



Solaris Driver

with Embedded RAINlink Technology

Installation and User's Guide

Drivers are available for Solaris versions 2.6, 7 and 8 for either x86 or Sparc platforms. The driver can be obtained from the ZNYX Networks web site: www.znyx.com. The distribution is available as a compressed, tar file in *pkgadd* format. The NetBlaster driver includes ZNYX Networks' embedded RAINlink technology, providing link aggregation and link failover features.

Platforms:	Sparc or x86-based platforms with standard PCI, PMC, or CompactPCI slots.
Operating Systems:	Solaris 8 (Sparc 32 bit/64 bit) Solaris 8 with hot-swap (Netra CT-Series) Solaris 8 (x86) Solaris 7 (Sparc 32 bit/64 bit) Solaris 7 (x86) Solaris 2.6 (Sparc) Solaris 2.6 (x86)

RETRIEVING DRIVERS

1. Go to the ZNYX web site at www.znyx.com and navigate under the Customer Service area to get the Solaris driver for your adapter. The current driver works for the following ZNYX models:

Model	Bus	Number of Ports
212	PMC	2
214	PMC	4
222	PMC	2
244	PMC	4
345Q	PCI	1
346Q	PCI	4
348Q	PCI	2
374	PCI	4
412	cPCI	2
414	cPCI	4
422	cPCI	2

424	cPCI	4
442 hot-swap	cPCI	2
444 hot-swap	cPCI	4
474 rear i/o & hot-swap	cPCI	4
478 rear i/o & hot-swap	cPCI	8

- The driver is available as a compressed tar file. Retrieve the file to a Solaris system, uncompress the file and untar. This will create a subdirectory in the current directory that contains the package name ZNYXnb.

For Intel driver:

```
uncompress ZNYXnb.intel.tar.Z
tar xvf ZNYXnb.intel.tar
```

For Sparc driver:

```
uncompress ZNYXnb.sparc.tar.Z
tar xvf ZNYXnb.sparc.tar
```

INSTALLING DRIVERS

- From the same directory where the package was untarred, enter the following command:

```
pkgadd -d . (NOTE: "dot" for current directory)
```

- The system will show you information about the package available at that location. You should see:

```
The following packages are available:

1  ZNYXnb          ZNYX NetBlaster Driver for Solaris 2.7
                        (i86pc) V2.3.7

Select package(s) you wish to process (or 'all' to process all packages).
(default: all) [?,??,q]: 1
```

- Hit the ENTER key to accept the default choice. The installation scripts will load and execute at this point:

```
Processing package instance <ZNYXnb> from </usr/znyx>

ZNYX NetBlaster Driver for Solaris 2.7
(i86pc) V2.3.7

Copyright (c) 2001 ZNYX Corporation
All Rights Reserved
```

- On Intel platforms, the script first makes a necessary check for memory configuration for the driver to support multiple interfaces. If you see the message below, you need to add the line "set lomempages=256" to the end of */etc/system*, and reboot. Answer "y" and follow the

prompts back out of the scripts. Make the change to */etc/system*, and reboot. Once you've rebooted, begin the package installation again with *pkgadd*. You should no longer see the message. If you do not see the message below, continue with the installation.

```
If you are installing this driver to support multiple interfaces, more
system resources are needed. You MUST add the following line to the file
/etc/system, reboot the system and restart this pkgadd.

set lomempages=256

Do you wish to abort the pkgadd to edit /etc/system(y) [y,n,?] y
```

5. The first question you will be asked is if you want to configure and bring up the interfaces.

```
Do you want to configure and bring up the interfaces [y,n,?] y
```

The NetBlaster driver includes ZNYX Networks' embedded RAINlink technology. If you are planning to enable RAINlink link aggregation, you should answer "no" to this question. You will configure and bring up your RAINlink adapters by completing this package installation, and then following the Configuring RAINlink for Solaris section that follows.

6. If you are simply bringing up the NetBlaster interfaces, answer yes to configure the interfaces now. Answering yes will allow you to give IP host names to each interface, and bring up the interfaces.
7. Next you will be prompted for the number of ports that you would like to configure:

```
Enter number of ports to configure (1-16) [1]: 2
```

8. Enter the number of ports that you would like to configure. If you have a ZNYX adapter with two ports, enter up to two. The maximum number of ports supported by the driver is 32.
9. The script will next request an IP name or address for each port.

NOTE: The IP name(s) you are supplying must exist for the interface to work properly. See your Network Administrator and obtain valid names before continuing. The script does not check for the validity or existence of the host names that you supply.

```
Enter hostname for interface znb0 [server-znb0]:
Enter hostname for interface znb1 [server-znb1]:

The following host names must be in the hosts database.
Check with your Network Administrator for more information.
```

```
server-znb0
server-znb1
```

10. Once all names have been supplied, the script continues.

```
## Processing package information.
## Processing system information.
## Verifying disk space requirements.
## Checking for conflicts with packages already installed.
## Checking for setuid/setgid programs.

This package contains scripts which will be executed with super-user
permission during the process of installing this package.

Do you want to continue with the installation of <ZNYXnb> [y,n,?]
```

