Token–Ring mini–HOWTO

Mike Phillips

mikep@linuxtr.net

Tom Gall

tom_gall@vnet.ibm.com

Mike Eckhoff

Revision History

Revision 5.00	2002–01–23	Revised by: mlp
Updated to reflect the current state of Token Ring with Linux		
Revision 4.3	29 March 2000	Revised by: tg
Brought up to date.		
Revision 4.1	7 January 1998	Revised by: me
v4.1 released by Mike		

This howto is designed to help you get up and running using a Token Ring adaptor to access the network. Generally speaking Section 3 will tell you which driver you need based on the adaptor card you have.

Table of Contents

1. Introduction....1
1.1. Special Thanks....1
1.2. Copyright Information....1
1.3. Disclaimer....2
1.4. New Versions....2
1.5. Credits....2
1.6. Feedback....2

2. Hardware requirements....4

3. Which driver should I use?....6
3.1. Drivers/Adapter Specifics....7
3.1.1. Kernel Module Aliases and Parameters....8
3.1.2. IBMTR Driver....8
3.1.3. Olympic Driver....9
3.1.4. Lanstreamer Driver....10
3.1.5. 3Com 3C359 Driver....11
3.1.6. SysKonnect adapters....13
3.1.7. PCMCIA....15
3.1.8. Madge Supplied Drivers....18
3.1.9. Olicom Drivers....19

4. Known problems....20

5. VMWare and Token Ring....21

6. Commonly asked Questions....22

A. GNU Free Documentation License....23

A.1. 0. PREAMBLE....24

A.2. 1. APPLICABILITY AND DEFINITIONS....25

A.3. 2. VERBATIM COPYING....26

A.4. 3. COPYING IN QUANTITY....27

A.5. 4. MODIFICATIONS....28

A.6. 5. COMBINING DOCUMENTS....30

A.7. 6. COLLECTIONS OF DOCUMENTS....31

A.8. 7. AGGREGATION WITH INDEPENDENT WORKS....32

A.9. 8. TRANSLATION....33

Table of Contents

A.10. 9. TERMINATION.....34

A.11. 10. FUTURE REVISIONS OF THIS LICENSE.....35

1. Introduction

Welcome to the Linux Token Ring mini−howto. We hope you find the information contained within helpful.

If you have any problems with the drivers that are not talked about in this howto, feel free to email me at <mikep@linuxtrnet>.

You may also wish to join the Linux on Token Ring Listserv by mailing <majordomo@linuxtr.net> with the body containing:

```
subscribe linux-tr
```

The latest and greatest information, drivers, patches, bug fixes, etc, etc can always be found at the Linux Token Project site.

1.1. Special Thanks

Thanks to Mark Swanson, Peter De Schrijver, David Morris, Paul Norton and everyone else who has contributed to the Token Ring code and drivers over the years.

Thanks also to the many people and companies who have provided hardware and technical documents to enable the drivers to be written in the first place.

Special Thanks to Mike Eckhoff the originator of this HOWTO, and Tom Gall for the previous version, and to Matthew Marsh for hosting the website and mailing list!

And, finally, thanks to all to subscribers to the linux−tr mailing list who have provided support, feedback, testing and thanks over the years. It wouldn't have been worth it without your continued support and gratitude.

1.2. Copyright Information

This documument is copyright (c) 1995−1998 by Michael Eckhoff, copyright(c) 2000 by Tom Gall and copyright (c) 2001 by Mike Phillips.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation, with no Invariant sections, with no Front−Cover Texts, and with no Back−Cover Texts.

A copy of the license is included in Appendix A

If you have any question, please contact <linux−howto@linuxdoc.org>

1.3. Disclaimer

No liability for the contents of this document can be accepted. Use the concepts, examples and other content at your own risk. As this is a new edition of this document, there may be errors and inaccuracies, that may of course be damaging to your system. Proceed with caution, and although this is highly unlikely, the authors do not take any responsibility for that.

All copyrights are held by their respective owners, unless specifically noted otherwise. Use of a term in this document should not be regarded as affecting the validity of any trademark or service mark.

Naming of particular products or brands should not be seen as endorsements.

You are strongly recommended to take a backup of your system before major installation and backups at regular intervals.

1.4. New Versions

The latest version of this document can always be found at Linux Token Ring Project under the Documentation section.

The latest version of this HOWTO will always be made available on the Linux Token Ring Project website, in a variety of formats:

- HTML,
- Plain text,
- Adobe Acrobat pdf,
- Postscript,
- SGML source,

1.5. Credits

In this version I have the pleasure of acknowleding:

IBM for providing hardware, technical documentation and technical support when the tech docs didn't quite seem enough.

Madge for providing their hardware to test with.

3Com for proving the technical documents to allow the 3c359 driver to be developed.

1.6. Feedback

Feedback is most certainly welcome for this document. Without your submissions and input, this document wouldn't exist. Please send your additions, comments and criticisms to the following email address: <mikep@linuxtr.net>

2. Hardware requirements

Make sure that you have a Token Ring card that is supported from the list below. Many PCI,ISA and even the odd MCA cards are now supported. Check http://www.linuxtr.net for the latest information.

