LINUX on zSeries

HiperSockets Support- October 31, 2001

LINUX kernel 2.4

LNUX-2004-01

LINUX on zSeries

HiperSockets Support- October 31, 2001

LINUX kernel 2.4

LNUX-2004-01

Note

Before using this document, be sure to read the information in “Notices” on page 21.

Second Edition – (October 2001)

This edition applies to the LINUX on zSeries kernel 2.4 patch (made in September 2001).

This edition replaces LNUX-2004-00.

Contents

About this book

This book describes how to configure the QETH module for LINUX on zSeries HiperSockets. Feedback on this topic can be sent to:

`Linux390@de.ibm.com`

The driver described herein has been developed for the zSeries 64-bit architecture and 31-bit architecture with version 2.4 of the LINUX kernel.

Assumptions

The following general assumptions are made about your background knowledge:

- You have an understanding of LINUX and zSeries terminology.
- You are familiar with LINUX device driver software.
- You have an understanding of basic computer architecture, operating systems, and programs.
- You are familiar with the devices attached to your system. (Detailed knowledge should not be required, as the code specific to the hardware is provided by IBM.)

Chapter 1. LINUX on zSeries QETH device driver for HiperSockets

This driver is subject to licence conditions as reflected in: “International License Agreement for Non-Warranted Programs” on page 23.

The QETH LINUX network driver supports HiperSockets virtual devices as well as the OSA-Express feature in QDIO mode on Fast Ethernet, Gigabit Ethernet and High Speed Token Ring. HiperSockets enable the zSeries to connect to virtual networks on a shared zSeries.

A HiperSockets device is controlled by the same device driver as the OSA-Express card. Most of the device driver parameters are common to the two devices.

The OSA-Express card is described in detail in *OSA-Express Customer’s Guide and Reference*, SA22-7403.

Naming conventions

Different cards used will generate different interface base names:

- Ethernet cards will generate an interface name starting with ″eth″
- Token Ring cards will generate an interface name starting with ″tr″
- HiperSockets devices will generate an interface name starting with ″hsi″

Numbers will be appended to the base names according to the following rules:

- If a device interface number, `devif_num`, is specified during device configuration, that number will be used. For example, a device configuration like:

  ```
  qeth7, <read_devno>, <write_devno>, <data_devno>
  ```

  will cause the interface name to be `eth7` for an Ethernet card, `tr7` for a Token Ring card, and `hsi7` for HiperSockets.

  If the `devif` number is `-1`, the next available number will be used. See “Configuring QETH for HiperSockets using the channel device layer” on page 2 for the description of `devif_num`.
- If chandev is instructed to use device number names (`use_devno_names`), the interface number will be the cuu number of the read channel. For example, if the read channel has the cuu `0xfd0c`, the interface name would be `eth0xfd0c` for an Ethernet card, `tr0xfd0c` for a Token Ring card, and `hsi0xfd0c` for a HiperSockets device. See “Commonly used options” on page 13 for the description of `use_devno_names`.

Introduction

You need two modules to configure HiperSockets as well as the OSA-Express feature in QDIO mode:

- The QDIO protocol governs the interface between the zSeries and the OSA-Express card. You need only load the QDIO module, no configuration is necessary.
- The QETH module controls the card itself. How to configure the QETH module is described in this chapter.

For HiperSockets three I/O subchannels must be available to the driver. One subchannel is for control reads, one for control writes, and the third is for data.

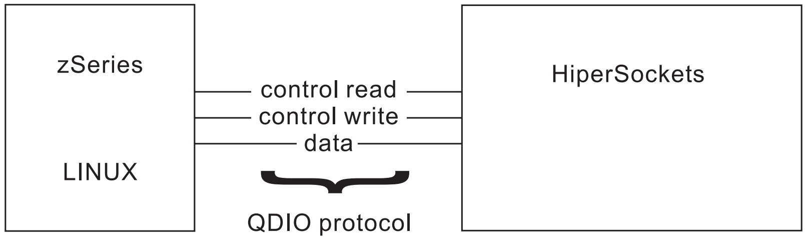


Figure 1. Channels and devices

QETH supported functions

The following functions are supported:

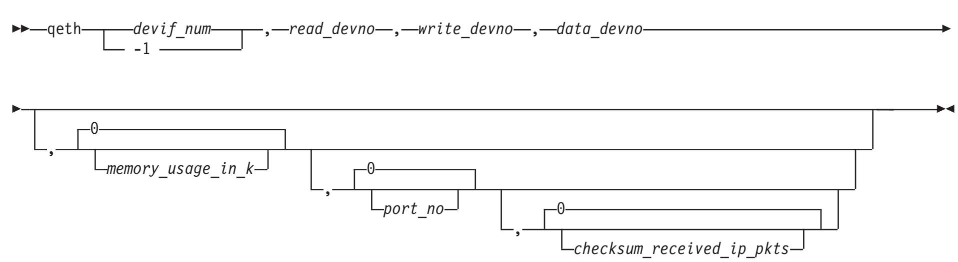
- Auto-detection of devices
- Primary and secondary routers
- Individual device configuration. It is possible to configure different triples of channels on the same CHPID differently. For example, if you have CHPID fc, then you can configure 0xfc00,0xfc01,0xfc02 differently from 0xfc04,0xfc05,0xfc06, for example, with different mem_usage values.
- IP Address Takeover

Configuring QETH for HiperSockets using the channel device layer

This section describes how to configure QETH for HiperSockets with the channel device layer. Only the most common options are given here to illustrate the syntax; see “Chapter 2. LINUX on zSeries Channel device layer” on page 11 for full details of all channel device options.