11. Answer “y” when prompted to continue running scripts requiring root privileges. The package will now finish installing the driver, and if you selected to configure and bring up interfaces, it will bring them up.

```
Installing ZNYX NetBlaster Driver for Solaris 2.7 as <ZNYXnb>

## Installing part 1 of 1.
/etc/rain/rainlink
/etc/rain <implied directory>
/etc/rain/rlconfig
/etc/rain/rlfe2
/etc/rain/rlfe4
/etc/rain/rlff2
/etc/rain/rlff4
/etc/rain/rlss2
/etc/rain/rlss4
/kernel/drv/znb
/kernel/drv/znb.conf
[ verifying class <none> ]
## Executing postinstall script.
Bringing up available boards now....

Installation of <ZNYXnb> was successful.
```

12. When the script completes, you should see the successful banner, and you are returned to the package installation screen. Enter "q" to exit the *pkgadd* program. Reboot the system.

CONFIGURING RAINLINK FOR SOLARIS

All the necessary installation and configuration programs for RAINlink for Solaris are placed on the system during the installation of the NetBlaster driver.

1. RAINlink for Solaris includes utilities and example configuration files for standard configurations in order to get you up and running quickly. The necessary files are placed onto your system during the NetBlaster driver installation in */etc/rain*. Change directories to */etc/rain* to begin the installation:

```
cd /etc/rain
```

2. An installation script is included called *rainlink*. The *rainlink* script can be used to activate, deactivate, or display the current RAINlink for Solaris configuration. The first time the *rainlink* script is run, the script will give you a list of default configurations to choose from suitable for most basic configurations of two or four port trunks or failover groups. Start the *rainlink* script by typing *rainlink -a* at the command prompt:

```
rainlink -a
```

3. The script dialogue will allow you to choose from a set of default configurations.

RAINlink

This script initializes or reinitializes the RAINlink layer within the ZNYX NetBlaster driver. An input script located at `/etc/rain/rainlink.conf` is used as input to ZNYX's `rlconfig` application. Either choose one of the following prepared input scripts which will be copied to `/etc/rain/rainlink.conf`, or exit and create your own custom script.

- 1 - System-to-System trunking between 4 ports.
- 2 - Fast EtherChannel trunking between 4 ports.
- 3 - Fast Failover between 4 ports.
- 4 - System-to-System trunking between 2 ports.
- 5 - Fast EtherChannel trunking between 2 ports.
- 6 - Fast Failover between 2 ports.
- 7 - Exit and create custom script.

Choose one of the above 7 choices(1-7):

4. Make your selection and continue. A note will be displayed informing you that the *rainlink* script will create a configuration file named, `/etc/rain/rainlink.conf`, which will be the default configuration file used from this point forward. Details of the contents of the configuration file are provided later. Hit Enter to continue:

```
NOTE: /etc/rain/rainlink.conf has been created. If a rainlink.conf file
exists rainlink will not prompt for one of the previous 7 choices
To modify the RAINlink configuration in the future, edit
/etc/rain/rainlink.conf, and either run rainlink again, or reboot the
system.
```

```
Hit any key to continue:
Bringing down all Net Blaster interfaces now....
Running rlconfig ...
#####TRUNK0 Successfully Created
TRUNK0 mode set
```

5. The next series of questions relate to bringing up the interfaces and starting the RAINlink daemon, rld. Starting the RAINlink daemon allows you access to the statistics and configuration of RAINlink interfaces from a remote location. Bring up the RAINlink interfaces, then answer whether or not you will use the RAINlink daemon, and then bring up the remaining non-RAIN interfaces if there are any. You will need to supply IP names or addresses for each configured interface you bring up.

```
Do you want to configure and bring up the RAINlink interfaces [y,n,?] y
Enter hostname for interface zrl0 [barbrady-zrl0]: server1
```

```
The following host names must be in the hosts database.
Check with your Network Administrator for more information.
```

```
server1
```

```
Do you want to start the RAINlink daemon to allow for network
monitoring[ y,n,?] y
```

```
Do you want to configure and bring up any non-RAINlink interfaces [y,n,?] y
```

```
Bring znb4 up [y,n,?] y
Enter hostname for interface znb4 [barbrady-znb4]: access1
```

```
The following host names must be in the hosts database.
Check with your Network Administrator for more information.
```

```
access1
```

```
Bringing up available boards now....
```

The script then exits. Your RAINlink and non-RAINlink interfaces should now be configured.

6. You can use *rainlink -s* to verify the status of your RAINlink configuration:

```
rainlink -s
```

```
trunk0<LAYER_3_MODE> = znb0<UP>, znb1<UP>, znb2<UP>,
znb3<UP>, znb4<UP> ==> zrl0
```

ADVANCED TRUNKING

You can create link aggregation groups (trunks), assign ports to these trunks and select a scheme of dynamic load balancing. The most straightforward method for configuring RAINlink for Solaris is to edit the `/etc/rain/rainlink.conf` file and run `rainlink -a`. The script de-installs the current ZNYX interfaces, runs a configuration application `rlconfig` with the `rainlink.conf` input script, and brings up the RAIN links. The `rainlink -a` script can also bring up any regular non-RAIN interfaces, which were not configured into the RAIN links. All these steps can be done individually. Details of using the `rlconfig` application are included in a later section.

