

Clinical Neutron Therapy System Installation and Operations

Jonathan Jacky¹

Radiation Oncology, Box 356043
University of Washington
Seattle, WA 98195-6043

Technical Report 99-08-01

August 1999

(Revised February 2000, April 2001, May 2002, November 2002)

¹email jon@u.washington.edu, telephone (206)-598-4117, fax (206)-598-6218

Abstract

This document is the installation and operations manual for the Clinical Neutron Therapy System (CNTS), intended for cyclotron engineering staff and computer system administrators. This manual describes the hardware, system software, and network required by CNTS, shows how to install and configure it, and explains how to revise the configuration to accommodate changes in the department's computing facilities. It discusses operations including startup and shutdown, troubleshooting and recovery procedures. It discusses all the control program configuration files, command files, executable program files and data files, describing their names, locations, ownership, protections, formats, contents, and use. It also describes some files that provide summaries of control system activity that can be useful for quality assurance and troubleshooting. This manual explains how the therapy control program interacts with the treatment planning program and the prescription viewer/archiver. It also discusses security issues.

©1998, 1999, 2001, 2002 by Jonathan Jacky

This work may not be copied or reproduced in whole or part for any commercial purpose. Permission to photocopy in whole or part without payment of fee is granted for nonprofit educational and research purposes provided that all such copies include the following notice: a notice that such copying is by permission of the authors; an acknowledgment of the authors of the work; and all applicable portions of this copyright notice. All rights reserved.

Contents

1	Computers and terminals	2
1.1	Control computer	2
1.1.1	Serial ports	4
1.1.2	Console port	5
1.2	Controllers	5
1.3	PLC watchdog	6
1.4	X terminal	6
1.5	Room display	7
1.6	Host computer	8
1.7	Department computer cluster	9
2	Network	11
2.1	Addressing	11
2.2	Network security	12
2.3	Hub and wiring	13

3	Configuration	15
3.1	Boot parameters	15
3.2	Command interpreter (target shell)	16
3.3	Startup command file	17
3.4	Startup process	18
4	Data files	20
4.1	Directories	20
4.2	Groups	21
4.3	Files in /cntsdata/data/iso	21
4.3.1	Directory	21
4.3.2	Directory contents	21
4.3.3	Reading and writing data files	23
4.3.4	Maintaining data files	24
4.3.5	accum.bak	29
4.3.6	accum.dat	30
4.3.7	calfields.dat	36
4.3.8	colors.dat	37
4.3.9	dosecal.dat	38
4.3.10	exper.dat	41
4.3.11	fonts.dat	41

4.3.12	keycodes.dat etc.	42
4.3.13	message.dat	44
4.3.14	mlccal.dat	44
4.3.15	oper.dat	46
4.3.16	ports.dat	47
4.3.17	prescr.bak	48
4.3.18	prescr.dat	49
4.3.19	run.dat	57
4.3.20	timeouts.dat	58
4.3.21	tol.dat	59
4.4	Files in /cntsdata/log/iso	60
4.4.1	Directory	60
4.4.2	Directory contents	60
4.4.3	Controller logs	61
4.4.4	Message logs	62
4.4.5	Stored fields	63
4.4.6	Treatment records	64
4.5	Files in /radonc/treatdata/cnts/data/iso	74
4.5.1	Directory	74
4.5.2	Directory contents	74
4.5.3	Archived prescriptions	74

4.5.4	Archived doses and fractions	75
4.6	Files in /radonc/treatdata/cnts/log/iso	75
5	Summary files	76
5.1	Automation	77
5.2	Prescription transfer summaries	77
5.2.1	prism*.log	77
5.3	Archiver summaries	78
5.3.1	preview*.log	78
5.3.2	prevlog.out	78
5.3.3	prevlog.err	79
5.4	Event log summaries	80
5.4.1	eventlog.out	80
5.4.2	eventlog.err	80
5.5	Treatment record summaries	80
5.5.1	treatrec.out	80
5.5.2	treatrec.err	81
5.6	Treatment histories and accumulated dose summaries	82
5.6.1	accum.out	83
5.6.2	accum.err	85
5.7	Statistical summaries	88

5.7.1	Operating statistics	88
5.7.2	Patient accruals and new fields	88
6	Startup and shutdown	91
6.1	Startup	92
6.1.1	Preliminaries	92
6.1.2	Start the controllers	92
6.1.3	Start the X terminal and room display	92
6.1.4	Start the control program	93
6.1.5	Clear the PLC watchdog interlock	93
6.1.6	Confirm that startup completed normally	94
6.2	Shutdown	94
6.2.1	Preliminaries	94
6.2.2	Shut down the control program	95
6.2.3	What happens at shutdown	95
6.2.4	Halting or crashing the control program	95
7	Interruptions and recovery procedures	97
7.1	Control program stops	97
7.1.1	Determining that the program has stopped	97
7.1.2	Determining how the program stopped	98
7.1.3	Determining when the program stopped	98

7.1.4	Recovering after the program stops during a therapy run	99
7.2	TMC, LCC, or DMC problems	99
7.2.1	Recovering after a DMC problem during a treatment run	100
7.3	PLC problems	100
7.4	X terminal problems	100
7.5	Host computer, file system, and network problems	101
7.6	Host shuts down	101
A	Hardware reference data	102
B	Host computer	103
B.1	CNTS user	103
B.2	Hosts file	104
B.3	Remote shell facility (<code>remsh</code>)	104
B.4	Trivial file transfer protocol (<code>tftp</code>)	105
B.5	Time server	105
B.6	Font server	106
B.7	Network File System (NFS)	107
B.8	Network security	107
C	Hub	109
C.1	Bridge	110

D	X terminal	111
D.1	Host configuration	111
D.2	Boot parameters	112
D.3	Terminal setup	113
D.3.1	Setup key (F3)	113
D.3.2	No window manager	113
D.3.3	Security	114
D.3.4	Font server	114
D.4	Boot process	115
D.4.1	Console	115
D.5	Tektronix X terminal boot parameters (summary)	116
D.6	Tektronix X terminal setup (summary)	117
D.7	Tektronix X terminal boot and startup messages	118
E	Room display	120
F	Control computer boot parameters	123
G	Console terminal setup	124
H	Control computer startup command file	125
I	Control computer boot and startup messages	128
J	Control program log messages at startup	134

List of Tables

1.1	Serial ports and controllers	4
2.1	Devices, IP addresses, and names on the CNTS private network	12
2.2	Devices and ports on the CNTS hub	13
4.1	Directory listing of /cntsdata/data/iso	22
4.2	Description of files in /cntsdata/data/iso	26
4.3	Reading files in /cntsdata/data/iso	27
4.4	Writing files in /cntsdata/data/iso	28
4.5	Leaf collimator calibration field settings in calfields.dat	36
4.6	Dosimetry calibration constants in dosecal.dat	39
4.7	Leaf collimator calibration constants in mlccal.dat	45
4.8	Settings in prescr.dat	51
4.9	Wedge selection and rotation codes in prescr.dat	54
4.10	Flags in prescr.dat	55
4.11	Timeout intervals in timeouts.dat	58

4.12	Tolerances in <code>tol.dat</code>	59
4.13	Description of files in <code>/cntsdata/log/iso</code>	61
4.14	Writing files in <code>/cntslog/log/iso</code>	62
4.15	Flags in treatment record	69
4.16	Pressure/temperature status codes	70
5.1	Entry for one field from <code>accum.out</code>	84
5.2	Operating statistics reported by <code>stats</code> program	89
5.3	Patient accruals and new fields reported by <code>new_pats</code> program	90
A.1	Devices, manufacturers and model numbers	102

List of Figures

1.1	CNTS computers, terminals, and networks	3
4.1	Sample calfields.dat file	36
4.2	Sample colors.dat file	37
4.3	Sample dosecal.dat file	40
4.4	Sample fonts.dat file	41
4.5	Sample keycodes.dat file	43
4.6	Beginning of a sample mlccal.dat file	45
4.7	Sample oper.dat file	46
4.8	Beginning of a sample ports.dat file	47
4.9	Sample timeouts.dat file	58
4.10	Sample tol.dat file	60
4.11	Sample stored field file, study00001fld12.dat	64

Introduction

This document is the installation and operations manual for the Clinical Neutron Therapy System (CNTS), intended for cyclotron engineering staff and computer system administrators. This manual describes the hardware, system software, and network required by CNTS, shows how to install and configure it, and explains how to revise the configuration to accommodate changes in the department's computing facilities. It discusses operations including startup and shutdown, troubleshooting and recovery procedures. It discusses all the control program configuration files, command files, executable program files and data files, describing their names, locations, ownership, protections, formats, contents, and use. It also describes some files that provide summaries of control system activity that can be useful for quality assurance and troubleshooting. This manual explains how the therapy control program interacts with the treatment planning program and the prescription viewer/archiver. It also discusses security issues.

A separate therapist's manual [5] tells how to use the program to treat patients. A separate reference manual [4] provides a comprehensive description of the program's user interface, internal operations, and communications with the therapy equipment it controls. A separate programmer's manual [3] provides information that would be needed to modify the program.

The control program is quite flexible regarding computer system configuration. The same executable program file can run without modification with different host computers, file systems, X terminals etc. when configured as described in this manual. Therefore it is not necessary to revise the control program to accommodate changes in the department's computing environment as computers are replaced, etc. Configuration details such as host names and IP addresses that appear in this document were the ones in use at this writing (April 2001) and may change in the future (they have already changed twice).

At this writing the new control system includes just one control system computer and one control program, for the isocentric treatment room. In other documents this control computer is called the *neutron therapy computer*. In the future we will add more computers and programs (for the cyclotron, etc.). This document provides some guidance for accommodating future expansion.

Chapter 1

Computers and terminals

A block diagram of the CNTS control computers, terminals, and networks appears in Fig. 1.1.

1.1 Control computer

The neutron therapy control computer is a single board computer with a 68040 microprocessor¹ [8]. It runs the VxWorks operating system from Wind River Systems, Inc [18] and uses the VX-Windows `xlib` implementation for VxWorks from VisiCom Laboratories, Inc [16]. The computer is housed in a VME crate² and connected by flat cables to a distribution panel called a *transition module*³ [7] in the back of the crate.

Boot ROM chips for VxWorks provided by Wind River must be installed in the computer.

The computer has no disk. Its program and data reside in the file system of a *host* computer that communicates with the computer on an Ethernet network. The Ethernet cable is connected at the transition module.

The computer has no display or keyboard. Instead, it uses the same Ethernet to communicate with an X terminal and a PC used as a display.

The computer senses and controls the therapy machine by receiving messages and sending com-

¹Motorola MVME167-004B with a 25MHz 68040 processor and 32MB memory .

²Mupac 5099KDL20FK-100 (sold by Carlo Gavazzi Electronic Packaging) .

³Motorola MVME712M

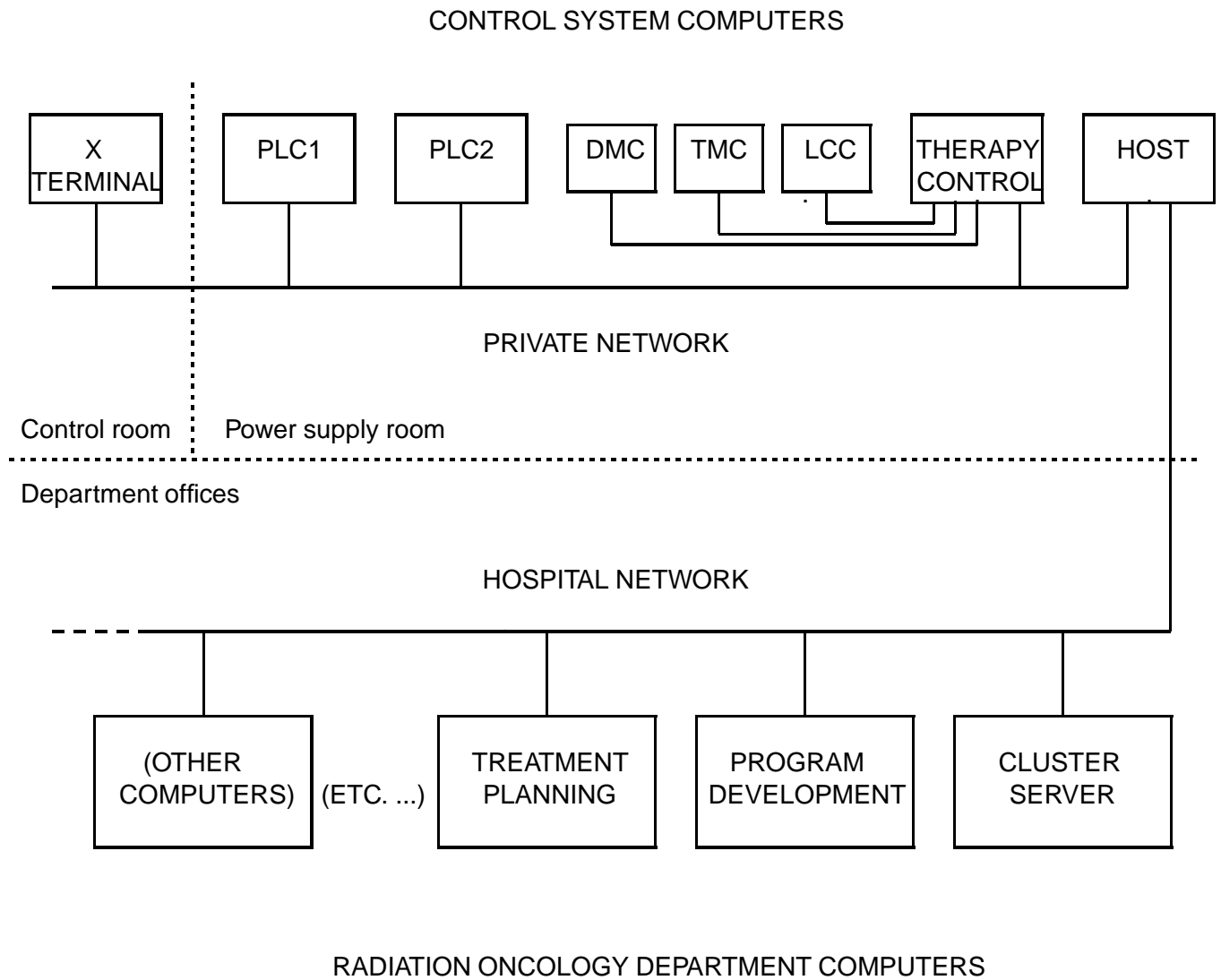


Figure 1.1: CNTS computers, terminals, and networks

Port	Ctlr	Description	Configuration	Jumper
1		Console terminal	DCE (DTE)	J1 (J11)
2	TMC	Treatment motion controller	DCE	J16
3	LCC	Leaf collimator controller	DCE	J13
4	DMC	Dose monitor controller	DCE	J18

Table 1.1: Serial ports and controllers

mands to several controllers. The computer communicates with some of these controllers over the Ethernet, with others over separate RS232 lines. The computer has no ports for controlling the therapy machine directly and it does not communicate with other cards in the VME crate over the VME bus.

The VME crate and the other controllers are housed in separate equipment racks in the power supply room, NW267.

The computer parallel printer port is not connected.

1.1.1 Serial ports

The computer has four RS232 serial communication ports. One is connected to the console terminal (not shown in fig. 1.1) and the other three are connected to controllers. Table 1.1 shows the port assignments.

Connections to the serial ports are made through the transition module. The port numbers appear next to the connectors on the transition module. We have added labels that identify the device that should be attached to each port⁴.

Each serial port must be configured by jumpers on the transition module to select DCE (data communications equipment, as needed to connect to a terminal, where the computer transmits outgoing data on pin 3 and receives incoming data on pin 2) or DTE (data terminal equipment, as needed to connect to a modem, vice versa). The console port has been configured as both DTE and DCE, depending on which cable is available for the console terminal, so a jumper is installed on header J11 or J1, as appropriate. At this writing the cables from the TMC, LCC and DMC make those devices look like terminals, so ports 2, 3 and 4 are configured as DCE. Therefore jumpers are installed on headers J16, J13 and J18. (Table 1.1, also Figs. 2-1 – 2-9, pps. 2-2 – 2-11 in [7]).

⁴The Motorola hardware manuals and the transition module itself label these ports 1, 2, 3, 4 as shown in Table 1.1 but the the software documentation including the Wind River manuals, our reports and the code itself number these same four ports 0, 1, 2, 3.

It is not necessary to set any switches or jumpers to set serial line characteristics (baud rate, parity, stop bits), these are all set by software.

1.1.2 Console port

The first serial port is the computer console port⁵. When the control computer is configured, it is necessary to connect a serial terminal (such as a VT100) to this port to set boot parameters (section 3.1). During routine operation (including reboots) a console terminal is not required, but it can be helpful for troubleshooting.

The control computer boot parameters must be set to auto-boot (this is the default). When auto-boot is enabled no input from the console keyboard is needed at startup so it is possible to start the program when no console terminal is connected.

When the control computer is reset, the ROM boot monitor sets the console port to 9600 baud, 8 data bits, 1 stop bit and no parity. The console terminal must be set to match. When the control computer boots, first the monitor, then the VxWorks kernel, then the VxWorks command interpreter (executing the startup command file), and finally the application control program write progress messages to the console port (Appendix I). After startup completes, when the program is running normally, all the log messages are written to the console port (they are also written to a file).

Messages may be written at the console port if problems occur, for example if the computer is unable to boot (section 3.4) or the program halts or crashes (section 7.1). This information appears nowhere else. If a console terminal is not connected, it is lost.

1.2 Controllers

Three controllers are custom devices provided by the cyclotron vendor, Scanditronix: the TMC (treatment motion controller), LCC (leaf collimator controller) and DMC (dose monitor controller) [10, 11, 12]. The control computer communicates with each of these over a separate RS232 cable.

Two controllers are commercially available programmable logic controllers (PLC's)[13, 14]. PLC1 mostly handles the cyclotron and beam lines while PLC2 handles therapy equipment. The control computer communicates with both of these over the Ethernet. Both PLC's are programmed using a personal computer that is usually connected to the private network (not shown in fig. 1.1).

⁵This is labelled port 1 on the transition module and in the Motorola hardware documentation, but port 0 in the software documentation

1.3 PLC watchdog

Components external to the control computer (hardware and PLC) ensure that the control system enters a safe state when the control program stops running.

The control program can stop in several ways: it *shuts down* when the operator commands it, it *halts* when the program detects certain conditions, and it may *crash* in response to hardware failures, programming errors, or disturbances in the environment (power, network, etc.).

PLC1 controls a watchdog interlock in the hardwired safety trace that prevents the beam from turning on. The PLC sets this interlock unless a continuous train of pulses (heartbeat) arrives from the running control program.

When the control program is running, it continually toggles the input to a timer network in the PLC. If the control program stops, or is excessively delayed, or communication between the control computer and the PLC is disrupted, the input dwells in one state, the timer times out, and the PLC sets the watchdog interlock.

At this writing the control program usually toggles the PLC about five times per second. The watchdog timeout is set for one half second. The control program is not strictly periodic so the timeout must be set much longer than the typical period.

The PLC watchdog interlock is indicated by a lamp under a reset button inside the HSIS (Hardware Safety Interlock System) cabinet in the power supply room. The button is labelled **THERAPY A WATCHDOG NOT OK/RESET**. When the interlock is set, the lamp is lit.

After the control program begins running the operator must clear the watchdog interlock by pressing this reset button. It is not cleared automatically when the control program begins running and it is not possible to clear it from the X terminal.

1.4 X terminal

Therapists and cyclotron staff operate the program by using an X terminal housed in the therapy control desk in the cyclotron control room, NN143E.

The program can use almost any X terminal or workstation running an X server. The program requires a color display and a keyboard. The program does not use the mouse (but most terminals and workstations will not work if the mouse is not connected). The program graphics cover 1024 by 768 pixels, and it looks best if the display resolution is also this size so the graphics fill the screen.

During routine operation the program should be run without a window manager, although it can run under one.

The control program reads several configuration files when it starts up. Different X terminals are supported by changing the contents of these files. The X terminal IP address appears in the startup file (section 3.3). Other files configure colors, fonts, and keyboard layout (sections 4.3.8, 4.3.11, and 4.3.12).

The X terminal communicates over Ethernet with the therapy control computer and also with a host computer. The terminal's control program and fonts are stored in the host computer file system. When the terminal is turned on, it *boots* (loads its control program from the host). Then when the control computer boots and the control program starts up, it commands the X terminal to load all its fonts from the host. At these times the host computer must be running and must be configured properly. After the X terminal has successfully booted and the control program has started up, it does not communicate with the host.

The therapy control program communicates almost continually with the X terminal: the program refreshes the terminal screen at least once each second, and the terminal sends each keystroke to the program.

The X terminal must already be running (must already have been successfully booted) when the control program begins running (when the therapy control computer boots) and must continue to run as long as the control program runs.

Instructions for configuring the X terminal in use at this writing⁶ appear in appendix D.

1.5 Room display

Therapists in the isocentric treatment room can view treatment parameters and other information on the room display (not shown in Fig. 1.1).

The room display is an ordinary PC⁷ with a color display. It runs the Linux operating system configured to behave like an X terminal, running an X server accessible to the control computer but no window manager. Its resolution is set to 1024 by 768. It does not normally use the PC keyboard or mouse (they are only used to configure the PC).

The room display is optional. To add or remove the room display, it is necessary to revise the startup

⁶Tektronix XP217C, 8MB, 17" color monitor, DEC LK401-style keyboard .

⁷Compaq Evo with TFT7020 17-inch flat panel display

file and restart the control program. The control program reads the IP address of the room display from the startup file (section 3.3). If this IP address is omitted, the control program will not attempt to connect to the room display. However, if the IP address is included but the room display does not respond to the connection attempt (it is not running or not configured properly), the control program will not work (it will not complete its startup sequence).

Unlike the X terminal, the room display PC is self-contained. It has its own disk and file system, so it does not need to obtain its control program or fonts from the host computer.

Some notes on configuring the Linux software on the PC to behave as an X terminal appear in appendix E.

1.6 Host computer

All CNTS programs and data reside in the file system of a host computer that communicates with the control computer and X terminal over Ethernet. The host provides these services to the control system:

1. Provides executable program files and configuration files to the X terminal when it boots, using `tftp` (trivial file transfer protocol).
2. Provides executable program files and the startup command file to the control computer when it boots, using a remote shell facility⁸.
3. Provides the date and time to the control computer when it boots, using the `time` service (on port 37, not the `daytime` service on port 13). This service can optionally be used to reset the time while the control program is running.
4. Provides fonts to the X terminal when control program starts, using `tftp`, NFS or an X font server. The font server is the preferred method.
5. Provides data files to the control program while it is running, using NFS (Network File System).

All services use the TCP/IP network protocol to communicate over Ethernet network hardware.

All services except the last are needed only when the control computer or the X terminal boots. In order to start (or restart) the control computer or X terminal, the host must be running and configured

⁸VxWorks also supports booting by `ftp`, `tftp`, and `BOOTP`, but we found these alternatives do not work with startup scripts so they are unsuitable for our system.

properly. After the control computer and X terminal have successfully booted, they can continue to run even if the host shuts down.

The last service is used frequently (but not continually) while the control program is running. When the program starts up it reads all input data files and stores their entire contents in control computer memory. The running program uses the in-memory contents. Most data files are read again when needed. For example, the entire prescription file is read into memory each time the operator invokes the **Select Patient** operation, because its contents may have changed since the last patient was selected. Output data files are written as data is generated. For example the treatment record file is written each time the beam turns on or off.