Cards that are reported to work:

3COM

- 3C389 PCMCIA
- 3C619, 3C619B or 3C619C Token Link
- 3C319 Velocity ISA
- 3C359 Velocity XL – PCI
- 3C339 Velocity PCI

IBM

- **PCI.** PCI Token Ring Adapter; PCI Wake on Lan Token Ring Adapter; 16/4 Token Ring PCI Adapter 2, Wake on Lan, and Wake on Lan Special; High Speed 100/16/4 Token Ring Adapter, Token Ring 16/4 Management Adapter.
- **Cardbus.** 16/4 Token Ring Adapter
- **LanStreamer.** PCI: Auto LanStreamer, Triple Lanstreamer; MCA: LanStreamer MC16, Lanstreamer MC32, AutoLanstreamer MC32, Dual Lanstreamer MC32
- **ISA.** Auto 16/4 Token Ring Adapter, 16/4 Token Ring Adapter, Turbo 16/4 Token Ring Adapter, Auto Wake Token Ring Adapter.
- **PCMCIA.** Turbo 16/4 PC Card, Turbo 16/4 PC Card 2, Auto 16/4 Credit Card Adapter, 16/4 Credit Card Adapter, 16/4 Credit Card Adapter II
- **Tropic MCA.** 16/4 Token Ring Adapter/A, Auto 16/4 Token Ring Adapter

Olicom

- RapidFire 3139, 3140, 3141, and 3540
- OC 3136
- OC 3137
- OC 3118
- OC 3129

Madge

- 51–02 Smart 16/4 PCI
- 20–03 16/4 Cardbus Adapter Mk2
- 51–04 Smart 16/4 PCI Ringnode Mk3
- 51–09 Smart 16/4 Fiber PCI Ringnode
- 51–07 Smart 100/16/4 PCI–HS Ringnode
- 51–05 Smart 100/16/4 PCI Ringnode
- 20–01 Smart 16/4 PCMCIA
- 60–07 Presto PCI 2000
- 60–06 Presto PCI Plus
- 60–05 Presto PCI
- 53–05 Smart Mk4 PCI Adapter (low profile)

- 31−40 Rapidfire 3140V2 16/4 PCI Adapter

SysKonnect

- TR4/16(+) SK−4190 ISA
- TR4/16(+) SK−4590 PCI
- TR4/16(+) SK−4591 PCI

SMC

- Tokencard Elite (8115T)
- Tokencard Elite/A MCA (8115T/A)

Intel

- TokenExpress PRO
- TokenExpress 16/4

Cards that may cause problems:

Token−Ring Network 16/4 Adapter II. This adapter will NOT work. Do not confuse this card with the IBM Token Ring adapter II (4mbit) which does. It is a DMA/Busmaster adapter for ISA.

3Com TokenLink Velocity ISA. You may or may not get this one to work. I have had reports of people running it without problems, and others who get errors left and right.

---

3. Which driver should I use?

The realm of Token Ring drivers on Linux has expanded quite a bit in last couple of years. It's not just ibmtr anymore! So as a result this map will tell you given a card which driver you should try and the recommended minimum kernel version (if any).

3COM

- 3C389 PCMCIA –– ibmtr_cs
- 3C619, 3C619B or 3C619C Token Link –– ibmtr
- 3C319 Velocity ISA –– try ibmtr
- 3C359 Velocity XL – PCI –– driver available from http://www.linuxtr.net
- 3C339 Velocity PCI –– tms380tr

IBM

- PCI Token Ring Adaptor –– olympic
- PCI Wake on Lan Token Ring Adaptor –– olympic
- 16/4 Token Ring PCI Adaptor 2, Wake On Lan, and Wake on Lan Special –– olympic
- High Speed 100/16/4 Token Ring –– olympic
- Turbo 16/4 ISA adapter –– ibmtr
- Token Ring Auto 16/4 ISA adapter –– ibmtr
- Token Ring Auto 16/4 adapter /A –– ibmtr
- Token Ring 16/4 adapter /A –– ibmtr
- Token Ring adapter /A –– ibmtr
- Token Ring adapter II (4 Megabit only) –– ibmtr
- 16/4 ISA Token Ring card (16bit) –– ibmtr
- 16/4 ISA Token Ring card (8bit) –– ibmtr
- All LANStreamer –– lanstreamer
- PCMCIA – Turbo 16/4 –– ibmtr_cs
- PCMCIA – 16/4 –– ibmtr_cs
- Cardbus – 16/4 – olympic, kernel v.2.4.3 or greater

Olicom

- RapidFire 3139, 3140, 3141, and 3540
- OC 3136
- OC 3137
- OC 3118
- OC 3129

For these Olicom cards, see their website http://www.olicom.com for drivers. You will need a 2.2.x series kernel.

Madge

- 51–02 Smart 16/4 PCI
- 20–03 16/4 Cardbus Adapter Mk2
- 51–04 Smart 16/4 PCI Ringnode Mk3
- 51–09 Smart 16/4 Fiber PCI Ringnode
- 51–07 Smart 100/16/4 PCI–HS Ringnode

- 51−05 Smart 100/16/4 PCI Ringnode
- 20−01 Smart 16/4 PCMCIA
- 60−07 Presto PCI 2000
- 60−06 Presto PCI Plus
- 60−05 Presto PCI

For these Madge cards you'll want to visit their site http://www.madge.com for drivers and get the 2.31 Madge drivers. You will need either a 2.0.36 or 2.2.5 as a minimum.

2.41 drivers:

- 51−05 Smart Mk4 PCI Adapter
- 53−05 Smart Mk4 PCI Adapter (low profile)
- 31−40 Rapidfire 3140V2 16/4 PCI Adapter
- 20−03 Smart 16/4 Cardbus Mk2
- 51−04 Smart 16/4 PCI Ringnode Mk3
- 60−07 Presto PCI 2000
- 60−06 Presto PCI Plus
- 60−05 Presto PCI

According to the Madge README file the 2.41 driver has been tested on uniprocessor and SMP kernel versions: 2.0.36, 2.2.5−15 ,2.2.10, 2.2.12−20, 2.4.2−2.

Other Madge cards are reportedly based on the Texas Instruments tms380 chipset and thus as of the 2.3.26 kernel you can try the tms380tr driver.

SysKonnect

- TR4/16(+) SK−4190 ISA
- TR4/16(+) SK−4590 PCI
- TR4/16(+) SK−4591 PCI

In the 2.2.x series of kernels try sktr. In the 2.3.x and greater series try the tms380tr driver.