To add a trunk, edit the file `/etc/rain/rainlink.conf`. The file is a plain text file that is extensively commented. Commented lines begin with a pound sign (#). Add a line for each trunk containing the list of ports to be included in that trunk. End each line with a semicolon. For example, to include ports 0 through 3 in trunk0, add an entry:

```
trunk0=znb0, znb1, znb2, znb3;
```

The order of entries is not important. You can use any available ports in any order. Port entries should not be duplicated in trunks. A trunk must consist of at least two ports, and cannot contain other trunks. The following is an acceptable way to configure two trunks:

```
trunk0=znb0, znb3;
trunk1=znb1, znb2;
```

For system-to-system trunking, enable IP Trunking mode. Both systems should be setup identically. In IP Trunking mode, packets are sent to the driver as large datagrams and fragmented across the active links. To enable IP Trunking mode, uncomment/add the following line for each configured trunk:

```
trunk0=ip_trunking_mode;
```

For system-to-switch trunking, the default method of load balancing scheme uses Layer 3 Protocols. To use only the “Layer 2 Protocols” for dynamic load balancing, comment out the “layer_3_mode” option, and uncomment the “layer_2_mode” option. The options are mutually exclusive.

```
# trunk0=layer_3_mode;
trunk0=layer_2_mode;
```

To explicitly choose Layer 3, uncomment (or add) the “Layer 3 Protocols” option, and comment out the “Layer 2 Protocols” option:

```
trunk0=layer_3_mode;
# trunk0=layer_2_mode;
```

For system-to-switch trunking, you can disable Balance mode. Balance mode is on by default, and is normally left enabled. Balance mode dynamically distributes the load across the ports equally. It can be disabled by adding a “not equals” to the option:

```
trunk0 != balance_mode;
```

An “equals” sign explicitly enables balance mode:

```
trunk0 = balance_mode;
```

Once you have assembled your *rainlink.conf* file, run *rainlink -a* to bring up RAINlink for Solaris with the new configuration.

```
rainlink -a
```

ADVANCED FAILOVER

You can create link failover groups, assign ports and trunks to these groups and select a modes of failover. Failover groups are configured in the same manner as trunks: Edit the */etc/rain/rainlink.conf* file and run *rainlink -a*. *rainlink -a* de-installs the current ZNYX interfaces, runs a configuration application *rlconfig* with the *rainlink.conf* input script, and brings up the RAIN links. The *rainlink -a* script can also bring up any regular non-RAIN interfaces, which were not configured into the RAIN links. All these steps can be done individually. Details of using the *rlconfig* application are included in a later section.

To add a failover group, edit the file */etc/rain/rainlink.conf*. The file is a plain text file that is extensively commented. Commented lines begin with a pound sign (#). Failover groups can contain individual ports, trunks, or both ports and trunks. Always configure trunks before failover groups. Add a line for each failover group containing the list of ports and trunks to be included in that group. End each line with a semicolon. For example, to build a failover group of ports 0 and 1:

```
failover0 = znb0, znb1;
```

To build a failover group of two trunks, first build the trunks, then the failover group:

```
trunk0 = znb0, znb1;  
trunk1 = znb2, znb3;  
failover0 = trunk0, trunk1;
```

The following order will not work. You must build the trunks first:

```
failover0 = trunk0, trunk1;  
trunk0 = znb0, znb1;  
trunk1 = znb2, znb3;
```

You can also mix trunks and ports in failover groups, as long as the trunks are built first:

```
trunk0 = znb0, znb1;  
failover0 = trunk0, znb2, znb3;
```

To remove a failover group, simply remove it, or comment it out, and run *rainlink -a*.

The default mode of failover is fast failover. In this mode, RAINlink for Solaris moves the traffic over to a redundant stand-by link in case of a link failure in as little as milliseconds. In addition to fast failover, Timeout mode can be enabled for a failover group. By enabling Timeout Mode, if no traffic is received in the specified time interval, the active port is automatically switched to another available link. You can

specify the timeout interval in milliseconds. To set Timeout Mode on a failover group with a timeout of 30 seconds, add a line for the failover group:

```
failover0 = timeout_mode 30000;
```

USING RLCONFIG

The *rlconfig* application views or changes the RAINlink for Solaris configuration. *Rlconfig* is installed into */etc/rain* during the NetBlaster driver installation. You can view the current RAINlink for Solaris configuration at any time, but the interfaces must be “down” in order to change the configuration See *ifconfig(1M)* for more explanation of how to bring an interface down.