The control program can continue to run if the host shuts down. It does not crash if it fails in an attempt to open, read, or write data files. It writes a log message indicating the failure and operations proceed. If a read operation fails, the program continues to use the previously loaded in-memory contents. If a write operation fails, the program usually stores the data in memory so it can be written out after the host becomes accessible again (see section 7 on recovery procedures).

Any computer that can provide a remote shell, `ftp`, time service, X fonts, and NFS over Ethernet could serve as the host. In fact, the services are completely independent so each service could be provided by a different computer: the control computer and X terminal could boot from different hosts, the fonts could reside on a third host and the data files could reside on a fourth host (however the time service must be provided by the control computer boot host).

At this writing there is just one host⁹. Instructions for configuring this host appear in appendix B.

1.7 Department computer cluster

Many computing activities related to neutron therapy including treatment planning and control program software development occur on the department computer cluster. The cluster is a collection of Hewlett-Packard workstations running the HP-UX operating system connected to the hospital network, another Ethernet network which is separate from the control system network (fig 1.1). All the computers in the cluster share a common file system and all boot from a computer called the *cluster server*.

The control system host computer is not a member of the cluster. It has its own file system and boots from its own local disk. The host computer can boot, run, and serve the control computer even when the cluster is shut down. However the host computer and the cluster computers are connected by the hospital network. The host computer uses NFS to enable the treatment planning program running

⁹Hewlett-Packard B160 workstation running HP-UX 10.20 , located in NW267.

on the cluster computers to update the control program prescription file.

The control program is developed on the cluster with Tornado [17], a product from the operating system vendor. The Tornado software enables developers to compile and link programs on their workstations and to boot, run and debug programs on other computers (on the hospital network) through their workstations. The Tornado software requires extensive software installation and configuration on the cluster in addition to the services described here. The completed program does not use Tornado and does not require any Tornado facilities to be installed on the host¹⁰. Each time a new control program version is produced, the files are copied from the cluster to the host.

¹⁰The control program VxWorks configuration does not include the *target agent* and it is not possible to connect to the control program through a Tornado *target server* running on a workstation ([17] pages 4–6, 12–13, [18] p. 18). Moreover, the Tornado software running on the cluster cannot reach the control system private network.

Chapter 2

Network

The control computer, PLCs, X terminal and host communicate over Ethernet hardware using the TCP/IP protocol. They are connected by a *private network* that is dedicated to the Clinical Neutron Therapy System. The host is also connected to the hospital network; control program data files and executable files are provided to the host over this connection. The host is connected to the two networks by two separate network interface cards (NICs) so the two networks are effectively isolated from each other (fig. 1.1 in chapter 1)¹

2.1 Addressing

Each device on the network has a unique built-in *Ethernet address* which is assigned by the manufacturer and cannot be changed. Devices communicate by exchanging *packets* of data which contain the Ethernet address of the recipient. Ethernet usually behaves like a single bus that connects each device to all the others at all times. Every packet written by any device usually reaches all devices. Each device's Ethernet interface hardware responds only to packets that are addressed to it, or to packets that contain a special *broadcast address*². Other packets are ignored by the interface hardware and do not interrupt or affect the device's central processor.

In control programs and configuration files each device is identified by its *IP address*, not its Ethernet address. Assigning IP addresses is an important configuration step. The IP address of a device is not built in, it is set (or changed) when a device is connected to a network, by using some device-

¹This kind of configuration is sometimes called a *private network using a dual homed host interface*.

²Broadcast packets are not unusual on the hospital network. Devices must broadcast when they do not possess the Ethernet address of the intended recipient, for example when using the ARP and BOOTP protocols [15]. At this writing the devices on the private network do not need to use broadcast packets.

Device	IP Address	Name
Control computer	192.168.0.7	elrond
X terminal	192.168.0.6	
Host computer	192.168.0.10	trombone
PLC1	192.168.0.9	
PLC2	192.168.0.5	
Room Display	192.168.0.11	
PLC programmer (PC)	192.168.0.8	

Table 2.1: Devices, IP addresses, and names on the CNTS private network

specific setup procedure³. Translation between IP addresses and Ethernet addresses is performed automatically by built-in TCP/IP software (in the VxWorks and host operating systems and in the X terminal control program).

IP addresses must be assigned with care to ensure that two devices do not receive the same address. An Internet standard recommends that IP addresses in small private networks should be chosen from the range 192.168.0.0 – 192.168.255.255⁴. Table 2.1 shows the addresses in use on the private network at this writing.

It is possible to assign names to act as synonyms for IP addresses. This can make it more convenient to type commands, etc. Table 2.1 shows names for two CNTS devices. The host name appears in the control program startup script (section 3.3). The names are assigned in the boot parameters which are stored in the control computer's nonvolatile memory (section 3.1). The control computer operating system software translates from these names to the corresponding IP addresses by referring to its own in-memory table. The translation does not require communication with any name server computer and it is not necessary for these names to be registered in a name server.

The host computer also has a different IP address on the hospital network, and a name that is registered in a name server computer. The control computer and other devices on the private network never use this IP address.

2.2 Network security

The CNTS private network is not connected to (or reachable from) any other network (the host computer never routes packets from the hospital network to the private network). This avoids most

³At this writing all CNTS devices use *static* (permanent) IP addresses.

⁴RFC 1918, <http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc1918.html>

Device	Port
Control computer (elrond)	4
X terminal (therapy console)	9
Room display	10
Host computer (trombone)	19
PLC1 (cyclotron, watchdog)	22
PLC programmer (PC)	23
PLC2 (therapy equipment)	24

Table 2.2: Devices and ports on the CNTS hub

network security and congestion issues.

The host is also connected to the hospital network, which is connected to the Internet. The University does not operate any “firewall” to keep connection requests from the Internet out of the hospital network. Instead, security is delegated to each computer. We configure the host so it only accepts connections to a few network services from a few computers in the Radiation Oncology department (appendix B). Once a connection is established, the host is protected in the usual way, with user accounts, passwords, file permissions, avoidance of vulnerable services and practices, and frequent backups of operating system and data files (for authorizations and protections pertinent to CNTS see Chapter 4 and Appendix B).

2.3 Hub and wiring

The control computer, PLCs, host computer, and X terminal are connected to a *hub*⁵. located in the cyclotron power supply room, mounted in the same cabinet as the PLC. Table 2.2 shows the hub ports in use at this writing.

The hub was configured when it was installed (appendix C). The configuration is permanent. It should not be necessary to change it. New devices can be plugged into unoccupied sockets at any time; the hub will detect them. (Of course it is still necessary to assign IP addresses to new devices.)

The network is wired with 10Base-T unshielded twisted pair (UTP) cables and RJ45 connectors (not 10Base-2 coaxial cables and BNC connectors). The network is wired to TIA/EIA Category 5 specifications which supports 100 Mbps signals.

On the front of the hub are twenty-four RJ45 sockets (the hub is mounted vertically in the rack so

⁵Cabletron MicroMACC-24E , but almost any hub would do.

the front of the hub faces up and the back faces down). For each socket there is a small green *link light* LED on the front of the hub. When a socket is connected to a device that is powered up, its LED lights up.

Sixteen of the sockets are occupied by short gray cables to a patch panel⁶ mounted in the same rack. Then from the patch panel, there are blue cables to several small *site boxes* located elsewhere in the power supply room and in the control room, NN143E. We plug our computers and terminals into the RJ45 sockets in the site boxes.

The control computer is connected to Ethernet through an AUI port (Application User Interface, a 15-pin D connector) on its transition module. The AUI cable is connected to a transceiver⁷, which connects to a black 10Base-T cable that runs under the floor to the site box under the floor panel near the rack that holds the hub.

The host computer has two network interface cards (NICs). One is connected to the private network through the hub. The other is connected to the hospital network through a cable connected to a socket in a site box under the floor panel behind the PDP11. The hospital network beyond this point is operated by Medical Center Information Systems (MCIS).

The X terminal has an RJ45 socket for direct connection to 10Base-T, so it does not require a transceiver. A short cable runs from the terminal to the site box inside the therapy control desk, under the terminal shelf near the floor.

Other site boxes in the control room are inside the cyclotron control desk and the fixed beam room control desk.

⁶Ortronics, similar to AMP 557870-1

⁷Cabletron 802.3 10Base-T Transceiver .

Chapter 3

Configuration

The control computer and its program are adapted to their environment by setting boot parameters and providing a startup command file. This enables the same executable program file to run without modification with different host computers, file systems, X terminals etc. Therefore it is not necessary to revise the control program to accommodate changes in the department's computing environment as computers are replaced, etc.

3.1 Boot parameters

Set boot parameters to select the host, the operating system file that contains the executable operating system code, and the startup command file that contains the initialization commands that configure the system when it starts up. Other boot parameters set the control computer's user name for authorization on the host, and its network IP address and name.

Boot ROM chips provided by the operating system vendor, Wind River Systems Inc., must be installed in the computer. To set boot parameters, attach a terminal to control computer console port. Change the jumpers in the control computer to enable it to communicate with the terminal (section 1.1.2). Then use the terminal setup screen to set these terminal characteristics:

```
baud rate 9600
8 bits per character
no parity
one stop bit
no new line
```

The vendor's manual tells how to set the boot parameters (section 3.5, pages 65 – 70 in [17], also section 5.8, pages 361 – 362 in [18]). In brief: press the VME crate reset button, and when `Press any key to stop auto boot . . .` appears on the console screen, press a key within seven seconds. At the `[VxWorks Boot]:` prompt, type `p` (print) to list the boot parameters. The screen should display these parameters, followed by the boot prompt.

```
boot device           :ei
processor number      :0
host name             :trombone
file name             :/cnts/tftpdir/wind/mv167/vxWorks
inet on ethernet (e) :192.168.0.7:ffff0000
host inet (h)         :192.168.0.10
user (u)              :cnts
flags (f)             :0x0
target name (tn)      :elrond
startup script (s)    :/cnts/tftpdir/code/iso/isostart
```

```
[VxWorks Boot]:
```

The program files are under the `/cnts/tftpdir` directory on the host computer (although at this writing we use `remsh` not `tftp` to boot).

To change parameters, type `c` (change) at the boot prompt. The boot program prompts for each parameter in order. At the prompt, type in a new value, or type a period (`.`) to clear the value, or type nothing to leave the value unchanged. Then press `RETURN` to be prompted for the next parameter. You must type a period when prompted for the `ftp password` to ensure the computer uses `remsh` to boot. Type `@` to start the boot process.

Changing boot parameters can be inconvenient because a console terminal is not always connected. It is easier to make changes on the host. For example, to boot a differently configured version of the operating system (for troubleshooting or development purposes) we rename a different directory under `/cnts/tftpdir/wind/` to `mv167`. To execute different initialization commands at startup, rename a different file under `/cnts/tftpdir/code/iso` to `isostart`.

3.2 Command interpreter (target shell)

Our configuration of the VxWorks operating system includes a command interpreter called the *target shell* (section 9.2, pages 457 – 464 in [18]). The target shell is needed to execute the startup command file.

After it has processed the startup command file, the target shell normally reads additional commands from the console port. The last line in our startup command file suspends the shell so it cannot process any other commands while the control program is running.

In our configuration the target shell cannot be accessed over the network (section 9.2.6, pages 461 – 462 in [18]).

3.3 Startup command file

When the control program boots, it executes the startup command file named in the boot parameters, normally `/cnts/tftpd/iso/isostart`. Edit this file to select the control program files, the user number and group number for authorization on the host, the file system on the host to mount for NFS, and the IP address of the X terminal.

The command file in use at this writing appears in appendix H. Some notable features are described in the following paragraphs.

The command `taskDelete tPortmapd` deletes the portmapper demon task. This task is automatically started by our configuration of VxWorks, but is not needed by our control program. After this command is executed the control program does not listen on any TCP ports. The control program uses dynamically-assigned UDP ports for NFS¹.

The command `putenv TIMEZONE etc.` identifies the time zone and the dates of the transitions between standard time and daylight savings time. The control computer uses this information to translate from its internal clock to the displayed time.

The command `nfsAuthUnixSet("trombone", 2200, 70, 0, &aup_gids)` identifies the control computer as user 2200 (username `cnts`) and group 70 (`prism`) to the host computer `trombone`, to obtain authorization for accessing NFS files.

The command `nfsMount("trombone", "/cntsdata", "/cntsdata")` NFS mounts the `/cntsdata` filesystem on the host `trombone`, and identifies it with the filesystem named `/cntsdata` in the control program.

The command `hostShow` writes out the control program's in-memory table of host names and IP numbers, which plays a role similar to the `/etc/hosts` file on a Unix system. This confirms that the control program does not need to use the name server.

¹Listening on TCP ports can present potential security problems when a machine is accessible from the Internet, but they are not a problem on a private network.

The command `ld 1,0,"/cnts/tftpdir/vxwindows/support.o"` loads the object file of support functions provided with the Visicom VX-Windows product.

The command `starttime()` executes a function in `support.o` that sets the control computer's clock by accessing the `time` server on the host.

The command `ld 0,0,"/cnts/tftpdir/vxwindows/X11.o"` loads the X Window System object file provided with the Visicom VX-Windows product.

The command `vxClientLibInit(1)` invokes a function in `support.o` that initializes the X Window System.

The command `ld 1,0,"/cnts/tftpdir/code/iso/iso.o"` loads the object file for the control program. All the application code we developed is compiled and linked into this object file.

The command `taskSpawn "tUser",99,8,20000,user,4,"192.168.0.6:0","192.168.0.9", "192.168.0.5","192.168.0.11:0"` starts the control program by spawning the `user` function in `iso.o`. The parameter `192.168.0.6:0` is the IP address and display number of the X terminal. The next two IP addresses identify PLC1 and PLC2, respectively. The last parameter `192.168.0.11:0` is the IP address and display number of the room display.

The room display is optional. To run the control program without the room display, change the sixth parameter of the `taskSpawn` command from 4 to 3 and remove the last parameter.

The command `taskSuspend` suspends the shell.

3.4 Startup process

The control computer boots using the remote shell utility, providing the `cnts` username but no password. The host must be configured as described in appendix B.3. For more about the VxWorks startup process section 8.3.4, pages 435 – 446 in [18].

Appendix I shows the messages that appear on the console terminal during a successful startup. The messages scroll by quickly. Type `ctrl-s` to stop the scrolling and `ctrl-q` to resume it.

The messages through `Memory size ...` are written by the control computer boot ROM monitor and the VxWorks operating system. The messages from `# isostart ...` through `taskSuspend` are written by the target shell as it interprets the startup command file. The re-

maining messages are from the control program itself². These messages include a listing of all tasks running on the control computer and all TCP and UDP ports in use (generated by the VxWorks `i` and `inetstatShow` functions).

While the program is running, progress messages are handled as log messages. After the startup sequence completes, the console terminal displays every log message that the program writes. They can also be viewed on the X terminal display message log screen, or they can be found later in the message log file (section 4.4.4). Appendix J shows the log messages written during a successful startup. If problems occur, error messages will be written instead.

The control computer may fail to boot or the control program may fail to complete its startup sequence. Messages on the console terminal or in the message log can help identify the problem. See section 3.5.4, pages 70 – 71 in [17] and also [19], stored on the department cluster under `/radonc/cnts/notes`.

²The message `trombone:/usr/vxwindows/lib/X11/Xcms.txt: No such file or directory` always appears. It does not indicate any problem.

Chapter 4

Data files

4.1 Directories

Control system data files appear in these directories on host computer:

`/cntsdata/data/iso` Prescription file, accumulating doses file, calibration files, configuration files

`/cntsdata/log/iso` Treatment records, message logs, controller logs, stored fields

Archived data files appear in these directories on the department cluster:

`/radonc/treatdata/cnts/data/iso` Archived prescriptions and accumulated doses

`/radonc/treatdata/cnts/log/iso` Archived treatment records and message logs

The control system data directories are set in the control program code (in `filenames.h`)¹

Other directories under `/cntsdata/data`, `/cntsdata/log`, etc. contain data files used for program development and testing. The `/cntsdata/data/iso-fix` directory stores copies of files in `/cntsdata/data/iso` made before and after editing them by hand. The contents of this directory serve as a record of the changes we make.

¹The `/cntsdata` file system is hard-coded in the program, however in the startup command file `isostart` the `nfsMount` command could associate a different file system on the server with the `/cntsdata` file system on the client (control computer). At this writing we use `/cntsdata` on the host as well.

4.2 Groups

The pertinent groups are `cnts` (group number 103) and `prism` (group number 70). Those who administer and maintain the CNTS data files must belong to both groups.

Users in the `prism` group have permission to read and write files that contain patient data, including the control program prescription file. This group includes therapists and dosimetrists, as well as software developers and system administrators.

The control program is configured to have the permissions of a user in the `prism` group. This is set in the startup command file `isostart`, in the `nfsAuthUnixSet` command.

Users in the `cnts` group have permission to read and write control program configuration and calibration files. This group includes cyclotron engineering staff, as well as some software developers and system administrators

4.3 Files in `/cntsdata/data/iso`

The `/cntsdata/data/iso` directory contains the prescription file, accumulating doses file, calibration files, and configuration files.

4.3.1 Directory

The `/cntsdata/data/iso` directory has owner, group and protection: `jon cnts drwxrwxr-x`.

Only members of the `cnts` group (or root) can create, delete (`rm`), or rename (`mv`) files in the directory. Some files in this directory must be accessible to users in the `prism` group so `world` has read and execute permission for this directory.

4.3.2 Directory contents

The `/cntsdata/data/iso` directory contains a fixed collection of files. Table 4.1 shows a directory listing. Table 4.2 provides a brief description of each file.

The directory listing should always look like Table 4.1. The file names are set in the control program

```

% ls -l /cntsdata/data/iso
total 170
-rw-rw-r--  1 jon      cnts           254 Apr  9 11:05 READ
-rw-rw----  1 jon      prism        3030 Jul  9 09:14 accum.bak
-rw-rw----  1 jon      prism        3702 Jul 14 10:10 accum.dat
-rw-rw-r--  1 jon      cnts          97 May 12 16:37 calfields.dat
-rw-rw-r--  1 jon      cnts         189 Apr  8 14:16 colors.dat
-rw-rw-r--  1 jon      cnts        1091 May 14 15:23 dosecal.dat
-rw-rw-r--  1 jon      cnts       16725 May 12 17:15 exper.dat
-rw-rw-r--  1 jon      cnts         189 Apr  8 14:16 fonts.dat
-rw-rw-r--  1 jon      cnts         902 Apr  8 14:16 keycodes.dat
-rw-rw-r--  1 jon      cnts        1046 Apr  8 14:16 keycodes.hp712
-rw-rw-r--  1 jon      cnts         996 Apr  8 14:16 keycodes.hp715
-rw-rw-r--  1 jon      cnts        1042 Apr  8 14:16 keycodes.hpb160
-rw-rw-r--  1 jon      cnts         902 Apr  8 14:16 keycodes.tekxp
-rw-rw-r--  1 jon      cnts          56 Jul 13 15:39 message.dat
-rw-rw----  1 jon      prism         859 Jul  7 17:47 mlccal.dat
-rw-rw-r--  1 jon      cnts         153 Jun 28 13:24 oper.dat
-rw-rw-r--  1 jon      cnts        1525 Apr  8 14:16 ports.dat
-rw-rw----  1 jon      prism       19843 Jul 14 09:48 prescr.bak
-rw-rw----  1 jon      prism       19842 Jul 14 09:48 prescr.dat
-rw-rw----  1 jon      prism         79 Jul 14 10:13 run.dat
-rw-rw-r--  1 jon      cnts         398 May 12 13:43 timeouts.dat
-rw-rw-r--  1 jon      cnts         127 Jun 21 12:13 tol.dat

```

Table 4.1: Directory listing of /cntsdata/data/iso

code (in `filenames.h`). If any files are missing, or there are any additional files, or the ownership or protection of any file is different, the control program may not work. Moreover this indicates an error by a software developer or system administrator, or a security problem.

The size and date of each file may change however. Most files in the `prism` group are changed several (or many) times every working day by the running control system and host utilities; files in the `cnts` group are only changed when calibrations or configurations are revised.

Files in the `prism` group provide `read` and `write` permission to the group so therapists and the control program itself can read and rewrite them. The leaf collimator calibration file `mlccal.dat` has to be in the `prism` group because it is rewritten by the control program. System administrators and software developers must also belong to this group so they can maintain the files as needed (for example to edit a corrupted file).

Files in the `cnts` group provide `read` and `write` permission to the group, so any software developer, engineer, or system administrator in this (small) group can maintain them (for example to edit configuration files or calibration files). Files in this group also provide `read` permission to `world` so the control program itself, which is in the `prism` group, can read them. However `world` does *not* have `write` permission for these files.

4.3.3 Reading and writing data files

Tables 4.3 and 4.4 show when each file is read and written, and by which program(s).

All input files are read when the control program starts up. Most input files are read again when some operation demands it. For example, the control program attempts to read the dosimetry calibration file each time the operator selects the dosimetry calibration screen in experiment mode. This ensures that the program works with up-to-date data after any revision. All of the calibration files and some other input files are read each time you press the **Refresh** key (except when a run is in progress).

If you suspect a problem with an input file, or you have revised a file, select the appropriate screen or press **Refresh** to force the program to attempt to read the file (or shut down and restart the control program if you must). Check the message log. It should say something like:

```
14-JUL-1999 06:37:10 Dose cal file, success at line 18
```

Then select the screen that displays the data from the file to confirm that the data is correct. If instead it says something like

```
14-JUL-1999 06:37:10 Dose cal file, open error at line 0
```

that usually means the file is missing or has the wrong name or has the wrong protection. If it says something like

```
14-JUL-1999 06:37:10 Dose cal file, conversion error at line  
6
```

(or ...range error ...) that usually means the file contents have the wrong format (or the wrong value). This indicates that the file has become corrupted (or was erroneously edited). If it says something like

```
14-JUL-1999 06:37:10 Dose cal file, timed out at line 0
```

that usually means the file server did not respond.

In case of any error like these, the control program will continue running with the earlier version of data stored in memory. If the errors occur at startup, the program often uses built-in defaults (details in sections 4.3.5 through 4.3.21).

4.3.4 Maintaining data files

All CNTS data files are text files, so they can all be edited with any text editor. All were initially created with a text editor.

During operation, the files marked with w (write) mode in Tables 4.2, 4.3 and 4.4 are updated by the CNTS software (control program and host utilities) and it should not normally be necessary to do anything else to maintain them. However you can resort to a text editor if it becomes necessary to revise them.

In all files, items within each line are separated by spaces and usually the number of spaces does not matter (data need not be aligned in columns). A few exceptions occur where items (such as patient names) may contain embedded spaces. Sometimes items appear in a file which are not used. These items are there for compatibility with earlier versions or to serve as place-holders for future enhancements. These items must be present and must have the right format (floating point number, character string or whatever) but their values do not matter.

The following sections describe all of the files. The meaning of the file contents is described in much more detail in the reference manual [4].