SMC

- Tokencard Elite (8115T)
- Tokencard Elite/A MCA (8115T/A)

Driver is included as part of the 2.3.38+ kernel.

Intel

- TokenExpress PRO
- TokenExpress 16/4

Support for these cards is currently under development. Check http://www.linuxtr.net for status.

3.1. Drivers/Adapter Specifics

Here we'll describe the different options and configurations available for each of the available drivers.

3.1.1. Kernel Module Aliases and Parameters

Most drivers accept arguments in the form of module paramters (with the exception of the special case of PCMCIA, which is fully described below).

Kernel modules are specified in the file /etc/conf.modules or /etc/modules.conf depending upon which version of modutils you've got.

You can directly modify this file or use the tools builtin to your specific distribution. These distribution specific tools are beyond the scope of this document, but you can always directly modify the modules.conf file by hand to get things up and running and then figure out how your distribution handles these files. For example, Debian has several files in the /etc/modutils directory and from these builds the modules.conf file.

Kernel modules aliases are utilized to associate a particular name with a kernel module.

For token ring, this is used to assign drivers for each of the token ring interfaces so that the system scripts know which driver to insert when you bring an interface up.

The format of the alias lines are:

```
alias interface module_name
```

Usually, the only line you'll need for the token ring networking would be something like:

```
alias olympic tr0
```

This binds the olympic driver to the tr0 interface so when you type

```
ifconfig tr0 up
```

if the tr0 interface is not already loaded, the system will insert the olympic driver, which in turn will find the network card and create the tr0 network device.

Kernel modules parameters are specified in the following format:

```
options module_name parameter_1=XXX [parameter2=YYY ...]
```

Where the modules_name is the name of the driver, i.e. olympic, ibmtr, 3c359 and the ` parameters are those available for each driver. See either the following sections for driver specifics or check out the drivers source code.

For example, if you wanted to set the Olympic driver to 16 mbps operation and with a default buffer size of 8192 bytes, you would use the following line:

```
options olympic ringspeed=16 pkt_buf_sz=8192
```

3.1.2. IBMTR Driver

IBM Tropic Chipset Based Token Ring Adapters

This is the original token ring driver in the kernel and supports almost all adapters that use the IBM Tropic chipset, including the IBM ISA, ISA/Pnp, and a multitude of adapters from other manufacturers.

The IBM Turbo 16/4 ISA/PnP adapter will, in fact, work fine with the ibmtr driver. In older drivers you had to run the card in Auto 16/4 compatability mode. The simplest way to set this is to use the LANAID disks sent with the card and run the command:

```
LANAIDC /FAST=AUTO16
```

You should then use LANAIDC or LANAID to configure the card according to documentation. The latest drivers for the Turbo Adapters will recognize these adapters and configure them straight out of the box. You may have to either turn off isapnp support in the kernel or modify your isapnp.conf file to enable the adapter.

Options:

Perusal of the ibmtr source code may leave you to believe that the adapter can take three parameters, however, in reality the driver doesn't take any. These parameters are a hang over from the early stages of the driver and are only intended to be used to force the driver to only test restricted åddresses when looking for adapters. The information on these options are included here for completeness only.

- *io*: Specify the I/O ports that the driver will check for the presence of any cards. All Tropic based ISA adapters, or adapters emulating the ISA cards will be found on either port 0xA20 or 0xA24. If you know that your adapter is configured for 0xA24 and/or that probing on port 0xA20 will cause problems with your machine, use io to force the driver to check a specific port only.

  The Turbo adapters (including the confusingly named latest Auto 16/4 cards) can have their io regions located anywhere permitted by the PnP specification. This location is found using the new turbo detection code and no parameters are required.

- *irq & mem*: The two options were used to tell the driver exactly which irq to use and where the shared ram for the adapter could be found. These two options are now totally redundant in the driver as the interrupt line and the location of the shared ram is obtained directly by interrogating the adapter.

3.1.3. Olympic Driver

IBM PCI Pit/Pit−Phy/Olympic chipset based token ring cards

Options:

The driver accepts four options: ringspeed, pkt_buf_sz, message_level and network_monitor.

These options can be specified differently for each card found, i.e if you have two olympic adapters in your machine and want to assign a ring speed of 16mbps to the first adapter, but a ring speed of 4mbps to the second adapter, your options line would read:

```
options olympic ringspeed=16,4
```

However, it should be noted that the driver assigns value to each adapter in the order they are discovered¸ which is usually the order there are present on the pci bus. A little trial and error may be required to be certain which adapter is receiving which configuration option.

- *ringspeed*: Has one of three settings 0 (default), 4 or 16. 0 will make the card autosense the ringspeed

and join at the appropriate speed, this will be the default option for most people. 4 or 16 allow you to explicitly force the card to operate at a certain speed. The card will fail if you try to insert it at the wrong speed. (Although some hubs will allow this so be *very* careful). The main purpose for explicitly setting the ring speed is for when the card is first on the ring. In autosense mode, if the card cannot detect any active monitors on the ring it will not open, so you must re−init the card at the appropriate speed. Unfortunately at present the only way of doing this is rmmod and insmod which is a bit tough if it is compiled in the kernel. The driver does support 100 mbps full duplex operation. This is automatically detected by the adapter when connected to an appropriate switch.