Options for *rlconfig* include:

Option:	Use:
-s	Displays the current RAIN configuration. Commands are not read from standard input with this option.
-t	Tear down all previously configured RAIN ports. Commands are not read from standard input with this option.
-p ppa	Displays statistics for the specified ppa device. Use 0 for ppa0, etc.
-l [#] file_name	Creates a file containing a list of the configured ZRL devices and the available ZNB devices. The “#” is used for version identification. This option is not intended for use by the user; it is used by the rainlink script to configure RAIN

To view the current configuration at any time, enter:

```
rlconfig -s
```

The resulting display shows you the status of the RAIN configuration.

```
trunk0<IP_TRUNKING> = znb0<UP>, znb1<UP>, znb2<UP>,
znb3<UP>, znb4<UP> ==> zrl0
```

Lines can be input directly to *rlconfig* from standard input, or from a script like */etc/rain/rainlink.conf*. To start *rlconfig* enter:

```
rlconfig
```

Enter commands one line at a time and conclude with Ctrl-d.

To use a previously prepared file of commands, redirect the file into *rlconfig*. For example:

```
rlconfig < /etc/rain/rainlink.conf
```

Typical usage would be to redirect a script of commands into *rlconfig* which instruct *rlconfig* to build and set modes on trunks and failover groups. Semicolons delimit commands. Spaces and new lines are ignored. Commands take the following form:

```
trunk<number> = znb<number> [ , znb<number> ... ] ;
failover<number> = znb | trunk<number> [ , znb|trunk<number> ...]

trunk<number> = mode ;
trunk<number> != mode ;

failover<number> = mode ;
failover<number> != mode ;
```

where <number> is a value between 0 and the maximum number of ports in the system. Trunks can consist of ports denoted by a comma-delimited list of znb<number>. A failover group can consist of ports or trunks. You must build a trunk prior to its use in a failover group.

Acceptable modes for trunks are:

Mode:	Appropriate for:
ip_trunking_mode	System-to-system trunking
layer_2_mode	System-to-switch trunking
layer_3_mode	System-to-switch trunking
balance_mode	System-to-switch load balancing enabled/disabled
timeout_mode <time>	Failover mode; time parameter in milliseconds
hub_mode	Failover mode; Only receives on the ACTIVE port. Use when connecting failover ports in the same collision domain, like to a hub. Do not use when connecting system-to-system, or system-to-switch.

To tear down the existing RAIN configuration, the interface must first be down. See ifconfig(1M):

```
ifconfig zr10 down
rlconfig -t
```

ADDITIONAL DRIVER CONFIGURATION

The znb driver controls the NetBlaster adapter's ethernet devices. Driver options can be edited in two different ways:

- Configure the znb driver parameters by editing the znb.conf file in /kernel/drv. Configuration options affect all znb devices.
- Use the ndd utility to temporarily change a parameter. Configuration options are done on a per-device basis. The change is lost when you reboot the system.

This section lists the available NetBlaster driver parameters, and describes how you can configure these parameters.

Driver Parameter Values and Definitions

This section describes the parameters and settings for the NetBlaster driver.

Parameter	Status	Description
link_status	Read only	Defines the current status
link_speed	Read only	Defines the current status
link_mode	Read only	Defines the current status
transceiver_inuse	Read only	Static Read-Only Parameter
ipg1	Read only	Static Read-Only Parameter
ipg2	Read only	Static Read-Only Parameter
use_int_xcvr	Read only	Static Read-Only Parameter
pace_size	Read only	Static Read-Only Parameter
adv_autoneg_cap	Read only	Static Read-Only Parameter
adv_100fdx_cap	Read only	Static Read-Only Parameter
adv_100hdx_cap	Read only	Static Read-Only Parameter
adv_10fdx_cap	Read only	Static Read-Only Parameter
adv_10hdx_cap	Read only	Static Read-Only Parameter
autoneg_cap	Read only	Static Read-Only Parameter
100fdx_cap	Read only	Static Read-Only Parameter
100hdx_cap	Read only	Static Read-Only Parameter
10fdx_cap	Read only	Static Read-Only Parameter
10hdx_cap	Read only	Static Read-Only Parameter
lp_autoneg_cap	Read only	Static Read-Only Parameter
lp_100fdx_cap	Read only	Static Read-Only Parameter
lp_100hdx_cap	Read only	Static Read-Only Parameter
lp_10fdx_cap	Read only	Static Read-Only Parameter
lp_10hdx_cap	Read only	Static Read-Only Parameter
instance	Read and write	Device instance
lance_mode	Read only	Static Read-Only Parameter
ipg0	Read only	Static Read-Only Parameter

Retrieving the Current Status

The read-only parameters described below explain the operational mode of the interface. These parameters define the current status.

Parameter	Values	Description
link_status	0	Current link status = Link down = Link up
	1	
link_speed	0	Valid only if the link is up = 10 Mbps = 100 Mbps
	1	

Parameter	Values	Description
link_mode	0 1	Valid only if the link is up = Half duplex = Full duplex

Retrieving the Static Read-Only Parameters

The static read-only parameters described below return the default values shown. These values are returned to improve compatibility with existing applications that use `ndd` to communicate with standard Solaris network drivers. They do not have particular meaning to the `znb` driver.