The driver will normally use auto-detection to find all HiperSockets in the system. (The `noauto` option can be used to exclude address ranges from auto-detection.) In some circumstances it may be necessary to configure the driver explicitly for a device. This is done with the command:

HiperSockets device identification



Note: All characters must be entered in lower case as shown, except in hexadecimal numbers where either case may be used.

Where:

devif_num
: is the device interface number. This is concatenated with `qeth`, for example `qeth1`.

  A value of -1 indicates that the next available number is to be allocated automatically.

read_devno
: is the read channel address (in hexadecimal preceded by 0x)

  This address must be an even number.

write_devno
: is the write channel address (in hexadecimal preceded by 0x)

  This address must be one greater than the read channel address.

data_devno
: is the data channel address (in hexadecimal preceded by 0x)

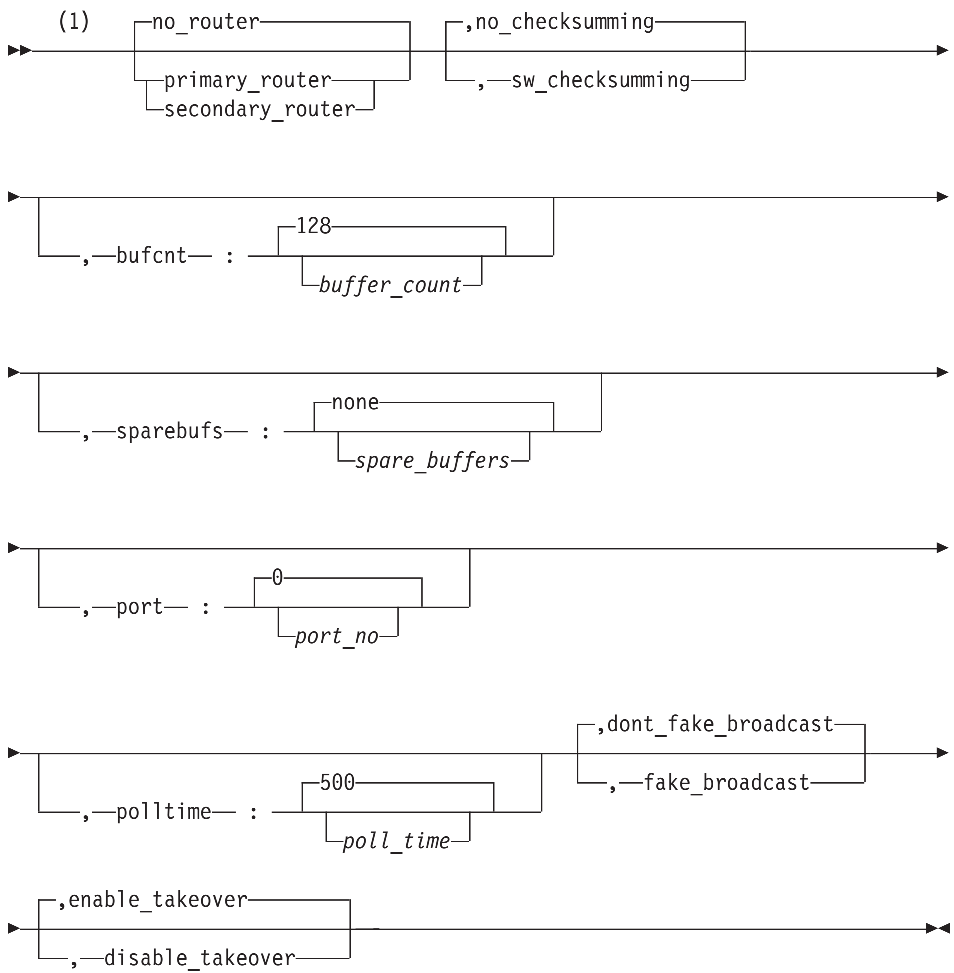
memory_usage_in_k
: is the number of kilobytes to be allocated for read and write buffers. (The allocation between read and write is determined by the driver.)

port_no
: is the relative port number of the device.

checksum_received_ip_pkts
: is 1 to perform software checksumming or 0 (the default) not to perform it.

HiperSockets QETH parameter syntax

qeth options



Notes:

1 All options except the first used must be preceded by a comma. All characters must be entered in lower case as shown, except in hexadecimal numbers where either case may be used.

The meanings of the parameters of this command are as follows:

primary_router | secondary_router | no_router

Specifies whether the device is used to interconnect networks. A ″Primary router″ is the principal connection between two networks; a ″Secondary router″ is used as backup in case of problems with the primary. Both of these options require the LINUX system to be configured as a router. The default for this parameter is ″No router″ – the HiperSockets will only be used to connect the LINUX on zSeries system to a single network.

It is possible to add routing status dynamically. This is done with the command:

```
echo primary_router ifname > /proc/qeth
```

or

```
echo secondary_router ifname > /proc/qeth
```

| *ifname* is the name of the interface in LINUX, for example `hsi0`.