- *pkt_buf_sz*: This is this initial receive buffer allocation size. This will default to 4096 if no value is entered. You may increase performance of the driver by setting this to a value larger than the network packet size, although the driver now re−sizes buffers based on MTU settings as well.
- *message_level*: Controls level of messages created by the driver. Defaults to 0 which only displays start−up and critical messages. Presently any non−zero value will display all soft messages as well. NB This does not turn debugging messages on, that must be done by modified the source code.
- *network_monitor*: Any non−zero value will provide a quasi network monitoring mode. All unexpected MAC frames (beaconing etc.) will be received by the driver and the source and destination addresses printed. Also an entry will be added in /proc/net called olympic_tr%d, where tr%d is the registered device name, i.e tr0, tr1, etc. This displays low level information about the configuration of the ring and the adapter. This feature has been designed for network administrators to assist in the diagnosis of network / ring problems. (This used to OLYMPIC_NETWORK_MONITOR, but has now changed to allow each adapter to be configured differently and to alleviate the necessity to re−compile olympic to turn the option on).

Multi−card. The driver will detect multiple cards and will work with shared interrupts, each card is assigned the next token ring device, i.e. tr0 , tr1, tr2. The driver should also happily reside in the system with other drivers. It has been tested with ibmtr.c running. I have had multiple cards in the same system, all sharing the same interrupt and working perfectly fine together. This is also true for the Cardbus Olympic adapters, I have quite happily had a Cardbus adapter and regular 16 bit PCMCIA token ring adapter working together in the same laptop.

Variable MTU size:. The driver can handle a MTU size upto either 4500 or 18000 depending upon ring speed. The driver also changes the size of the receive buffers as part of the mtu re−sizing, so if you set mtu = 18000, you will need to be able to allocate 16 * (sk_buff with 18000 buffer size) call it 18500 bytes per ring position = 296,000 bytes of memory space, plus of course anything necessary for the tx sk_buff's. Remember this is per card, so if you are building routers, gateway's etc, you could start to use a lot of memory real fast.

3.1.4. Lanstreamer Driver

IBM PCI/MCA Lanstreamer chipset based token ring cards

Options:

The driver accepts three options: ringspeed, pkt_buf_sz, message_level and network_monitor.

These options can be specified differently for each card found, i.e if you have two olympic adapters in your machine and want to assign a ring speed of 16mbps to the first adapter, but a ring speed of 4mbps to the second adapter, your options line would read:

```
options lanstreamer ringspeed=16,4
```

```
```

However, it should be noted that the driver assigns value to each adapter in the order they are discovered¸ which is usually the order there are present on the pci/mca bus. A little trial and error may be required to be certain which adapter is receiving which configuration option.

- *ringspeed*: Has one of three settings 0 (default), 4 or 16. 0 will make the card autosense the ringspeed and join at the appropriate speed, this will be the default option for most people. 4 or 16 allow you to explicitly force the card to operate at a certain speed. The card will fail if you try to insert it at the wrong speed. (Although some hubs will allow this so be *very* careful). The main purpose for explicitly setting the ring speed is for when the card is first on the ring. In autosense mode, if the card cannot detect any active monitors on the ring it will not open, so you must re−init the card at the appropriate speed. Unfortunately at present the only way of doing this is rmmod and insmod which is a bit tough if it is compiled in the kernel. switch.
- *pkt_buf_sz*: This is this initial receive buffer allocation size. This will default to 4096 if no value is entered. You may increase performance of the driver by setting this to a value larger than the network packet size, although the driver now re−sizes buffers based on MTU settings as well.
- *message_level*: Controls level of messages created by the driver. Defaults to 0 which only displays start−up and critical messages. Presently any non−zero value will display all soft messages as well. NB This does not turn debugging messages on, that must be done by modified the source code.

Network Monitor. The Lanstreamer driver does support a network monitor mode similar to the olympic driver, however it is a compile time option and not a module parameter. To enable the network monitor mode, edit lanstreamer.c and change the line:

```
#define STREAMER_NETWORK_MONITOR 0
```

to read:

```
#define STREAMER_NETWORK_MONITOR 1
```

All unexpected MAC frames (beaconing etc.) will be received by the driver and the source and destination addresses printed. Also an entry will be added in /proc/net called streamer_tr. This displays low level information about the configuration of the ring and the adapter. This feature has been designed for network administrators to assist in the diagnosis of network / ring problems.

Multi−card. The driver will detect multiple cards and will work with shared interrupts, each card is assigned the next token ring device, i.e. tr0 , tr1, tr2. The driver should also happily reside in the system with other drivers.

Variable MTU size:. The driver can handle a MTU size upto either 4500 or 18000 depending upon ring speed. The driver also changes the size of the receive buffers as part of the mtu re−sizing, so if you set mtu = 18000, you will need to be able to allocate 16 * (sk_buff with 18000 buffer size) call it 18500 bytes per ring position = 296,000 bytes of memory space, plus of course anything necessary for the tx sk_buff's. Remember this is per card, so if you are building routers, gateway's etc, you could start to use a lot of memory real fast.

3.1.5. 3Com 3C359 Driver

3COM PCI TOKEN LINK VELOCITY XL TOKEN RING CARDS

Currently the 3c359 driver in not included in the standard kernel source. To utlize the driver, you must download the driver from the Linux Token Ring Project web site and patch your kernel.

Once you've downloaded the file, you can patch your kernel with the following commands:

```
cd /usr/src/linux
patch -p1 < 3c359-2.4.16.patch
```

or, if the patch file is gzipped:

```
zcat 3c359-2.4.16.patch | patch -p1
```

Then just run make config|menuconfig|xconfig and select the 3c359 driver from the token ring drivers section of the kernel configuration and then compile and install the kernel and/or modules as usual.

Options:

The driver accepts three options: ringspeed, pkt_buf_sz, message_level.

These options can be specified differently for each card found, i.e if you have two olympic adapters in your machine and want to assign a ring speed of 16mbps to the first adapter, but a ring speed of 4mbps to the second adapter, your options line would read:

```
options 3c359 ringspeed=16,4
```

However, it should be noted that the driver assigns value to each adapter in the order they are discovered¸ which is usually the order there are present on the pci bus. A little trial and error may be required to be certain which adapter is receiving which configuration option.