Parameter	Value
transceiver_inuse	0
ipg1	8
ipg2	4
use_int_xcvr	0
pace_size	0
adv_autoneg_cap	1
adv_100fdx_cap	1
adv_100hdx_cap	1
adv_10fdx_cap	1
adv_10hdx_cap	1
autoneg_cap	1
100fdx_cap	1
100hdx_cap	1
10fdx_cap	1
10hdx_cap	1
lp_autoneg_cap	1
lp_100fdx_cap	1
lp_100hdx_cap	1
lp_10fdx_cap	1
lp_10hdx_cap	1
lance_mode	0
ipg0	16

Setting Parameters Using the `ndd` Utility

Use the `ndd` utility to configure parameters that are valid until you reboot the system. The `ndd` utility supports any networking driver, which implements the Data Link Provider Interface (DLPI).

The following sections describe how you can use the `znb` driver and the `ndd` utility to modify (with the `-set` option) or display (without the `-set` option) the parameters for each NetBlaster device.

Identifying Device Instances

Before you use the `ndd` utility to get or set a parameter for a NetBlaster device, you must specify the device instance for the utility.

To Specify the Device Instance for the ndd Utility

1. Check the `/etc/path_to_inst` file to identify the instance associated with a particular device.

```
# grep znb /etc/path_to_inst
"/pci@1f,2000/pci@2/ethernet @0,1" 4 "znb"
"/pci@1f,2000/pci@2/ethernet @1,1" 5 "znb"
"/pci@1f,2000/pci@2/ethernet @2,1" 6 "znb"
"/pci@1f,2000/pci@2/ethernet @3,1" 7 "znb"
```

In the example above, the four `ethernet@x,1` instances are from a ZNYX quad-port FastEthernet PCI adapter installed in slot 2. For clarity, the instance numbers are **bold**.

2. Use the instance number to select the device.

```
# ndd -set /dev/znb instance instance#
```

The device remains selected until you change the selection.

Non-Interactive and Interactive Modes

You can use the `ndd` utility in two modes:

- Non-interactive
- Interactive

In non-interactive mode, you invoke the utility to execute a specific command. Once the command is executed, you exit the utility. In interactive mode, you can use the utility to get or set more than one parameter value. (Refer to the *ndd (1M)* man page for more information.)

Using the ndd Utility in Non-Interactive Mode

This section describes how to modify and to display parameter values. To modify a parameter value, use the `-set` option. If you invoke the `ndd` utility with the `-set` option, the utility passes value, which must be specified down to the named `/dev/znb` driver instance, and assigns it to the parameter:

```
# ndd -set /dev/znb parameter value
```

To display the value of a parameter, specify the parameter name (and omit the value).

When you omit the `-set` option, a query operation is assumed and the utility queries the named driver instance, retrieves and prints the value associated with the specified parameter:

```
# ndd /dev/znb parameter
```

Using the ndd Utility in Interactive Mode

To modify a parameter value in interactive mode, specify `ndd /dev/znb`, as shown below. The `ndd` utility then prompts you for the name of the parameter:

```
# ndd /dev/znb
name to get/set ?
```

Enter the parameter name or ? to view all parameters. After entering the parameter name, the ndd utility prompts you for the parameter value (see Driver Parameter, Status, and Descriptions through Read-Only Link Partner Capabilities).

To list all the parameters supported by the znb driver, type:

```
ndd /dev/znb \?
```

Example of Listing All Parameters Supported by the znb Driver

```
# ndd /dev/znb \?

?                (read only)
transceiver_inuse (read only)
link_status      (read only)
link_speed       (read only)
link_mode        (read only)
ipg1             (read only)
ipg2             (read only)
use_int_xcvr     (read only)
pace_size        (read only)
adv_autoneg_cap  (read only)
adv_100fdx_cap   (read only)
adv_100hdx_cap   (read only)
adv_10fdx_cap    (read only)
adv_10hdx_cap    (read only)
autoneg_cap      (read only)
100fdx_cap       (read only)
100hdx_cap       (read only)
10fdx_cap        (read only)
10hdx_cap        (read only)
lp_autoneg_cap   (read only)
lp_100fdx_cap    (read only)
lp_100hdx_cap    (read only)
lp_10fdx_cap     (read only)
lp_10hdx_cap     (read only)
instance         (read and write)
lance_mode       (read only)
ipg0             (read only)
```

Setting Parameters Using the znb.conf File

You can also specify the properties for all the znb devices by editing the znb.conf file in the /kernel/drv directory. Uncomment the desired option by removing the "#" sign, and then edit the line to include the appropriate values:

Parameter	Usage
hal_debug=1; zxe_debug=3; sal_debug=3;	To enable debugging output, set the zxe_debug, hal_debug or sal_debug values to a non-zero level from 1 to 4. The larger the number, the more console output. Setting debug levels will slow down system performance.
mode=10; fulldup;	Default operation is autonegotiation. To force operation at a specific speed use the mode and fulldup flags. Mode can be either 10 or 100. If fulldup is present, the driver will be forced to full duplex operation;
port_mode10=0,1,2,3; port_mode100=0,1,2,3; port_halfdup=0,1,2,3; port_fulldup=0,1,2,3;	To force operation speeds on a per port basis use port_mode and port_fulldup. There usage is similar to the mode and fulldup flags but they take a port as their first parameter. fulldup has been changed from a boolean parameter to now take a port and a boolean flag.
no_promiscuous_loop;	Setting no_promiscuous_loop causes the driver to not loop back transmit packets when promiscuous mode is set

See the `znb.conf` file for details of setting the parameters. The Solaris man page for *driver.conf* (4) includes additional details.