It is not possible to reset routing status with the current hardware.

sw_checksumming | no_checksumming
Specifies whether error detection is to be performed by the driver, or is not required.

buffer_count
Specifies the number of inbound buffers used. Valid values for *buffer_count* are 16 to 128. The default is 128.

This may be used to overcome problems with memory shortage. The size of each buffer is 64 kilobytes.

spare_buffers
Specifies the number of spare buffers to reserve. The default is none. These buffers are pre-allocated and can be used as a safety valve if excessive load fills the normal buffer pool.

port_no
Specifies the port number on the `CHPID`. The default port number is 0.

poll_time
Specifies the maximum duration of background polling (in microseconds) used by `QDIO`. The default is 500.

dont_fake_broadcast | fake_broadcast
`fake_broadcast` sets the 'broadcast capable' device flag. This is necessary for the `gated` routing daemon.

enable_takeover | disable_takeover
allow/do not allow IP address takeover.

HiperSockets channel device layer configuration example

|

```
qeth1,0x7c00,0x7c01,0x7c02,4096,-1
```

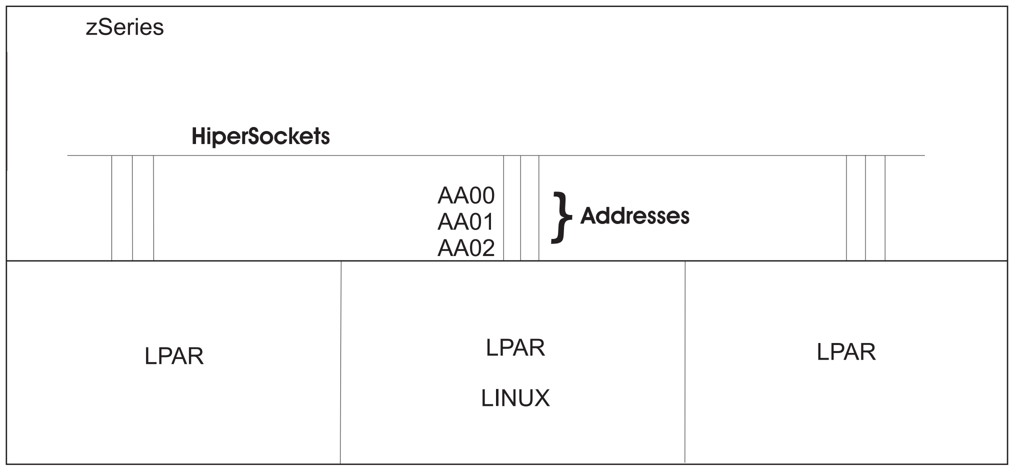
| This tells the channel device layer to force `qeth1` (if detected) to use device addresses 7c00, 7c01 and 7c02, allocate four megabytes of buffer space, and use the default port.

HiperSockets examples

1: Basic configuration

In this example a single HiperSockets is being used to connect a LINUX on zSeries system to a network.

Hardware configuration – HiperSockets connecting LINUX on zSeries to a network.



Software configuration – HiperSockets connecting LINUX on zSeries to a network.

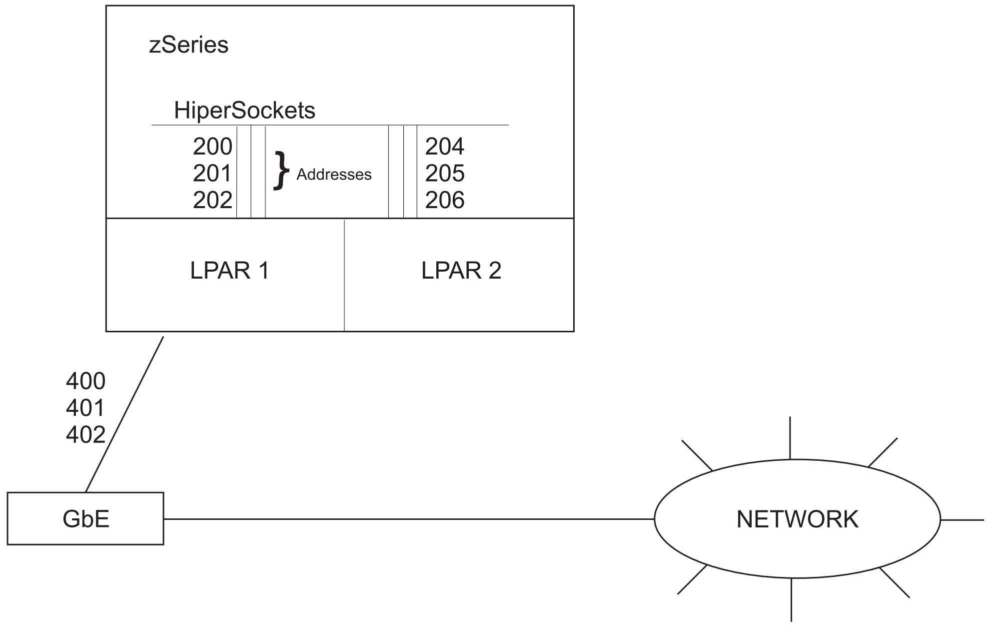
With the channel device layer the load commands for this configuration are:

```
qeth-1,0xAA00,0xAA01,0xAA02
```

2: Router configuration

This example shows how LINUX systems running on different LPARs in a zSeries may use HiperSockets to communicate with a network or to act as a router between networks.