File	Mode	Description
READ	d	Documentation
accum.bak	w*	Copy of an earlier accum.dat
accum.dat	r,w [†]	Accumulated doses and fractions
calfields.dat	r	LCC calibration field sizes
colors.dat	r	X terminal color names
dosecal.dat	r	DMC calibration factors
exper.dat	r	Experiments: studies, fields, machine setups
fonts.dat	r	X terminal font names
keycodes.dat	r	X terminal keypad and function key keycodes
keycodes.hp712	r*	HP 712 keypad and function key keycodes
keycodes.hp715	r*	HP 715 keypad and function key keycodes
keycodes.hpb160	r*	HP B160 keypad and function key keycodes
keycodes.tekxp	r*	X terminal keypad and function key keycodes
message.dat	r	Help screen text (message of the day)
mlccal.dat	r,w	MLC calibration factors
oper.dat	r	Operator authorization
ports.dat	r	X-ray port shapes
prescr.bak	w*	Copy of an earlier prescr.dat
prescr.dat	r,w [†]	Prescriptions: patients, fields, machine setups
run.dat	r,w	Run in progress/Run completed indicator
timeouts.dat	r	Setup timeout, start timeout, other timeouts
tol.dat	r	Tolerance windows for leaves and other settings

- d Documentation only
- r Read by control program and prescription utility program
- w Written by control program only
- w[†] Written by control program and prescription utility
- r* Read by host prescription utility program only
- w* Written by host prescription utility program only
- w[†] Written by treatment planning program and prescription utility

Table 4.2: Description of files in /cntsdata/data/iso

File	Mode	Read when . . .
READ	d	(documentation only, not read by CNTS software)
accum.bak	w*	(not read by CNTS software)
accum.dat	r,w [†]	Startup only
calfields.dat	r	LCC cal. screen selected in experiment mode, startup, refresh
colors.dat	r	Startup only
dosecal.dat	r	Dosimetry cal. screen selected in experiment mode, startup, refresh
exper.dat	r	Patient (actually study) list selected in experiment mode only
fonts.dat	r	Startup only
keycodes.dat	r	Startup only
keycodes.hp712	r*	Prescription utility startup only
keycodes.hp715	r*	Prescription utility startup only
keycodes.hpb160	r*	Prescription utility startup only
keycodes.tekxp	r*	Prescription utility startup only
message.dat	r	Help screen selected, startup, refresh
mlccal.dat	r,w	LCC cal. screen selected in experiment mode, startup, refresh
oper.dat	r	Logout, startup, refresh
ports.dat	r	LCC cal. screen selected in experiment mode, startup, refresh
prescr.bak	w*	(not read by CNTS software)
prescr.dat	r,w [†]	Patient list selected in therapy mode, startup
run.dat	r,w	Startup only
timeouts.dat	r	Startup only
tol.dat	r	LCC cal. screen selected in experiment mode, startup, refresh

- d Documentation only
 - r Read by control program and prescription utility program
 - w Written by control program only
 - w[†] Written by control program and prescription utility
 - r* Read by host prescription utility program only
 - w* Written by host prescription utility program only
 - w[†] Written by treatment planning program and prescription utility
- refresh means when the operator presses the Refresh key (except during a run)

Table 4.3: Reading files in /cntsdata/data/iso

File	Mode	Written when ...
accum.bak	w*	Patient is archived
accum.dat	r,w [†]	Run finishes in therapy mode Write File at Fields screen in therapy mode
mlccal.dat	r,w	Patient is archived Write File when LCC calibration complete
prescr.bak	w*	Fields transferred from treatment planning system Patient is archived
prescr.dat	r,w [†]	Field is superceded Fields transferred from treatment planning system Patient is archived
run.dat	r,w	Field is superceded Beam turns on or run finishes in therapy mode

- d Documentation only
- r Read by control program and prescription utility program
- w Written by control program only
- w[†] Written by control program and prescription utility
- r* Read by host prescription utility program only
- w* Written by host prescription utility program only
- w[†] Written by treatment planning program and prescription utility

Table 4.4: Writing files in /cntsdata/data/iso

4.3.5 accum.bak

The file `accum.bak` is a copy of `accum.dat` written by the `preview` prescription viewer/archiver utility program before each patient is archived. It is not read by the CNTS software. Its contents may be useful if `accum.dat` is corrupted by the prescription utility. This might happen in the unlikely event that the prescription utility crashes while it is rewriting `accum.dat`, or the prescription utility and the control program attempt to rewrite `accum.dat` at the same time (at this writing these have never happened).

The `accum.bak` file is usually not a current backup of `accum.dat` because the control program rewrites `accum.dat` at the end of every treatment run, but it does not rewrite `accum.bak` before doing this.

You cannot copy `accum.bak` over `accum.dat` to recover if a therapist inadvertently archives a patient who is still under treatment, because `accum.dat` may have been rewritten in the meantime. In that situation, you must use the procedure described in section 4.3.6.

4.3.6 accum.dat

The file `accum.dat` records the accumulating fractions and doses for each field. The control program displays the accumulated fractions and total dose in the `FRAC` and `TOTAL MU` columns of the therapy fields display, respectively. This file also records the dose delivered so far today for each field; if this is nonzero the control program displays it in a warning dialog box when you select the field.

The control program also maintains the accumulated fractions and doses in memory and always uses its in-memory copy. The `accum.dat` file exists so this information can be restored when the control program starts up again after a shutdown. The `accum.dat` file also provides a way to recover if the in-memory data becomes invalid (see section 4.3.6 below for how this can happen).

The control program only reads `accum.dat` when it starts up. If the program is unable to read `accum.dat`, it assigns a missing data flag to the fraction, total dose, and today's dose for every field; these are displayed as blanks on the therapy fields display, and a warning appears when the therapist selects any field.

The control program rewrites the entire `accum.dat` file after each treatment run to keep its contents up to date.

The `preview` prescription viewer/archiver program rewrites `accum.dat` when it archives a patient (at this time it removes the patient from `accum.dat`).

You can force the program to rewrite the entire `accum.dat` file at any time with the current in-memory information by pressing the **Write File** key at the fields screen (for any patient) in therapy mode.

Consistency between `accum.dat` and `prescr.dat`

The control program makes the in-memory database consistent with `prescr.dat` each time the program reads that file (usually when the therapist presses **Select Patient**). The program makes `accum.dat` consistent with the in-memory database each time it rewrites that file (usually at the end of a treatment run).

From the time the therapist transfers new fields to `prescr.dat`, until the end of the first treatment run after that, there will be fields present in `prescr.dat` that are absent from `accum.dat`. That is alright. However there should never be fields present in `accum.dat` which are absent from `prescr.dat`. That would indicate that the in-memory database or the file itself has become corrupted.

When a field is removed from `prescr.dat`, it must also be removed from `accum.dat`. This is the only time that a field should be removed from `accum.dat`. The preview prescription viewer/archiver program removes fields from both files when a patient is archived, maintaining consistency between the two files.

Creating `accum.dat`

If `accum.dat` is missing or badly corrupted you should not attempt to create a new one from nothing because it can be difficult to ensure that it is consistent with `prescr.dat`.

Create a new `accum.dat` that is consistent with `prescr.dat` as follows: Shut down the control program, remove the old `accum.dat` if any, then create an empty file named `accum.dat`. Restart the control program, select any patient, then select the fields screen. Press **Write File**. This writes out a properly formatted `accum.dat` with an entry for each patient and field in `prescr.dat`, with all the accumulated doses and fractions set to zero. You only need to replace certain zeroes with the pertinent fractions and doses.

Editing `accum.dat`

It may become necessary to edit the `accum.dat` file. It is possible that the in-memory information may become incorrect. For example, certain (rare) DMC communication errors can cause the control program to record the incorrect dose for a treatment run. The control program detects and indicates these errors (see below) but cannot correct them. In this situation it is necessary to determine the correct dose (from the panel LED's or mechanical counters) and edit it into `accum.dat`.

It is also possible that the contents of `accum.dat` may become out of date. For example, if contact with the file server is lost the control program will continue updating the in-memory information but not the file. If the program shuts down before contact with the file server resumes, the file would be out-of-date when the program starts up again. In this situation it is necessary to edit the file (based on the paper chart or other information outside the computer) to bring it up-to-date before restarting the program.

When you edit `accum.dat`, make copies of the file before and after the edits in `iso-fix` directory. Give each copy a unique name (for example `accum.dat.sep08-1`) to ensure that it does not overwrite any other file. These copies will serve as a record of what you did.

After you edit `accum.dat`, the only way to make the changes effective is to shut down and restart the control program. The control program only reads `accum.dat` when it starts up.

Restoring archived data to accum.dat

It may happen that a therapist archives a patient by mistake before that patient has completed treatment. In that situation you must restore the accumulated doses and fractions from the archive file (see section 4.5), as follows:

First restore `prescr.dat` (section 4.3.18). Then press **Select Patient** to make the in-memory data consistent with the restored `prescr.dat`. Then select the fields screen and press **Write File** to write out an `accum.dat` that contains the restored patient and fields, with accumulated fractions and doses set to zero. Then edit `accum.dat`, replacing zeroes with the actual values recorded in the archive file. Then shut down and restart the control program to read the restored `accum.dat`.

Make copies of `accum.dat` before and after the restore operation in the `iso-fix` directory, taking care to name them so they do not overwrite any other files there.

Format of accum.dat

Here are the first several lines of a sample `accum.dat` file:

```

10 19 08 099 149 12 68 19-AUG-1999 15:35:06 149 lines 12 patients 68 fields
11 2439 Valjean, Jean                2-08-93-06          20-Jul-1999 1 12
21 1 L-25-A Lao                      I N N T           1
25 9 8 351.0 312.5 39.0 39.2 0
21 2 R-40-A Rao                      I N N T           2
25 9 7 738.0 574.1 82.0 0.0 0
.
.
.
```

The first two characters in each line are a two-digit record identification number. There are only four kinds of records: timestamp, patient identification, field identification, and field data. After the time stamp on the first line, there are blocks of data consisting of a single patient identification record followed by several pairs of field identification and field data records. There is just one block of data for each patient.

Patient identification record

There is one record to identify each patient. Each patient appears only once in `accum.dat`. The control program only reads the first two numbers from this record, if you are editing the file you do not need to correct the other items. The two numbers at the end of the record are the index of this patient in the in-memory data structure, and the number of fields for this patient. They were included for diagnostic purposes during program development.

```

11  2439 Valjean, Jean                2-08-93-06        20-Jul-1999  1 12
|      |      |                      |                  |      |
|      |      |                      |                  |      |
|      |      |                      |                  |      N
|      |      |                      |                  |
|      |      |                      |                  | Patient index
|      |      |                      |                  |
|      |      |                      |                  | Date of first data transfer
|      |      |                      | Hospital ID
|      |      | Patient name
| Patient number assigned in treatment plan
Patient record identifier

```

Field identification record

There are two records for each field. These records follow the patient record and appear in order of ascending field sequence number. The first of the two records is the field identifier. The control program only reads the first two numbers from this record, you do not need to correct the other items. The number at the end of the record is the index of this field in the in-memory data structure, included for diagnostic purposes.

```

21  1 L-25-A Lao                      I N N T  1
|      |      |                      |          |
|      |      |                      |          | Field index
|      |      |                      |          | (not used)
|      |      | Field name assigned in treatment plan
|      | Field sequence number assigned by control program
Field record identifier

```

Field data record

The second of the two field records contains the fractions and doses. All of the items in this record are read by the control program. If you are editing the file you must ensure that all of these items are present and have the correct values.

```

25  9  8  351.0  312.5  39.0  39.2  0
|   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |   Warning flag
|   |   |   |   |   |   |   |   (0 = no warning, 1 = suspect, 2 = crashed)
|   |   |   |   |   |   |   |   MU delivered on the day of the timestamp
|   |   |   |   |   |   |   |   Prescribed daily MU
|   |   |   |   |   |   |   |   Total MU delivered so far
|   |   |   |   |   |   |   |   Prescribed total MU
|   |   |   |   |   |   |   |   Fractions delivered so far
|   |   |   |   |   |   |   |   Precribed total fractions
Dose record identifier

```

The last item in this record is a warning flag. The control program sets this flag when some event (such as a DMC communication error) occurs that suggests the information in this record may not be correct. If the flag is clear (zero), the MU and fractions delivered so far are displayed normally on the control program field list screen. The flag is one if a DMC error occurred during a run, and is two if the `run.dat` file suggests that the program crashed during a run (section 4.3.19). If the flag is set (any nonzero value) asterisks appear on the field list screen. If the therapist selects the field in this condition, a dialog box appears, warning that the dose record for this field may be incomplete. If you are editing this record and are confident that the values are correct, you should clear the flag to zero.

Seq.	Contents	Default
1	Setting for leaves 0 – 19 in large calibration field	-10.0
2	Setting for leaves 20 – 39 in large calibration field	10.0
3	Setting for leaves 0 – 19 in midline calibration field	0.0
4	Setting for leaves 20 – 39 in midline calibration field	-2.0

Table 4.5: Leaf collimator calibration field settings in `calfields.dat`

4.3.7 `calfields.dat`

The `calfields.dat` file stores the leaf positions for the two calibration field shapes used during the leaf collimator calibration procedure. The control program does not display this information anywhere, the only way to find out for sure what values are in use is to perform the calibration procedure.

The program reads `calfields.dat` when it starts up, each time the leaf collimator calibration screen is selected in experiment mode, and each time the operator presses the Refresh key (except when a run is in progress). If the program is unable to read `calfields.dat` when it starts up, it initially uses the built-in defaults shown in Table 4.5. (If it is unable to read the file, it writes a log message to indicate this.)

The file contains a list of four leaf settings, one per line, and comment lines that begin with `#`. Leaf settings are identified only by their sequence order in the file (not counting comment lines) as shown in Table 4.5. Fig. 4.1 shows the `calfields.dat` file in use at this writing.

```
#
# calfields.dat: large leaves 0-19, large 20-39, small 0-19, small 20-39
#
-11.0
11.0
-1.0
5.0
```

Figure 4.1: Sample `calfields.dat` file

4.3.8 colors.dat

The `colors.dat` file stores the names of the colors used in the control program displays. It is possible to adjust the color scheme by changing this file, without changing program code. This capability is needed to make the displays look right, because the same color name can appear quite different on different models of display hardware.

The control program reads this file when it starts up. If the program is unable to read `colors.dat`, it uses the built-in defaults which are the same as in Fig. 4.2.

This file is simply a list of 23 color names, one to a line. The names must be chosen from the list recognized by the X window system (see appendix D in volume two of [9] or the file `/usr/lib/X11/rgb.txt` on the host computer). Fig. 4.2 shows the `colors.dat` file in use at this writing. These colors look best on the Tektronix XP217 X terminal.

```
white
black
gray
AntiqueWhite
LightGray
red
LightPink
SpringGreen
PaleGreen
SeaGreen
ForestGreen
blue
LightBlue
navy
yellow
LightYellow
gold
SaddleBrown
tan
firebrick
orchid
lavender
coral
```

Figure 4.2: Sample `colors.dat` file

4.3.9 dosecal.dat

The `dosecal.dat` file stores the dosimetry calibration constants that the program uses to calculate calibration factors that are written into the DMC at the beginning of every run. The control program displays these constants in the panel on the right on the dosimetry calibration display.

The program reads `dosecal.dat` when it starts up, each time the dosimetry calibration display is selected in experiment mode, and each time the operator presses the Refresh key (except when a run is in progress). If the program cannot read `dosecal.dat` when it starts up, it assigns a missing data flag to each constant. In this condition the control program will not set up the DMC.

The first line contains the date that the file was last updated in `dd-mmm-yyyy` format, a space, then the name of the engineer who is responsible for the contents of the file. These appear on the dosimetry calibration screen.

The rest of the file contains a list of calibration constants, one per line, and comment lines that begin with `#`. Comment lines can appear anywhere after the first line.

Calibration constants are identified only by their sequence order in the file (not counting comment lines). Note that the order of the constants in the file is *not* the same as the order they appear on the dosimetry calibration display. There are no labels or keywords in the file. We use comment lines to annotate the file but the comment lines are ignored by the program. Table 4.6 describes the dosimetry calibration constants, listed in the order they must appear in the file. Figure 4.3 shows the `dosecal.dat` file in use at this writing.

If the values read from the file do not lie within the ranges shown in Table 4.6 the program sets the dosimetry calibration interlock. In this condition the control program will not set up the DMC. The only way to clear this interlock is to correct the file and command the control program to read it again.

Seq.	Name	Units	Minimum	Maximum	Example
1	DOSE RATE	MU/min	0.0	99.9	60.0
2	TIME FACTOR		1.00	2.00	1.49
3	CAL GAIN 1		600	800	711
4	CAL GAIN 2		600	800	721
5	P/T DEADLINE	HOURS	1	72	12
6	XCFAC		10000	40000	32000
7	YCFAC		-40000.0	-10000.0	-20000
8	XRFACT		0	2000	1000
9	YRFACT		-2000	0	-1000
10	MAXRSET		2000	10000	9999
11	MINRSET		0	700	0
12	IONFAC		0	10000	5000
13	SERVMIN		0	0	0
14	SERVMAX		0	10	5
15	LOWFAC		-10	0	-1
16	HIGHFAC		0	50000	32000
17	DOSE R. TIME WIN.	SEC	0.0	300.0	10.0
18	LOW DOSE R. FCT		0.0	1.0	0.1

Table 4.6: Dosimetry calibration constants in dosecal.dat

```
12-May-1999 risler
#
# dosecal.dat
#
#       5-Apr-1999  J. Jacky,R. Risler Change calvolt1,2 phys name and date
#       12-Apr-1999  J. Jacky,R. Emery calvolt2 720->715 try to avoid dose diff
#       12-May-1999  J. Jacky,R. Risler new calvolt1, calvolt2
#
# first, the four editable cal factors: d_rate, t_fac, calvolt1, calvolt2
#
60.0
1.49
711
721
#
# next, the read-only cal factors:
# pt_deadline, xcfact, ycfact, xrfact, yrfact, maxrset, minrset,
# ionfac, servmin, servmax, lowfac, highfac
#
12
32000
-20000
1000
-1000
9999
0
5000
0
5
-1
32000
#
# drate_sec, drate_frac: after drate_sec, require drate_frac of expected dose
#
10.0
0.1
```

Figure 4.3: Sample dosecal.dat file

4.3.10 exper.dat

The `exper.dat` file stores all the experimental studies and fields used in experiment mode.

The control program reads the `exper.dat` file when the operator presses the **Select Patient** key in experiment mode.

The format of `exper.dat` is the same as `prescr.dat` (section 4.3.18). The control program does not use certain items in experiment mode, but they still must be present in the file.

4.3.11 fonts.dat

The `fonts.dat` file stores the names of the fonts used in the control program displays. It is possible to change the fonts by changing this file, without changing any program code. Font names are stored in a file because different host computers provide different collections of fonts.

The control program reads this file when it starts up. If the program is unable to read `fonts.dat`, it uses the built-in defaults which are the same as in Fig. 4.4.

The file is simply is list of 6 font names (with * wildcards), one to a line. The font names are chosen from the list displayed by executing the `xlsfonts` command on the host computer. Fig. 4.4 shows the `fonts.dat` file in use at this writing. These fonts are provided by the HP-UX version 10.20 font server.

```
*-helvetica-bold-r-*-240-75-*  
*-helvetica-bold-r-*-180-75-*  
*-helvetica-medium-r-*-180-75-*  
*-helvetica-bold-r-*-140-75-*  
*-lucidatypewriter-medium-*-180-75-*  
*-courier-bold-r-*-18-180-75*
```

Figure 4.4: Sample `fonts.dat` file

4.3.12 `keycodes.dat` etc.

The `keycodes.dat` file stores the keycodes for the keypad keys and function keys. Keycodes are stored in a file (rather than depending on the X keysym mechanism) so each operation can be assigned to the key at a particular physical location on the keyboard, regardless of the keysym assigned to that key (especially, regardless of whether the keysym might have been reassigned in some nonstandard way by some other software).

The control program reads this file when it starts up. If the program is unable to read `keycodes.dat`, it uses the built-in defaults which are the same as in Fig 4.5.

The file contains a list of 28 keycodes in decimal notation, one to a line. After the keycode on each line there is a comment that describes the function of that key, but the control program does not read this information. Keycodes are identified only by their sequence order in the file. Fig. 4.5 shows the `keycodes.dat` file in use at this writing, which is for the Tektronix XP217 X terminal with the DEC LK401-style keyboard. The comments in the file indicate the sequence order. Each line is always associated with the same control program function (for example the fifth line is always the keycode for **Select Patient**) but the keycode will be different for different model keyboards.

To find the keycodes for each key, execute the `xmodmap -pk` command on the host while using the X terminal or workstation of interest as your X console.

In addition to `keycodes.dat`, the `/cntsdata/data/iso` directory contains `keycodes.hp712`, `keycodes.hp715`, `keycodes.hp160`, and `keycodes.tekxp` for HP 712, 715 and B160 workstations and the Tektronix XP217 X terminal (`keycodes.dat` is a copy of `keycodes.xp217`). These other keycodes files are required by the `preview` neutron prescription viewer/archiver utility. They must not be deleted.

```
16, /* PF1, refresh */
126, /* PF2, help */
103, /* PF3, msg_page */
95, /* PF4, log_message */
116, /* KP7, select_patient */
125, /* KP8, select_field */
133, /* KP9, field_summary */
140, /* KP-, store_field */
115, /* KP4, gantry_psa */
123, /* KP5, filter_wedge */
124, /* KP6, leaf_collim */
132, /* KP-commma ("separator"), dose_intlk */
113, /* KP1, override_cmd */
122, /* KP2, auto_setup */
130, /* KP3, cancel_run */
129, /* KP-enter, select_op */
120, /* KP-0, cancel */
121, /* KP-decimal, login */
15, /* F1, expt_mode */
23, /* F2, dose_cal */
106, /* F3, collim_cal (<shift>-F3 on Tek kbd) */
39, /* F4, large field */
47, /* F5, small field */
55, /* F6, write file */
63, /* F7, tmc_page */
71, /* F8, lcc_page */
79, /* F9, dmc_page */
87 /* F10, plc_page */
```

Figure 4.5: Sample keycodes .dat file

4.3.13 message.dat

The `message.dat` file contains the text that appears in the message of the day section of the help screen. To change the message, simply edit this file.

The control program reads this file when it starts up, each time the help screen is selected, and each time the operator presses the Refresh key (except when a run is in progress).

4.3.14 mlccal.dat

The `mlccal.dat` file contains the calibration constants for the leaf collimator. The control program displays these constants on the leaf collimator calibration screen.

The program reads `mlccal.dat` when it starts up, each time the leaf collimator calibration screen is selected in experiment mode, and each time the operator presses the Refresh key (except when a run is in progress). If the program cannot read `mlccal.dat` when it starts up, it assigns a missing data flag to each constant. In this condition the control program will not set up the LCC.

The control program rewrites `mlccal.dat` when the operator invokes the **Write File** operation at the end of the calibration procedure.

The file contains a list of 120 calibration constants (three for each leaf), one per line, and comment lines that begin with #. Calibration constants are identified only by their sequence order in the file (not counting comment lines) as shown in Table 4.7. Fig. 4.6 shows the beginning of a sample `mlccal.dat` file.

The first line of the file looks like a comment line but its contents are significant. It shows who did the most recent calibration, and when. This line is written by the control program when the file is updated at the end of the calibration procedure. After the initial "# " (pound sign and blank), this line shows the username of the operator who was logged in, and the date and time of the update. The username and date must each be a string not containing any spaces, but separated by spaces, occupying no more than 80 characters in all. The control program displays these two items on the LCC Calibration screen (it ignores any text following the date).

If the values read from the file do not lie within the ranges shown in Table 4.7 the program sets the collimator calibration interlock. In this condition the control program will not set up the LCC. The only way to clear this interlock is to correct the file and command the control program to read it again.