- *ringspeed*: Has one of three settings 0 (default), 4 or 16. 0 will make the card autosense the ringspeed and join at the appropriate speed, this will be the default option for most people. 4 or 16 allow you to explicitly force the card to operate at a certain speed. The card will fail if you try to insert it at the wrong speed. (Although some hubs will allow this so be *very* careful). The main purpose for explicitly setting the ring speed is for when the card is first on the ring. In autosense mode, if the card cannot detect any active monitors on the ring it will open at the same speed as its last opening. This can be harardous if this speed does not match the speed you want the ring to operate at.
- *pkt_buf_sz*: This is this initial receive buffer allocation size. This will default to 4096 if no value is entered. You may increase performance of the driver by setting this to a value larger than the network packet size, although the driver now re−sizes buffers based on MTU settings as well.
- *message_level*: Controls level of messages created by the driver. Defaults to 0 which only displays start−up and critical messages. Presently any non−zero value will display all soft messages as well. NB This does not turn debugging messages on, that must be done by modified the source code.

Multi−card. The driver will detect multiple cards and will work with shared interrupts, each card is assigned the next token ring device, i.e. tr0 , tr1, tr2. The driver should also happily reside in the system with other drivers. It has been tested with ibmtr.c running. I have had multiple cards in the same system, all sharing the same interrupt and working perfectly fine together.

Variable MTU size:. The driver can handle a MTU size upto either 4500 or 18000 depending upon ring speed. The driver also changes the size of the receive buffers as part of the mtu re−sizing, so if you set mtu = 18000, you will need to be able to allocate 16 * (sk_buff with 18000 buffer size) call it 18500 bytes per ring position = 296,000 bytes of memory space, plus of course anything necessary for the tx sk_buff's. Remember this is per card, so if you are building routers, gateway's etc, you could start to use a lot of memory real fast.

3.1.6. SysKonnect adapters

Information for the SysKonnect Token Ring ISA/PCI Adapter is courtesy Jay Schulist <jschlst@samba.org>

The Linux SysKonnect Token Ring driver works with the SysKonnect TR4/16(+) ISA, SysKonnect TR4/16(+) PCI, SysKonnect TR4/16 PCI, and older revisions of the SK NET TR4/16 ISA card.

Latest information on this driver can be obtained on the Linux−SNA WWW site. Please point your browser to: http://www.linux−sna.org

Important information to be noted:

- 1. Adapters can be slow to open (~20 secs) and close (~5 secs), please be patient.
- 2. This driver works very well when autoprobing for adapters. Why even think about those nasty io/int/dma settings of modprobe when the driver will do it all for you!

This driver is rather simple to use. Select Y to Token Ring adapter support in the kernel configuration. A choice for SysKonnect Token Ring adapters will appear. This drives supports all SysKonnect ISA and PCI adapters. Choose this option. I personally recommend compiling the driver as a module (M), but if you you would like to compile it staticly answer Y instead.

This driver supports multiple adapters without the need to load multiple copies of the driver. You should be able to load up to 7 adapters without any kernel modifications, if you are in need of more please contact the maintainer of this driver.

Load the driver either by lilo/loadlin or as a module. When a module using the following command will suffice for most:

```
# modprobe sktr
```

This will produce output similar to the following: (Output is user specific)

```
sktr.c: v1.01 08/29/97 by Christoph Goos
tr0: SK NET TR 4/16 PCI found at 0x6100, using IRQ 17.
tr1: SK NET TR 4/16 PCI found at 0x6200, using IRQ 16.
tr2: SK NET TR 4/16 ISA found at 0xa20, using IRQ 10 and DMA 5.
```

Now just setup the device via ifconfig and set and routes you may have. After this you are ready to start sending some tokens.

Errata. For anyone wondering where to pick up the SysKonnect adapters please browse to http://www.syskonnect.com

Below is the setting for the SK NET TR 4/16 ISA adapters

```
    ***************************
    ***   C O N T E N T S   ***
    ***************************

1) Location of DIP-Switch W1
2) Default settings
3) DIP-Switch W1 description
```

```
          ============================================================
          CHAPTER 1     LOCATION OF DIP-SWITCH
          ============================================================

          +-------------------------------------------------------------------+
          |+------+                          +-----+           +---+          |
          ||------|                      W1 +-----+     +----+ |   |          |
          ||------|                                     |    | |   |       +---+
          ||------|              +-----------+          +----+ |   |       |  ||
          ||------|              |           |          +---+  +---+       +---+
          ||------|              | TMS380C26 |          |   |                |
          |+------+              |           |          +---+                |-+
          |                      |           |                               | |
          |                      +-----------+                               | |
          |                                                                  | |
          |                                                                  |-+
          |                                                                  |
          |                                                                  |
          |                                                                  |
          |                                                                  |
          +------------+----------------+--+----------------------+---------+
                       +----------------+  +----------------------+
```

```
          ============================================================
          CHAPTER 2     DEFAULT SETTINGS
          ============================================================

    W1    1  2  3  4  5  6  7  8
  +-------------------------------+
  | ON    X                       |
  | OFF      X  X  X  X  X  X  X |
  +-------------------------------+

  W1.1 = ON               Adapter drives address lines SA17..19
  W1.2 - 1.5 = OFF        BootROM disabled
  W1.6 - 1.8 = OFF        I/O address 0A20h
```

```
          ============================================================
          CHAPTER 3     DIP SWITCH W1 DESCRIPTION
          ============================================================

  +---+---+---+---+---+---+---+---+  ON
  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
  +---+---+---+---+---+---+---+---+  OFF
  |AD | BootROM Addr. |  I/O      |
  +-+-+-------+-------+-----+-----+
    |         |             |
    |         |             +------ 6     7     8
    |         |                     ON    ON    ON       1900h
    |         |                     ON    ON    OFF      0900h
    |         |                     ON    OFF   ON       1980h
    |         |                     ON    OFF   OFF      0980h
    |         |                     OFF   ON    ON       1b20h
    |         |                     OFF   ON    OFF      0b20h
    |         |                     OFF   OFF   ON       1a20h
    |         |                     OFF   OFF   OFF      0a20h    (+)
    |         |
    |         |
    |         +-------- 2     3     4     5
```

```
        |                   OFF   x     x     x       disabled  (+)
        |                   ON    ON    ON    ON      C0000
        |                   ON    ON    ON    OFF     C4000
        |                   ON    ON    OFF   ON      C8000
        |                   ON    ON    OFF   OFF     CC000
        |                   ON    OFF   ON    ON      D0000
        |                   ON    OFF   ON    OFF     D4000
        |                   ON    OFF   OFF   ON      D8000
        |                   ON    OFF   OFF   OFF     DC000
        |
        |
        +----- 1
               OFF    adapter does NOT drive SA<17..19>
               ON     adapter drives SA<17..19>  (+)


        (+) means default setting
```