Hardware configuration – HiperSockets and LINUX on zSeries as router.



In this example it is assumed that LINUX is configured as a router in both LPARs.

Software configuration – HiperSockets and LINUX on zSeries as router.

LPAR 1 – uses subchannels 200 - 202 for connecting to HiperSockets and 400 - 402 to connect to the OSA-Express feature. There it is able to route packets in and out. Specifying `primary_router` for HiperSockets is in general not necessary:

```
add_parms,0x10,0x400,0x402,primary_router
qeth-1,0x400,0x401,0x402
qeth-1,0x200,0x201,0x202
```

LPAR 2 – uses subchannels 204 - 206 as a network client:

```
qeth-1,0x204,0x205,0x206
```

HiperSockets – Preparing the connection

Activating the HiperSockets connection

The network devices can be activated with the `ifconfig` command. The right MTU size is selected automatically.

For details of the ifconfig command, see the ifconfig manpage.

An example of the use of `ifconfig` is:

```
ifconfig hsi0 192.168.100.11 netmask 255.255.255.0
```

HiperSockets IP Address Takeover

Unless the `disable_takeover` flag has been set it is possible to add and remove ranges of IP addresses for HiperSockets by writing to the `/proc/qeth_ipa_takeover` file. When a command is written to this file the driver calls on the OSA ″Address Takeover″ mechanism. This overrides any previous allocation of the specified address to another LPAR or card. If another LPAR on the same card has already registered for that IP address this association will be removed.

Note: Using HiperSockets, only IP addresses of another LINUX operating system in the same CEC can be taken over, not IP addresses of other zSeries operating systems. IP takeover must be enabled both on the image taking over the address, and on the image that gives up its address.

The registered addresses are held in this file in plain text and can be read to see the current associations.

Only one command at a time can be written to the file. Subsequent commands in the same write action are ignored.

The following commands are available:
- `add4` *<addr>*/*<mask_bits>*[:*<interface>*]
- `inv4`
- `del4` *<addr>*/*<mask_bits>*[:*<interface>*]

`add4` adds an address range. `del4` deletes an address range. *<addr>* is an 8 character hexadecimal IP address. *<mask_bits>* specifies the number of bits which are set in the network mask. *<interface>* is optional and specifies the interface name to which the address range is bound.

For example

```
echo add4 c0a80a00/24 > /proc/qeth_ipa_takeover
```

activates all addresses in the 192.168.10 subnet for address takeover.

`inv4` inverts the selection of address ranges done with add4. Issue `inv4` once to set all addresses which have been specified with `add4` not to use the takeover mechanism; all other IPv4 addresses will be set to use it.

Note that the address is not actually taken over until a corresponding `ifconfig` command is executed; for example

```
ifconfig hsi0 192.168.10.5
```

sets the IP address 192.168.10.5 on the device `hsi0`, and removes it from other LPARs if necessary.

The IP address must be different to that previously set on the device or no action will be taken. To take over a device at the same address `ifconfig` must be called twice; the first time with a dummy address (for example 0.0.0.0) to notify the device of the takeover and the second time with the original address to reset it. To re-capture the `hsi0` device with the IP address in the previous example you could use:

```
ifconfig hsi0 0.0.0.0
ifconfig hsi0 192.168.10.5
```

The second line alone will not take over the device from another LPAR if the IP address is the same as that set by the other LPAR.

QETH restrictions

- The MTU range is 576 – 57344. This may be restricted by the framesize announced by the microcode.
- There is a restriction in LINUX that the packet size of a multicast packet can not be greater than the MTU size of the interface used.

Chapter 2. LINUX on zSeries Channel device layer

The channel device layer provides a common interface to LINUX on zSeries channel devices. You can use this interface to configure the devices and to handle machine checks (devices appearing and disappearing).

The drivers using the channel device layer at the time of writing are:

1. **LCS** – Ethernet, Token Ring, non-QDIO OSA
2. **CTC/ESCON** – high speed serial link
3. **QETH** – OSA-Express feature in QDIO mode and HiperSockets.

The *raison d'être* of the channel device layer is to draw together the configuration of the drivers and to resolve conflicts. These could, for example, result in the `LCS` and `CTC` drivers fighting over 3088/08 and 3088/1F devices (which could be either 2216/3172 `LCS` compatible devices or `ESCON/CTC`). To resolve the clashing without the channel device layer each of these device drivers had to be configured separately, with a check for conflicts performed by eye.

The channel device layer is used on a per-driver basis, not on a system basis. For example a CTC driver which is not configured to use the channel device layer can be used in conjunction with a LCS driver which is configured to use it.

Description

The current configuration of the channel device layer is held (in human readable form) in the file `/proc/chandev`.

You can pass arguments to the channel device layer in three ways:

1. Piping them to `/proc/chandev`, for example:

   ```
   echo reprobe >/proc/chandev
   ```

   will cause un-initialized channel devices to be probed.
2. Editing them into `/etc/chandev.conf` – this will only take effect after a reboot of after executing the sequence of commands mentioned in "Read configuration" on page 18.
3. Using the `'chandev='` keyword on the LINUX boot command line, for example:

   ```
   chandev=noauto,0x0,0x480d;noauto,0x4810,0xffff
   ```

   will exclude all devices from auto-detection except for subchannels 0x480e and 0x480f.

Multiple options can be passed, separated by commas, but no spaces are allowed between parameters. You can also add comments to the configuration file. Comment lines must be prefixed with a '#' character.