Lines	Contents	Minimum	Maximum
1 – 40	MINPOS for leaves 0 – 39	200.0	400.0
41 – 80	MAXPOS for leaves 0 – 39	0.0	200.0
81 – 120	SCAFAC for leaves 0 - 39	-3276.8	3276.8

Table 4.7: Leaf collimator calibration constants in mlccal.dat

```
# DORMAN 25-MAR-2002 12:08:45
# mlccal file --- all minpos, then all maxpos, then all scafac
#
294.5
293.9
292.0
290.6
294.0
296.2
293.2
297.0
295.6
296.9
291.2
292.7
290.1
291.0
293.1
298.0
294.1
296.6
292.8
291.7
.
.
.
```

Figure 4.6: Beginning of a sample mlccal.dat file

4.3.15 oper.dat

The `oper.dat` file contains the operator identification and the authorization that is checked when an operator logs in and also when an operator selects experiment mode.

The program reads this file when it starts up, each time an operator logs out, and each time the operator presses the Refresh key (except when a run is in progress). If the control program cannot read `oper.dat` at startup, it uses the built-in default with one user who is authorized for experiment mode and whose username and password are empty strings (just type RETURN).

The file consists of a sequence of lines with one operator per line. There can be up to twenty lines. On each line there is an operator username, that operator's password, and an authorization flag. Usernames and passwords are case sensitive, by convention we always assign usernames and passwords in uppercase. Usernames and passwords can be up to 16 characters long. The authorization flag is an integer. If the integer is nonzero, that operator is authorized for experiment mode. Fig. 4.7 shows a sample `oper.dat` file.

```
JON AAAAA 1
DAVE BBBBB 1
RISLER CCCCC 1
ROB DDDDD 1
DORMAN EEEEE 1
KUAN FFFFF 1
BOB GGGGG 0
ANGUS HHHHH 0
LAURA IIIII 0
JEFF JJJJJ 0
MONTY KKKKK 0
```

Figure 4.7: Sample `oper.dat` file

4.3.16 ports.dat

The `ports.dat` file contains the leaf settings for the X-ray port film fields (useful ports are not exactly square, they have stepped sides).

The program reads this file at startup, each time the leaf collimator calibration screen is selected in experiment mode, and each time the operator presses the Refresh key (except when a run is in progress). If the program cannot read `ports.dat` at startup, it initially uses the built-in defaults which are the same as shown in Fig. 4.8.

The file contains an entry for each of the five port fields in this order: 10 cm, 16.5 cm, 20.5 cm, 24.5 cm and 28.5 cm \times 32.5 cm. Each entry begins with one line of identification. This line must begin with “21 ”, after that any string may appear. The control program does not read the contents of this line, it identifies port fields by their sequence order in the file. Next come four lines of leaf settings in the same format as in `prescr.dat` (section 4.3.18). Fig. 4.8 shows the beginning of the `ports.dat` file in use at this writing.

```

21  10 x 10 PORT
24 0  -5.8  -5.4  -5.2  -5.0  0.0  0.0  0.0  0.0  0.0  0.0
24 1  -5.6  -5.4  -5.2  -5.0  0.0  0.0  0.0  0.0  0.0  0.0
24 2   5.6   5.4   5.2   5.0  0.0  0.0  0.0  0.0  0.0  0.0
24 3   5.8   5.4   5.2   5.0  0.0  0.0  0.0  0.0  0.0  0.0
21  16.5 x 16.5 PORT
24 0  -9.45  -9.05  -8.85  -8.65  -8.45  -8.25  0.0  0.0  0.0  0.0
24 1  -9.25  -9.05  -8.85  -8.65  -8.45  -8.25  0.0  0.0  0.0  0.0
24 2   9.25   9.05   8.85   8.65   8.45   8.25  0.0  0.0  0.0  0.0
24 3   9.45   9.05   8.85   8.65   8.45   8.25  0.0  0.0  0.0  0.0
21  20.5 x 20.5 PORT
24 0 -11.65 -11.25 -11.05 -10.85 -10.65 -10.45 -10.25 0.0 0.0 0.0
.
.
.

```

Figure 4.8: Beginning of a sample `ports.dat` file

4.3.17 `prescr.bak`

The file `prescr.bak` is a copy of `prescr.dat` written by the `preview` prescription viewer/archiver utility program before each patient is archived and before a field is superceded. It is not read by the CNTS software. Its contents may be useful if `prescr.dat` is corrupted by the prescription utility. This might happen in the unlikely event that the prescription utility crashes while it is rewriting `prescr.dat` (at this writing this has never happened).

The `prescr.bak` file is usually not a current backup of `prescr.dat` because the treatment planning program often appends new patients and fields to `prescr.bak`, but it does not rewrite `prescr.bak` before doing this.

You cannot copy `prescr.bak` over `prescr.dat` to recover if a therapist inadvertently archives a patient who is still under treatment, because new fields may have been added to `prescr.dat` in the meantime. In that situation, you must use the procedure described in section 4.3.18.

4.3.18 `prescr.dat`

The `prescr.dat` file contains all the patients and fields, including all the prescribed settings. This information appears on the patients screen, the fields screen, the field summary screen, and the various subsystem screens. Items in `prescr.dat` identify the treatment plan where each field originated. This may be useful for quality assurance purposes.

The control program reads `prescr.dat` at startup and each time the patients screen is selected in therapy mode. The `prism` treatment planning program appends new patients and fields to the end of `prescr.dat` each time a therapist performs a prescription transfer operation. The `preview` prescription viewer/archiver utility rewrites `prescr.dat` each time a therapist archives a patient or supersedes a field.

Capacity of `prescr.dat`

There are built-in upper limits to the number of patients and fields the control program can read. It can read at most 100 patients and 500 fields total, and at most 40 fields for a single patient. If any of these limits are exceeded in `prescr.dat` the control program writes a log message, rejects the entire file, and continues to use the previously loaded in-memory contents.

Consistency of `prescr.dat`

The `prescr.dat` file consists of blocks of data, where each block contains a patient identifier and one or more fields for that patient. It is usual to have several blocks for the same patient at different locations in the file (a patient's blocks need not be contiguous).

For each patient, the control program internally identifies each field by its sequence order in the file, disregarding fields for other patients (so the first field for each patient is numbered 1). To maintain consistency (between `prescr.dat` and `accum.dat` for example) each field must keep the same sequence number (within its patient's fields) as long as the patient is under treatment.

Consistency is ensured by only appending new fields to the end of the file, never rearranging them or deleting any until the patient completes treatment. Then all of the patient's fields must be removed at one time (in a single operation). This discipline is observed by the `prism` treatment planning program (which adds patients and fields) and the `preview` archiver program (which removes them). In the unlikely event that it becomes necessary to edit `prescr.dat` by hand, you must observe the same discipline.

Restoring archived data to `prescr.dat`

It may happen that a therapist archives a patient by mistake before that patient has completed treatment. In that situation you must restore the patient's data from the archive (see section 4.5). Simply append that patient's entire archive file to the end of `prescr.dat`. After that, you must restore the accumulated doses and fractions (section 4.3.6).

Make copies of `prescr.dat` before and after the restore operation in the the `iso-fix` directory, taking care to name them so they do not overwrite any other files there.

Format of `prescr.dat`

Here are the first several lines of a sample `prescr.dat` file:

```

11 2539 Karenina, Anna                8-81-23-97          3-Aug-1999
12 NO PHYS NAME                       0.0      0.0
13 This is a 3-Dimensional Tx Plan
21 1 L-45-A H&N                        I N N T 2539 6      3-Aug-1999 6:21:43
22 11 0 1331.0      0.0 121.0 2 0 0 270.0 3-Aug-1999 laura X T 0
23 120.0 50.0 50.0 180.0 180.0 45.0 45.0
24 0 -3.2 -3.2 -2.8 -2.4 0.0 0.0 0.0 0.0 0.0 0.0
24 1 -3.2 -3.2 -3.2 -2.3 0.0 0.0 0.0 0.0 0.0 0.0
24 2 4.0 4.0 4.0 4.0 0.0 0.0 0.0 0.0 0.0 0.0
24 3 4.0 4.0 3.7 3.7 0.0 0.0 0.0 0.0 0.0 0.0
21 2 L-45-P H&N                        I N N T 2539 6      3-Aug-1999 6:21:43
22 11 0 1364.0      0.0 124.0 2 2 0 270.0 3-Aug-1999 laura X T 0
.
.
.
```

The first two characters in each line are a two-digit record identification number. There are seven kinds of records: patient identification (11), prescription information (12), comment (13), field identification (21), field dose (22), treatment motions (23) and leaf positions (24).

The file consists of blocks of data, where each block begins with a patient identification, prescription information, and comment record in that order, followed by one or more fields, each consisting of a field identification, field dose, and treatment motion record, then four leaf position records, in that order.

Name	Units	Minimum	Maximum
FRAC		0	99
TOTAL MU	MU	0.0	9999.9
DOSE	MU	0.0	999.8
WEDGE TYPE		0	3
WEDGE ROT		0	3
LEAF 0 – 19	cm	-15.00	3.5
LEAF 20 – 39	cm	-3.5	15.00
GANTRY	deg	0.0	359.9
COLLIMATOR	deg	75.0	284.0
TURNTABLE	deg	62.0	298.0

Table 4.8: Settings in `prescr.dat`

There can be (usually are) more than one block for each patient, separated by blocks for other patients (all of a patient's data need not be contiguous in the file).

Table 4.8 shows the prescribed settings stored in `prescr.dat`. The coordinate system conventions for the settings appear in Appendix B of [6]. If a setting value read from the file does not lie within the range shown in the table, the control program will accept the file but it will display a warning message when the field is selected and indicate that the setting is not ready.

Field identification record

In this record the field sequence number is merely the sequence number within the block; each block starts over at 1, so when there is more than one block for the patient, these field numbers are not unique within that patient. The control program does not display the field sequence number from the file. Instead it computes a unique field sequence number and displays that on the fields screen. This computed sequence number does not appear in `prescr.dat` (it does appear in `accum.dat`).

```

21  1 L-45-A H&N                I N N T 2539 6 3-Aug-1999 6:21:43
|    | |                      |         | |
|    | |                      |         | |   Plan time stamp in Prism
|    | |                      |         | |   Case number in Prism
|    | |                      |         | |   Patient number in Prism
|    | |                      |         | |   Not used
|    | |                      |         | |
|    | | Field name in Prism treatment plan
|    | | Field sequence number within the block, assigned by Prism
Field identification record number

```

The control program displays the field name on the fields screen. The field name must be separated from the field sequence number by one space, then the next thirty characters are taken as the field name (so the field name can include embedded blanks). At least nine more padding characters must follow the field name. The control program does not use the contents of these characters, but they must be there (at this writing the treatment planning program always writes “ I N N T ”).

The items at the end of the record identify the treatment plan in the Prism treatment planning system where this field originated: patient number, case number, and plan time stamp (plans in Prism are uniquely identified only by their time stamp). These may be useful for quality assurance purposes. These items are optional; if they are not present, the control program will still accept the file. The control program does not display these items.

This record also shows the date the field was transferred from the treatment planning system into the control system, and the username (on the treatment planning computer) of the therapist who performed the transfer. The program displays these on the fields screen. These items are optional; if they are not present, the control program will still accept the file.

Three optional flags appear at the end of this record (Table 4.10). If the completion flag is S (superceded) or F (film), the control program displays that same flag in the STATUS column on the fields screen; if the flag is X the program displays a blank in this position if the prescribed dose has not been delivered, or C (completed) if the prescribed dose has been delivered, or E (exceeded) if the prescribed total dose has been exceeded.

Completion flag	Description
S (superceded)	An operator has marked the field as superceded (by another field), whether or not the prescribed total dose has been delivered.
F (film)	An operator has marked the field as a film field not used for treatment, just for X-ray films
X	Otherwise (more codes may be defined in the future)

Origin flag	Description
T (transferred)	Transferred from treatment planning system
X	Otherwise (more codes may be defined in the future)

Place-holder	Description
0	Place-holder (more codes may be defined in the future)

Table 4.10: Flags in `prescr.dat`

Treatment motion record

At this writing only the turntable rotation angle and the gantry rotation angle are used by the control program. The program displays these settings on the field summary screen and the gantry/couch screen. The other items in this record must be present and are always written by the treatment planning program.

```

23 120.0 50.0 50.0 180.0 180.0 45.0 45.0
|      |      |      |      |      |      |
|      |      |      |      |      |      |      Gantry stop angle (not used)
|      |      |      |      |      |      |      Gantry rotation
|      |      |      |      |      |      |      Couch top rotation (not used)
|      |      |      |      |      |      |      Turntable rotation
|      |      |      |      |      |      |      Couch longitudinal position (not used)
|      |      |      |      |      |      |      Couch lateral position (not used)
|      |      |      |      |      |      |      Couch vertical position (not used)
|      |      |      |      |      |      |      Treatment motion record identification number

```

Leaf position record

There are four of these records for each field, with sequence number 0 for leaves 0 through 9, 1 for leaves 10 through 19, 2 for leaves 20 through 29 and 3 for leaves 30 through 39. The control program displays these settings on the leaf collimator screen.

```

24 0 -3.2 -3.2 -2.8 -2.4 0.0 0.0 0.0 0.0 0.0 0.0
|   |   |   |   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |   |   |   |      Leaf 10*a+9
|   |   |   |   |   |   |   |   |   |   |      Leaf 10*a+8
|   |   |   |   |   |   |   |   |   |   |      Leaf 10*a+7
|   |   |   |   |   |   |   |   |   |   |      Leaf 10*a+6
|   |   |   |   |   |   |   |   |   |   |      Leaf 10*a+5
|   |   |   |   |   |   |   |   |   |   |      Leaf 10*a+4
|   |   |   |   |   |   |   |   |   |   |      Leaf 10*a+3
|   |   |   |   |   |   |   |   |   |   |      Leaf 10*a+2
|   |   |   |   |   |   |   |   |   |   |      Leaf 10*a+1
|   |   |   |   |   |   |   |   |   |   |      Leaf 10*a (leaf 0 when seq. number is 0, leaf 10 when it is 1, etc.)
|   |   |   |   |   |   |   |   |   |   |      Sequence number a (0, 1, 2 or 3)
|   |   |   |   |   |   |   |   |   |   |      Leaf position record identification number

```

4.3.19 run.dat

The run.dat file indicates when the control program may have crashed during a treatment run (at this writing this has never happened).

The control program reads run.dat when it starts up. It writes the file when the beam turns on and then rewrites at the end of the run.

When the program starts up, if run.dat still indicates that a run is in progress, it is evidence that the program crashed during that run, so the control program sets the incomplete flag for the field identified in the file. Asterisks will appear on the fields screen and the therapist will be warned that the treatment record for that field may be incomplete. If the control program cannot read run.dat (it is missing or unreadable), the program does not set the incomplete flag for any fields.

During a run, run.dat records the run in progress:

```
1 2498 7, started Karenina, Anna , ANT SUPRACLAV
| | | | | | |
| | | | | | Field name assigned in plan
| | | | | Patient name
| | | | Run status
| | | | Field sequence number assigned by control program
| | | Patient number assigned in treatment planning program
Run in progress flag (0 = no run in progress, 1 = run in progress)
```

Between runs, run.dat records the most recently completed run:

```
0 2498 7, finished Karenina, Anna , ANT SUPRACLAV
| | | | | | |
| | | | | | Field name assigned in plan
| | | | | Patient name
| | | | Run status
| | | | Field sequence number assigned by control program
| | | Patient number assigned in treatment planning program
Run in progress flag (0 = no run in progress, 1 = run in progress)
```

If the program crashes during a run, you should revise both accum.dat and run.dat before restarting the program.

Seq.	Default	Units	Description
1	5	seconds	X server timeout (must be longer than file server timeout)
2	5	seconds	File server timeout, most files (after open, read, write)
3	300	seconds	File server timeout, log files (4500 lines)
4	15	minutes	DMC setup timeout (after auto setup DMC)
5	30	seconds	DMC start timeout (after all interlocks clear, CON START)
6	10	seconds	Beam plug timeout (after DMC timer enable off, FC1 closes)
7	2	seconds	Settling time for DMC relays (sensed through PLC)

Table 4.11: Timeout intervals in `timeouts.dat`

4.3.20 `timeouts.dat`

The `timeouts.dat` file contains the intervals that the program waits before timing out in several situations. The control program reads this file when it starts up. If it cannot read the file, it uses the built-in defaults shown in Table 4.11.

The file contains a list of seven timeout intervals, one per line, and comment lines that begin with `#`. The events associated with each interval are identified by their sequence order in the file (Table 4.11). Fig. 4.9 shows the `timeouts.dat` file in use at this writing.

```
# timeouts.dat
#
# 25-Jan-1999 J. Jacky First version
# 12-May-1999 J. Jacky Increase file timeout from 10 to 15 sec, X 10 to 20
#                   X timeout *must* be longer than file timeout!
#
# x_timeout (sec), file_timeout_sec, file_longtimeout_sec
20
15
300
#
# setup_min, start_sec, plugwait_sec, relay_sec
#
15
30
10
3
```

Figure 4.9: Sample `timeouts.dat` file

Seq.	Default	Units	Description
1	0.10	cm	Open leaves (difference between prescribed, actual)
2	0.10	cm	Closed leaves (difference between prescribed, actual)
3	0.10	deg	Gantry rotation (difference between prescribed, actual)
4	0.10	deg	Collimator rotation (difference between prescribed, actual)
5	0.10	deg	Turntable rotation (difference between prescribed, actual)
6	0.01		P/T correction factor (difference between manual, automatic)
7	1.0	mbar	Pressure (difference between sensors)

Table 4.12: Tolerances in `tol.dat`

4.3.21 `tol.dat`

The `tol.dat` file contains the tolerances or differences that the control program permits in several situations. The program displays the pressure tolerance on the dosimetry calibration screen, the others are not displayed anywhere.

The control program reads this file when it starts up, each time the leaf collimator calibration screen is selected in experiment mode, and each time the operator presses the Refresh key (except when a run is in progress). If the program cannot read the file when it starts up, it initially uses the built-in defaults shown in Table 4.12.

The file contains a list of seven tolerances, one per line, and comment lines that begin with #. The setting associated with each tolerance is identified by its sequence order in the file (Table 4.12). Fig. 4.10 shows the `tol.dat` file in use at this writing.


```
#
# tolerances file: open leaves, closed leaves, gantry, collim, turnt
#
0.15
0.20
0.5
0.7
0.7
#
# pt_factor, press
#
0.01
1.0
```

Figure 4.10: Sample `tol.dat` file

4.4 Files in `/cntsdata/log/iso`

The `/cntsdata/log/iso` directory contains treatment records, message logs, controller logs, and stored fields.

4.4.1 Directory

The `/cntsdata/log/iso` directory has owner, group and protection: `jon prism drwxrwx---`.

Members of the `prism` group have write access to this directory, so they can create, delete (`rm`) and rename (`mv`) files in the directory. This is necessary because the control program is configured to have the permissions of a user in the `prism` group, and the control program must be able to create files here.

World has no access to this directory.

4.4.2 Directory contents

The `/cntsdata/log/iso` directory contains an ever-changing collection of files. When the control program is running normally, at least two new files are created here every day. More can be created at the operator's command. Occasionally, large numbers of files are moved out to an archive

File	Description
dmcyyyy-mmdd-hhmmss.dat	DMC logs
iso-dyyyy-mmdd-hhmmss.log	Message logs written on demand
isoyyyy-mmdd-hhmmss.log	Message logs written automatically
isotr-dyyyy-mmdd-hhmmss.log	Treatment records written on demand
isotryyyy-mmdd-hhmmss.log	Treatment records written automatically
lccyyyy-mmdd-hhmmss.dat	LCC logs
patiiiifldjj.dat	Stored therapy fields
plcyyyy-mmdd-hhmmss.dat	PLC logs
studyiiiifldjj.dat	Stored experiment fields
tmcyyyy-mmdd-hhmmss.dat	TMC logs

An example of `studyiiiifldjj.dat` is `study00001fld12.dat`.

An example of `isotryyyy-mmdd-hhmmss.log` is `isotr1999-0907-064734.log`.

Table 4.13: Description of files in `/cntsdata/log/iso`

directory (section 4.6).

All of the files in this directory are written by the control program. Their owner, group and protection are: `cnts prism -rw-rw-rw-`.

Table 4.13 provides a brief description of each kind of file. Table 4.14 shows when a new version of each kind is created.

There are only a few kinds of files, but the directory usually contains many files of each kind. The control program gives each file a unique name, usually based on the date and time it was created. For example the treatment record file `isotr1999-0907-064734.log` was created on Sept. 7, 1999, at 34 seconds after 6:47 AM. The usual directory listing in alphabetic order shows files in order of their creation.

The following sections describe the files.

4.4.3 Controller logs

Controller log files record the message traffic (in both directions) between the control program and the attached controllers (DMC, LCC, TMC and PLC). The formats of these messages are described in vendors' documentation [10, 11, 12, 14] and summarized in the reference manual [4]. The logs also contain time stamps written at one minute intervals.

File	New file when ...
dmcyyyy-mmdd-hhmmss.dat	Write File at DMC screen
iso-dyyyy-mmdd-hhmmss.log	Write File at message log screen
isoyyyy-mmdd-hhmmss.log	Startup, then midnight every night
isotr-dyyyy-mmdd-hhmmss.log	Write File at message log screen
isotr-yyyy-mmdd-hhmmss.log	Startup, then midnight every night
lccyyyy-mmdd-hhmmss.dat	Write File at LCC screen
patiiiifldjj.dat	Store Field in therapy mode
plcyyyy-mmdd-hhmmss.dat	Write File at PLC screen
studyiiiifldjj.dat	Store Field in experiment mode
tmcyyyy-mmdd-hhmmss.dat	Write File at TMC screen

Table 4.14: Writing files in /cntslog/log/iso

The program writes a controller log file when the operator presses the **Write File** key while the screen for that controller is visible on the display.

The control program writes out the entire contents of its in-memory controller message history buffer to the file. The buffer holds up to 4500 lines of messages. If the program has not been running long enough to fill the buffer, the file will be shorter.

4.4.4 Message logs

Message log files record events that occur as the control program runs. A short fragment from one of these files appears in Appendix J.

The control program creates and writes most message log files automatically. These files have names of the form `isoyyyy-mmdd-hhmmss.log` (for example `iso1999-0907-064734.log`). The program creates a message log file when it starts up, then writes messages for the rest of the day. The file remains open all day. At midnight the program closes the file and creates a new one, which remains open through the following day. This continues, with a new file created each night, until the program shuts down. As a result, most message log file names indicate a time near midnight (such as `iso1999-0910-000005.log`).

It is possible that the control program may lose contact with the file server for some time (because the server is down, or there is a network problem). During this time the program does not write messages to the log file, but they continue to accumulate in the in-memory buffer. If the file server is not available at midnight, the control program does not attempt to open a new file, and the old one will span two days (or more). When contact with the server is re-established, the program writes a

message like this:

```
03-SEP-1999 12:40:31 File server responding, reading and writing files  
again
```

It indicates that messages preceding this one are missing from the file.

You can write out the entire contents of the in-memory message buffer (up to 4500 messages) by pressing the **Write File** key while the message log screen is displayed. This provides a way to save the messages that occurred when the file server was inaccessible (and all the others as well). Files created this way are named `iso-dyyyy-mmdd-hhmmss.log` (with a `-d` for “on demand”), for example `iso-d1999-0907-104607.log`.

4.4.5 Stored fields

Stored field files contain actual machine settings recorded at the operator’s demand.

The program creates a stored field file when the operator presses the **Store Field** key³.

The names of the stored field files include the number of the patient (or study) that was selected when the operator invoked **Store Field**, and the (unique) number of the next higher field for that patient (study). For example `study00001fld12.dat` is field 12 for experimental study 1. The field number is incremented on each **Store Field** operation, so there will be unique filenames even if the same initial field is stored multiple times.