3.1.7. PCMCIA

3.1.7.1. Introduction

PCMCIA Token Ring adapters will work on all versions of the Linux kernel. Unfortunately, the road to hell is often paved with melting snowballs ;−) and there are a myriad of different combinations that can be used to get the adapters to work, all with different options, different requirements and different issues. Hopefully with this document you will be able to figure out which combinations of ingredients are required and how to get them up and running on your machine.

3.1.7.2. History

In the 2.0.x and 2.2.x kernels days, pcmcia was only available as an external package, created and maintained by David Hinds. When the only stable kernel available was 2.0.36, life was pretty easy and with a few simple configuration options the adapters would work.

With the advent of 2.2.x, ibmtr.c was completely updated, which broke the pcmcia driver (ibmtr_cs.c). The pcmcia driver was updated to work with the new ibmtr driver and the 2.2.x kernels. This is where the first level of complication starts. As the pcmcia_cs package is stand alone, it has to support the various different kernels, so instead of being able to have different versions of drivers in different versions of the kernel source, the pcmcia_cs drivers must work with all kernel versions. This not only creates some ugliness in the driver itself but also causes confusion as to which version of pcmcia_cs works for the latest kernel.

At this point, everything was working fine, and then come along the 2.3.x develpment series of kernels. The 2.3.x kernels provided their own support for pcmcia and the ibmtr_cs driver was included in the kernel proper. So now there were two ways of getting pcmcia token ring support, either using the kernel drivers themselves or using the pcmcia_cs package, not too much of a problem because only developers were using the 2.3.x kernels. Of course this all changed when the 2.4 kernel was released and a lot more users started using the kernel.

During late 2000, early 2001, significant development work was done on both the standard ibmtr driver and

the pcmcia driver. Original pcmcia updates including using high memory and hot−eject support. These initial updates were only for the 2.2.x kernels, and hence only included in the pcmcia_cs package. Later development saw great improvements in ibmtr and ibmtr_cs for the 2.4.x kernels. So as of writing, 1/23/02 , there are many different combinations of kernel version and driver floating around especially considering that different distributions have released different versions of the 2.4 kernels.

3.1.7.3. 2.0.x kernels

If you are using one of the 2.0.x kernels, then I salute your perserverance and really you should have got the pcmcia drivers configured and working by now ;−)

You will have to use the pcmcia_cs package and play with the /etc/pcmcia/config.opts, see the section below about config.opts fun. Just about any version of pcmcia_cs that's been released in the last 2/3 years will work fine.

3.1.7.4. 2.2.0 – 2.2.6 kernels

These were the series of kernels where the pcmcia driver didn't work at all. It's probably just easiest to upgrade the kernel to a later version.

If you really do need to get this up and running, then a recent pcmcia_cs is required and you should be able to grab the ibmtr.c and ibmtr.h from a 2.2.7 − 2.2.16 kernel and use them (note no greater than 2.2.16 !!)

You have to do the config.opts mangling, see the section on setting all this up.

3.1.7.5. 2.2.7 – 2.2.16 kernels

These kernels are well supported, simply use the pcmcia_cs package and play with the config.opts file.

3.1.7.6. 2.2.17 – 2.2.19 kernels

The pcmcia driver was updated for these kernel to eliminate the need for the config.opts mangling. You'll need pcmcia_cs at least 3.1.24, although it is probably better just to grab the latest version.

Simply compile up pcmcia_cs and you're done. No need to play with config.opts, in fact if you've been running a previous version that did have the ibmtr_cs line in config.opts it would be a *very good* idea to remove or comment out the line. The new driver allocates the entire 64k for shared ram and it needs to be aligned on a 64k boundary, if you've got a previous srambase value not on a 64k boundary, the driver will barf and the kernel will panic.

3.1.7.7. 2.4.0 – 2.4.4 (non Redhat) kernels

Use the built−in kernel pcmcia driver and play with config.opts.

If you want to use the latest and greatest version of the driver with the high memory and hot−swap support you can download the patch and patch up your kernel. Then the line in config.opts can be removed and everything will work fine.

3.1.7.8. 2.4.4−ac11 > kernels

These kernels include the new drivers so simply compile up the drivers, ensure that there is no configuration line in config.opts and away you go.

3.1.7.9. 2.4.2 mangled, i.e. Redhat 7.1

When RedHat released 7.1 with the 2.4.2 kernel they modified the kernel (as they always do) and included the updated ibmtr/ibmtr_cs driver from the web site. If you're lucky this may work straight out of the box (again no need for the ibmtr_cs line in config.opts), if not then it is probably easiest to upgrade to the latest 2.4.x kernels and use the drivers there. (The reason being that while I will work out how to get around a distribution caused problem, I will not provide support for them, I'll answer questions and give help because I'm a nice guy, but I am not going to provide driver updates against distributions. Official support is for the drivers in the kernels available from the official kernel mirrors.

