraTool>kiratool  -l  192.168.2.6  -u  snoopy
-p beagle mac set FE0000510200
```

```
Successfully set MAC address to FE0000510200
```

or

```
C:\Program  Files\KiraTool>kiratool  -l  192.168.2.6  -u  snoopy
-p beagle mac set fe:00:00:51:02:00
```

```
Successfully set MAC address to fe:00:00:51:02:00
```

ip [show]

Shows currently configured IP address:

```
C:\Program  Files\KiraTool>kiratool  -l  192.168.2.6  -u  snoopy
-p beagle ip
```

```
IP address: 192.168.2.6
```

ip set <ip address>

The module's IP address can be set manually, use this method if you want to configure the module IP address using the "static" method, see the "ipsrc" command below:

or the module can get an IP address via DHCP or the BIOS of the host.

```
C:\Program  Files\KiraTool>kiratool  -l  192.168.2.6  -u  snoopy
-p beagle ip set 192.168.2.5
```

```
Successfully set IP address to 192.168.2.5
```

netmask [show]

Display the netmask currently used by the module:

```
C:\Program  Files\KiraTool>kiratool  -l  192.168.2.6  -u  snoopy
-p beagle netmask
```

```
Subnet mask: 255.255.255.0
```

netmask set <netmask>

You can set the netmask using the normal IP dot notation. Please take note that changing the netmask can change the behavior of the module with regards to broadcasting. If you "widen" the netmask then broadcasts by the module can use more network bandwidth.

```
C:\Program  Files\KiraTool>kiratool  -l  192.168.2.6  -u  snoopy
-p beagle netmask set 255.255.0.0
```

```
Successfully set Subnet mask to 255.255.0.0
```

gw [show]

This shows the currently used default routing gateway for the module:

```
C:\Program  Files\KiraTool>kiratool  -l  192.168.2.6  -u  snoopy
-p beagle gw
```

```
Default gateway: 192.168.2.1
```

gw set <ip address>

This will set a new default routing gateway:

```
C:\Program  Files\KiraTool>kiratool  -l  192.168.2.6  -u  snoopy
-p beagle gw set 192.168.2.3
```

```
Successfully set Default gateway to 192.168.2.3
```

ipsrc [show]

This command is very useful, it allows you to inspect which method the module uses in order to get its IP address:

```
C:\Program  Files\KiraTool>kiratool  -l  192.168.2.6  -u  snoopy
-p beagle ipsrc
```

```
IP source: DHCP Address
```

ipsrc set [static|dhcp|bios|none]

The four methods available work as follows:

- ***dhcp*** allows the module to get the IP configuration from the locally resident DHCP server. This should be in the same broadcast domain, as the module, otherwise the DHCP lookup will not work. DHCP also sets other basic information like the netmask, the IP address and the gateway address.
- ***static*** allows only static setting of the module's IP address.
- **none** means unspecified.
- ***bios*** is the method, where the module will look into the BIOS of the host in order to find the IP address.

Example:

```
C:\Program  Files\KiraTool>kiratool  -l  192.168.2.6  -u  snoopy
-p beagle ipsrc set static
```

```
Successfully set IP source to static
```

Firmware Commands

The KT also allows you to manipulate the module's firmware.

fw [ver]

Shows the version of the firmware

```
C:\Program Files\KiraTool>kiratool -l 192.168.2.5 -u snoopy
-p beagle fw

Firmware version: 4.2.0

Build number:     5359

Hardware ID:      0x20

Firmware tag:     Beta

OEM:              raritan
```

fw validate

This command allows you to check if the firmware binary file is compatible with your module. It is a very good idea to check this BEFORE you attempt to upgrade the module's firmware!

In order to perform the check, you need to know the exact name and location on your hard drive of the firmware binary.

```
C:\Program Files\KiraTool>kiratool -l 192.168.2.6 -u snoopy
-p beagle fw validate

F:\fw-kira-kimamdg4-amddc-peppercon_040200-5359.bin

Starting Firmware Validation

Uploading Firmware File

0% ------------------- 50% ------------------ 100%

**************************************************

Upload complete.

Validating Firmware

Firmware file is valid.
```

fw upgrade

Obviously this is the upgrade command corresponding to the above validation. Please note that it is quite possible to "upgrade" the firmware with one of the same version. This is often useful to re-install the firmware.

```
C:\Program Files\KiraTool>kiratool -l 192.168.2.6 -u snoopy
-p beagle fw upgrade

F:\fw-kira-kimamdg4-amddc-peppercon_040200-5359.bin

Starting Firmware Upgrade

Uploading Firmware File
```

```
0% ------------------ 50% ----------------- 100%
**************************************************
Upload complete.
Flashing Firmware (takes about 1min)
Successfully upgraded firmware.
```

Test Commands

You can use the test command to perform several self-tests on the module. You can specify to test “everything” or you can skip certain tests. Depending on your module, some tests may simply not be available, because your module might not have the appropriate function.

test <test>

Execute the test labeled ***test*** on the module. For example the most basic test is the `device` test, it checks if the module is responding.

```
C:\Program  Files\KiraTool>kiratool  -l  192.168.2.5  -u  snoopy  -p
beagle test device
device: ok (firmware 4.2.0, Build 5359)
```

test all [-s test_to_skip]

This executes all of the tests. It includes the following subtests provided they are available on your module in the following order:

1. **device** – is the device available
2. ***ddc info*** – the DDC interface
3. ***video status*** – the status of the video inputs and outputs. May fail if there is no remote video connected.
4. ***ipmb bmc*** – the Board Management Controller (if available)
5. ***fml esb2*** – tests FML interface and the ESB2
6. ***usb status*** – the status of the USB interface (for SCSI over USB)
7. ***nic status*** – the status of the network interface (LAN)

The example below shows you the output of the test all command on my test module (an OPMA module with no remote connections):

```
C:\Program   Files\KiraTool>kiratool   -l   192.168.2.5   -u   snoopy
-p beagle test all
device: ok (firmware 4.2.0, Build 5359)
Could not query DDC from device: Self test not supported
ddc info: error
```