The format of a stored field file is exactly the same as one block of the prescription file `prescr.dat` (section 4.3.18), consisting of patient identification records and field data records for one field. The patient (or study) number and field number are the same as appear in the file name. The patient (or study) name corresponds to the patient (study) number, and the field name is entered by the operator during the **Store Field** operation. The prescribed doses and fractions are set to zero, and at this writing the optional items in the field data records are omitted. Fig 4.11 shows the stored field file `study00001fld12.dat`.

The stored field files are in the proper format so they could be appended to the prescription file `prescr.dat` (section 4.3.18) or the experiment file `exper.dat` (section 4.3.10). At this writing there is no CNTS software that performs this function.

³At this writing, we have removed the **Store Field** operation from the program, but there are still some stored field files in the archive.

```

11      1 Square fields                9-87-85-66          4-Aug-97
12 Selected by risler                  0.0          0.0
13 Written by therapy control program
21 12 large field, 1st iteration      I N N T
25  0  0    0.0    0.0 -10000.0    0.0  0
23 160.9   31.7   68.4  180.2  180.0    0.0    0.0
24 0 -10.0 -10.0 -10.0 -10.0 -10.0 -10.0  -9.9 -10.0 -10.1 -10.0
24 1  -9.9 -10.1 -10.3 -10.0  -9.9  -9.9 -10.0  -9.9 -10.0 -10.0
24 2  10.1  10.1  10.3  10.0  10.0  10.0  10.1  10.1  10.1  10.0
24 3  10.1  10.0  10.1  10.0  10.0  10.1  10.1  10.1  10.1  10.1

```

Figure 4.11: Sample stored field file, `study00001fld12.dat`

4.4.6 Treatment records

Treatment record files record the state of the treatment (or experiment) session and therapy machine each time the beam turns on or off in the isocentric treatment room, and each time a run finishes (and also at some other times, see below).

The control program creates and writes most treatment record files automatically. These files have names of the form `isotryyyy-mmdd-hhmmss.log` (for example `isotr1999-0907-064734.log`). The program creates a new treatment record file each time it starts up, and at midnight each night (the same times that it creates a new message log file, see section 4.4.4). On days when there are no runs, the file is created but it remains empty.

A treatment record is written whenever the DMC timer enabled signal turns on (off). This indicates that FC1 has opened (closed) while the beam plug is open and usually indicates that the beam has turned on (off) in the isocentric treatment room. The contents of the records makes it possible to distinguish records written during treatment or experiment runs, and records written during test mode.

A treatment record is also written whenever the operator indicates that a run has finished by pressing a key at the therapy console in response to a control program message box (CONFIRM at the end of a normal run or EXIT RUN after various interruptions or exceptional conditions). At this time the beam should already be off. This treatment record stores the official dose for the run (which is usually a fraction of one MU higher than the dose recorded in the beam-off record).

In a treatment run with no interruptions, three treatment records are written: a beam-on record, a beam-off record, and a run-finished record. If the treatment run is interrupted and resumed, there will be an additional pair of beam-off/beam-on records for each interruption/resumption. There should be just one run-finished record after the last beam-off record in the run.

It is possible that the control program may lose contact with the file server for some time (because the server is down, or there is a network problem). During this time the program does not write treatment records to the file, so there is a gap in the file where records are missing. These gaps are not indicated by any message in the treatment record file itself, but they occur at the same time as the corresponding gaps in the message log file, which are indicated there by `File server responding ... again` messages (section 4.4.4).

When contact with the server is re-established, you can write out the entire contents of the in-memory treatment record buffer (up to 4500 lines) by pressing **Write File** key while the message log screen is displayed (this also writes out the message log buffer, see section 4.4.4). This provides a way to save the records that occurred when the file server was inaccessible (and all the others as well), in the same format as the automatically written treatment record files. Files created this way are named `isotr-dyyyy-mmdd-hhmmss.log` (with a `-d` for “on demand”), for example `isotr-d1999-0907-104607.log`.

Treatment record format

Each treatment record records pertinent information about the state of the treatment session and therapy machine at the time it was written. Each record contains 15 lines. There is a blank line after each record.

Some of the items in the treatment records originate in data files in `/cntsdata/data/iso`, usually `prescr.dat` (or `exper.dat`) but also `dosecal.dat`. Others are measured or computed by the control program. Items in each record identify the treatment plan file on the host computer where the prescribed data originated. This may be useful for quality assurance purposes.

A utility program that displays and prints the contents of the treatment records in several easy-to-read formats is planned.

Here are three treatment records written during a run with no interruptions. A line-by-line description follows.

```
07-MAY-2002 09:38:38 Beam on - start of run
Patient 3948 Doe, John , 6 s.30.L boost
KIM 8 1 888.0 111.1 14 26-Apr-2002 16:04:22 30-Apr-2002 monty
111.0 111.0 0.0 0.0 0.0 3.22 3.22 0.00 60.0 0 X T 0 0.0 0.0
1 0 1023.8 21.8 0.989 704 716 712 724 risler 17-Apr-2002
60 deg 60 deg 90 deg 90 deg SMALL SMALL
90.0 90.0 270.0 270.1 240.0 240.3 50.90 62.70 120.10 180.60
-3.50 -3.47 -3.00 -3.23* 0.00 0.01 0.00 0.01 0.00 -0.01
0.00 0.00 0.00 0.01 0.00 -0.04 0.00 -0.01 0.00 0.01
-4.50 -4.49 -4.50 -4.43 -4.20 -4.18 -2.70 -2.73 0.00 -0.01
0.00 -0.01 0.00 -0.02 0.00 -0.01 0.00 0.00 0.00 0.13
3.40 3.43 2.70 2.67 2.00 2.04 1.30 1.34 0.00 -0.02
0.00 -0.02 0.00 -0.04 0.00 -0.02 0.00 0.03 0.00 0.01
3.20 3.23 2.90 2.92 0.00 0.04 0.00 0.04 0.00 -0.02
0.00 0.00 0.00 0.03 0.00 -0.02 0.00 0.00 0.00 -0.01
```

```
07-MAY-2002 09:40:40 Beam off in ended phase
Patient 3948 Doe, John , 6 s.30.L boost
KIM 8 1 888.0 111.1 14 26-Apr-2002 16:04:22 30-Apr-2002 monty
111.0 111.0 0.0 111.0 110.4 3.22 3.22 2.01 60.0 0 X T 0 0.0 0.0
1 0 1023.8 21.8 0.989 704 716 712 724 risler 17-Apr-2002
60 deg 60 deg 90 deg 90 deg SMALL SMALL
90.0 90.0 270.0 270.1 240.0 240.3 50.90 62.70 120.10 180.60
-3.50 -3.47 -3.00 -3.23* 0.00 0.01 0.00 0.01 0.00 -0.01
0.00 0.00 0.00 0.01 0.00 -0.04 0.00 -0.01 0.00 0.01
-4.50 -4.49 -4.50 -4.43 -4.20 -4.18 -2.70 -2.73 0.00 -0.01
0.00 -0.01 0.00 -0.02 0.00 -0.01 0.00 0.00 0.00 0.14
3.40 3.44 2.70 2.66 2.00 2.04 1.30 1.34 0.00 -0.02
0.00 -0.02 0.00 -0.04 0.00 -0.02 0.00 0.03 0.00 0.01
3.20 3.24 2.90 2.92 0.00 0.04 0.00 0.04 0.00 -0.03
0.00 0.00 0.00 0.03 0.00 -0.03 0.00 0.00 0.00 -0.01
```

```
07-MAY-2002 09:41:00 Run finished in termwait phase
Patient 3948 Doe, John , 6 s.30.L boost
KIM 8 2 888.0 222.3 14 26-Apr-2002 16:04:22 30-Apr-2002 monty
111.0 111.0 111.2 111.2 111.3 3.22 3.22 2.02 60.0 0 X T 0 111.2 110.6
1 0 1023.8 21.8 0.989 704 716 712 724 risler 17-Apr-2002
60 deg 60 deg 90 deg 90 deg SMALL SMALL
90.0 90.0 270.0 270.1 240.0 240.3 50.90 62.70 120.10 180.60
-3.50 -3.47 -3.00 -3.23* 0.00 0.01 0.00 0.01 0.00 -0.01
0.00 0.00 0.00 0.01 0.00 -0.04 0.00 -0.01 0.00 0.01
-4.50 -4.49 -4.50 -4.43 -4.20 -4.18 -2.70 -2.73 0.00 -0.01
0.00 -0.01 0.00 -0.02 0.00 -0.01 0.00 0.00 0.00 0.14
...
```

Line 1: Record identification

```
07-MAY-2002 09:38:38 Beam on - start of run
|
|
|           Event type
Date, time record written in dd-mmm-yyyy hh:mm:ss format
```

Line 2: Patient and field identification

Here the field name comes from the treatment plan, but the field sequence number is assigned by the control program.

```
Patient  3948 Doe, John           , 6 s.30.L   boost
|
|           |           |           |
|           |           |           |
|           |           |           | Field name in plan
|           |           |           | Field sequence number assigned by program
|           | Patient name (or study name)
|           | Patient number assigned in treatment planning system, or study number
Patient (therapy mode) or Study (experiment mode)
```

Line 3: Fractions, total dose, treatment plan

The items at the end of this record identify the treatment plan where this field originated. This may be useful for quality assurance purposes. The first item in the record is the control system operator who performed the treatment and the item at the end is the planning system user who transferred the prescription. The meaning of the other items is described in the reference manual [4].

```
KIM  8   1  888.0  111.1 14 26-Apr-2002 16:04:22 30-Apr-2002 monty
|
|
|           |           |           | | |
|           |           |           |
|           |           |           |           |           |
|           |           |           |           |           | Field transfer date
|           |           |           |           |           | Plan timestamp in treatment planning system
|           |           |           |           |           | Case number in treatment planning system
|           |           |           |           |           | Total dose delivered to date
|           |           |           |           |           | Prescribed total dose
|           |           |           |           |           | Fractions delivered to date
Oper Prescribed fractions
```


appear at the end of the field dose record in `prescr.dat` (section 4.3.18).

Warning flag	Description
0	No warning
1	Accumulated dose may not be correct
2	Control program may have crashed during a treatment run

Completion flag	Description
C (completed)	The total dose delivered is greater than or equal to the prescribed total dose.
S (superceded)	An operator has marked the field as superceded (by another field), whether or not the prescribed total dose has been delivered.
X	Otherwise (more codes may be defined in the future)

Origin flag	Description
T (transferred)	Transferred from treatment planning system
X	Otherwise (more codes may be defined in the future)

Place-holder	Description
0	Place-holder (more codes may be defined in the future)

Table 4.15: Flags in treatment record

Code	Description
0	OK
1	Pressure 1 unavailable
2	Pressure 2 unavailable
3	Pressure 1 outside range
4	Pressure 2 outside range
5	Pressures outside range
6	Pressure sensors differ
7	No pressure information available
8	P/T factor differs from manual
9	P/T factor outside range
10	Temperature outside range
11	Manual correction factor expired

Table 4.16: Pressure/temperature status codes

Line 5: Dosimetry calibration

```

1 0 1023.8 21.8 0.989 704 716 712 724 risler 17-Apr-2002
| | | | | | | | | | |
| | | | | | | | | | Date dosecal.dat last revised
| | | | | | | | | | User who last revised dosecal.dat file
| | | | | | | | | | adjust calvolt2 factor loaded into DMC
| | | | | | | | | | adjusted calvolt1 factor loaded into DMC
| | | | | | | | | | calvolt2 constant from dosecal.dat file
| | | | | | | | | | calvolt1 constant from dosecal.dat file
| | | | | | | | | | Pressure/Temperature correction factor
| | | | | | | | | | Temperature, degrees C
| | | | | | | | | | Pressure, mbar
| | | | | | | | | | Pressure/temperature status code
| | | | | | | | | | Pressure/temperature mode (0 = manual, 1 = automatic)

```

The adjusted `calvolt` factors are obtained from the corresponding constants by dividing by the P/T correction factor.

The pressure/temperature status codes are described in Table 4.16. Some of these codes can only appear in records written when manual pressure/temperature mode has been selected (see the reference manual [4]).

Line 6: Filter/Wedge

60 deg	60 deg	90 deg	90 deg	SMALL	SMALL				
									Actual flattening filter selection
									Prescribed flattening filter selection
									Actual wedge rotation
									Prescribed wedge rotation
									Actual wedge selection
									Prescribed wedge selection

Here the actual position information originates from switches in the treatment head.

Line 7: External motions

90.0	90.0	270.0	270.1	240.0	240.3	50.90	62.70	120.10	180.60
									Actual top rot.
									Actual couch height
									Actual couch longitude
									Actual couch lateral position
									Actual turntable rotation
									Prescribed turntable rotation
									Actual collimator rotation
									Prescribed collimator rotation
									Actual gantry rotation
									Prescribed gantry rotation

Here the actual positions originate from potentiometers or encoder signals from the therapy equipment.

Line 8-15: Leaves

-3.50	-3.47	-3.00	-3.23*	0.00	0.01	0.00	0.01	0.00	-0.01
				Leaves 2, 3, 4 prescribed and Actual positions					
				Leaf 1 has been overridden by the operator					
				Actual leaf 1 position					
				Prescribed leaf 1 position					
				Actual leaf 0 position					
				Prescribed leaf 0 position					

Here the actual value of leaf 1 differs from the prescribed value by more than the allowed tolerance; the asterisk indicates that the operator has overridden this setting to allow the treatment to proceed. Any setting which has been overridden will be marked in this way.

Seven more rows in the same format show the remaining leaves.

Accumulating doses and fractions

In each treatment run at least three treatment records are written (beam-on, beam-off, run-finished) but usually only lines 1, 3 and 4 should show significant changes as dose accumulates:

Here is the beam-on record, showing the prescribed and preset dose and treatment time. The accumulated dose and elapsed time are still zero.

```
07-MAY-2002 09:38:38 Beam on - start of run
Patient  3948 Doe, John           , 6 s.30.L    boost
KIM  8   1  888.0   111.1 14 26-Apr-2002 16:04:22 30-Apr-2002 monty
111.0 111.0    0.0   0.0   0.0  3.22  3.22  0.00  60.0  0 X T  0   0.0   0.0
...
```

Here is the beam-off record. It shows the accumulated dose in both channels and the elapsed time for the run. However the number of fractions, the total dose delivered, and the daily dose have not yet been incremented.

```
07-MAY-2002 09:40:40 Beam off in ended phase
Patient  3948 Doe, John           , 6 s.30.L    boost
KIM  8   1  888.0   111.1 14 26-Apr-2002 16:04:22 30-Apr-2002 monty
111.0 111.0    0.0 111.0 110.4  3.22  3.22  2.01  60.0  0 X T  0   0.0   0.0
...
```

Here is the run-finished record. Here the number of fractions, the total dose and the daily dose have all been incremented. This is the first run of the day for this field so the daily dose 111.2 is the same as the official dose for the run. See the reference manual [4].

```
07-MAY-2002 09:41:00 Run finished in termwait phase
Patient  3948 Doe, John           , 6 s.30.L    boost
KIM  8   2  888.0   222.3 14 26-Apr-2002 16:04:22 30-Apr-2002 monty
111.0 111.0  111.2 111.2 111.3  3.22  3.22  2.02  60.0  0 X T  0 111.2 110.6
...
```

4.5 Files in `/radonc/treatdata/cnts/data/iso`

The `/radonc/treatdata/cnts/data/iso` directory on the department cluster contains archived prescriptions, doses and fractions.

There should be enough capacity in the disk file system to store many years of archives on-line.

4.5.1 Directory

The `/radonc/treatdata/cnts/data/iso` directory has owner, group and protection: `jon prism drwxrwx---`.

Members of the `prism` group have write access to this directory, so they can create, delete (`rm`) and rename (`mv`) files in the directory. This is necessary because the therapists belong to the `prism` group, and they create the files here (by running the `preview` prescription viewer/archiver program).

World has no access to this directory.

4.5.2 Directory contents

The `/radonc/treatdata/cnts/data/iso` directory contains an ever-growing collection of files. Two new files are created here each time a therapist archives a patient.

All of the files in this directory are written by the `preview` prescription viewer/archiver program. The owner of each file is the therapist who archived the patient. The group and protection are: `cnts prism -rw-rw-r--`.

4.5.3 Archived prescriptions

Each archived prescription file has a name of the form `prescr-nnnnn-yyyy-mmdd.dat`, composed of the patient number, and the date when the patient was archived. For example, `prescr-02535-1999-0723` is the prescription for patient 2535, archived on July 23, 1999. The date is included so each file will have a unique name, even if the same patient returns after completing a previous course of treatment.

Each prescription archive file is an exact copy of all that patient's data blocks in the prescription file

`prescr.dat` (section 4.3.18) at the time the archive operation occurred. The blocks appear in the archive file in the same order they appeared in the prescription file.

If it becomes necessary to restore a patient's prescription from the archive, follow the instructions in section 4.3.18.

4.5.4 Archived doses and fractions

Each archived dose file has a name of the form `accum-nnnn-yyyy-mmdd.dat`, composed of the patient number, and the date when the patient was archived, for example `accum-02535-1999-0723.dat`. Each archived dose file is created at the same time as the corresponding archived prescription.

Each dose archive file is an exact copy of that patient's data block in the accumulation file `accum.dat` (section 4.3.6) at the time the archive operation occurred.

If it becomes necessary to restore a patient's doses and fractions from the archive, follow the instructions in section 4.3.6.

4.6 Files in /radonc/treatdata/cnts/log/iso

The `/radonc/treatdata/cnts/log/iso` directory on the department cluster contains treatment records, message logs etc. moved out of `/cntslog/log/iso` (section 4.4) for permanent storage here.

There should be enough capacity in the disk file system to store many years of records and logs on-line.

Chapter 5

Summary files

Several utility programs write files that can be useful for maintenance, troubleshooting, and quality assurance. Some of these files record activity pertinent to the control program, such as adding or archiving prescriptions. Others are summaries of therapy control program activity.

The programs that write these files are separate from the therapy control program, and the files are not needed to operate the control program. Therefore these programs and files are optional, and they may change as our needs evolve. This chapter describes the files and their contents at this writing (May 2002).

Currently, these files are all in `/cntsdata/log/iso`. They are:

`prism*.log` (for example `prism2002-0503-082946.log`)

`preview*.log` (for example `preview2002-0503-083027.log`)

`prevlog.err`
`prevlog.out`

`eventlog.err`
`eventlog.out`

`treatrec.err`
`treatrec.out`

`accum.err`
`accum.out`

You can get a quick look at how well the system has been running recently by executing the command `tail accum.err` and so on for the other `.err` files. You can see a day-by-day history for all the patients currently under treatment by executing `more accum.out`.

The `prism*.log` and `preview*.log` files are written as needed (frequently), the others are updated automatically every night. They are written in the same directory that contains the treatment records and log files written by the therapy control program. Usually, each file remains there for several months, until all of the patients referenced in the file have been archived. Eventually, we archive older files by moving them to `/radonc/treatdata/cnts/log/iso`.

5.1 Automation

Some of these files are created or maintained by running scripts automatically. The scripts run on the cluster server, not the therapy control computer. The scripts are stored in the `/usr/local/cnts` directory in the cluster file system. The file `crontab.nightly` in that directory contains instructions that direct the Unix `cron` facility to execute the scripts shortly before midnight every night. See the Unix documentation on `crontab` for detailed instructions and an explanation of the file format. In brief: a user must be authorized in `/var/adm/cron/cron.allow`, then that user can start the nightly jobs by executing the command `crontab crontab.nightly` once.

The scripts send email to alert staff about situations that might need attention. Their email addresses are in the file `mail-warning`.

The following sections describe the files in more detail.

5.2 Prescription transfer summaries

5.2.1 `prism*.log`

Each time a therapist uses the Prism treatment planning program to add new prescription data to the control program prescription data file `prescr.dat`, Prism creates a file whose name has the form `prismyyyymmdd-hhmmss.log`. This file contains one line that records that the transfer took place, and records the username (on the treatment planning computer) of the therapist who performed the transfer.

For example, the file `prism2002-0501-134605.log` contains this line:

```
01-May-2002 13:46:05 Prism fields appended to prescr.dat by monty
```

The new prescription data might contain one field or several, but always contains data for just one patient. Each time data is transferred for a different patient, another file is written.

The file name format was chosen so that files appear in chronological order in most contexts. For example, the command `cat prism*.log` writes a chronological listing of prescription transfer activity.

In the future we may write more detailed prescription transfer logs (including the list of fields that were transferred) in `/prismdata/cnts-logs` in the cluster filesystem.

5.3 Archiver summaries

5.3.1 `preview*.log`

The prescription viewer and archiver program (named **preview**) creates a log file each time it runs. Each file name has the form `previewyyyy-mmdd-hhmmss.log`. Each of these files records significant activities that occur during the program run, which might include archiving one or more patients (removing the patients and all of their fields from the prescription data file).

5.3.2 `prevlog.out`

The file `prevlog.out` is a cumulative summary of the archiver activity recorded in all the `preview*.log` files. To see a chronological listing of recent archiver activity, you can use a command like `tail -n 50 prevlog.out | more`.

The file `prevlog.out` is maintained each night by the `prevlog-cmd` script, which merely appends pertinent lines from all of that day's `preview*.log` files.

Here is an excerpt from `prevlog.out`:

```
07-APR-2002 23:40:02 prevlog i preview
08-APR-2002 23:40:02 prevlog i preview
08-Apr-2002 10:48:43 Archiving patient 3887 Karenina, Anna
09-APR-2002 23:40:02 prevlog i preview
```

```
09-Apr-2002 11:23:10 Archiving patient 3887 Karenina, Anna
09-Apr-2002 11:23:12 Archived successfully
09-Apr-2002 11:24:03 Archiving patient 3893 Sharpe, Becky
09-Apr-2002 11:24:03 Archived successfully
10-APR-2002 23:40:00 prevlog i preview
11-APR-2002 23:40:00 prevlog i preview
```

There are only three kinds of lines in `prevlog.out`. The lines that contain `prevlog i preview` indicate when the script ran. Here they indicate that the script ran once every night, shortly before midnight. Following lines indicate activity earlier that same day.

A successful archive operation is indicated by two consecutive lines containing `Archiving patient` and `Archived successfully`. This example shows two successful archive operations on April 9, 2002.

An unsuccessful archive operation is indicated by a line containing `Archiving patient not` followed by `Archived successfully`. This example shows an unsuccessful archive operation on April 8, 2002.

5.3.3 `prevlog.err`

The file `prevlog.err` is a cumulative summary of errors reported by the previewer/archiver program, also maintained by `prevlog-cmd`.

Here is an excerpt from `prevlog.err`, showing more information about the unsuccessful archive operation on April 8, 2002.

```
07-APR-2002 23:40:02 prevlog i preview
08-APR-2002 23:40:02 prevlog i preview
08-Apr-2002 10:50:03 Couldn't open (w) archive file /radonc/treatdata/cnts/dat
08-Apr-2002 10:50:03 Files may be corrupted
09-APR-2002 23:40:02 prevlog i preview
10-APR-2002 23:40:00 prevlog i preview
```

At the time when the archive operation was attempted, the workstation where the previewer/archiver was running could not reach the file server, apparently due to an intermittent network problem at the workstation. The warning `Files may be corrupted` was painful: `prescr.dat` had been truncated to zero length. We had to restore `prescr.dat` from `prescr.bak`.

5.4 Event log summaries

5.4.1 eventlog.out

At this writing the file `eventlog.out` is always empty. The `eventlog-cmd` script opens it but writes nothing into it. We may use it for something in the future.