To be consistent with other hotpluggable architectures the script pointed to by `/proc/sys/kernel/hotplug` (this will normally be `/sbin/hotplug`) will be called automatically on startup or on a device machine check as follows:.

```
/sbin/hotplug chandev <start starting_devnames>
                      <machine_check (devname last/pre_recovery_status)
                                     (current/post_recovery_status)>.
```

The channel device layer does not open stdin stdout or stderr so it is advisable that you open them at the start of your script, as in this sample which starts devices as they become available:

```
#!/bin/bash
exec >/dev/console 2>&1 0>&1
# Remove the comment symbol from the line below for debugging purposes.
# echo $*
if [ "$1" = "chandev" ] && [ "$2" = "start" ]
then
    shift 2
    while [ "$1" != "" ]  && [ "$1" != "machine_check" ]
    do
        isup='ifconfig $1 2>/dev/null | grep UP'
        if [ "$isup" = "" ]
        then
            ifup $1
        fi
        shift
    done
fi
```

For example if devices tr0 and ctc0 become active at a time when eth0 and eth1 are subject to a gone machine check and eth2 is subject to a revalidate machine check (which is normally fully recoverable) the parameters passed to hotplug would be:

```
/sbin/hotplug chandev start tr0 ctc0
                  machine_check eth0 gone gone eth1 gone gone
                                eth2 revalidate good
```

This script can be used, for example, to call /etc/rc.d/init.d/network start when a device appears. (This makes the ipldelay kernel boot parameter obsolete when LINUX is running native.) It may also be used to recover from bad machine checks if the default machine check handling is inadequate. The machine checks that can be presented as parameters to the channel device layer are good, not_operational, no_path, revalidate and device_gone.

The channel device layer will wait a few seconds after machine checks before running /sbin/hotplug because a machine check on one device is often followed by checks on others. It is better to handle multiple devices with a single script, rather than with individual scripts for each device which could compete for resources.

Channel device layer options

Terminology

devno a 16 bit unsigned number (usually expressed as hexadecimal) which uniquely identifies a subchannel connected to a device.

force list
a term (specific to channel device layer) describing a range of *devno* which are to be configured specifically (as opposed to configuration by auto-detection).

auto machine check recovery bitfield

The bits in this field signify:

not_operational
0x1

no_path
0x2

revalidate
0x4

gone 0x8

chan_type bitfield

The bits in this field signify:

ctc 0x01

escon 0x02

lcs 0x04

osad 0x08 – reserved, not used in this release

qeth 0x10

A single device driver may handle more than one type of device. In this case the values corresponding to each device handled are summed to create the parameter

Device identification (QDIO)

For the syntax of the HiperSockets with the channel device layer see “Configuring QETH for HiperSockets using the channel device layer” on page 2.

Commonly used options

These options are used to set up the system.

add parameters

```
►►─add_parms─,─chan_type─┬──────────────────────────┬─,─string──────►◄
                         └─,─lo_devno─,─high_devno─┘
```

chan_type is defined in ***chan_type bitfield*** in the terminology on page 13.

This is for device driver specific options which are passed as a string to the driver and are not dealt with by the channel device layer. This string cannot contain spaces. ***lo_devno*** and ***high_devno*** are optional parameters to specify a range.

The *string* is interpreted by the driver (see the particular driver chapter for details).

delete parameters

```
►►─del_parms─┬──────────────────────────────────────────┬──────►◄
             └─,─chan_type─┬──────────────────────────┬─┘
                           └─,─exact_match─┬────────┬─┘
                                           └─,─lo_devno─┘
```

chan_type is defined in the terminology on page 13.

This deletes some or all device driver specific options. If `chan_type` is not specified all the strings will be deleted. If ***exact_match*** is set to '1' the driver parameters will only be removed where ***chan_type*** is exactly equal. If ***exact_match*** is set to '0' the parameters are to be removed where any bit matches ***chan_type***. ***lo_devno*** is an optional parameter to specify that the delete is only to happen if this parameter matches a ***lo_devno*** in a defined range.

no auto-detection

```
►►──noauto──┬──────────────────────────────┬──────────────────────────►◄
            └─,──lo_devno──,──high_devno───┘
```

This stops auto-detection of channel devices in the given range of device numbers. `noauto` without a device range will stop auto-detection of all channel devices.

use device names

```
►►──use_devno_names──────────────────────────────────────────────────►◄
```

This instructs the channel device layer to assign device names based on the `cuu` number of the read channel. For example a token ring read channel with `cuu` number 0x7c00 would be assigned an interface name of `tr0x7c00`. This may be used to avoid device name conflicts. The default is to generate device names in sequence, so the default name for the channel above might be `tr2`.