```
video status: failed
IPMB BMC status test failed: Self test not supported
ipmb bmc: error
FML ESB2 status test failed: Self test not supported
fml esb2: error
usb status: failed
nic status: ok (link: up, duplex: full, speed: 100 MBit)
```

Test Types

Here is a more detailed listing of the available tests.

Note: *the tests are organized in a hierarchical fashion. If you exclude a top-level test like* **`–s nic`** *from the testing then ALL of the* **`nic`** *tests will be skipped! Conversely if you specify a top-level test to be done, then also all of the available sub-tests will be executed.*

- **`video <subtest>`** - tests video interface (digital video input and output)
  - *`status`* – checks detected video signal and resolution
  - *`crc`* – calculate CRC sum over the captured screen
- **`ddc <subtest>`** – test DDC interface
  - *`info`* – queries EDID information from the device and compares it to the EDID information known by the OS (only available under Windows)
- **`ipmb <subtest>`** – test IPMB interface
  - *`bmc`* – test whether a BMC responds over IPMB
  - *`evalboard`* – test whether IPMB connection between two evaluation board work
- **`fml <subtest>`** – test FML interface
  - *`esb2`* – test whether an ESB2 is responding on FML when TPT (TCP PassThrough) is active
  - *`evalboard`* – test whether FML connection between two evaluation board work
- **`usb <subtest>`** – test USB interface
  - *`status`* – test whether the device's USB modul is enumerated
- **`nic <subtest>`** – tests network interface
  - *`status`* – test NIC status and parameters
  - *`loopback`* – test NIC loopback functionality
  - *`ping <host>`* – Test whether pinging a host works

Test Return Codes

All of the above tests return an error code if they fail and a zero (0) code when they succeed:

- 0 – zero is returned if ALL of the specifed tests executed successfully
- -1 (minus one) is returned when an error occurs (except for the test command itself). Be careful: in some operating systems this is converted to 127 or another value. Be sure to check carefully!
- Other values are returned when a specific test produces an error. See table below.

Note: *if the* **`test all`** *command fails, then the returned error code is that of the first failed test. Testing will continue even if errors are encountered in previous tests.*

Table 3 - 1. Return Codes Overview

Test	**Return Code**
device	1
video status	2
video crc	3
ddc info	4
ipmb ddc	5
fml esb2	6
usb status	7