5.4.2 eventlog.err

The file `eventlog.err` is a cumulative summary of significant event log entries including startup, shutdown, and a few kinds of faults. It is maintained each night by the `eventlog-cmd` script, which merely appends pertinent lines from all of that day's `iso*.log` files. Here is a brief excerpt.

```
23-APR-2002 23:50:00 eventlog i iso
24-APR-2002 23:50:00 eventlog i iso
24-APR-2002 20:01:18 Single sample corrupted dose A 269.9 MU
25-APR-2002 23:50:00 eventlog i iso
26-APR-2002 23:50:00 eventlog i iso
27-APR-2002 23:50:00 eventlog i iso
28-APR-2002 23:50:00 eventlog i iso
29-APR-2002 23:50:00 eventlog i iso
29-APR-2002 12:14:38 Operator requested shutdown
29-APR-2002 12:16:04 Starting
30-APR-2002 23:50:00 eventlog i iso
01-MAY-2002 23:50:00 eventlog i iso
```

5.5 Treatment record summaries

5.5.1 treatrec.out

The file `treatrec.out` is always empty. The `treatrec-cmd` script opens it but writes nothing into it. We may use it for something in the future.

5.5.2 treatrec.err

The file `treatrec.err` is a cumulative summary of significant treatment record entries and other information that can be inferred from treatment records. It is maintained each night by the `treatrec-cmd` script,

Here is an excerpt from `treatrec.err`:

```
11-APR-2001 16:25:29 error reading treatment record at line 2
12-APR-2001 23:55:00 treatrec
12-APR-2001 10:49:36 pat 3353 fld 16: run dose 9.5 differs from preset 120.0
13-APR-2001 23:55:00 treatrec
14-APR-2001 23:55:01 treatrec
15-APR-2001 23:55:01 treatrec
16-APR-2001 23:55:01 treatrec
16-APR-2001 16:43:12 pat 2318 fld 1: run dose 4.0 differs from preset 102.0
17-APR-2001 23:55:01 treatrec
18-APR-2001 23:55:01 treatrec
19-APR-2001 23:55:01 treatrec
19-APR-2001 10:40:03 pat 3353 fld 17: unfinished run at end of file
20-APR-2001 23:55:01 treatrec
20-APR-2001 10:38:04 pat 3353 fld 17: dose warning flag set in log
21-APR-2001 23:55:01 treatrec
```

The following paragraphs describe all the entries that can occur in `treatrec.err`. They are shown here without their usual preceding timestamps. Some entries might suggest that more investigation is needed. You can use each entry's time stamp to find more details in the event log files `iso*.log` and treatment record files `isotr*.log`.

`treatrec`

Indicates when the script ran. The following entries refer to events earlier that same day.

`unrecognized record type`

`error reading treatment record`

The script could not interpret a line in the treatment record file. Perhaps the line is incorrectly formatted or the file is corrupted.

on follows another on

off does not follow on

run finished does not follow off

Treatment records do not occur in the expected sequence (beam on, beam off, run finished, but it is not unusual to see beam on again after beam off).

total dose value exceeds prescribed value

daily dose value exceeds prescribed value

fractions value exceed prescribed value

Self-explanatory.

run dose value differs from preset value

The run did not deliver the preset dose. This is not unusual; it occurs whenever an operator stops the run deliberately for any reason, or an interlock stops the run, or the machine delivers an unexpectedly low dose rate.

dose warning flag set in log

The control program detected a fault. The dose in this treatment record might not be correct.

incomplete record at end of file

unfinished run at end of file

A treatment record file ends in the midst of a run. Perhaps the program shut down, halted, or crashed during the run.

5.6 Treatment histories and accumulated dose summaries

The files `accum.err` and `accum.out` monitor the records of accumulating total doses and fractions maintained by the therapy control program. In addition to summarizing the data, they warn of possible errors. They are maintained each night by the `accum-cmd` script.

The therapy control program keeps an in-memory record of the accumulated total dose and number of fractions for each field under treatment. At the end of each treatment run it updates the in-memory record for the current field with the dose delivered during that run, and also writes out all of the in-memory records (for all fields) to the `accum.dat` file in `/cntsdata/data/iso`. At the end of each treatment run, the control program also appends a record to the end of the current treatment record file `isotr*.log` in `/cntsdata/log/iso` that records the dose delivered during that run. These treatment records also contain the information for that field from the in-memory records at that time.

It is possible that any of these files might become corrupted or contain incorrect data (this has happened a few times) so it is important that they be checked. Therapists often check the in-memory records (displayed on the control program field list screen) against their paper records, but it is also possible to check the files against each other. For each field, the sum of all the treatment run doses and treatment days in all of the pertinent treatment records should equal the total dose and number of fractions in `accum.dat`. The `accum-cmd` script performs this comparison and reports its findings in the `accum.err` and `accum.out` files.

5.6.1 `accum.out`

The file `accum.out` shows a day-by-day history for all the patients currently under treatment. Although it is named `accum.out`, it does not contain any information from `accum.dat`; all of its contents are from `run finished` treatment records for that field in the entire collection of treatment record files ¹.

The `accum.out` file contains a section for each field currently under treatment (that is, for every field in `accum.dat`). For some fields, the section only says (for example)

```
field 4 not found
```

This means there are no treatment records for that field. This occurs when treatments have not yet begun, or when the field is a film field, or for other reasons.

If treatment records can be found, there is a table. Table 5.1 shows an example.

The column headings describe the prescription. For this field, the prescribed number of fractions is 8, the prescribed daily dose is 90 MU, and the prescribed total dose is 720 MU. The prescription is read from the most recent treatment record (it should be the same in every treatment record for this field). The fractions and total dose always appear twice (the column headings are always the same

¹The `accum-cmd` script invokes the `accum` script, which in turn invokes the `pat-field` script, which repeatedly invokes the `treatrec` function in `Treatrec.pm` and generates `accum.out`.

		Patient 3957 Karenina, Anna				
		Field 2 lt lat lat				
		Fractions		Daily	Total	Total
		Acc	Calc	Meas	Accum	Calc
		8	8	90	720	720
		--	--	----	-----	-----
25-APR-2002	11:06:44	1	1	45.0	45.0	45.0
25-APR-2002	11:09:58	1	1	45.1*	90.1	90.1
26-APR-2002	10:37:32	1	1	0.0	90.1	90.1
26-APR-2002	10:40:27	2	2	90.1	180.2	180.2
30-APR-2002	10:46:39	3	3	90.1	270.3	270.3
01-MAY-2002	10:52:49	4	4	90.2	360.5	360.5
02-MAY-2002	10:52:45	5	5	90.2	450.7	450.7
		--	--	----	-----	-----
Totals		5	5		450.7	450.7

Table 5.1: Entry for one field from `accum.out`

but the column contents may differ).

The rest of the table is a day-by-day treatment history, with one row for each treatment run for that field. The first columns show the date and time when each run ended. There might be more than one run on the same day. In this example there are two runs on April 25 and April 26. This is not unusual, it occurs whenever it was not possible to deliver the entire prescribed daily dose in one run.

The two columns labelled `Fractions` show the fraction number, determined by two different methods. The fraction number should increment by one on each day when dose is delivered, no matter how many runs occur on that day. The fraction number does not increment until a run occurs that delivers at least 0.5 MU (the first run on April 26 was a false start where no dose was delivered). The `Acc` column shows the number of fractions recorded in the treatment record for that run, which is a copy of the in-memory value that was stored in the control program at the time when the treatment record was written. The `Calc` column shows the number of fractions calculated by the script from the entire collection of treatment records up to and including the run on this line. The calculation is performed by counting the number of days where treatment records show that dose was delivered for this field. These two values should agree, but they might not agree if the collection of treatment records is not consistent with the in-memory value. For example, if treatment record files for some days are missing, they will not be counted, and the number in the `Fractions Acc` column will be larger than in `Fractions Calc`.

The column labeled `Daily Meas` shows the dose actually delivered during each run, as recorded

in the `run finished` recorded (this is the “official dose for the run”). If the preset dose for the run was different from the prescribed dose that appears at the top of the column, an asterisk appears after the dose (see the second run on April 25).

The two columns labelled `Total` show the total accumulated dose, determined by two different methods. The `Accum` column shows the number of total accumulated dose recorded in the treatment record for that run, which is a copy of the in-memory value that was stored in the control program at the time when the treatment record was written. This value is followed by an asterisk if the dose warning flag is set in the treatment record. The `Calc` column shows total accumulated dose calculated by the script from the entire collection of treatment records up to and including the run on this line. The calculation is performed by adding up the dose for each run for this field. These two values should agree, but they might not agree if the collection of treatment records is not consistent with the in-memory value. For example, if treatment record files for some days are missing, their dose contribution will not be counted, and the number in the `Total Accum` column will be larger than in `Total Calc`.

5.6.2 `accum.err`

The file `accum.err` is a cumulative summary of status messages and error messages about `accum.dat`. It is maintained each night by the `accum-cmd` script. Here is an excerpt:

```
13-NOV-2000 23:00:03 accum.dat 09-NOV-2000 14:40:29: 90 lines 7 pats 41 flds
14-NOV-2000 23:00:03 accum.dat 14-NOV-2000 14:42:44: 105 lines 8 pats 48 flds
14-NOV-2000 23:17:02 accum.dat pat 3149 fld 4: dose warning flag set
15-NOV-2000 23:00:03 accum.dat 15-Nov-2000 15:24:33: 120 lines 7 pats 56 flds
15-NOV-2000 23:20:25 accum.dat pat 3149 fld 4: dose warning flag set in log
16-NOV-2000 23:00:03 accum.dat 16-NOV-2000 13:15:53: 130 lines 7 pats 61 flds
16-NOV-2000 23:22:29 accum.dat pat 3149 fld 4: 128.6 dose claimed, 136.4 found
17-NOV-2000 23:00:04 accum.dat 17-NOV-2000 15:44:02: 140 lines 9 pats 65 flds
17-NOV-2000 23:24:52 accum.dat pat 3149 fld 4: 160.8 dose claimed, 168.6 found
```

Here is an explanation of each kind of message:

```
13-NOV-2000 23:00:03 accum.dat 09-NOV-2000 14:40:29: 90 lines 7 pats 41 flds
```

This line is written each time the script runs. The first timestamp shows when the script ran (usually, once every night shortly before midnight). The second timestamp shows when the `accum.dat` file was most recently updated (that is, when the most recent treatment run finished). The rest of the

line shows the number of lines, patients, and fields in the file. The numbers change from day to day as patients and fields are added, then archived.

```
14-NOV-2000 23:17:02 accum.dat pat 3149 fld 4: dose warning flag set
```

This line indicates that the dose warning flag for this field was set in the `accum.dat` file when the script examined the file at the time indicated (the timestamp indicates when the script ran, not when the flag was first set). This occurs when the control program detects certain errors during a treatment run.

The script will write this message every day until someone clears the flag by editing the `accum.dat` file and rebooting the control program.

```
15-NOV-2000 23:20:25 accum.dat pat 3149 fld 4: dose warning flag set in log
```

This line indicates that the dose warning flag for this field was set in a treatment record when the script examined the record at the time indicated (the timestamp indicates when the script ran, not when the treatment record was written). This occurs when the in-memory warning flag for this field was set when the treatment record was written, as a result of the control program detecting errors during a run.

The message is only written when the warning flag is set in the most recent treatment record for that field so the same treatment record will not be reported repeatedly in `accum.err`, unless it is the last treatment record for that field.

This message can occur even after the warning flag in `accum.dat` has been cleared.

```
16-NOV-2000 23:22:29 accum.dat pat 3149 fld 4: 128.6 dose claimed, 136.4 found
```

This message indicates that there is a discrepancy between the treatment records and the `accum.dat` file. The `dose claimed` is the total dose found in `accum.dat` at the indicated time, while the `dose found` is the total dose calculated by adding up the delivered dose in every run finished treatment record for this beam.

Here are examples of other messages that can appear in `accum.err`.

```
05-MAY-2002 23:00:23 accum.dat pat 3957 fld 1: 546.7 dose found, 455 prescribed
```

The total dose calculated from the treatment records exceeds the product of the prescribed daily dose in `accum.dat` and the number of fractions claimed in `accum.dat`.

```
05-MAY-2002 23:00:23 accum.dat pat 3957 fld 1: 546.7 dose claimed, 455 prescribe
```

The total dose claimed in `accum.dat` exceeds the product of the prescribed daily dose in `accum.dat` and the number of fractions calculated from the treatment records.

```
05-MAY-2002 23:00:23 accum.dat pat 3957 fld 1: 5 fractions claimed, 6 found
```

The number of fractions claimed in `accum.dat` differs from the number calculated from the treatment records.

```
06-MAY-2002 15:04:53 accum.dat pat 3953 fld 6: 2 fractions claimed, 1 prescribed
```

The number of fractions claimed in `accum.dat` differs from the number prescribed in `accum.dat`.

```
06-MAY-2002 15:04:53 accum.dat pat 3953 fld 6: 2 fractions found, 1 prescribed
```

The number of fractions calculated from the treatment records exceeds the number prescribed in `accum.dat`.

In addition to writing each of the preceding messages to `accum.err`, the script also emails them to the staff members whose addresses appear in `mail-warning`.

The script writes these additional messages to `accum.err`, but does not email them. The following messages all mean that the script was unable to interpret the data in the `accum.dat` file; perhaps it is incorrectly formatted.

```
14-APR-2000 23:00:46 accum.dat line 1: couldn't match timestamp
06-FEB-2001 23:50:42 accum.dat line 191: couldn't match field
07-FEB-2001 23:56:40 accum.dat line 198: couldn't match patient
20-MAY-2001 23:01:07 accum.dat line 54: couldn't match date
09-OCT-2001 23:07:30 accum.dat line 101: couldn't match doses
```

The following message means that the script was unable to interpret the data in the `accum.out` file; perhaps it is incorrectly formatted.

```
21-JAN-2002 23:00:30 accum.dat pat 3779 fld 5: couldn't match totals
```

The following message means that the `accum.dat` file contains more lines than claimed in its own first line.

```
18-OCT-2000 23:41:04 accum.dat: 219 lines claimed, 249 found
```

The control program works correctly in this situation; it only reads then number of lines claimed (219 in this example), and ignores the remaining lines. We have observed that sometimes after a patient is archived and the `accum.dat` file is supposed to become shorter, old data remains at the end of the file, as if the file is not being truncated to zero length when it is rewritten. This only happens rarely and we have not been able to reproduce it on demand. The problem clears itself when another patient is archived later.

5.7 Statistical summaries

A few utility programs collect and report statistics. These programs are kept in `/radonc/cnts/qa` in the cluster file system. They run (on a cluster computer) when invoked by typing at the command line. Their output goes wherever the command line directs (there is no standard destination, but some sample output is saved in `/radonc/cnts/qa`). Directions appear in the program headers.

5.7.1 Operating statistics

Table 5.2 shows some statistics for the first year of operation with the new therapy control system, reported by the `stats` program. These statistics were collected from the archived event logs.

5.7.2 Patient accruals and new fields

Table 5.3 shows monthly patient accruals and new fields, reported by the `new_pats` program. These statistics were collected from the archived prescription files.

1-JUL-1999 THROUGH 30-JUN-2000

THERAPY MODE

Treatment days	212		
Treatment sessions	1485		
Treatment fields	3918		
Treatment runs (DMC setups)	4016		
Treatment cycles (beam on/off)	4395		
Treatment sessions per day	7.0	+/-	2.6
Treat. session time, minutes	28.1	+/-	16.7
Treatment fields per session	2.6	+/-	0.9
Treat. field time, minutes	9.3	+/-	13.3
Treat. run time, minutes	2.29	+/-	11.98
Treat. cycle time, minutes	1.74	+/-	2.30
DMC setups	5360		
Leaf collimator setups	5996		
Wedge selection setups	2136		
Wedge rotation setups	2755		
Flattening filter setups	1057		
Fields w/more than one run	83		
Fields with one run, percent	97.9		
Fields w/more than one cycle	238		
Fields with one cycle, percent	93.9		
Proton interlocks	167		
Treat. timeouts (low dose rate)	23		
Beam off at therapy console	35		
Other interlocks	253		

EXPERIMENT MODE

... omitted, fit table on one page ...

TOTALS

... omitted, fit table on one page ...

Table 5.2: Operating statistics reported by stats program

Year-Mo	Patients	Fields
-----	-----	-----
1999-07	8	87
1999-08	12	73
1999-09	6	81
1999-10	10	108
1999-11	2	57
1999-12	9	78
2000-01	8	49
2000-02	12	115
2000-03	10	195
2000-04	5	52
2000-05	5	60
2000-06	7	117
2000-07	1	14
-----	-----	-----
Total	95	1088

Table 5.3: Patient accruals and new fields reported by new_pats program

Chapter 6

Startup and shutdown

The control program is designed to run indefinitely. It is usually left running overnight and on weekends.

These are the only operations where it should be necessary to shut down and restart the control program:

1. Power down or reset the control computer (for service or replacement etc.)
2. Power down or reset the X terminal or room display
3. Restart to recover from a PLC shutdown or communication error
4. Restart to load a new control program version
5. Restart to read a new version of the startup command file (with new configuration information)
6. Restart to read a new version of a data file that is only read at startup, such as `accum.dat`

Other operations can be accomplished without restarting the control program (chapter 7). In particular, it should not be necessary to restart the control program to recover after the TMC, LCC or DMC is powered down or reset.

The program does not reset the DMC when it restarts, so any values displayed on the LEDs and mechanical counters will remain there. The program does not reset the DMC until the operator selects a field.

You should not restart the control program by simply toggling the power or pressing the reset button (this does no harm but in the log files it will appear that the program crashed). You should do an orderly shutdown first (see below) and then start up according to the following procedure.

6.1 Startup

6.1.1 Preliminaries

You do not need to enter any information during a normal startup. All configuration must be done earlier (chapter 3).

Before you can start the control program or the X terminal, the host computer must be running and all network hardware (hub, bridge and transceiver) between the host, the X terminal, and the control computer must be connected and working.

6.1.2 Start the controllers

For normal operation, the PLC must be connected and working before you start the control program. It is possible to start the control program when the PLC is disconnected or not working, but in this condition the PLC watchdog interlock will be set and many operations will be disabled. If you start the PLC later the control program will not be able to synchronize with it, so you must restart the control program again.

It is usual to start the control program when the TMC, LCC, and DMC are already connected and working. However it is not necessary. It is possible to start the control program first, then start these controllers later and then synchronize each with the control program (chapter 7).

6.1.3 Start the X terminal and room display

Before you can start the control program, the X terminal in the therapy console must be running. The room display must also be running, if the control program is configured to use it (in the startup command file).

To start the X terminal, simply turn on the power switch on its controller box (which is separate from the monitor and is located on a shelf inside the therapy control console).

It takes about 30 to 45 seconds for the X terminal to boot. While it is booting it displays a blue background screen and many messages scroll by (Appendix D.7). When it has completed booting it shows the the blank gray *root weave* background (not the blue boot screen or a setup screen). In this condition it is ready for you to boot the control computer.

If the control program crashes, the control program display may remain frozen (not updating) on the X terminal screen (the on-screen clock is stopped). In this condition also you can boot the control computer; it is not necessary to reboot the X terminal.

To start the room display, just turn on the power and wait for Linux to boot.

6.1.4 Start the control program

Go to the VME crate in the power supply room. The VME crate must be powered up.

Boot the control computer by pressing the reset button on the control computer front panel, or the reset button on the VME crate. It is not necessary to toggle the VME crate power.

The control program will start running automatically after the control computer boots. It usually takes about 35 seconds for the control computer to boot and the control program to complete its startup sequence. Startup may take longer if the host computer is busy (running an animated screen saver prolongs the boot process appreciably). If it takes much longer than usual (two minutes or more) there is a problem with the host computer or network.

While the control computer is booting and the control program is starting up, many messages are written to the console port (Appendix I). After the startup sequence completes, the console terminal displays every log message that the program writes (Appendix J).

When startup has completed successfully the blue CNTS login screen appears on the X terminal in the control room (but you cannot see this in the power supply room).

6.1.5 Clear the PLC watchdog interlock

After the control program starts up, the PLC watchdog interlock will be set (section 1.3). You must clear this interlock before the beam can be turned on in the treatment room.

Open the HSIS (Hardware Safety Interlock System) cabinet in the power supply room. Press the button labelled **THERAPY A WATCHDOG NOT OK/RESET** to clear the interlock.

If the control program has successfully completed its startup sequence, the lamp in the button turns off, indicating that the watchdog interlock is now clear. If the lamp does not turn off, perhaps startup has not yet completed. Try pressing the reset button again after a few seconds.

6.1.6 Confirm that startup completed normally

Appendix J shows the log messages written during a normal startup. Other messages may indicate problems. In particular, there should be many messages of the form

```
14-JUL-1999 06:37:10 Dose cal file, success at line 18
```

If instead they say something like

```
14-JUL-1999 06:37:10 Dose cal file, open error at line 0
```

there is a problem. See section 4.3.3 for details and remedies.

6.2 Shutdown

You should not shut down the control program by simply turning off the control computer power. Do an orderly shutdown.

6.2.1 Preliminaries

Only shut down when you are sure you can start up again. Make sure the host is running and is accessible (just press **Select Patient** to confirm you can read the prescription file). The control program can run when the host is not running or not accessible, but it cannot reboot in that condition.

You may wish to use **Write File** to write out the in-memory controller logs, or the message log and treatment records, or the `accum.dat` file. When the program restarts the in-memory contents are lost.

Use the **Log Message** operation to write a log message describing the reason for the shutdown.

6.2.2 Shut down the control program

You can only shut down the control program at the X terminal.

You can only shut down the program in experiment mode when no run is in progress (not all users are authorized to enter experiment mode).

Press `ctrl-shift-Q` (press the control key, the shift key, and the Q key all at the same time).

Press **Confirm** at the `Are you sure?` dialog box.

6.2.3 What happens at shutdown

At shutdown the control program

1. Writes `Operator Operator requested shutdown` at the end of the message log.
2. Closes the message log file and treatment log file
3. Closes the connection to the X terminal
4. Writes `Have a nice day` to the console port

The control program graphics disappear from the X terminal screen, which reverts to the blank gray weave.

The messages indicate the program was stopped by a controlled shutdown instead of a halt or crash.

The PLC watchdog sets.

After a shutdown, the control program can only be restarted by rebooting the control computer.

6.2.4 Halting or crashing the control program

It is possible to halt or crash the program from the X terminal (this capability was provided for testing). To halt the program, type `ctrl-shift-A` (A for *Abort*, instead of Q for *Quit*). To crash the program, type `ctrl-shift-K` or `-L` or `-M`.

You can only halt or crash the program in experiment mode when no run is in progress.

Chapter 7

Interruptions and recovery procedures

It should be possible to resume after any problem by restarting the control program (chapter 6.1), but in many situations this drastic remedy is not necessary and it may cause useful information to be lost. This chapter describes the least disruptive and most informative way to recover from several kinds of interruptions.

7.1 Control program stops

The control program can stop in several ways: it *shuts down* when the operator commands it, it *halts* when the program detects certain conditions, and it may *crash* in response to hardware failures, programming errors, or disturbances in the environment (power, network, etc.).

The program does not restart spontaneously. To recover, you must perform the startup procedure (chapter 6.1), but first try to determine how, when and why the program stopped.

7.1.1 Determining that the program has stopped

When the control program has stopped, the X terminal screen is blank (shows the “gray weave” pattern) or frozen (not updating). The screen is frozen if the clock stops counting for more than 15 seconds (the screen might freeze for up to 15 seconds while the control program tries to access files, but then the file access will time out and the clock will resume counting).