3.1.7.10. 2.4.x kernels and pcmcia_cs

There is no need to use pcmcia_cs with the 2.4 kernels to get the token ring adapters up and running, but I appreciate that some of you may need to use pcmcia_cs to get other adapters working that are not supported properly in the kernel.

The pcmcia_cs package will not work with the latest drivers, it may work with the 2.4.0−2.4.4 drivers. I am currently in two minds about providing support with pcmcia_cs for the 2.4 kernels, you can ask me directly or check the web site every now and then so see if anything has changed.

3.1.7.11. Config.opts mangling (or how to send yourself insane)

This is the hardest part to getting the pcmcia adapters working with the drivers that need the ibmtr_cs line in /etc/pcmcia/config.opts. No set of values is guaranteed to work the same on a different machine. It really is a case of trial and error but forewarned and forearmed with a little bit of knowledge can make the process a whole lot easier.

"Hey, I don't care, just give me something that works"

OK, try this, it works in most situations, if it doesn't you have to read the rest of the section anyway. Just insert the following line in /etc/pcmcia/config.opts

```
modules "ibmtr_cs" opts "mmiobase=0xd2000 srambase=0xd4000"
```

restart pcmcia and insert the adapter.

"OK, that didn't work, bring on the pain"

The pcmcia driver need to allocate two areas of memory to operate properly. All areas of memory allocated must be aligned on the same boundary as the size of the area being aligned, i.e. a block 8K in size must be on an 8K boundary (0xc8000, 0xca000, 0xcc000, 0xce000, 0xd0000, 0xd2000) and for a 16K block must be on a 16K boundary (0xc8000, 0xcc000, 0xd0000, 0xd4000). All memory areas must be allocated within the ISA address space, 0xC0000−0xDFFFF). Theoretically you should be able to use anywhere within this area, although experience has shown that most machines hide stuff in the 0xc0000−0xc9fff area. Some machines have even been known to use the 0xd0000−0xd1fff area without telling anybody (some thinkpads !!). So you really want to stick with memory allocations in the 0xcc000 − 0xdffff range.

Of course, the two memory areas cannot overlap either ;)

The first area of memory is an 8K area for the memory mapped input/output (MMIO) and must be placed on an 8K boundary. This area of memory is not usually the cause of any problems and can be placed pretty much anywhere, recommended values are: 0xcc000, 0xd0000,0xd2000,0xd4000.

The second area of memory can be sized to fit your desires, this is the area of memory where the incoming and outgoing packets are stored and received. The driver defaults to a 16K memory size and must be placed on a 16K boundary. Good areas are: 0xd0000,0xd4000,0xd8000.

Once you've decided which areas of memory you are goin to try, you need to add the correct line to the /etc/pcmcia/config.opts file. Configuration lines in this file take the format of:

```
module "module_name" opts "option1=opt1_value option2=opt2_value ...."
```

In our case module_name is ibmtr_cs. There are three options that be set with the ibmtr_cs driver, mmiobase, srambase and sramsize.

If they are not set they will revert to the defaults in the driver, which in 9 cases out of 10 won't work for you. sramsize rarely has to be set unless you are looking for that last little bit of performance from your adapter.

So, having decided upon your values, let's say 0xd2000 for the MMIO and 0xd4000 for the shared memory you would build a config.opts line like this:

```
module "ibmtr_cs" opts "mmiobase=0xd2000 srambase=0xd4000"
```

The pcmcia_cs package must be restarted for these new options to take effect, usually with:

```
/etc/init.d/pcmcia restart or /etc/rc.d/init.d/pcmcia/restart
```

depending upon which run level organization your distribution adheres to.

Then just plug it in and see if it works. If not you'll just have to go back and change the values for mmiobase and srambase until you find a combination that works. Or, you can upgrade to a kernel/pcmcia_cs version that support high memory allocation, where all this config.opts nonsense is not required and you can just happily plug your adapter in and watch it run.

3.1.8. Madge Supplied Drivers

Madge released 2.31 of their driver in 1999 and 2.41 in late 2001. Both drivers can be downloaded from the Madge web site and the 2.41 driver is also available from the Linux Token Ring Project web site.

Once the drivers have been downloaded, see the README file that comes with the drivers for instruction on how to built and install the drivers. The only other issue some people find with the drivers is a failure to build the tool chain due to an incorrect version of the newt libraries. If you get a compiler error relating to newt.h change the madge−source/include/mtok/config.h file so that the #define NEWNEWT line reads:

```
#define NEWNEWT 1
```

This will ensure the tools use the correct newt libraries during the build process.

A patch is available from the Linux Token Ring Project web site for the 2.31 drivers to enable them to work with the 2.4.x kernels.

3.1.9. Olicom Drivers

Back when Olicom were still in business they did produce a Linux driver that does actually work. Trying to find the driver these days is a bit tough. If the ftp.olicom.com site is still up and running, the driver can be found there.

The driver is a combination of GPL source code and proprietary binary low level code. The driver only works with the 2.0.36 and 2.2.x kernels. It should be possible to port this driver to the 2.4.x kernels...

---

4. Known problems

See www.linuxtr.net for the latest greatest set of bugs. Generally speaking the biggest problem that I've seen (with ibmtr) is that if you pull your connection from the wall the 2.0.x series of kernels would generally not recover.

This has been fixed in the latest version of ibmtr and the driver should now recognize when the link cable has been detached.