nic status	8
nic loopback	9
nic ping	10
nic broadcast	11
fml evalboard	12
ipmb evalboard	13

Chapter 4. KiraTool Installation Guide

This chapter explains the different installation methods of how to install the KT under different operating systems.

Installing the KiraTool under different OS

Windows Version

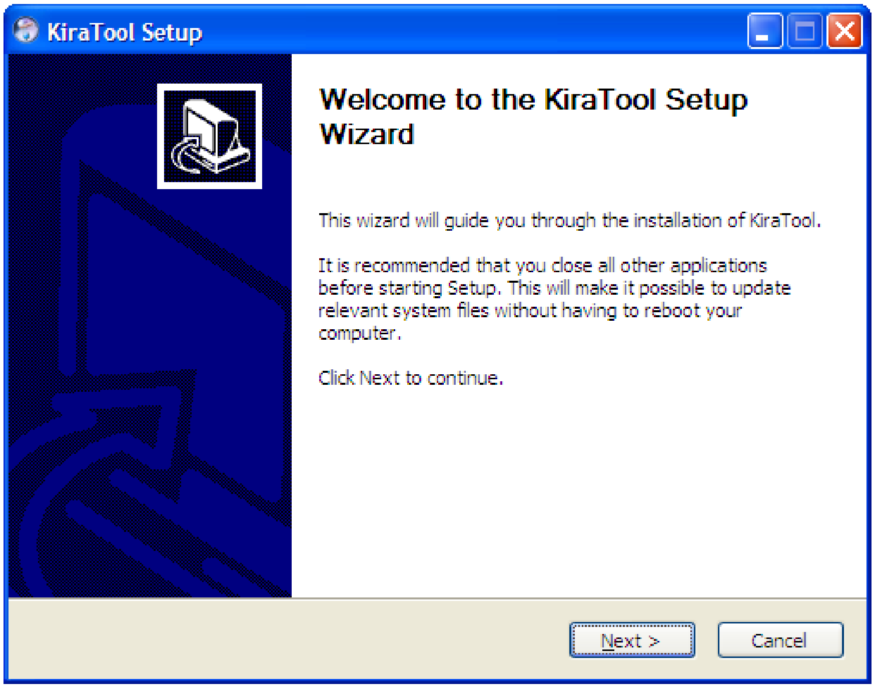
We discuss the Windows installation first, as this is the one most customers are likely to use. We recommend that you only install the Windows version on a modern Windows operating system, i.e. Windows 2000, Windows XP or higher.

When you receive KT for Windows you will find a self-extracting executable file: “KiraTool 1.5.10.EXE”

Invoke the file by double-clicking it with your left mouse-button.

You will be greeted by the following installer screen:

Figure 4 - 1. KiraTool Setup Welcome Screen



Welcome to the KiraTool Setup Wizard

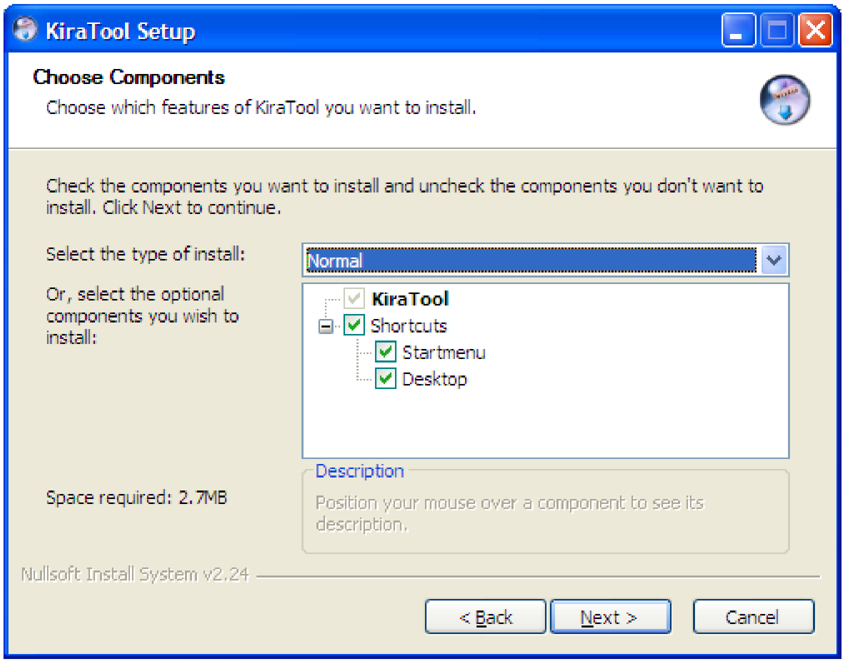
This wizard will guide you through the installation of KiraTool.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

Click **NEXT** to continue:

Figure 4 - 2. KiraTool Setup Choose Components Screen



KiraTool Setup

Choose Components

Choose which features of KiraTool you want to install.

Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

Select the type of install: Normal

Or, select the optional components you wish to install:

- ☑ KiraTool
- ☑ Shortcuts
  - ☑ Startmenu
  - ☑ Desktop

Space required: 2.7MB

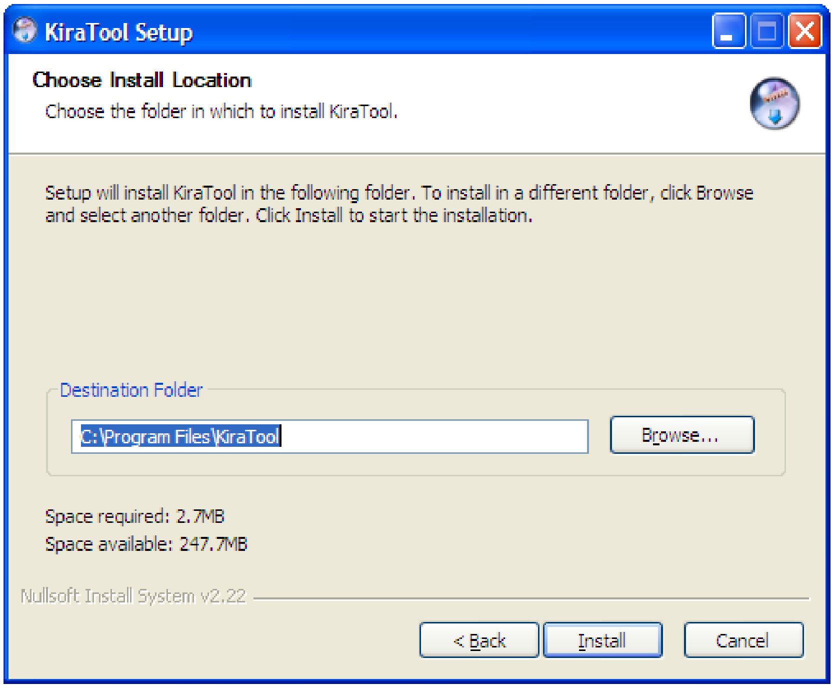
Description

Position your mouse over a component to see its description.

The above installer screen allows you to select the components to install. Generally you should accept the defaults.

Once you have selected **NEXT**, the installation will ask you for the location to install the KT to, generally this will be in the suggested default location of C:\Program Files:

Figure 4 - 3. KiraTool Setup Install Location Screen



KiraTool Setup

Choose Install Location

Choose the folder in which to install KiraTool.

Setup will install KiraTool in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.

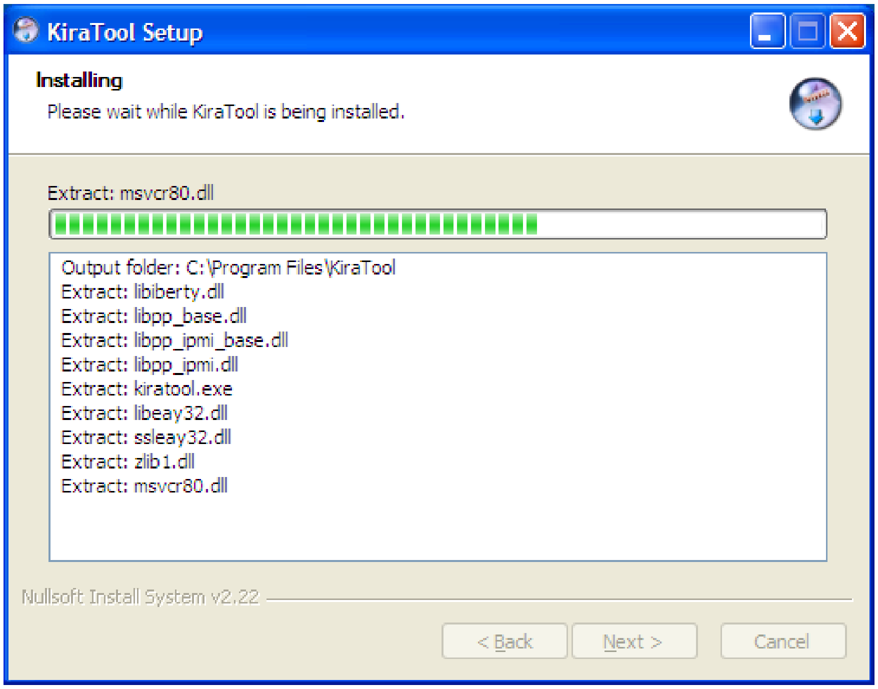
Destination Folder: C:\Program Files\KiraTool

Space required: 2.7MB

Space available: 247.7MB

When you accept the suggestion, the installer will proceed to extract and copy the files:

Figure 4 - 4. KiraTool Setup Installing Screen



Installing

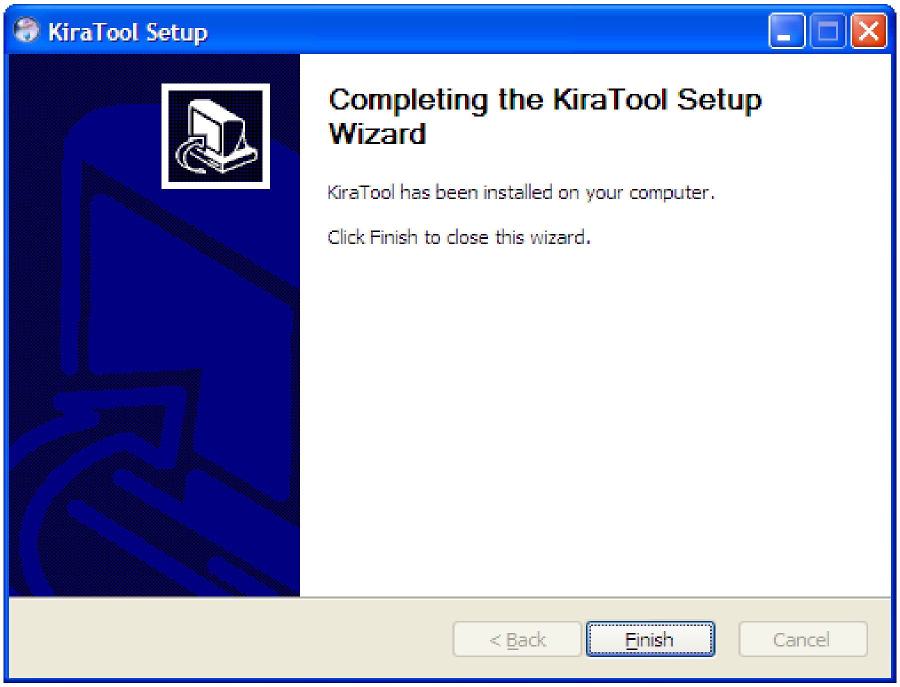
Please wait while KiraTool is being installed.

Extract: msvcr80.dll