When the control program has stopped, the PLC watchdog interlock sets.

7.1.2 Determining how the program stopped

Messages on the console terminal can provide useful information for troubleshooting. It is important to write them down before restarting the program because they scroll away when the program restarts. If a terminal is not connected at the console port, this information is lost.

While the program is running, it writes every log message to the console terminal. When the control program shuts down, the last log message it writes is `Operator requested shutdown`. When the program halts or crashes, it does not write this message (or any others).

When the program halts or crashes, the last few log messages that appear on the terminal might not be written to the log file, so it is important to write them down.

After the last log message, the program writes additional progress messages that only appear on the console terminal (not in the message log file). The very last of these is `Have a nice day`.

When the control program halts, it writes a message like this:

```
X server not responding
Assertion failed: 0, file user.c, line 424

Task 368ff4 tUser handling SIGABRT, number 6
```

The control program source code at the indicated file and line number may reveal more about the reason for the halt.

When the control program crashes, it may write a message something like this:

```
Task 358690 tTMC handling SIGSEGV, number 11
```

7.1.3 Determining when the program stopped

The time-stamped log messages at the end of the message log file and treatment records at the end of the treatment record file indicate the last operations the program performed. There may be a few more time-stamped log messages on the console terminal that do not appear in the file. The `run.dat` file indicates whether the last therapy run was in progress when the program stopped (section 4.3.19).

7.1.4 Recovering after the program stops during a therapy run

If the program halts or crashes during a therapy run the the beam will remain on for up to a second until the PLC watchdog interlock sets. The DMC continues to work normally after the control program stops, monitoring and displaying the accumulating dose and treatment time.

When the dose on the LED displays stops increasing after the beam turns off, write down the dose for the run. Edit the `accum.dat` file to enter the dose for this run. (section 4.3.6). Edit the `run.dat` file to indicate that the run is finished (section 4.3.19).

Restart the control program. The program does not reset the DMC when it restarts, so any values displayed on the LEDs and mechanical counters will remain there. The program does not reset the DMC until the operator selects a field.

If you do not edit `run.dat` to indicate that the run is completed, after the control program restarts it will display a warning each time a therapist selects that field. To clear that warning, you must edit `accum.dat` and restart the program again (section 4.3.6). It is usually best to do this after the end of treatment on the same day.

7.2 TMC, LCC, or DMC problems

It should not be necessary to restart the control program to recover from problems with the TMC, LCC, or DMC.

If a controller is disconnected, powered down, reset, or transmits an error message (or any other unexpected message) the control program sets the therapy sum interlock, displays a controller error interlock indicator, and writes a log message.

To recover from this condition, select a field at the X terminal, or press the **Refresh** key. This causes the control program to send a reset command to all three controllers and re-establish communication with each. This should clear any controller error interlocks.

If selecting a field at the X terminal does not clear the controller error, confirm that the controller is powered up and connected, and perform any recommended recovery procedures (check cable connections, push the hardware reset button, etc.). Then select a field or press **Refresh** again.

7.2.1 Recovering after a DMC problem during a treatment run

If a DMC problem occurs during a therapy run, the beam may remain on for up to three seconds until the therapy sum interlock becomes effective. During this interval additional dose accumulates, which may not be recorded correctly by DMC or the control program. You must determine the dose for the run (the dose monitoring LED's, the mechanical counters, or some other method).

After the run, the control program will display asterisks in place of the accumulated dose for that field and will display a warning each time a therapist selects it. To clear the warnings you must edit `accum.dat` to enter the correct dose for the run and clear the warning flag, and then restart the program (section 4.3.6).

7.3 PLC problems

If a PLC communication error occurs the control program keeps running and indicates the error on the PLC screen but most operations cannot be performed.

The only way to recover is to restart the control program while the PLC is running and connected.

7.4 X terminal problems

Turning off the X terminal monitor does not affect the control program. It is usually turned off overnight and on weekends.

Turning off the X terminal controller does not cause the program to halt. The user interface is inaccessible to the operator but the program continues working just as it would if the operator were not present.

Turning on or rebooting the X terminal controller while the program is running causes the program to halt.

If the X terminal shuts down or restarts while the program is running the only way to recover is to restart the control program after the X terminal is on, connected and running.

7.5 Host computer, file system, and network problems

The control program continues running when the host computer or its file system are inaccessible (this might occur because the host is down, or because there is a network problem).

When the control program attempts to open, read, or write a file it waits up to 15 seconds for the host to respond (the interval is set in `timeouts.dat`, see section 4.3.20). During this interval the program does not respond to keystrokes and the on-screen clock stops updating. It looks as if the program has crashed, except the PLC watchdog interlock does not set.

If the file server does not respond, the program reads from (or writes to) an in-memory buffer instead. It displays a warning message to the operator and writes a log message like

```
14-JUL-1999 06:37:10 Dose cal file, timed out at line 0
```

After this the program does not attempt to write log messages, treatment records, or any other files. It remains in this condition until the user causes another attempt to read a file (usually by invoking the **Select Patient** operation). The program attempts to read the file. If this attempt also fails, the program continues in the same condition. If this attempt succeeds, the program displays another message and writes a log message

```
14-JUL-1999 12:40:31 File server responding, reading and writing
files again
```

It resumes writing log messages and treatment records. There is a gap in the message log file and the treatment record file but the in-memory contents are complete. Press **Write File** to write the in-memory contents out to files (sections 4.4.4 and 4.4.6)

7.6 Host shuts down

When the host is shut down, the X terminal and control computer keep running (see above) but they cannot be restarted.

When the host is restarted, all its services become available again automatically. No special action on the host is needed to re-establish contact with the control program.

Appendix A

Hardware reference data

Device	Manufacturer and model number
Control computer	Motorola MVME167-004B with a 25MHz 68040 processor and 32MB memory
Transition module	Motorola MVME712M
Crate	Mupac 5099KDL20FK-100 (sold by Carlo Gavazzi Electronic Packaging)
Hub	Cabletron MicroMACC-24E
Transceiver	Cabletron 802.3 10Base-T Transceiver
Patch panel	Ortronics, similar to AMP 557870-1
X terminal	Tektronix XP217C, 8MB, 17" color monitor, DEC LK401-style keyboard
Room display	Compaq Evo with TFT7020 17-inch flat panel display
Host computer	Hewlett-Packard B160 workstation running HP-UX 10.20

Table A.1: Devices, manufacturers and model numbers

Appendix B

Host computer

The host computer provides `remsh`, `tftp`, `time` service, a font server and NFS. The host is connected to the Internet so certain security measures are advisable. Here are instructions for setting these up under HP-UX 10.20. The procedures for other Unix-like operating systems would be similar.

More information about HP-UX can be found in the Hewlett Packard manual set kept in NN184. Information about the configuration in the Radiation Oncology Department is kept in the gray loose-leaf binder labelled `UW Radonc / HP Software Config + Support` kept in NN184.

B.1 CNTS user

To obtain access to the host, the control computer must be authorized. This is accomplished by creating an account for the control computer on the host:

<code>cnts</code>	user name
<code>2200</code>	user ID
<code>20</code>	group ID
<code>/radonc/cnts</code>	login directory

The `cnts` user only exists to provide authorizations for the control computers. Logins for the `cnts` user must be enabled in `/etc/passwd`, so there should be a (non-empty) password for

this user, but it should not normally be necessary for any human user to log in as `cnts`. The `cnts` user need not own any files, it only needs permission to read files and to execute directories under `/radonc/cnts` and `/cnts/tftpdir` in order to boot and run the operating system and control program.

The `/etc/passwd` file assigns the `cnts` user the user ID 2200 and the group ID 20 (the `users` group, as defined in the `/etc/group` file). The control program startup file assigns the running control program the user ID 2200 also, but assigns the group ID 70 (the `prism` group). It is possible for the same user to belong to several groups. Membership in the `prism` group enables the running control program to read and write data files.

B.2 Hosts file

There must be an entry in `/etc/hosts` for the control computer, as follows:

```
192.168.0.7 elrond elrond
```

When this entry is not present, we sometimes experience long delays when the control computer boots (about ten minutes). Perhaps the host attempts to look up the control computer in a name server.

It is apparently not necessary to have an entry in this file for the X terminal (at this writing we do not have one and we have never experienced boot delays). Perhaps there is a difference because the control computer boots using `remsh` but the X terminal uses `tftp`.

It is not necessary to have an `/etc/hosts.equiv` file (despite the suggestion in the HP `remsh` manual pages). It is better for security not to have one.

B.3 Remote shell facility (`remsh`)

At this writing the control computer boots using `remsh`.

The HP-UX remote shell facility is called `remsh`. Use `remsh` where the Wind River documentation refers to `rsh`.

To obtain authorization for using `remsh`, the control computer provides the `cnts` username. Logins must be enabled for the `cnts` user in `/etc/passwd`. Its login directory `/radonc/cnts`

must contain an `.rhosts` file. The `cnts` user must have read permission for this file. The `.rhosts` file must contain an entry for the control computer:

```
elrond cnts
```

This entry must contain the name assigned in `/etc/hosts`. The boot fails if the IP address appears instead.

The `cnts` user's login directory need not contain any `.cshrc` or `.login` file. If any messages are written to the standard output when `remsh` connects, the boot will fail. So if there is a `.cshrc` file, it cannot execute any `echo` commands, etc.

B.4 Trivial file transfer protocol (tftp)

The X terminal boots using `tftp`.

To enable `tftp`, there must be an account for user `tftp`. At this writing the home directory is `/cnts/tftpdir`. For security reasons, logins for the `tftp` user should be disabled in `/etc/passwd`.

World must have execute permission for this directory and the directories above it (`/` and `/cnts`). Files made available by `tftp` are placed in this directory, or in directories beneath it. World must have read permission for these files.

The Tektronix X terminal boot files are stored under `/cnts/tftpdir/tekxp/boot` (appendix D.1, and appendix D.7 for a complete list of files).

Using `tftp` does not require any authorization for the X terminal.

At this writing the control computer boots using `remsh`, not `tftp`. However the control computer program files are stored under `/cnts/tftpdir/code/iso` so it, too, could boot using `tftp` in the future.

B.5 Time server

The control computer sets its clock when it boots, using the `time` server (on port 37, not the `daytime` server on port 13).

The `time` server is provided with most TCP/IP implementations ([15], p. 16). It is a standard component of HP-UX 10.20. No special configuration effort is needed.

B.6 Font server

The X terminal loads its fonts from the X font server.

The X font server `xfstpd` can be started at any time by the command

```
/usr/bin/X11/xfstpd -config /etc/X11/fs/config -port 7000 -daemon
```

To indicate that the font server should start when the system boots, there is a file `/etc/rc.config.d/xfstpd` with owner `root`, group `sys`, protection `-r-xr-xr-x` and these contents:

```
#
# xfstpd configuration: set RUN_X_FONT_SERVER to a non-zero value to
# initiate a font server at run level 2.
#
# Installed at /etc/rc.config.d/xfstpd
#

RUN_X_FONT_SERVER=1
CONFIG_FILE_NAME=/etc/X11/fs/config
```

To actually start the server at boot time, there is a file `/sbin/init.d/xfstpd` (too long to include here) with owner `bin`, group `bin` and protection `-r-xr-xr-x`.

The font server must be identified in one of the configuration files that the X terminal loads at boot time. The file `/mnt/tftpdir/tekxp/boot/config/fonts.tbl` must contain the line

```
tcp/192.168.0.10:7000
```

The X terminal must be configured to use the font server (section D.3.4)

B.7 Network File System (NFS)

The control program accesses its data files using NFS.

To obtain authorization for using NFS, the control computer provides the `cnts` user ID 2200 and the `prism` group ID 70. Permissions must allow users in the `prism` to read all data files and rewrite certain data files in `/cntsdata/data/iso`. Permissions must allow users in the `prism` group to create new files in `/cntsdata/log/iso`. See section 4 for more details about permissions.

The host must export the filesystem `/cntsdata` for read and write access by the control computer. If the host is not a member of the department cluster, it must also export `/cntsdata` to the treatment planning computers in the cluster so the treatment planning program can update the prescription file.

The host should not export `/cntsdata` to any other computers.

In HP-UX, you can use the SAM (System Administration) utility to export the filesystem (use the `Network -> NFS` panel). You must identify the control computer as `elrond`, the name assigned in `/etc/hosts`. If you use the IP address instead, the control computer can't read files in `/cntsdata` that don't have world read permission (it can read other files, curiously enough).

B.8 Network security

The University does not operate any "firewall" to keep connection requests from the Internet out of the hospital network. Instead, security is delegated to each computer. We configure the host so it only accepts connections to a few network services from a few computers in the Radiation Oncology Department (and from the private network). We accomplish this by adding entries to the file `/var/adm/inetd.sec`. This achieves the effect of a firewall customized for our host.

For example the following lines allow all the computers on the private network and three computers in the Department (named `host1` etc.) to make connections for `tftp`, `remsh`, and `time` service.

```
tftp    allow 192.168.* host1 host2 host3
shell   allow 192.168.* host1 host2 host3
time    allow 192.168.* host1 host2 host3
```

The authorizations in the `inetd.sec` only allow connections. After that each service's own security mechanisms are still in effect. For example, after `inetd.sec` allows a connection to the

shell (remsh) service, it is still necessary for the remote user to be authorized as described in section B.3 to execute a shell command.

The `inetd.sec` security mechanism only guards against connections to services which are started from `/etc/inetd.conf`. Other services must be protected by other means. In particular, the host should not allow remote connections to `sendmail` (which could allow the host to be used as a mail relay by spammers). To prevent this, set `SENDMAIL_SERVER=0` in `/etc/rc.config.d/mailservs`. Even with this precaution in effect, it is still possible to send network mail out from the host.

Appendix C

Hub

The hub is a Cabletron MicroMACC-24E [2]. It provides connections for 24 devices. It was selected and configured by Medical Center Information Systems (MCIS) staff¹. (In fact this hub is probably far more elaborate than we need, and almost any hub would do.)

The configuration is permanent. It should never be necessary to change it. New devices can be plugged into unoccupied sockets at any time; the hub will detect them. It is not necessary to enter new configuration information into the hub when devices are attached or removed.

To configure the hub, a terminal is connected by a serial line (not an Ethernet connection) to the COM1 port on the front of the hub and configurable information such as the IP address is entered. This information is stored in nonvolatile memory in the hub; even when the hub is powered down, this information should remain onboard indefinitely.

It is possible to monitor hub activity remotely over the network, using SNMP (simple network management protocol). This can only be done from one particular remote computer, the *network management platform*, whose IP address is entered into the hub during the configuration procedure. The remote computer must also provide a password, another item that is entered during the configuration procedure. Software on the remote computer is able to communicate with the hub, display the data etc. This remote computer can only read status information from the hub, it cannot change the configuration or reboot the hub. Monitoring the hub from the remote computer does not load the hub or delay transmission through the hub. The remote management computer is located at the MCIS Northgate office.

This is some of the configuration information that was entered when the hub was installed:

¹Marc Hudson, 206-685-3772, hudsonm@mcis.washington.edu, installed and configured the hub on August 30, 1996.

128.95.181.3	hub IP address
255.255.255.0	subnet mask
128.95.150.24	remote management platform IP address
(known to MCIS staff)	hub password, enables remote monitoring

The built-in Ethernet addresses are 00-00-1D-1D-04-63 (hub)² and 00-00-1D-1A-FD-7B (bridge).

C.1 Bridge

The hub contains a BRIM-E6 bridge/router interface module and an EPIM-T 10Base-T Ethernet port interface module [1].

The 24 ordinary connections in the hub constitute an Ethernet segment. It is possible to connect another Ethernet segment to the hub through the bridge. The bridge filters packets based on Ethernet addresses (not IP addresses). The only packets that enter the hub through the bridge are those addressed to devices connected to the hub (or *broadcast* packets intended to reach every device on the subnet). The only packets that leave the hub through the bridge are those addressed to devices elsewhere (including broadcast packets). Network activity among other devices beyond the bridge does not affect devices connected to the hub. Devices on the hub can communicate with each other even if the bridge is disconnected.

The bridge was included because we initially did not have a private network. The control computer, X terminal, and host were all connected to the the hospital network. The bridge limited traffic and provided some isolation from network problems elsewhere. In the present configuration the private network provides a better solution to these problems so the bridge is not necessary.

²The spare hub Ethernet address is 00-00-1D-1D-04-63

Appendix D

X terminal

The X terminal is am Tektronix model XP217C with 8MB of memory, a 17" color monitor, and a DEC LK401-style keyboard. Its control program is Tektronix XpressWare V8.0.

The X terminal is assigned IP address 192.168.0.6 and boots from the host at IP address 192.168.0.10.

The X terminal is configured to display 1024 by 768 pixels and run without a window manager.

More information about this terminal appears on the department cluster in the directories under /home/tftpdir/tekxp. To read the on-line manual, run the viewer program by executing the script /home/tftpdir/tekxp/doc/bin/frame4/bin/viewer and open the document /home/tftpdir/tekxp/doc/online/top.doc. Manual pages in PostScript extracted from the viewer are under /doc/ps. Printed manual pages are collected in an orange looseleaf binder with a hand-lettered label Tek Xterm / Xpressware V.8.0, usually kept in NN184.

D.1 Host configuration

The terminal uses tftp to boot from the host. Instructions for configuring the host to support tftp appear in appendix B.

The X terminal control program executable file must be on the host whose IP address is given in the ihost boot parameter, under the home directory of the tftp (/cnts/tftpdir at this writing),

at the relative location given in the `bpath` boot parameter, `/tekxp/boot/os.350`. Therefore the absolute path to the file is `/cnts/tftpdire/tekxp/boot/os.350`. World must have read permission for this file and execute permission for all the directories above it.

D.2 Boot parameters

Some of the X terminal configuration items are boot parameters.

To view the X terminal boot parameters, turn on the power to begin the boot process, then touch the space bar to stop the boot in progress (indicated by moving bar graph). The screen should display these boot parameters, followed by the `BOOT>` prompt.

```
TekXpress Boot Monitor V5.5
8192 Kbytes memory installed

Hardware 08:00:11:04:16:94 Ethernet address

IADDR= 192.168.0.6      xterm IP address
IHOST= 192.168.0.10    host IP address
IMASK= 255.255.255.0  subnet mask
IGATE= 128.95.181.100 gateway IP address (irrelevant)
BPATH= /tekxp/boot/os.350 path to terminal control program on host

DNODE = 0.0
BMETHOD= TFTP          boot method, tftp
BDELAY= DISABLED
BAFROM= NVRAM
```

To set boot parameters, enter information at the boot prompt (type a carriage return after each line):

```
iaddr 192.168.0.6      xterm IP address
bpath /tekxp/boot/os.350 path to terminal control program on host
imask 255.255.255.0  subnet mask
ihost 192.168.0.10    host IP address
igate 128.95.181.100 gateway IP address (irrelevant)
bmethod tftp          select tftp boot method
```

<code>se monset</code>	select monitor settings
<code>...</code>	(shows a menu of monitor settings)
<code>se monset 11</code>	select setting 11 (1024x768 75Hz 17")
<code>nvsave</code>	save to non-volatile memory, wait for prompt to continue
<code>boot</code>	continue boot sequence

Other boot parameters are described in the manual pages.

D.3 Terminal setup

Some of the X terminal configuration items are entered into forms and menus reached by the `Setup` key. You must use the mouse for these operations. The control program itself never uses the mouse, but the mouse must be connected or the terminal will not work.

D.3.1 Setup key (F3)

The Tektronix terminal F3 function key is the terminal Setup key. Terminal setup functions are only available after the terminal boots but before the control program begins running. When the control program is running pressing the F3 key invokes a control program function instead (it causes the leaf collimator calibration display to appear, see [4]).

When the terminal is used with a window manager running on a host on the department HP cluster, the F3 key may not bring up the setup menu. The F3 key apparently has no effect when the HP login screen is displayed, and when the user logs in, a script may reconfigure the terminal, including the F3 key. To recover the Setup function, it is necessary to bypass the usual login method. From the HP login screen select `Options`, then `Session`, then `Failsafe Session`. Log in. The F3 key should work.

D.3.2 No window manager

For routine operation at the therapy console, the X terminal should not run with a window manager. This is configured as follows:

1. Press the F3 key. A menu pops up.
2. Select Setup.
3. From the Configuration Summaries pull down menu, select X Environment.
4. From the pull-down menu provided by selecting the first item to the right of Host Connect Method, select Disabled.
5. Select Return to Main Menu.
6. Select Save Settings to NVRAM. This takes several seconds, do not interrupt.
7. Select Reboot Terminal (or maybe just Reset Server?)

To go back to having a host provide the window manager, follow above to Host Connect Method. Instead of Disabled, select XDMCP Direct and enter the IP Address of the host (192.168.0.10 for trombone). Then continue as above.

D.3.3 Security

The X terminal must only accept connections from the control computer (except when it boots and loads fonts from the host).

Select Setup/F3, Configuration Summaries, X Environment. On that page there is an Enable Access Control button. Select Yes. The terminal displays the message Allow only hosts listed in the Display Access Table to connect to the display.

Then, from Setup/F3, select Network Tables and Utilities, Display Access Table. Select Add Entry and enter the control computer IP address 192.168.0.7. This address should then appear in the Host address table. The only other entry in the table should be the X terminal itself, 192.168.0.6.

D.3.4 Font server

The X terminal is configured to load fonts from the X server on the host.

Press Setup/F3, select Network Tables and Utilities, Font Directory. Enter

```
tcp/192.168.0.10:7000
```

This is the host IP address (the host name `trombone` will not work) and the port number for the X font server. This entry should appear in the `Font directories` table. The only other entries should be `/tekxp/boot` and `resident/`.

This same line must appear in the file `/cnts/tftpdire/tekxp/boot/config/fonts.tbl` on the host and the font server must be running on the host (section B.6).

D.4 Boot process

To boot the X terminal, just turn the power on, or type `boot` or `b` at the `BOOT>` prompt, or select `Reboot Terminal` from the setup main menu page. Booting takes 30 to 45 seconds.

The progress messages that are written when the X terminal boots appear in section D.7.

D.4.1 Console

The X terminal control program writes progress messages when it boots and when it loads fonts. These messages appear in a console window which is not usually visible. To display the console window, Press `Setup/F3`, then select `Console`. The console window remains visible until you select `Console` again.

D.5 Tektronix X terminal boot parameters (summary)

```
TekXpress Boot Monitor V5.5  
8192 Kbytes memory installed
```

```
Hardware 08:00:11:04:16:9a Ethernet address
```

```
IADDR= 192.168.0.6      xterm IP address  
IHOST= 192.168.0.10    host IP address  
IMASK= 255.255.255.0   subnet mask  
IGATE= 128.95.181.100  gateway IP address (irrelevant)  
BPATH= /tekxp/boot/os.35 path to terminal control program on host
```

```
DNODE = 0.0  
BMETHOD= TFTP          boot method, tftp  
BDELAY= DISABLED  
BAFROM= NVRAM
```

To set resolution to 1024 by 768 pixels, at the boot prompt type `se monset` and select menu entry number 11.

D.6 Tektronix X terminal setup (summary)

Press F3 to bring up the Setup Main Menu. Then select

No window manager:

```
Configuration Summaries
  X Environment
    Host Connect Method
      Disabled
```

Security, restrict access:

```
Configuration Summaries
  X Environment
    Enable Access Control
      Yes
```

Security, allow access only to control computer:

```
Network Tables and Utilities
  Display Access Table
    Add Entry
      192.168.0.7
```

Font server:

```
Network Tables and Utilities
  Font Directory
    Add Entry
      tcp/192.168.0.10:7000
```

This same line must appear on the host in the file `/cnts/tftpdire/teexp/boot/config/fonts.tbl`

D.7 Tektronix X terminal boot and startup messages

Attempting to boot with TFTP ...