REMOVING DRIVERS

1. Remove the driver with the standard Solaris package removal utility, *pkgm*.

```
pkgm ZNYXnb
```

RAINlink includes the following files and utilities to aid in configuration and monitoring:

- rainlink
- rlconfig
- rlalarm
- rlstats
- rld

This section includes Unix-style man pages to be used as reference.

rainlink(1M)

NAME

rainlink – Activate/Deactivate RAINlink

SYNOPSIS

```
/etc/rain/rainlink [ -a | -d | -s | -u ]
```

DESCRIPTION

The *rainlink* command script initializes and reinitializes the RAINlink layer within the ZNYX NetBlaster driver. An input script located in */etc/rain/rainlink.conf* is used as input to ZNYX's rlconfig application. A list of default choices is presented if the */etc/rain/rainlink.conf* input script does not exist. The script can be used to bring up the RAINlink and non-RAINlink interfaces. It can also be used to display the current RAINlink configuration.

OPTIONS

- | | |
|----|---|
| -a | Activate RAINlink features. Creates an <i>/etc/rain/rainlink.conf</i> file if it doesn't exist from a list of default choices, brings down the existing NetBlaster interfaces, tears down the existing RAINlink configuration, builds the new RAINlink configuration, brings up the RAINlink interfaces, and finally any other NetBlaster interfaces. |
| -d | Deactivate RAINlink features. Tears down the existing RAINlink interfaces, and brings up NetBlaster interfaces. |
| -s | Show RAINlink configuration (same as <i>/etc/rain/rlconfig -s</i>) |
| -u | Display usage (same as no parameters) |

FILES

/etc/rain/rainlink.conf

SEE ALSO

rlconfig(1M)

rlconfig(1M)

NAME

rlconfig - Configure ZNYX Redundant Array of Independent Netports (RAIN).

SYNOPSIS

```
/etc/rain/rlconfig [ -s ] [ -t ] [ -p ppa ]  
                  [ -l # list_file ] < input_file
```

DESCRIPTION

The *rlconfig* application is used to configure multiple ZNYX NetBlaster ports into trunks or failover groups. With exception of the *-s* option, all interfaces affected by *rlconfig* must be in the down states. See *ifconfig(1M)* for explanation on bringing down interfaces. The application reads standard input for commands.

A trunk is a grouping of two or more ports that can do port aggregation. Port aggregation can be put in a mode where multiple clients connecting to a server can utilize the bandwidth of multiple ports seamlessly, or in a mode where a server connecting to a server can seamlessly utilize the higher bandwidth of multiple ports.

A failover group is a grouping of ports where only one port is active at any one time. The remaining ports in the failover group are in standby in case the active port goes down. Typically, a failover group would only switch from one link to another if the physical link were lost. By enabling timeout mode, if no traffic is received in the specified time interval, the active port is automatically switched to another available link. The timeout mode time interval is settable by the user in increments of milliseconds.

OPTIONS

- | | |
|--------------------|---|
| -s | Displays the current RAIN configuration. Commands are not read from standard input with this option. |
| -t | Tear down all previously configured RAIN ports. Commands are not read from standard input with this option. |
| -l [#] file_name | Creates a file containing a list of the configured ZRL devices and the available ZNB devices. The “#” is used for version identification. This option is not intended for use by the user; it is used by the <i>rainlink</i> script to configure RAIN |
| -p ppa | Displays hardware level statistics for the specified ppa. |

USAGE

Typically, a script of commands is directed into *rlconfig* which instruct *rlconfig* to build and set modes on trunks and failover groups. Semicolons delimit commands. Spaces and new lines are ignored. Commands take the following form:

```
trunk<number> = znb<number> [ , znb<number> ... ] ;  
failover<number> = znb<number> | trunk<number> [ , znb<number>  
|  
trunk<number> ...];
```

```
trunk<number> = mode ;
trunk<number> != mode ;

failover<number> = mode ;
failover<number> != mode ;
```

where <number> is a value between 0 and the maximum number of ports in the system. Trunks can consist of ports denoted by a comma-delimited list of znb<number>. A failover group can consist of ports or trunks. You must build a trunk prior to its use in a failover group.

Acceptable modes for trunk and failover configurations:

Mode:	Appropriate for:
ip_trunking_mode	System-to-system trunking
layer_2_mode	System-to-switch trunking
layer_3_mode	System-to-switch trunking
balance_mode	System-to-switch load balancing enabled/disabled
timeout_mode <time>	Failover mode; time parameter in milliseconds

DISPLAYS

The “-s” option of the *rlconfig* displays the current status of the RAINlink interfaces. The display includes any configured modes for trunks or failover groups, and the status of the interfaces in the form:

```
trunk<number> <mode> = znb<number>, znb<number> ... ==> zrl<number>
```

Where number is the trunk, group, or interface identifier, and mode is one of the acceptable modes. For example:

```
trunk0<IP_TRUNKING> = znb0<UP>, znb1<UP> ==> zrl0
```

The interface status can be:

Name:	Meaning:
UP	Interface is UP
DOWN	Interface is DOWN
ACTIVE	The interface is UP, and is the ACTIVE member of a failover group
STANDBY	The interface is UP, but is not the ACTIVE member of a failover group.