```
Output folder: C:\Program Files\KiraTool
Extract: libiberty.dll
Extract: libpp_base.dll
Extract: libpp_ipmi_base.dll
Extract: libpp_ipmi.dll
Extract: kiratool.exe
Extract: libeay32.dll
Extract: ssleay32.dll
Extract: zlib1.dll
Extract: msvcr80.dll
```

The final installer screen confirms successful installation:

Figure 4 - 5. KiraTool Setup Finished Screen



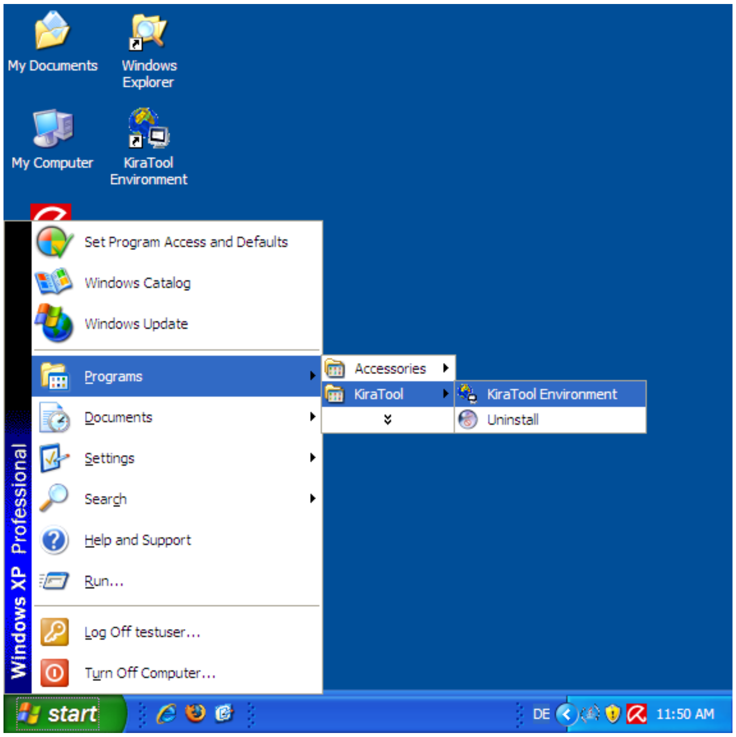
Completing the KiraTool Setup Wizard

KiraTool has been installed on your computer.

Click Finish to close this wizard.

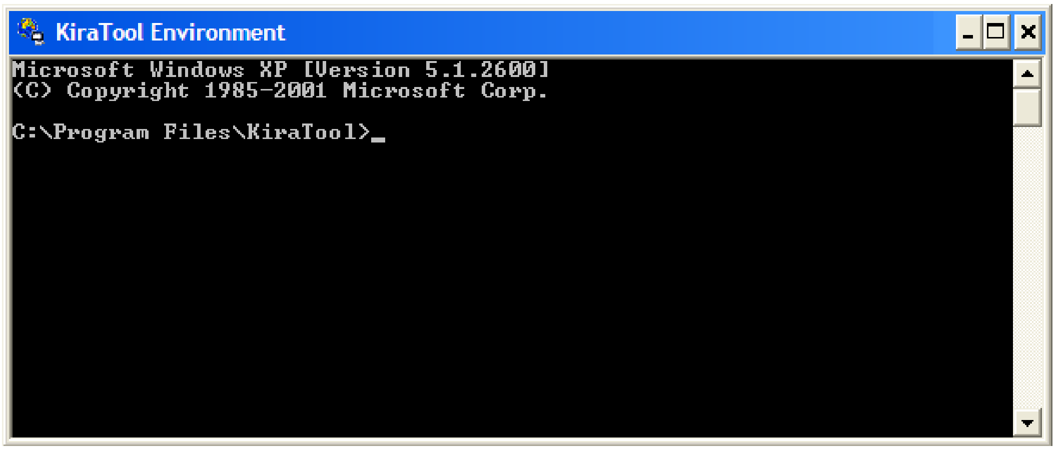
Once you have selected **FINISH** you will find a “KiraTool Environment” shortcut on your desktop. The picture below shows the shortcut (in the background) and the fact that you can also invoke KT from the Windows XP Start Menu:

Figure 4 - 6. Start the KiraTool under Windows XP



When you start the “KiraTool Environment”, you will find yourself in a normal Windows Command Line box in the correct directory to start the KT:

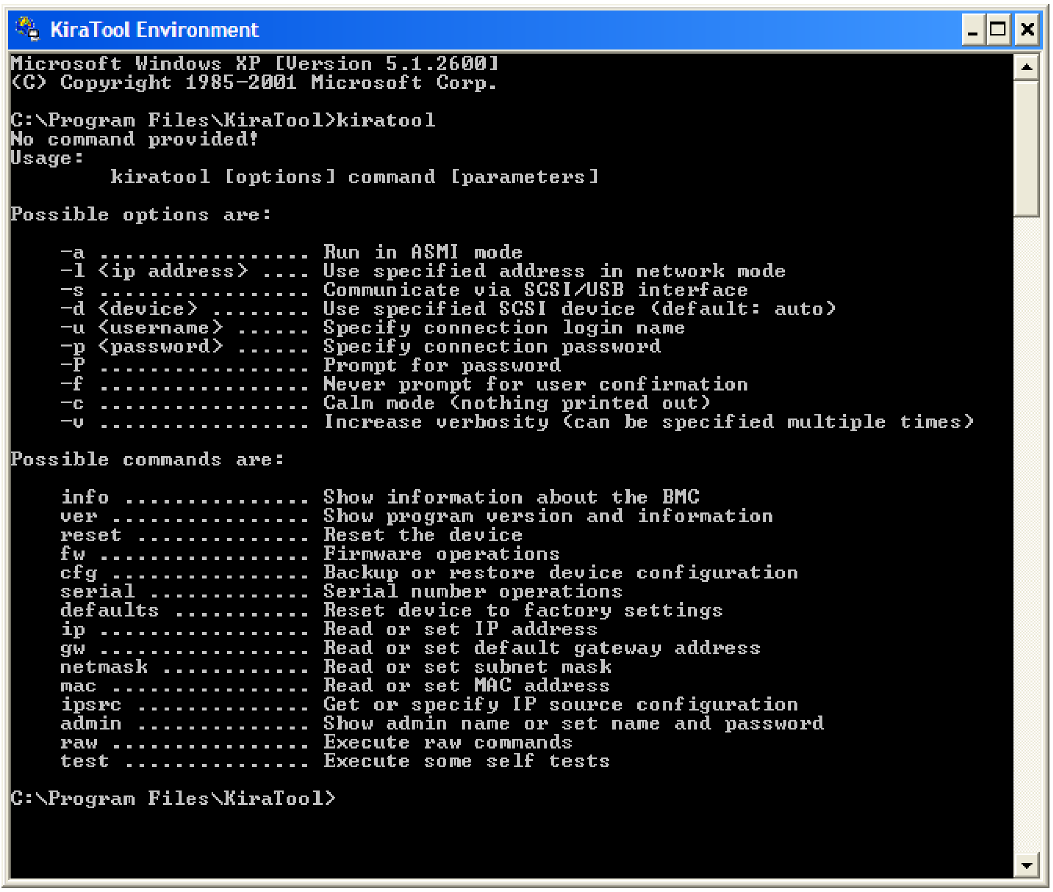
Figure 4 - 7. Starting the KiraTool under Windows XP



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Program Files\KiraTool>_
```