Requesting file 'tekxp/boot/os.350' from IP host 192.168.0.10

Starting the X terminal code

Opening for TFTP read: /tekxp/boot/xge8.350 on host 192.168.0.10 Found

Opening for TFTP read: /tekxp/boot/rconf.350 on host 192.168.0.10 Found

Opening for TFTP read: /tekxp/boot/xp.cnf on host 192.168.0.10 Found

Opening for TFTP read: /tekxp/boot/gateway.tbl on host 192.168.0.10 Found

Opening for TFTP read: /tekxp/boot/hosts.tbl on host 192.168.0.10 Found

Opening for TFTP read: /tekxp/boot/nodes.tbl on host 192.168.0.10 Found

Opening for TFTP read: /tekxp/boot/nfs.tbl on host 192.168.0.10 Found

remote configuration complete

xpshd: protocol started

Opening for TFTP read: /tekxp/boot/launcher.300 on host 192.168.0.10 Found

Opening for TFTP read: /tekxp/boot/config/fonts.tbl on host 192.168.0.10
Found

Opening for TFTP read: /tekxp/boot/config/rgb.txt on host 192.168.0.10 Found

Opening for TFTP read: /tekxp/boot/setup.350 on host 192.168.0.10 Found

Opening for TFTP read: /tekxp/boot/config/xhosts.tbl on host 192.168.0.10
Found

Opening for read: default.keymap Found

Opening for TFTP read: /tekxp/boot/config/remap.dir on host 192.168.0.10
Found

Searching for font directory: /tekxp/boot

Opening for TFTP read: /tekxp/boot/fonts.dir on host 192.168.0.10 Found

Opening for TFTP read: /tekxp/boot/fonts.alias on host 192.168.0.10 Found

Searching for font directory: resident/

Opening for read: fonts.dir found

Opening for read: fonts.alias found

Opening font: "Tekfixed" from FS 192.168.0.10:7000 Not Found

Opening for read: Tekfixed.snf.Z Found

Opening font: "TekCursor" from FS 192.168.0.10:7000 Not Found

Opening for read: TekCursor.snf.Z Found

Using MOUSE as core input device.

Opening font: "TekDefault" from FS 192.168.0.10:7000 Not Found

Opening for read: TekDefault.snf.Z Found

Opening for TFTP read: /tekxp/boot/config/system.launcher on host 192.168.0.10 F

Launcher: using auto configuration data.

When the terminal has successfully booted but the therapy control program is not yet running, the screen should show an empty gray background (actually a fine mesh pattern called the *root weave*) and a black X-shaped cursor. This indicates that no window manager is running and the terminal is ready for the control program to begin running.

When the therapy control program begins running (when the control computer boots) the X terminal loads the fonts used by the control program. These messages are written at the console.

```
Opening font: "*-helvetica-bold-r-*-240-75-*" from FS 192.168.0.10:7000 Found
Opening font: "*-helvetica-bold-r-*-180-75-*" from FS 192.168.0.10:7000 Found
Opening font: "*-helvetica-medium-r-*-180-75-*" from FS 192.168.0.10:7000
Found
Opening font: "*-helvetica-bold-r-*-140-75-*" from FS 192.168.0.10:7000 Found
Opening font: "*-lucidatypewriter-medium-*-180-75-*" from FS 192.168.0.10:70
00 Found
Opening font: "*-courier-bold-r-*-18-180-75*" from FS 192.168.0.10:7000 Found
```

Appendix E

Room display

The room display is an ordinary PC with a 17-inch flat panel color display (Compaq Evo with TFT7020). It runs the Linux operating system configured to behave like an X terminal, running an X server accessible to the control computer but no window manager. Its resolution is set to 1024 by 768. It does not normally use the PC keyboard or mouse (they are only used to configure the PC).

Unlike the X terminal, the room display PC is self-contained. It has its own disk and file system, so it does not need to obtain its control program or fonts from the host computer.

The PC was configured by Mark Wagner of the UW Radiation Oncology Department in June 2002. Here are his notes on the configuration (some of the IP addresses have since changed):

1. Install Red Hat 7.3 + errata
(<http://rhn.redhat.com/errata/rh73-errata.html>).
2. Install Ximian GNOME (add Xsession gdm session type).
3. Disable unneeded services:

```
for F in netfs nfslock portmap
do
/etc/rc.d/init.d/$F stop
chkconfig $F off
chkconfig --list $F
done
```

3. Add users:

```
for F in cnts markwag jon dreid risler
do
useradd -m $F
passwd $F
done
```

4. Enable telnet:

```
Edit /etc/xinetd.d/telnet
```

```
disable = no
only_from = 128.95.181.34/32
```

```
/etc/rc.d/init.d/xinetd restart
```

5. Restrict ssh and other serices that use tcp wrappers:

```
Edit /etc/hosts.allow
```

```
sshd: 128.95.181.
```

```
Edit /etc/hosts.deny
```

```
all: all
```

6. Set automatic login:

```
gdmconfig
```

```
# go to "Automatic login" tab
# set up startup and timed login
```

7. Create blank cursor:

```
Edit /usr/local/lib/xc_none:
```

```
#define xc_none_width 16
#define xc_none_height 16
static unsigned char xc_none_bits[] = {
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
```

```
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00};
```

8. Set cnts' .xsession

```
Edit /home/cnts/.xsession
```

```
-- cut here --
```

```
# Turn off screen blanking
```

```
/usr/X11R6/bin/xset -dpms
/usr/X11R6/bin/xset s off
```

```
# Make background/pointer black
```

```
/usr/bin/X11/xsetroot -solid black
/usr/bin/X11/xsetroot -cursor /usr/local/lib/xc_none /usr/local/lib/xc_none
```

```
# Allow remote systems access to display
```

```
/usr/X11R6/bin/xhost +128.95.181.34
```

```
while [ ! -f $HOME/logout ]; do
/bin/sleep 10
done
```

```
rm $HOME/logout
```

```
exit 0
```

```
-- cut here --
```

```
chmod 755 /home/cnts/.xsession
```

Appendix F

Control computer boot parameters

See section 3.5, pages 65 – 70 in [17].

```
boot device           :ei
processor number      :0
host name             :trombone
file name             :/cnts/tftpdire/wind/mv167/vxWorks
inet on ethernet (e) :192.168.0.7:ffff0000
host inet (h)         :192.168.0.10
user (u)              :cnts
flags (f)             :0x0
target name (tn)     :elrond
startup script (s)   :/cnts/tftpdire/code/iso/isostart
```


Appendix G

Console terminal setup

When the control computer is reset, use to set boot parameters:

baud rate 9600
8 bits per character
no parity
one stop bit
no new line

When the control program is running, it changes the console terminal line to these settings to communicate with the PLC. You may wish to change the terminal itself to the same settings to view PLC message traffic and any messages written at shutdown or halt.

baud rate 9600
7 bits per character
even parity
one stop bit
new line

Appendix H

Control computer startup command file

```
# isostart.192 - displays on private control network on 192.*.*.* subnet
# isostart.65 - displays on .65 xterm

# isostart - Isocentric room control program startup
#     Delete portmapper so TCP, UDP ports 111 aren't open
#     Spawn user, then suspend shell
#
# This file is read by the control computer at boot time
# Enter path to this file as "file name" in boot parameters (Tornado manual)
# Runs the application standalone, without Tornado: just load .o files and run
#
# Edit this startup file to change:
# - User and group id's for NFS authorization
# - Host name and mount point for NFS files
# - Directory for X windows .o files
# - Directory for application .o file(s)
# - Name(s) of application .o file(s), name of application top-level function
# - IP address of X console used by application (xterm or workstation)
#
# 22-May-1998  J. Jacky  Standalone startup
# 14-Aug-1998  J. Jacky  Ready for acceptance testing
# 21-Jun-1999  J. Jacky  Change group from cnts (130) to prism (70) in nfsAuth.
# ??-Nov-1999  J. Jacky  Mount files from trombone not oboe
# 4-Oct-2000   J. Jacky  Fix daylight savings time change date from Oct 1 to 29
#                (daylight savings returns on April 1 2001)
# 5-Jan-2001   D. Reid   Changed IP address for X-Term
```

```
# 3-Apr-2002 J. Jacky Fix daylight savings again to April 7 and Oct 27 2002
# 29-Apr-2002 J. Jacky Add PLC IP address arguments to user call
# 12-Aug-2002 d. reid, j. jacky Add IP address for the room display
```

```
taskDelete tPortmapd
```

```
# If you are using a local file system or NFS make
# sure the following line is not commented.
```

```
putenv "USE_LOCAL_FS"
```

```
putenv "TIMEZONE=PST8PDT::480:040702:102702"
```

```
# Mount nfs file system
```

```
aup_gids = 0 /* just one element to this "array", not used because ngids=0 */
nfsAuthUnixSet("trombone", 2200, 70, 0, &aup_gids) /* cnts user, prism group*/
nfsMount("trombone", "/cntsdata", "/cntsdata")
```

```
# Write nfs mount info to stdout
```

```
hostShow
```

```
nfsAuthUnixShow
```

```
nfsDevShow
```

```
# Load X window support
```

```
ld 1,0,"/cnts/tftpsdir/vxwindows/support.o"
```

```
# initialize the time used by the time support utilities
```

```
# (it seems this synchronizes the target time with host time)
```

```
starttime()
```

```
# Load Xlib
```

```
ld 0,0,"/cnts/tftpsdir/vxwindows/X11.o"
```

```
# Setup the X library initialization routines.
```

```
vxClientLibInit(1)
```

```
# Load application
```

```
ld 1,0,"/cnts/tftpsdir/code/iso/iso.o"
```

```
# Spawn application on X console at IP address.
```

```
# .10 is bilbo HP workstation, .65/.66 is Tek xterm, .182 is tuba
```

```
# Task will now run on the private control network -- 192.168.0.6
```

```
# Next two IP addresses are Modicon 1 and 2
```

```
taskSpawn "tUser",99,8,20000,user,4,"192.168.0.6:0","192.168.0.9","192.168.0.5",
```

```
# Tie up the shell so doesn't it try to read PLC traffic
```

```
# Exit *doesn't* work here -
```

```
taskSuspend
```

Appendix I

Control computer boot and startup messages

These are the messages that appear on the terminal attached to the console port when the control computer boots.

Numbers that represent machine addresses and program segment sizes may differ from those shown in this example.

```
VxWorks System Boot
```

```
Copyright 1984-1995 Wind River Systems, Inc.
```

```
CPU: Motorola MVME167  
Version: 5.3  
BSP Version: 1.1/1  
Creation Date: Nov 16 1995
```

```
Press any key to stop auto boot...
```

```
7
```

```
auto-booting ...
```

```
boot device           :ei  
processor number      :0  
host name             :trombone
```

```
file name :/cnts/tftplib/wind/mv167/vxWorks
inet on ethernet (e)      :192.168.0.7:ffff0000
host inet (h)             :192.168.0.10
user (u)                  :cnts
flags (f)                 :0x0
target name (tn)         :elrond
startup script (s)       :/cnts/tftplib/code/iso/isostart
```

```
Attaching network interface ei0... done.
Attaching network interface lo0... done.
Loading ... 400432 + 51296 + 24000
Starting at 0x20000 ...
```

```
Attaching network interface ei0... done.
Attaching network interface lo0... done.
Loading symbol table from trombone:/cnts/tftplib/wind/mv167/vxWork
s.sym ...
```

VxWorks Development System

VxWorks Version 5.3.1

KERNEL: WIND Version 2.5
Copyright Wind River Systems, Inc. 1984 - 1997

CPU Motorola MVME167. Processor #0.
Memory size 0x2000000. BSP Version 1.1/3.

```
# isostart.192 - displays on private control network on 192.*.*.* subnet
# isostart.65 - displays on .65 xterm

# isostart - Isocentric room control program startup
#       Delete portmapper so TCP, UDP ports 111 aren't open
#       Spawn user, then suspend shell
#
# This file is read by the control computer at boot time
# Enter path to this file as "file name" in boot parameters (Tornado manual)
# Runs the application standalone, without Tornado: just load .o files and run
#
# Edit this startup file to change:
# - User and group id's for NFS authorization
# - Host name and mount point for NFS files
```

```

# - Directory for X windows .o files
# - Directory for application .o file(s)
# - Name(s) of application .o file(s), name of application top-level function
# - IP address of X console used by application (xterm or workstation)
#
# 22-May-1998 J. Jacky Standalone startup
# 14-Aug-1998 J. Jacky Ready for acceptance testing
# 21-Jun-1999 J. Jacky Change group from cnts (130) to prism (70) in nfsAuth.
# ??-Nov-1999 J. Jacky Mount files from trombone not oboe
# 4-Oct-2000 J. Jacky Fix daylight savings time change date from Oct 1 to 29
#                (daylight savings returns on April 1 2001)
# 5-Jan-2001 D. Reid Changed IP address for X-Term
# 3-Apr-2002 J. Jacky Fix daylight savings again to April 7 and Oct 27 2002
# 29-Apr-2002 J. Jacky Add PLC IP address arguments to user call
# 12-Aug-2002 d. reid, j. jacky Add IP address for the room display

```

```

taskDelete tPortmapd
value = 0 = 0x0

```

```

# If you are using a local file system or NFS make
# sure the following line is not commented.
putenv "USE_LOCAL_FS"
value = 0 = 0x0

```

```

putenv "TIMEZONE=PST8PDT::480:040102:102902"
value = 0 = 0x0

```

```

# Mount nfs file system
aup_gids = 0 /* just one element to this "array", not used because ngids=0 */
new symbol "aup_gids" added to symbol table.
aup_gids = 0x1ffe6fc: value = 0 = 0x0
nfsAuthUnixSet("trombone", 2200, 70, 0, &aup_gids) /* cnts user, prism group*/
value = 0 = 0x0
nfsMount("trombone", "/cntsdata", "/cntsdata")
value = 0 = 0x0
\

```

```

# Write nfs mount info to stdout

```

```

hostShow
hostname      inet address      alias
-----
elrond        192.168.0.7
localhost     127.0.0.1

```

```

trombone          192.168.0.10
value = 0 = 0x0
nfsAuthUnixShow
machine name = trombone
user ID          = 2200
group ID         = 70
value = 19 = 0x13
nfsDevShow
device name      file system
-----
/cntsdata        trombone:/cntsdata
value = 0 = 0x0

# Load X window support
ld 1,0, "/cnts/tftpdire/vxwindows/support.o"
value = 33183960 = 0x1fa58d8

# initialize the time used by the time support utilities
# (it seems this synchronizes the target time with host time)
starttime()
value = 0 = 0x0

# Load Xlib
ld 0,0, "/cnts/tftpdire/vxwindows/X11.o"
value = 33163272 = 0x1fa0808

# Setup the X library initialization routines.
vxClientLibInit(1)
X11  libs loaded
Xext libs not loaded
Xmu  libs not loaded
Xt   libs not loaded
Xaw  libs not loaded
Xm   libs not loaded
value = 21 = 0x15

# Load application
ld 1,0, "/cnts/tftpdire/code/iso/iso.o"
value = 32642008 = 0x1f213d8

# Spawn application on X console at IP address.
# .10 is bilbo HP workstation, .65/.66 is Tek xterm, .182 is tuba

```



```
# Task will now run on the private control network -- 192.168.0.6
# Next two IP addresses are Modicon 1 and 2
taskSpawn "tUser",99,8,20000,user,4,"192.168.0.6:0","192.168.0.9","192.168.0.5",
value = 32943308 = 0x1f6acc
```

```
# Tie up the shell so doesn't it try to read PLC traffic
# Exit *doesn't* work here -
```

```
taskSuspend
Timeout file /cntsdata/data/iso/timeouts.dat, success at line 7
nargs 4, disp 192.168.0.6:0, task_opts 0, script (null), delay_msec 0
Calling vxlibinit
vxlibinit status 0
Font file /cntsdata/data/iso/fonts.dat, success at line 6
Color file /cntsdata/data/iso/colors.dat, success at line 2
Called XOpenDisplay on host 192.168.0.6:0, returned 0x3bca2c
Called XOpenDisplay on room host 192.168.0.11:0, returned 0x3bc9f3
trombone:/usr/vxwindows/lib/X11/Xcms.txt: No such file or directory
Keycode file /cntsdata/data/iso/keycodes.dat, success at line 28
```

NAME	ENTRY	TID	PRI	STATUS	PC	SP	ERRNO	DELAY
tRootTask	_usrRoot	1fbe038	0	DELAY	248a6	1fbdef0	30065	50
tExcTask	_excTask	1fddf40	0	PEND	52ca2	1fdde9c	0	0
tShell	_shell	1fb7a74	1	SUSPEND	24368	1fb8ab4	1c0001	0
tNetTask	_netTask	1fd8b5c	50	PEND	52490	1fd8b04	0	0
tPLC	_PLC_Task	1f83a58	90	READY	268f8	1f82c08	3d0004	53
tIntlk	_Intlk_Task	1f99f94	90	PEND	52490	1f99f58	0	0
tDMC	_DMC_Task	1f88a1c	90	PEND+T	52490	1f88840	3d0004	1
tLCC	_LCC_Task	1f8d9e0	92	PEND+T	52490	1f8d7f0	3d0004	33
tTMC	_TMC_Task	1f929a4	92	PEND+T	52490	1f92778	3d0004	3
tUser	_user	1fb51c0	99	PEND+T	52490	1fb4f4c	0	576

Active Internet connections (including servers)

PCB	Proto	Recv-Q	Send-Q	Local Address	Foreign Address	(state)
1fab08c	TCP	0	0	192.168.0.7.1025	192.168.0.6.6000	ESTABLISHED
1fdad0c	UDP	0	0	0.0.0.0.991	0.0.0.0.0	
1fdaf8c	UDP	0	0	0.0.0.0.1002	0.0.0.0.0	

Attempt #0:Connecting to Modicon 1 with a 10 second timeout

```
TCP/IP socket open to Modicon PLC1  
Attempt #0:Connecting to Modicon 2 with a 10 second timeout  
TCP/IP socket open to Modicon PLC2  
Returning from Open Port
```

After this startup sequence, the console terminal displays every log message that the program writes. Appendix J shows the first several log messages.

Appendix J

Control program log messages at startup

```
01-JUL-1999 15:49:37 Starting
01-JUL-1999 15:49:37 Log file /cntsdata/log/iso/iso1999-0701-154937.log, file no
01-JUL-1999 15:49:37 Treatment record file /cntsdata/log/iso/isotr1999-0701-1549
01-JUL-1999 15:49:37 Dose cal file, success at line 18
01-JUL-1999 15:49:37 Collimator cal file, success at line 120
01-JUL-1999 15:49:37 Tolerances file, success at line 7
01-JUL-1999 15:49:37 Ports file, success at line 25
01-JUL-1999 15:49:37 Collimator cal fields file, success at line 4
01-JUL-1999 15:49:38 Operator file, success at line 11
01-JUL-1999 15:49:38 Prescription file, success at line 205
01-JUL-1999 15:49:38 Accumulated dose file, success at line 61
01-JUL-1999 15:49:38 Run file, success at line 1
01-JUL-1999 15:49:38 Daily message file, success at line 4
01-JUL-1999 15:49:38 tIntlk spawned, task id 32908448
01-JUL-1999 15:49:38 TMC serial port /tyCo/1 open
01-JUL-1999 15:49:38 tTMC spawned, task id 32887992
01-JUL-1999 15:49:38 LCC serial port /tyCo/2 open
01-JUL-1999 15:49:38 tLCC spawned, task id 32867460
01-JUL-1999 15:49:38 DMC serial port /tyCo/3 open
01-JUL-1999 15:49:38 tDMC spawned, task id 32846928
01-JUL-1999 15:49:38 tPLC spawned, task id 32826272
01-JUL-1999 15:49:38 Isocentric therapy console ready
```

Bibliography

- [1] Cabletron Systems, Inc., P.O. Box 5005, Rochester, NH 03867-0505. *BRIM-E6 User's Guide*, 1994.
- [2] Cabletron Systems, Inc., P.O. Box 5005, Rochester, NH 03867-0505. *MicroMACC-22E, MicroMMAC-24E, MicroMMAC-32E, and MicroMMAC-34E, 10Base-T Intelligent Stackable Hub User's Guide*, 1994.
- [3] Jonathan Jacky. Clinical neutron therapy system implementation. Technical Report 2001-03-01, Department of Radiation Oncology, University of Washington, Box 356043, Seattle, Washington 98195-6043, USA, March 2001.
- [4] Jonathan Jacky and Ruedi Risler. Clinical neutron therapy system reference manual. Technical Report 99-10-01, Department of Radiation Oncology, University of Washington, Box 356043, Seattle, Washington 98195-6043, USA, October 1999.
- [5] Jonathan Jacky and Ruedi Risler. Clinical neutron therapy system therapist's guide. Technical Report 99-07-01, Department of Radiation Oncology, University of Washington, Box 356043, Seattle, Washington 98195-6043, USA, July 1999.
- [6] Ira J. Kalet, Jonathan Unger, Jonathan Jacky, and Mark Phillips. Prism system capabilities and user interface specification, version 1.2. Technical Report 97-12-02, Radiation Oncology Department, University of Washington, Seattle, Washington, 1997.
- [7] Motorola, Inc., Computer Group, 2900 South Diablo Way, Tempe, Arizona 85282-9602. *MVME712M Transition Module and P2 Adapter Board User's Manual*, MVME712M/D2 edition, 1993.
- [8] Motorola, Inc., Computer Group, 2900 South Diablo Way, Tempe, Arizona 85282-9602. *MVME167 Single Board Computer Installation Guide*, MVME167IG/D4 edition, 1997.
- [9] Adrian Nye. *Xlib Programming Manual*. O'Reilly and Associates, Inc., Sebastopol, CA, 1988.
- [10] Scanditronix AB. Dose monitor controller, DMC. In *MC 50 Manual*. Uppsala, Sweden, 1984. Volume 9B, Part 9.8.

- [11] Scanditronix AB. Leaf collimator controller, LCC. In *MC 50 Manual*. Uppsala, Sweden, 1984. Volume 9B, Part 9.6.
- [12] Scanditronix AB. Treatment motion controller, TMC. In *MC 50 Manual*. Uppsala, Sweden, 1984. Volume 9B, Part 9.7 (labeled 9.10 on most pages).
- [13] Schneider Electric, Inc., One High Street North Andover, MA. 01845. *Modicon TSX Quantum Automation Series Hardware Reference Guide*, Nov. 1996.
- [14] Schneider Electric, Inc., One High Street North Andover, MA. 01845. *Quantum Automation Series NOE x0 Ethernet Modules User Guide*, Nov. 1999.
- [15] W. Richard Stevens. *TCP/IP Illustrated, Volume 1: The Protocols*. Addison Wesley, 1994.
- [16] Visicom Laboratories, Inc., San Diego, California. *VisiCom X-Window System: VX-Windows 3.0 for VxWorks User's Manual*, 1996.
- [17] Wind River Systems, Inc., Alameda, California. *Tornado User's Guide (Unix Version) 1.0*, 1995.
- [18] Wind River Systems, Inc., Alameda, California. *VxWorks Programmer's Guide 5.3.1*, 1997.
- [19] Wind River Systems, Inc. WTN 32: Trouble shooting booting problems. Wind Technical Note for VxWorks 5.1.x, 1998. <http://www.wrs.com/csdocs/wtn32.html>.