Power user options

These options are used for maintenance or fine-tuning.

delete no-auto ranges

```
►►──del_noauto──┬────────────┬───────────────────────────────────────►◄
                └─,──devno───┘
```

Delete the range containing `devno`, or all `noauto` ranges if `devno` is not given.

delete forced device

```
►►──del_force──,──read_devno─────────────────────────────────────────►◄
```

Remove a forced channel device from the force list.

do not use device names

```
►►—dont_use_devno_names——————————————————————————————————————►◄
```

Cancel a **use_devno_names** command.

add a device

```
►►—add_model—,—chan_type—,—cu_type—,—cu_model—,—dev_type—,—dev_model—,——►

►—max_port_no—,—automatic_machine_check_handling——————————————————————►◄
```

Probe for the device specified. '-1' may be used as a wildcard for any of the parameters except ***chan_type*** or ***automatic_machine_check_handling***. Set ***max_port_no*** to zero ('0') for non LCS devices.

chan_type and ***automatic_machine_check_handling*** are defined in the terminology on page 12.

delete a device

```
►►—del_model—,—cu_type—,—cu_model—,—dev_type—,—dev_model——————————►◄
```

Remove the device specified. '-1' may be used as a wildcard for any of the parameters.

delete all devices

```
►►—del_all_models————————————————————————————————————————————►◄
```

Remove all devices.

auto-detect any devices

```
►►—non_cautious_auto_detect——————————————————————————————————►◄
```

Attempt to auto-detect devices even if their type/model pairs do not unambiguously identify the device. For example 3088/1F's can either be `CTC/ESCON` or `3172 LCS` compatible devices. If the wrong device driver attempts to probe these channels there may be long delays on startup or even a kernel lockup, so use this option with caution.

auto-detect known devices

```
►►──cautious_auto_detect──────────────────────────────────────────────►◄
```

Do not attempt to auto-detect devices unless their type/model pairs unambiguously identify the device. (This is the default behavior.)

machine check recovery

```
►►──auto_msck──┬──────────────────────────────┬─,──auto_msck_recovery──────►◄
               └─,──lo_devno──,──high_devno───┘
```

Specify the kind of machine check recovery to be performed over a range of devices. ***auto_msck_recovery*** is defined in the terminology on page 12.

delete machine check recovery

```
►►──del_auto_msck──┬────────────┬──────────────────────────────────►◄
                   └─,──devno───┘
```

Delete machine check recovery for the range of devices including ***devno***, or all machine check recovery if ***devno*** is not specified.

null model information

```
►►──reset_clean───────────────────────────────────────────────────►◄
```

Reset all model information, forced devices and `noauto` lists to null.

default model information

```
►►──reset_conf──────────────────────────────────────────────────►◄
```

Reset all model information, forced devices and `noauto` lists to default settings.

empty model information

```
►►──reset_conf_clean────────────────────────────────────────────►◄
```

Reset all model information, forced devices and `noauto` lists to empty.

shutdown device

```
►►──shutdown──┬─────────────────┬───────────────────────────────►◄
              └─┬─device_name─┬─┘
                └─read_devno──┘
```

Shut down the particular device identified by ***device_name*** or ***read_devno***, de-register it and release its interrupts. If no parameter is given all devices are shut down.

reprobe

```
►►──reprobe─────────────────────────────────────────────────────►◄
```

Call probe method for channels whose interrupts are not owned.

unregister general probe

```
►►──unregister_probe──┬────────────────┬────────────────────────►◄
                      └─probefunc_addr─┘
```

Unregister a probe function, or unregister all probe functions if no address given.

unregister specific probe

```
►►—unregister_probe_by_chan_type—chan_type—————————————————►◄
```

Unregister all probe functions which match the `chan_type` bitfield exactly. This is useful if you want a configuration to survive a kernel upgrade.

read configuration

```
►►—read_conf———————————————————————————————————————————————►◄
```

Read instructions from `/etc/chandev.conf`.

This is used to make the channel device layer read from `/etc/chandev.conf` on boot, or to cause the channel device layer to re-read its configuration during operation.

do not read configuration

```
►►—dont_read_conf——————————————————————————————————————————►◄
```

Do not read instructions from `/etc/chandev.conf` on boot.

For example the following sequence of commands piped to `/proc/chandev` should have the same effect as rebooting for channel devices:

- `shutdown`
- `reset_conf`
- `read_conf`
- `reprobe`

See also

If you wish to write a driver which is compatible with the channel device layer see:

- `/linux/include/asm-s390/chandev.h` – for the API (which is commented), and
- `/linux/drivers/s390/misc/chandev.c` – for the code.

Files

/proc/chandev
This holds the current configuration. Use

```
cat /proc/chandev
```

to see the configuration, and

```
echo command >/proc/chandev
```

to enter a new command.

/etc/chandev.conf
This file can be used to configure the channel device layer kernel parameters.

/sbin/hotplug
This is a user script or executable which is run whenever devices come online or go offline ('appear' or 'disappear').

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information about the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

Advanced Peer-to-Peer Networking	OS/390
APPN	OSA
CICS	PowerPC
Common User Access	RACF
e-business	RAMAC
ECKD	S/390
ESA/390	Seascape
ESCON	System/390
IBM	VM/ESA
Micro Channel	VSE/ESA
Multiprise	VTAM
MVS	z/OS
OS/2	zSeries

UNIX is a registered trademark in the United States and other countries licensed exclusively through The Open Group.

LINUX is a registered trademark of Linus Torvalds and others.

Microsoft, Windows NT, and MSDOS are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.

International License Agreement for Non-Warranted Programs

Part 1 - General Terms

PLEASE READ THIS AGREEMENT CAREFULLY BEFORE USING THE PROGRAM. IBM WILL LICENSE THE PROGRAM TO YOU ONLY IF YOU FIRST ACCEPT THE TERMS OF THIS AGREEMENT. BY USING THE PROGRAM YOU AGREE TO THESE TERMS. IF YOU DO NOT AGREE TO THE TERMS OF THIS AGREEMENT, PROMPTLY RETURN THE UNUSED PROGRAM TO THE PARTY (EITHER IBM OR ITS RESELLER) FROM WHOM YOU ACQUIRED IT TO RECEIVE A REFUND OF THE AMOUNT YOU PAID.