This allows you to start the actual KT:

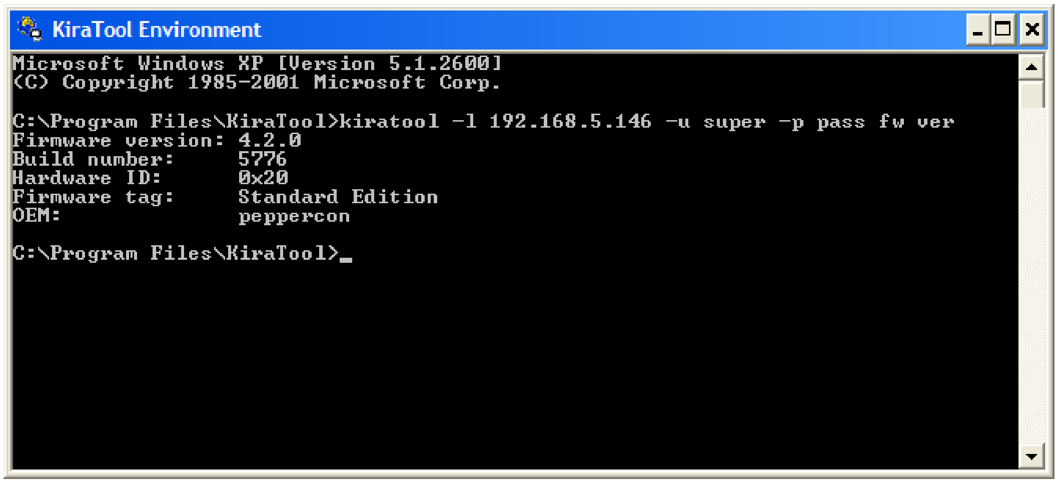
Figure 4 - 8. Running the KiraTool Online Help



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Program Files\KiraTool>kiratool
No command provided!
Usage:
        kiratool [options] command [parameters]

Possible options are:

    -a ................. Run in ASMI mode
    -l <ip address> .... Use specified address in network mode
    -s ................. Communicate via SCSI/USB interface
    -d <device> ........ Use specified SCSI device (default: auto)
    -u <username> ...... Specify connection login name
    -p <password> ...... Specify connection password
    -P ................. Prompt for password
    -f ................. Never prompt for user confirmation
    -c ................. Calm mode (nothing printed out)
    -v ................. Increase verbosity (can be specified multiple times)

Possible commands are:

    info ............... Show information about the BMC
    ver ................ Show program version and information
    reset .............. Reset the device
    fw ................. Firmware operations
    cfg ................ Backup or restore device configuration
    serial ............. Serial number operations
    defaults ........... Reset device to factory settings
    ip ................. Read or set IP address
    gw ................. Read or set default gateway address
    netmask ............ Read or set subnet mask
    mac ................ Read or set MAC address
    ipsrc .............. Get or specify IP source configuration
    admin .............. Show admin name or set name and password
    raw ................ Execute raw commands
    test ............... Execute some self tests

C:\Program Files\KiraTool>
```

or you can send commands:

Figure 4 - 9. Working with the KiraTool



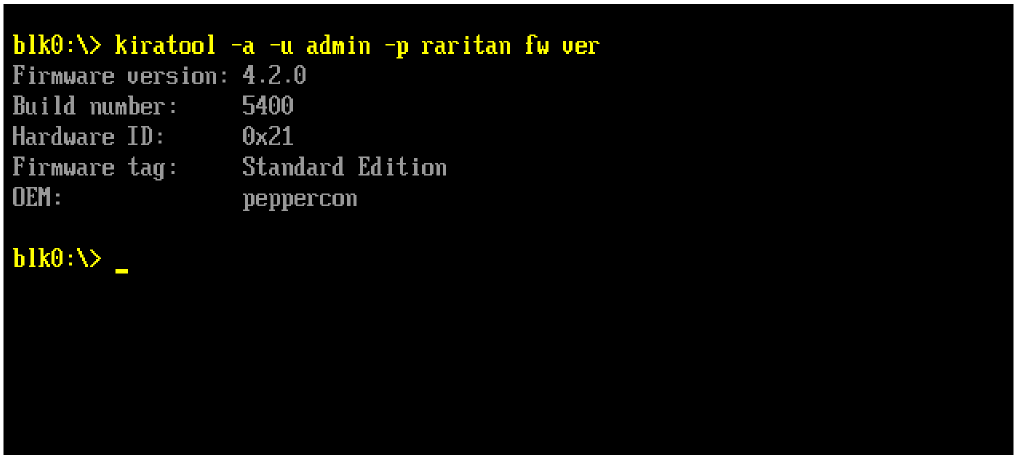
```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Program Files\KiraTool>kiratool -l 192.168.5.146 -u super -p pass fw ver
Firmware version: 4.2.0
Build number:     5776
Hardware ID:      0x20
Firmware tag:     Standard Edition
OEM:              peppercon

C:\Program Files\KiraTool>_
```

DOS and EFI Versions

The DOS and EFI versions of KT are extremely simple to install.

Simply copy the executable file to your local hard disk drive and invoke it. The picture below shows a typical invocation of the KT for EFI:

Figure 4 - 10. Working with KiraTool under EFI

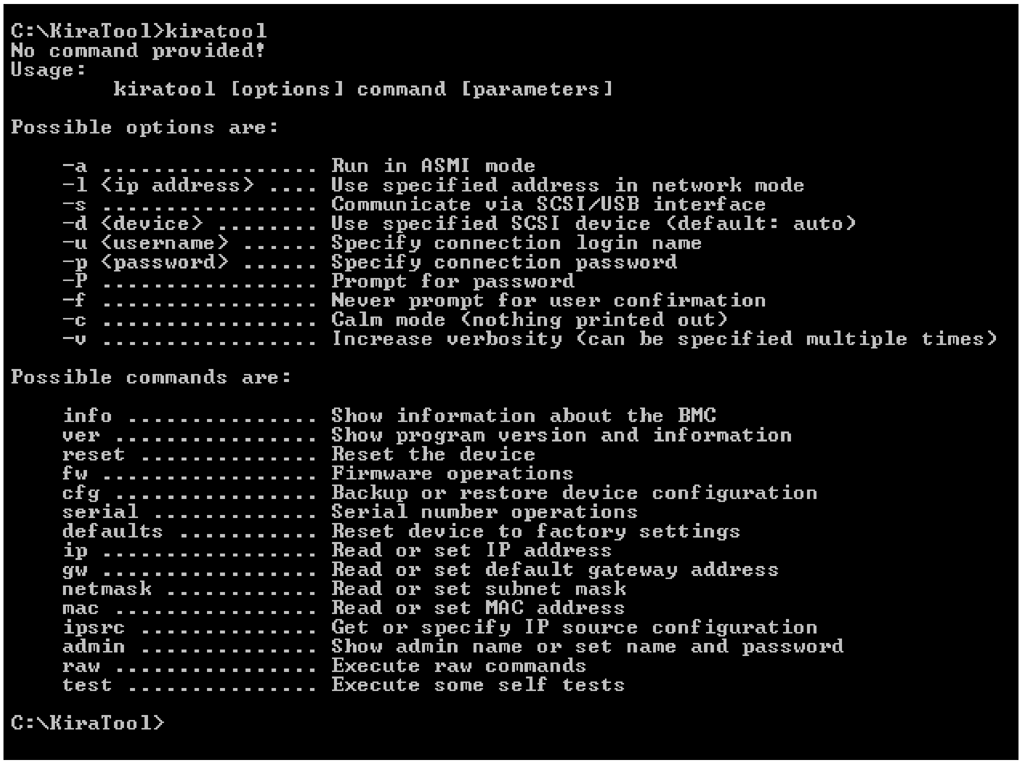