There are some laptops that don't want to work with the Olympic Cardbus adapter, for some reason the driver never sees the open interrupt from the card. I don't think this is a problem with the driver, but with the Cardbus subsystem, for some people this problem has simply gone away with a newer kernel and I personally have never seen it on the laptops I've used in the development of the driver (Sony Vaio Z505 and Dell Latitude CPx500).

---

5. VMWare and Token Ring

Thanks to Scott Russell scottrus@raleigh.ibm.com for this little "trick"

One of the bummers about VMWare is if you are on a Token−Ring adapter, your VMWare system can't have a real TCP/IP address. Turns out this isn't the case. Here's how to do it.

- In the info below we'll call your linux box 'linux.mycompany.biz.com'
- Register another ip address, I'll call it 'vmware.mycompany.biz.com'
- Make sure FORWARD_IPV4=true in your /etc/sysconfig/network file. If you have to change it you can dynamically turn on the feature as root

```
cat 1 > /proc/sys/net/ipv4/ip_forward
```

- Alias the second ip to the TR adapter. You end up with something like this from /sbin/ifconfig:

```
tr0 linux.mycompany.biz.com
tr0:0 vmware.mycompany.biz.com
vmnet1 192.168.0.1
```

- Make sure you can ping both ip addresses from another box. If you cannot then this next step will not work.
- Use ipchains/iptables to redirect incoming traffic for the tr0:0 interface to your vmnet1 interface. (When I did this I only redirected specific ports from tr0:0 to vmnet1.)

Now any outside system your 'NT' box appears to be on the TR. In bound traffic can find it as well as out.

---

6. Commonly asked Questions

Here are a collection of commonly asked questions that arise from time to time on the linux−tr mailing list. If your question isn't answered here or elsewhere in this document, feel free to ask away on the mailing list.

Q: DHCP doesn't work with my Token Ring adapter.
Q: I can't set the LAA on my adapter with ifconfig tr0 hw tr 4000DEADBEEF.
Q: My Linux machine is on a bridged network and I'm having connectivity issues with machine beyond the bridge.
Q: Can I use a Linux machine to bridge between token ring and ethernet ?
Q: OK, if I can't bridge, how do I connect my Token Ring and ethernet networks ?

Q: DHCP doesn't work with my Token Ring adapter.

A: Certain dhcp servers and clients do not work properly with token ring drivers. This is especially true with the 2.4 kernels. During the development of the 2.3.x series of kernels the internal type for token ring was changed to accomodate multicast support over token ring. The solution is to upgrade your dhcp client/server to a version that supports token ring and/or the latest kernel versions.

Q: I can't set the LAA on my adapter with ifconfig tr0 hw tr 4000DEADBEEF.

A: Firstly, double check that your adapter/driver support setting the LAA, and that you've supplied a valid LAA. Also, most drivers will only allow this to be set before the adapter is opened onto the ring. Again, this is related to the change in the internal type for token ring in the 2.4 kernels. A patch is available from the web site for nettools that fixes this and allows the LAA to be set.

Q: My Linux machine is on a bridged network and I'm having connectivity issues with machine beyond the bridge.

A: The token ring source routing code in the kernel uses the spanning tree algorithm. Contact your network administrator to enable this protocol on the bridges.

Q: Can I use a Linux machine to bridge between token ring and ethernet ?

A: The simply answer in no. Briding network topologies in software is incredibly complicated and while it is possibly, nobody has written the code to do it. If you must bridge there are several manufacturers that produce hardware bridges (most notably Cisco).

Q: OK, if I can't bridge, how do I connect my Token Ring and ethernet networks ?

A: A cheap linux box with a token ring and ethernet adapter makes an excellent router. There is no difference between setting up a token ring/ethernet router and an ethernet/ethernet router. You can do masquerading (NAT) and filtering on the router as per usual. For more details see the Netfilter howto.

A. GNU Free Documentation License

A.1. 0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other written document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

A.2. 1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you".

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front−matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.

The "Cover Texts" are certain short passages of text that are listed, as Front−Cover Texts or Back−Cover Texts, in the notice that says that the Document is released under this License.

A "Transparent" copy of the Document means a machine−readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup has been designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard−conforming simple HTML designed for human modification. Opaque formats include PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine−generated HTML produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A.3. 2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

A.4. 3. COPYING IN QUANTITY

If you publish printed copies of the Document numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front−Cover Texts on the front cover, and Back−Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine−readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly−accessible computer−network location containing a complete Transparent copy of the Document, free of added material, which the general network−using public has access to download anonymously at no charge using public−standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

A.5. 4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- **A.** Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- **B.** List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has less than five).
- **C.** State on the Title Page the name of the publisher of the Modified Version, as the publisher.
- **D.** Preserve all the copyright notices of the Document.
- **E.** Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- **F.** Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- **G.** Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- **H.** Include an unaltered copy of this License.
- **I.** Preserve the section entitled "History", and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- **J.** Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if

the original publisher of the version it refers to gives permission.

- **K.** In any section entitled "Acknowledgements" or "Dedications", preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

- **L.** Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

- **M.** Delete any section entitled "Endorsements". Such a section may not be included in the Modified Version.

- **N.** Do not retitle any existing section as "Endorsements" or to conflict in title with any Invariant Section.

If the Modified Version includes new front−matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties−−for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front−Cover Text, and a passage of up to 25 words as a Back−Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front−Cover Text and one of Back−Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version .

A.6. 5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled "History" in the various original documents, forming one section entitled "History"; likewise combine any sections entitled "Acknowledgements", and any sections entitled "Dedications". You must delete all sections entitled "Endorsements."

A.7. 6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

A.8. 7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an "aggregate", and this License does not apply to the other self–contained works thus compiled with the Document , on account of their being thus compiled, if they are not themselves derivative works of the Document. If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document's Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

A.9. 8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

A.10. 9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

A.11. 10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.