The Program is owned by International Business Machines Corporation or one of its subsidiaries (IBM) or an IBM supplier, and is copyrighted and licensed, not sold.

The term ″Program″ means the original program and all whole or partial copies of it. A Program consists of machine-readable instructions, its components, data, audio-visual content (such as images, text, recordings, or pictures), and related licensed materials.

This Agreement includes Part 1 - General Terms and Part 2 - Country-unique Terms and is the complete agreement regarding the use of this Program, and replaces any prior oral or written communications between you and IBM. The terms of Part 2 may replace or modify those of Part 1.

1. Licence

   Use of the Program

   IBM grants you a nonexclusive licence to use the Program.

   You may 1) use the Program to the extent of authorizations you have acquired and 2) make and install copies to support the level of use authorized, providing you reproduce the copyright notice and any other legends of ownership on each copy, or partial copy, of the Program.

   If you acquire this Program as a program upgrade, your authorization to use the Program from which you upgraded is terminated.

   You will ensure that anyone who uses the Program does so only in compliance with the terms of this Agreement.

   You may not 1) use, copy, modify, or distribute the Program except as provided in this Agreement; 2) reverse assemble, reverse compile, or otherwise translate the Program except as specifically permitted by law without the possibility of contractual waiver; or 3) sublicense, rent, or lease the Program.

   Transfer of Rights and Obligations

   You may transfer all your license rights and obligations under a Proof of Entitlement for the Program to another party by transferring the Proof of Entitlement and a copy of this Agreement and all documentation. The transfer of your license rights and obligations terminates your authorization to use the Program under the Proof of Entitlement.

2. Proof of Entitlement

   The Proof of Entitlement for this Program is evidence of your authorization to use this Program and of your eligibility for future upgrade program prices (if announced) and potential special or promotional opportunities.

3. Charges and Taxes

IBM defines use for the Program for charging purposes and specifies it in the Proof of Entitlement. Charges are based on extent of use authorized. If you wish to increase the extent of use, notify IBM or its reseller and pay any applicable charges. IBM does not give refunds or credits for charges already due or paid.

If any authority imposes a duty, tax, levy or fee, excluding those based on IBM's net income, upon the Program supplied by IBM under this Agreement, then you agree to pay that amount as IBM specifies or supply exemption documentation.

4. No Warranty

   SUBJECT TO ANY STATUTORY WARRANTIES WHICH CAN NOT BE EXCLUDED, IBM MAKES NO WARRANTIES OR CONDITIONS EITHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION, THE WARRANTY OF NON-INFRINGEMENT AND THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, REGARDING THE PROGRAM OR TECHNICAL SUPPORT, IF ANY. IBM MAKES NO WARRANTY REGARDING THE CAPABILITY OF THE PROGRAM TO CORRECTLY PROCESS, PROVIDE AND/OR RECEIVE DATE DATA WITHIN AND BETWEEN THE 20TH AND 21ST CENTURIES.

   The exclusion also applies to any of IBM's subcontractors, suppliers, or program developers (collectively called ″Suppliers″).

   Manufacturers, suppliers, or publishers of non-IBM Programs may provide their own warranties.

5. Limitation of Liability

   NEITHER IBM NOR ITS SUPPLIERS WILL BE LIABLE FOR ANY DIRECT OR INDIRECT DAMAGES, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST SAVINGS, OR ANY INCIDENTAL, SPECIAL, OR OTHER ECONOMIC CONSEQUENTIAL DAMAGES, EVEN IF IBM IS INFORMED OF THEIR POSSIBILITY. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE EXCLUSION OR LIMITATION MAY NOT APPLY TO YOU.

6. General

   Nothing in this Agreement affects any statutory rights of consumers that cannot be waived or limited by contract.

   IBM may terminate your licence if you fail to comply with the terms of this Agreement. If IBM does so, you must immediately destroy the Program and all copies you made of it.

   You agree to comply with applicable export laws and regulations.

   Neither you nor IBM will bring a legal action under this Agreement more than two years after the cause of action arose unless otherwise provided by local law without the possibility of contractual waiver or limitation.

   Neither you nor IBM is responsible for failure to fulfill any obligations due to causes beyond its control.

   IBM does not provide program services or technical support, unless IBM specifies otherwise.

   The laws of the country in which you acquire the Program govern this Agreement, except 1) in Australia, the laws of the State or Territory in which the transaction is performed govern this Agreement; 2) in Albania, Armenia, Belarus, Bosnia/Herzegovina, Bulgaria, Croatia, Czech Republic, Georgia, Hungary, Kazakhstan, Kirghizia, Former Yugoslav Republic of Macedonia (FYROM), Moldova, Poland, Romania, Russia, Slovak Republic, Slovenia, Ukraine, and Federal Republic of Yugoslavia, the laws of Austria govern this Agreement; 3) in the United Kingdom, all disputes relating to this Agreement will be governed by

English Law and will be submitted to the exclusive jurisdiction of the English courts; 4) in Canada, the laws in the Province of Ontario govern this Agreement; and 5) in the United States and Puerto Rico, and People's Republic of China, the laws of the State of New York govern this Agreement.