```
blk0:\> kiratool -a -u admin -p raritan fw ver
Firmware version: 4.2.0
Build number:     5400
Hardware ID:      0x21
Firmware tag:     Standard Edition
OEM:              peppercon

blk0:\> _
```

The DOS Version is very similar: again just copy the executable onto your hard drive.

Then when you invoke it, you will see something like:

Figure 4 - 11. Working with KiraTool under DOS



```
C:\KiraTool>kiratool
No command provided!
Usage:
        kiratool [options] command [parameters]

Possible options are:

    -a .................. Run in ASMI mode
    -l <ip address> .... Use specified address in network mode
    -s .................. Communicate via SCSI/USB interface
    -d <device> ........ Use specified SCSI device (default: auto)
    -u <username> ...... Specify connection login name
    -p <password> ...... Specify connection password
    -P .................. Prompt for password
    -f .................. Never prompt for user confirmation
    -c .................. Calm mode (nothing printed out)
    -v .................. Increase verbosity (can be specified multiple times)

Possible commands are:

    info ................ Show information about the BMC
    ver ................. Show program version and information
    reset ............... Reset the device
    fw .................. Firmware operations
    cfg ................. Backup or restore device configuration
    serial .............. Serial number operations
    defaults ............ Reset device to factory settings
    ip .................. Read or set IP address
    gw .................. Read or set default gateway address
    netmask ............. Read or set subnet mask
    mac ................. Read or set MAC address
    ipsrc ............... Get or specify IP source configuration
    admin ............... Show admin name or set name and password
    raw ................. Execute raw commands
    test ................ Execute some self tests

C:\KiraTool>
```

Linux Version

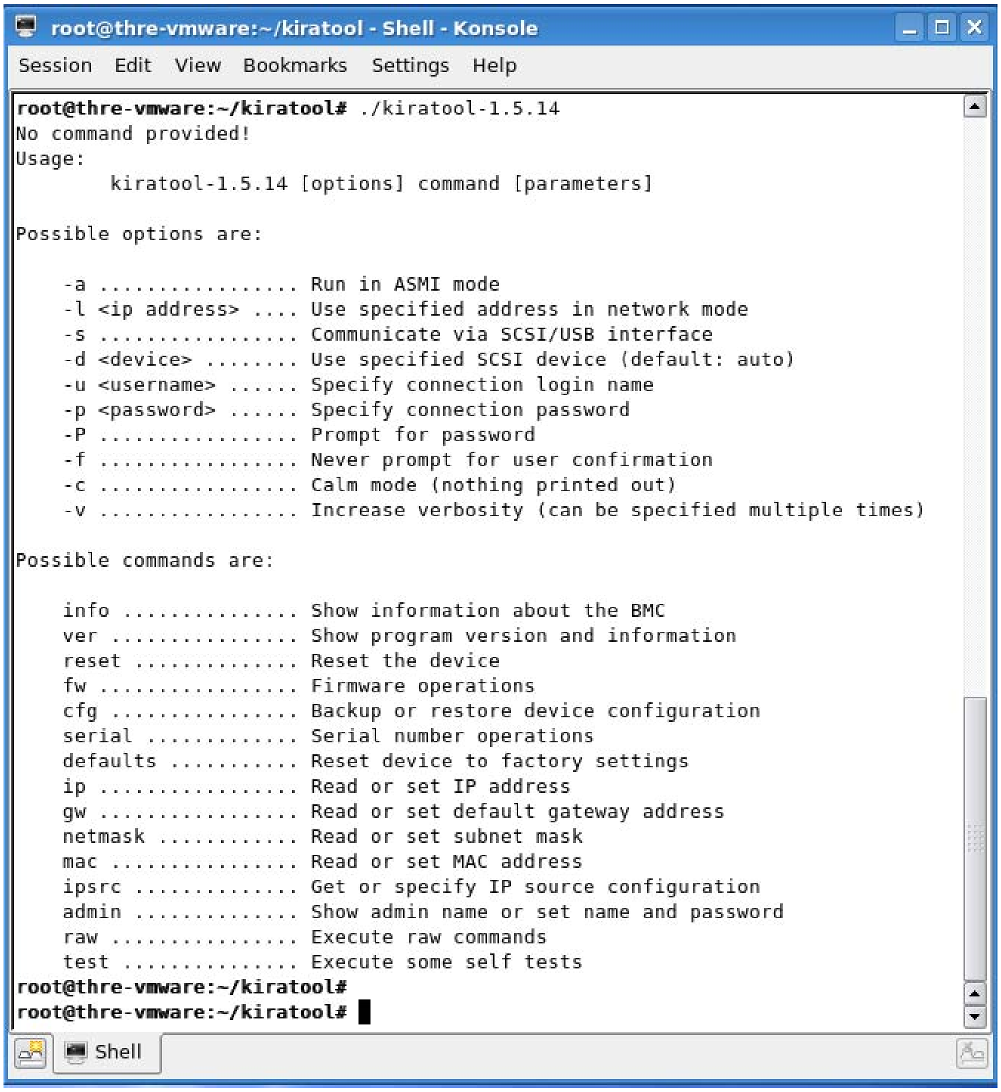
The Linux Version is also very similar: again just copy the executable to a directory that is found in your **`$PATH`** environment variable. Usually **`/usr/local/bin`** is a good idea. As such system directories are protected you need to be root in order to do this:

```
linux# cp kiratool /usr/local/bin
linux# chmod 755 /usr/local/bin/kiratool
```

In order to work, KiraTool needs the **`sg`** kernel module installed. Please load this kernel module before running KiraTool:

```
modprobe sg
```

Figure 4 - 12. Working with KiraTool under Linux



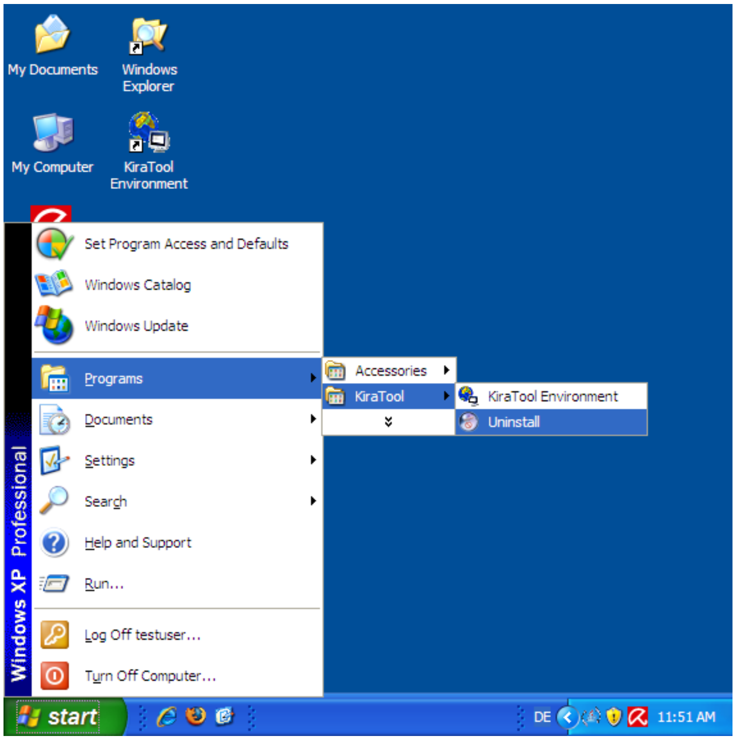
```
root@thre-vmware:~/kiratool# ./kiratool-1.5.14
No command provided!
Usage:
        kiratool-1.5.14 [options] command [parameters]