The “-p” option of *rlconfig* displays the hardware level statistics for the specifies ppa. The display includes current ppa state, fault state, link configuration and current link state. Transmit and receive statistics are also displayed.

The following are valid values for the above states:

Name:	Possible states:
State:	HS_STATE_NOT_INIT HS_STATE_STOPPED HS_STATE_RUNNING
Fault:	HS_FAULT_NONE HS_FAULT_INTERNAL HS_FAULT_EXTERNAL
Link configuration	HS_LINK_AUTO HS_LINK_TP HS_LINK_BNC HS_LINK_AUI HS_LINK_TPFD HS_LINK_TX HS_LINK_TXFD HS_LINK_T4
Current link state:	HS_LINK_DOWN HS_LINK_TP HS_LINK_BNC HS_LINK_AUI HS_LINK_TPFD HS_LINK_TX HS_LINK_TXFD HS_LINK_T4

FILES

/etc/rain/rainlink.conf

SEE ALSO

rainlink(1M)

rlalarm(1M)

NAME

rlalarm – RAINlink switch and event monitoring application

SYNOPSIS

```
/etc/rain/rlalarm [-d <debug_level> [-s <dev_id> <ppa>] [-r <dev_id>] [-m <ppa>]
[-p <dev_id> <ppa> <ppa> ...] [-h <host_name>
```

DESCRIPTION

rlalarm is a utility based on the RAINlink Management API (RMAPI) that allows manipulation of RAINlink devices as well as a method for event notifications of a specified device. The device can be either a RAINlink or MAC level device, and can be located locally or reachable remotely via a RAINlink server daemon (rld).

OPTIONS

-s dev_id ppa	Switch ACTIVE <i>ppa</i> of device <i>dev_id</i> to the specified <i>ppa</i> .
-r dev_id	Display RAINlink notifications from specified <i>dev_id</i> .
-m ppa	Display MAC level notifications from specified <i>ppa</i> .
-p dev_id ppa ppa ...	Set precedence of device <i>dev_id</i> on failover group to order of <i>ppa</i> list. First entry in list has highest precedence.
-d debug_level	Sets debug level (1 through 4). Larger number increases verbosity.
-h host_name	Connect to remote host.

USAGE

Rlalarm allows manipulation and monitoring of RAINlink devices. To switch the active port in a failover group (zrl0) from port0 to port1, enter:

```
rlalarm -s 0 1
```

The 0 references the RAINlink (zrl) device, the 1 references the port (or ppa, znb1) to make ACTIVE in the failover group.

The "-r" and "-m" options allow you to receive notifications of event changes at either the RAINlink level or MAC level.

The "-p" option allows a method for setting a precedence within a failover group. This allows the user to configure a failover group to always return to a specified ppa, if it becomes available later. The default operation for a RAINlink failover group is to only switch when the ACTIVE port no longer is available. For example, once you've configured a failover group of two ports,

```
failover0=znb0, znb1;
```

Use "rlalarm" to specify that whenever znb0 is available, it should be the ACTIVE port:

```
rlalarm -p 0 0 1
```

You can also connect to a remote host running the RAINlink server daemon *rld*. Provide the remote hostname or ip address as an argument to rlalarm:

```
rlalarm -h 10.0.0.1 -m 0
```

This would attempt a TCP connection to remote host with IP address 10.0.0.1, and if successful retrieve the MAC level statistics for znb0 (ppa 0).

FILES

None

SEE ALSO

rld(1M)

rlstats(1M)

NAME

rlstats – Displays statistics for NetBlaster or RAINlink devices

SYNOPSIS

```
/etc/rain/rlstats [[-m|-s|-o|-i] <ppa>] [-d <level>] [-z]
```

DESCRIPTION

The *rlstats* application is used to display statistics and configuration information for RAINlink devices.

OPTIONS

-m <i>ppa</i>	Displays MAC layer statistics for specified <i>ppa</i> .
-s <i>ppa</i>	Turns speedometer on for specified <i>ppa</i> .
-r <i>ppa</i>	Sets speedometer polling rate (in ms) for specified <i>ppa</i> . Use in conjunction with -s <i>ppa</i> .
-i <i>ppa</i>	Retrieve hardware information for specified <i>ppa</i>
-o <i>ppa</i>	Display speedometer statistics for the specified <i>ppa</i>
-d <i>debug_level</i>	Sets debug level (1 through 4). Larger number increases verbosity.
-z	Displays RAINlink configuration

USAGE

Use *rlstats* to display statistics for a particular port or RAINlink configuration.

FILES

None

SEE ALSO

rld(1M)

rld(1M)

NAME

rld – RAINlink redirector daemon

SYNOPSIS

```
/etc/rain/rld [-d debug_level] [-p port] [-f] [-l log_level]
```

DESCRIPTION

The *rld* server is used for remote management of RAINlink facilities.