Part 2 - Country-unique Terms

AUSTRALIA:

No Warranty (Section 4):

The following paragraph is added to this Section:

Although IBM specifies that there are no warranties, you may have certain rights under the Trade Practices Act 1974 or other legislation and are only limited to the extent permitted by the applicable legislation.

Limitation of Liability (Section 3):

The following paragraph is added to this Section:

Where IBM is in breach of a condition or warranty implied by the Trade Practices Act 1974, IBM's liability is limited to the repair or replacement of the goods, or the supply of equivalent goods. Where that condition or warranty relates to right to sell, quiet possession or clear title, or the goods are of a kind ordinarily acquired for personal, domestic or household use or consumption, then none of the limitations in this paragraph apply.

GERMANY:

No Warranty (Section 4):

The following paragraphs are added to this Section:

The minimum warranty period for Programs is six months.

In case a Program is delivered without Specifications, we will only warrant that the Program information correctly describes the Program and that the Program can be used according to the Program information. You have to check the usability according to the Program information within the ″money-back guaranty″ period.

Limitation of Liability (Section 5):

The following paragraph is added to this Section:

The limitations and exclusions specified in the Agreement will not apply to damages caused by IBM with fraud or gross negligence, and for express warranty.

INDIA:

General (Section 6):

The following replaces the fourth paragraph of this Section:

If no suit or other legal action is brought, within two years after the cause of action arose, in respect of any claim that either party may have against the other, the rights of the concerned party in respect of such claim will be forfeited and the other party will stand released from its obligations in respect of such claim.

IRELAND:

No Warranty (Section 4):

The following paragraph is added to this Section:

Except as expressly provided in these terms and conditions, all statutory conditions, including all warranties implied, but without prejudice to the generality of the foregoing, all warranties implied by the Sale of Goods Act 1893 or the Sale of Goods and Supply of Services Act 1980 are hereby excluded.

ITALY:

Limitation of Liability (Section 5):

This Section is replaced by the following:

Unless otherwise provided by mandatory law, IBM is not liable for any damages which might arise.

NEW ZEALAND:

No Warranty (Section 4):

The following paragraph is added to this Section:

Although IBM specifies that there are no warranties, you may have certain rights under the Consumer Guarantees Act 1993 or other legislation which cannot be excluded or limited. The Consumer Guarantees Act 1993 will not apply in respect of any goods or services which IBM provides, if you require the goods and services for the purposes of a business as defined in that Act.

Limitation of Liability (Section 5):

The following paragraph is added to this Section:

Where Programs are not acquired for the purposes of a business as defined in the Consumer Guarantees Act 1993, the limitations in this Section are subject to the limitations in that Act.

PEOPLE'S REPUBLIC OF CHINA:

Charges (Section 3):

The following paragraph is added to the Section:

All banking charges incurred in the People's Republic of China will be borne by you and those incurred outside the People's Republic of China will be borne by IBM.

UNITED KINGDOM:

Limitation of Liability (Section 5):

The following paragraph is added to this Section at the end of the first paragraph:

The limitation of liability will not apply to any breach of IBM's obligations implied by Section 12 of the Sales of Goods Act 1979 or Section 2 of the Supply of Goods and Services Act 1982.

GNU General Public Licence, Version 2, June 1991

> **DISCLAIMER**
> Elements of LINUX for zSeries and S/390 which utilise internal details of the zSeries and S/390 systems remain the intellectual property and copyright of IBM, notwithstanding the licence below. This right will be waived for specific elements in the case that the documentation specific to the element indicates that it is published under the GNU General Public Licence.

LINUX for zSeries and S/390 is licensed under the GNU General Public Licence which is reproduced below:

Copyright (C) 1989, 1991
Free Software Foundation, Inc.
59 Temple Place,
Suite 330,
Boston,
MA 02111-1307
USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps:

1. copyright the software, and

2. offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU General Public Licence: Terms and conditions for copying, distribution and modification

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The ″Program″, below, refers to any such program or work, and a ″work based on the Program″ means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term ″modification″.) Each licensee is addressed as ″you″.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

   You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a. You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   b. You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

   c. If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:
   a. Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
   b. Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
   c. Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

   The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

   If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program 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.
5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.
6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.
7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

   If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

   It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

   This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.
8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public 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.

   Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and ″any later version″, you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

    NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM ″AS IS″ WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

    END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the ″copyright″ line and a pointer to where the full notice is found.

<one line to give the program’s name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

```
This program is free software; you can redistribute it and/or modify it under the
terms of the GNU General Public License as published by the Free Software Foundation;
either version 2 of the License, or (at your option) any later version.
This program is distributed in the hope that it will be useful, but WITHOUT ANY
WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A
PARTICULAR PURPOSE. See the GNU General Public License for more details.
You should have received a copy of the GNU General Public License along with this
program; if not, write to the Free Software Foundation, Inc., 59 Temple Place,
Suite 330, Boston, MA 02111-1307 USA
```

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type 'show w'.
This is free software, and you are welcome to redistribute it under certain
conditions; type 'show c' for details.
```

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than 'show w' and 'show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program 'Gnomovision'
(which makes passes at compilers) written by James Hacker.
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

Index

A

C

D

G

H

N

O

Q

R

T

LNUX-2004-01