Possible options are:

    -a ................ Run in ASMI mode
    -l <ip address> .... Use specified address in network mode
    -s ................ Communicate via SCSI/USB interface
    -d <device> ........ Use specified SCSI device (default: auto)
    -u <username> ...... Specify connection login name
    -p <password> ...... Specify connection password
    -P ................ Prompt for password
    -f ................ Never prompt for user confirmation
    -c ................ Calm mode (nothing printed out)
    -v ................ Increase verbosity (can be specified multiple times)

Possible commands are:

    info .............. Show information about the BMC
    ver ............... Show program version and information
    reset ............. Reset the device
    fw ................ Firmware operations
    cfg ............... Backup or restore device configuration
    serial ............ Serial number operations
    defaults .......... Reset device to factory settings
    ip ................ Read or set IP address
    gw ................ Read or set default gateway address
    netmask ........... Read or set subnet mask
    mac ............... Read or set MAC address
    ipsrc ............. Get or specify IP source configuration
    admin ............. Show admin name or set name and password
    raw ............... Execute raw commands
    test .............. Execute some self tests
root@thre-vmware:~/kiratool#
root@thre-vmware:~/kiratool# █
```

Uninstalling the KiraTool

Windows Uninstall

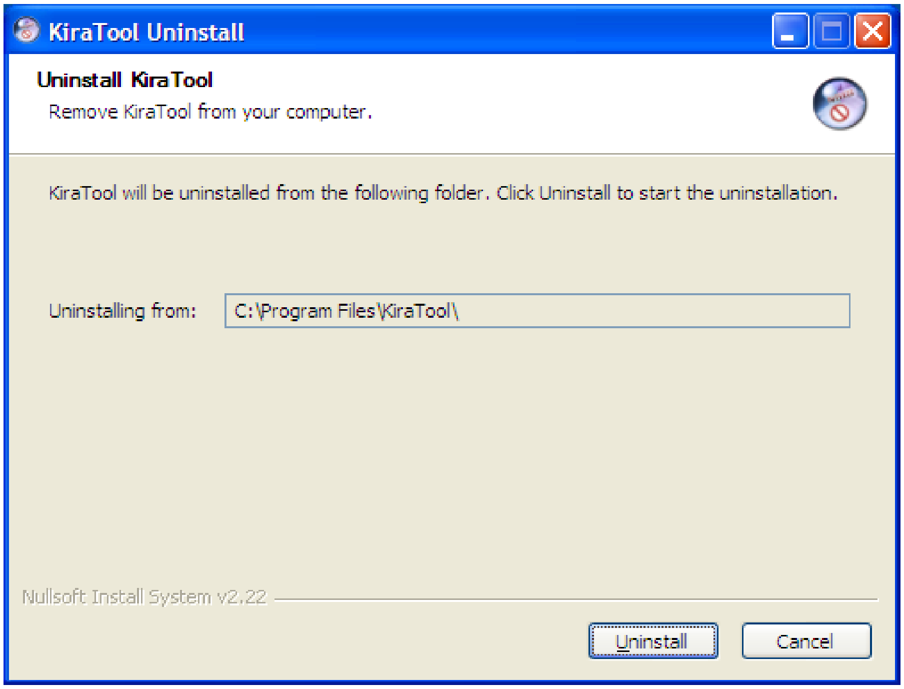
Generally uninstalling the KiraTool is easy. First start the uninstall wizard:

Figure 4 - 13. Uninstall the KiraTool under Windows



As usual you will be greeted with the wizards screen:

Figure 4 - 14. KiraTool Uninstall Wizard



Uninstall KiraTool

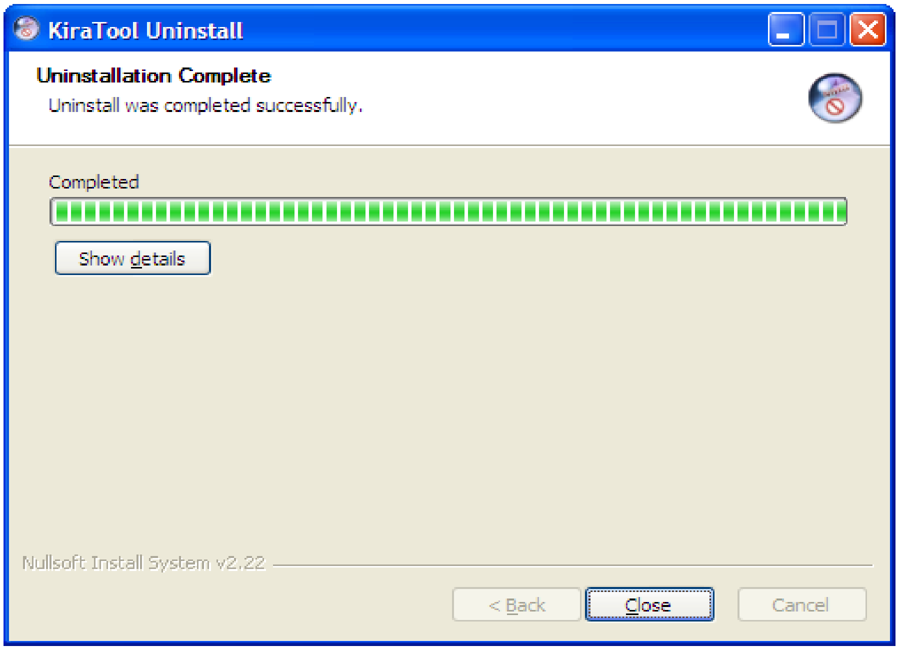
Remove KiraTool from your computer.

KiraTool will be uninstalled from the following folder. Click Uninstall to start the uninstallation.

Uninstalling from: C:\Program Files\KiraTool\

When you click **UNISTALL** you will start the removal of the KT. At the end of the uninstall, you will see a confirmation screen.

Figure 4 - 15. Finished KiraTool Uninstall Wizard


KiraTool Uninstall

Uninstallation Complete

Uninstall was completed successfully.

Completed

DOS and EFI Uninstall

Like under Linux, remove the installed binary from the disk.

Linux Uninstall

This is easy. Simply remove the files you placed in the system directories. You need to be root to do this:

```
linux# rm /usr/local/bin/kiratool
```

Appendix A. KiraTool Commands

Supported Operating Systems

- Windows (2000 or newer)
- EFI Shell
- Linux
- DOS

Supported Interfaces

- Remote: LAN (only Windows and Linux version)
- Local: - SCSI over USB
  - SMI (KCS)

Supported Functionality

- Network configuration (IP/mask/gw/MAC)
- Changing admin's name & password
- Showing serial number
- Resetting to factory defaults
- Firmware information and upgrade
- Device self-test

Usage