OPTIONS

-d debug_level	Sets debug level (1 through 4). Larger number increases verbosity
-p port	Sets TCP port to listen on. Default is 7000
-f	Run daemon in the foreground
-l log_level	Log debug messages at level (1 through 4) to log file /etc/rain/rld.log

FILES

None.

SEE ALSO

rlarm(1M), rlstats(1M)

Release Notes (v2.3.7)

CHANGES SINCE PREVIOUS RELEASE (2.3.5)

- Integrated modifications to the RMAPI library. Files included `rmapi.c`, `psdb_io.c` and `psdb_fio.c`. Fixes included graceful failure if -1 is passed in as an `rmapi_fd`, ignoring `EINTR` from `rmapi_get` and handling failure from `rmapi_get`.
- Fixed HAL multicast delete bug where a zero entry is copied over the removed item. Need to decrement then copy, not copy then decrement. Result was a zero copied and the last entry is lost.
- No longer separating the Solaris high availability versions. (i.e. no more 'N' versions.)
- Fixed bug in `rlk` so that receiving a packet on the standby link does not incorrectly keep the primary link from timing out.

CHANGES SINCE PREVIOUS RELEASE (2.3.4)

- Driver now attaches to ZNYX specific PCI Subdevice and Subvendor ID's. Not "ethernet".
- Fixed interrupt not serviced problem when bringing down an interface.
- Fixed problem with putting more than 4 entries in the per port speed settings.
- Added `SIGUSR` and `SIGUSR2` as valid signals for RMAPI signal mode.

CHANGES SINCE PREVIOUS RELEASE (2.1.9)

- Added support for ZX220, ZX240, ZX420, and ZX370 Series boards.
- Added support for Solaris 8 (x86, sparc 32-bit and sparc 64-bit)
- Discontinued support for 21140-based non-Q cards (ZX345, ZX346, ZX348)
- Added support for RAINlink Management API (RMAPI). Included example applications `rlalarm`, `rlstats`, and `rld`. Included example source in package under `/etc/rain/src`
- Added ability to set individual ports to different speeds and duplex in `/kernel/drv/znb.conf`

CHANGES SINCE PREVIOUS RELEASE (2.0.14)

- Added support for ZX440 and ZX470 Series boards.
- Added support for Solaris 7 (x86, sparc 32-bit and sparc 64-bit)
- Added "hub mode" to RAINlink failover groups. Hub mode limits reception of packets to only the ACTIVE link. This is useful when multiple ports of a failover group to the same collision domain, like a hub. It should not be used when directly connected system-to-system, or system-to-switch.
- Added "-p" option to `rlconfig` utility, to display port information and statistics.
- Fixed bug in older ICS-based PCI cards (non-Q cards) that prevented forced line speeds from working properly.
- Fixed bug in older ICS-based PCI cards (non-Q cards) that prevented autonegotiation with the Netgear FS108 switch.

DISCREPANCIES

- System-to-System trunks with large numbers of members require large system resources, and start to become ineffective uses of resources. We recommend not building trunks larger than 8 ports, with 4 ports offering the most efficient use.
- A trunk or failover group connected to a hub or switch (non-FEC) generates warning messages:

```
WARNING: IP: Hardware address x:x:x:x:x:x trying to  
be our address n.n.n.n
```

You can eliminate this message by setting the failover group in hub mode.

```
failover0=hub_mode;
```

- The driver does not support power management on sparc systems. Type:

```
uadmin 3 8
```

The command causes the system to do a power management suspend. Our driver does not support this type of driver detach and causes the power down to be aborted.

- The Solaris operating system keeps track of which adapter is in which slot. If you move the adapter from one slot to another, or remove an adapter, you will need to remove the package and remove references to the driver in the */etc/path_to_inst* system file.

1. Remove the package:

```
pkgrm ZNYXnb
```

2. Remove references to the znb driver from the file: */etc/path_to_inst*.

```
cp /etc/path_to_inst /etc/path_to_inst.sav  
grep -v znb /etc/path_to_inst.sav > /etc/path_to_inst
```

OTHER HELPFUL COMMANDS

- For version information about the driver, you can use:

```
modinfo | grep znb  
pkginfo -l ZNYXnb
```



ZNYX Networks
48421 Milmont
Fremont, CA 94539 USA

Telephone: (510) 249-0800 or (800) 724-0911
Fax: (510) 656-2460
Email: sales@znyx.com
support@znyx.com
Website: www.znyx.com

Solaris Installation and User's Guide
Document # DC0105-12
July 18, 2001

© 2000-2001 ZNYX Networks. All rights reserved worldwide. Information in this document is subject to change without prior notice. ZNYX Networks, RAIN, RAINlink and NetBlaster are trademarks or registered trademarks of ZNYX Networks in the United States and/or other countries. All other marks, trademarks or service marks are the property of their respective owners.

ZNYX Networks may have patents, pending patent applications, trademarks, copyrights or other intellectual property rights covered in the subject matter of this document. By furnishing this document, ZNYX Networks does not license nor waive its license to those intellectual property rights except as expressly provided in a written license agreement from ZNYX Networks. Information in this document is subject to change without prior notice.