```
kiratool [options][cmd args]
```

Table A - 1. Options Overview

Options	
`-a`	must be used for KIRA based ASMI modules
`-l <ip>`	use `<ip>` address to connect to the module
`-s`	use IPMI-over-SCSI interface
`-d <device>`	use specified SCSI device; default: auto-detect
`-u <username>`	user name for login
`-p <password>`	password for login
`-P`	prompt for password
`-f`	force: never prompt for user confirmation
`-v`	verbose: increase verbosity level by one step, may be mentioned more than once for extra output
`-c`	calm: does not print out anything (silent)
`-h / -?`	help: shows help and usage information

Table A - 2. Commands Overview

Commands	
`ver`	shows version of KiraTool
`info`	shows vendor and device ID of the connected device
`ipsrc set static | dhcp | bios | none`	sets IP address source
`ipsrc [show]`	shows current IP address source
`ip set <ip addr>`	sets IP address (e.g. 192.169.1.123)
`ip [show]`	shows current IP address
`netmask set <netmask>`	sets netmask (e.g. 255.255.255.0)
`netmask [show]`	shows current netmask
`gw set <gw addr>`	sets gateway address (e.g. 192.169.1.1)
`gw [show]`	shows current gateway address
`mac set <mac addr>`	sets MAC address (e.g. "FE:00:00:12:34:56" or "FE0000123456")
`mac [show -c]`	shows current MAC address (-c = compact mode, e.g. "87654321DCBA" instead of "87:65:43:21:DC:BA")
`fw upgrade [-h] [-o] <fw bin file>`	upgrades firmware (-h = cross-hwid, -o = cross-oem)
`fw validate [-h] [-o] <fw bin file>`	checks firmware compatibility (-h = cross-hwid, -o = cross-oem)
`fw [ver]`	shows firmware version information
`serial [show]`	shows device's serial number
`defaults`	resets all settings to factory defaults
`reset`	hard-resets the module

Commands

`cfg backup <filename>`	backup the device's configuration to a file
`cfg restore <filename>`	restore the device's configuration from a file
`cfg get <key>`	read and show the given configuration key
`cfg set <key> <value>`	sets the given configuration key to the given value
`admin name <name>`	changes new admin name
`admin passwd <passwd>`	changes admin's password
`admin [show]`	shows admin's name
`fni [show]`	show status of IPMI over FML forwarding
`fni set <on/off>`	turn IPMI over FML forwarding on of off
`raw <hex bytes>`	send raw command and prints raw response (<netfn> <cmd> [<d1>] [<d2>] ... [<dN>]: e.g. 06 01)
`test <test>`	performs module self test and shows results (return value is ==0 on success and =0 in failure)
`device`	tests whether the device is vailable at all
`video <subtest>`	tests video interface (DVO/DVI)
`status`	checks detected video signal and resolution
`crc`	calculate CRC sum over the captured screen
`ddc <subtest>`	tests DDC interface
`info`	queries EDID information from the device and compares it to the EDID information known by the OS (only available under Windows)
`ipmb <subtest>`	tests IPMB interface
`bmc`	test whether a BMC responds over IPMB
`fml <subtest>`	tests FML interface
`esb2`	test whether an ESB2 is responding on FML when TPT (TCP Pass-Through) is active
`usb [-c <channel>] <subtest>`	tests USB interface
`status`	test whether the device's USB module is enumerated
`nic [-c <channel>] <subtest>`	test NIC interface
`status`	test NIC status and parameters
`loopback`	test NIC loopback functionality
`ping <host>`	Test whether pinging a host works
`broadcast`	Broadcast ping (not yet implemented)
`all`	performs all tests and subtests one after another
`-s <test to skip>`	Single tests can be skipped using the -s parameter. You can both skip a whole component (e.g. -s ddc) and skip a single test (e.g. -s video crc) Included tests in sequence: 1. ddc info 2. video status 3. ipmb bmc 4. fml esb2 5. usb status 6. nic status

Return Codes

To let the caller know whether an error occurred and what went wrong, KT delivers a return code back the caller:

- If everything went well (all tests passed) a value of 0 (zero) is returned
- For all commands except the “test” command a -1 (minus one) is returned if an error occurs

Note: in some operating systems this is converted to 127 or another value. Please check carefully!

- If a particular test fails, the return code indicates which test failed, according to the table below:

Table A - 3. Return Codes Overview

Test	Failure Return Code
Device	1
video status	2
video crc	3
ddc info	4
ipmb ddc	5
fml esb2	6
usb status	7
nic status	8
nic loopback	9
nic ping	10
nic broadcast	11
fml evalboard	12
ipmb evalboard	13

Note: *if a test fails, KT will return the return code of the first error encountered back to the caller. However KT will not stop executing the tests. It is therefore possible, that the remaining tests pass without problems, but you still receive an error code.*

Appendix B. Raritan, Inc. Warranty Information

Limited Warranty

Raritan Inc. manufactures its hardware products from parts and components that are new or equivalent to new in accordance with industry-standard practices. Raritan warrants that the hardware products including the firmware will be free from defects in materials and workmanship under normal use. Any implied warranties on the Raritan firmware and hardware are limited to 24 months, respectively, beginning on the date of invoice. Some states/jurisdictions do not allow limitations on duration of an implied warranty, so the above limitation may not apply to you. Additionally Raritan grants a special warranty for 6 months.

Customer Remedies

Raritan's entire liability and exclusive remedy shall be, at Raritan's option, either (a) return of the price paid, or (b) repair or replacement of the firmware or hardware that does not meet this Limited Warranty and which is returned to Raritan with a copy of your receipt. Damage due to shipping the products to you is covered under this warranty. Otherwise warranty does not cover damage due to external causes, including accident, abuse, misuse, problems with electrical power, servicing not authorized by Raritan , usage not in accordance with product instructions, failure to perform required preventive maintenance and problems caused by use of parts and components not supplied by Raritan . Any replacement hardware will be warranted for the remainder of the original period or thirty (30) days, whichever is longer. Raritan will repair or replace products returned to Raritan's facility. To request warranty service you must inform Raritan within the warranty period. If warranty service is required, Raritan will issue a Return Material Authorization Number. You must ship the products back to Raritan in their original or an equivalent packaging, prepay shipping charges, and insure the shipment or accept the possibility of loss or damage during shipment.

No Other Warranties

To the maximum extend permitted by applicable law, Raritan disclaim all other warranties, either express or implied, including, but not limited to implied warranties of merchantability and fitness for a particular purpose, with regard to the firmware, the accompanying written materials, and any accompanying hardware. This limited warranty gives you specific legal rights. You may have others, which vary from state/jurisdiction to state/jurisdiction.

No Liability For Consequential Damages

To the maximum extent permitted by applicable law, in no event shall Raritan be liable for any damages whatsoever (including without limitation, special, incidental, consequential or indirect damages for personal injury, loss of business information, or any other pecuniary loss) arising out of the use of or inability to use this product, even if Raritan has been advised of the possibility of such damages. In any case, Raritan's entire liability under any provision of this agreement shall be limited to the amount actually paid by you for the firmware and/or hardware. Because some states/jurisdictions do not allow the exclusion or limitation of liability for consequential or incidental damages, the above limitation may not apply to you.