



*Models 4920 and 4921 Series of PC  
Telecommunication Packages  
Volume B*

# ***USER'S GUIDE***

---

PN: 961-021-012  
*Revision A*  
*July 1995*

## " NOTICE

This publication contains information proprietary to Intermec Technologies Corporation. It is being supplied to you with the express understanding that the information contained herein is for the benefit of the contracting party only, and may not be copied, distributed, or displayed to third parties without the express written consent of Intermec Technologies Corporation, and shall be returned to Intermec Technologies Corporation upon written request. If a purchase, license, or nondisclosure agreement has been executed, the terms of that agreement shall govern this document.

This publication is furnished for information only, and the information in it is subject to change without notice. Although every effort has been made to provide complete and accurate information, Intermec Technologies Corporation assumes no responsibility or liability for any errors or inaccuracies that may appear in this document.

Disclaimer of Warranties. The sample source code included in this document is presented for reference only. The code does not necessarily represent complete, tested programs. The code is provided **"AS IS WITH ALL FAULTS."** **ALL WARRANTIES ARE EXPRESSLY DISCLAIMED, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.**

We welcome your comments concerning this publication. Although every effort has been made to keep it free of errors, some may occur. When reporting a specific problem, please describe it briefly and include the book title and part number, as well as the paragraph or figure number and the page number.

Send your comments to:  
Intermec Technologies Corporation  
Publications Department  
550 Second Street SE  
Cedar Rapids, IA 52401

INTERMEC, NORAND, NOR\*WARE, and PEN\*KEY are registered trademarks of Intermec Technologies Corporation.

© 1991 Intermec Technologies Corporation. All rights reserved.



This publication printed on recycled paper.

### Acknowledgments

*AT*, *DOS*, *IBM*, *Micro Channel*, *PS/2*, *Personal System/2*, and *OS/2* are registered trademarks of International Business Machines Corporation.

*Hayes* is a registered trademark of Hayes Microcomputer Products Inc.

*Metacomp* is a trademark of Metacomp, Inc.

*Norton Disk Doctor* and *Norton Utilities* are registered trademarks of Peter Norton Computing Inc.

Some information in this guide has been reproduced courtesy of Metacomp, Inc.

# CONTENTS

## SECTION 8

<b>Interpreting Error Messages</b> .....	<b>8-1</b>
Error Message Format .....	8-1
Errors .....	8-2

## SECTION 9

<b>Understanding the Communication Control File</b> .....	<b>9-1</b>
Port Quantity .....	9-2
File Processing .....	9-3
Communication Processor Card .....	9-3
Host Port .....	9-4
Port .....	9-4
Device .....	9-4
Protocol .....	9-5
Host Asynchronous Port .....	9-6
Speed .....	9-6
Databits .....	9-7
Parity .....	9-7
Stopbits .....	9-7
Bus Port .....	9-8
HHC Ports .....	9-8
Port .....	9-9
Minor .....	9-9
Protocol .....	9-10
Chans .....	9-10
Speed .....	9-11
Modemtype .....	9-11
Reset .....	9-12
Config .....	9-12

NPCP Port .....	9-13
ADCCP Port .....	9-13
Minaddr .....	9-14
Maxaddr .....	9-14
TTY Port .....	9-15
Single .....	9-16
Databits .....	9-16
Parity .....	9-16
Stopbits .....	9-17
Combination TTY or ADCCP Port .....	9-17
Ymodem Port .....	9-19
Sample Model 4920 Communication Control File .....	9-19
Sample Model 4921 Communication Control File .....	9-22
Using Communication Control File Menu .....	9-24
Definitions .....	9-25
Procedure .....	9-25
<b>SECTION 10</b>	
<b>Understanding the System Control File .....</b>	<b>10-1</b>
Ending Communication Parameters .....	10-2
Ending Communication Methods .....	10-3
Time .....	10-3
HHC Count .....	10-4
HHC Requests .....	10-4
Predetermined Number .....	10-4
HHC Counting Method Parameters .....	10-4
Communication End Parameters .....	10-5
Endtime .....	10-5
Enddays .....	10-6
Endreq .....	10-6
Endterms .....	10-7
Endidl .....	10-7
Endunsched .....	10-7
Tcomretries .....	10-8
Terminal Identifying Parameters .....	10-8
Termidpos .....	10-9
Termidlen .....	10-9
IDLCheck .....	10-9
IDLPos .....	10-10

VersCheck .....	10-10
VersPos .....	10-10
VersLen .....	10-11
Data Format Parameters .....	10-11
Uplconcur .....	10-11
Uplfixed .....	10-12
Uplreclen .....	10-12
Upllogical .....	10-13
Uplincbad .....	10-13
Uplincbadsess .....	10-14
Uplincdup .....	10-14
Upldelimit .....	10-14
Uplcrlf .....	10-15
Uploadfname .....	10-15
Uploadback .....	10-15
DEXfname .....	10-16
Dnlfixed .....	10-16
Dnlreclen .....	10-16
Dnltrunc .....	10-17
Dnlpack .....	10-17
System-Wide Display Options .....	10-18
Maincolor .....	10-18
Alt1color .....	10-19
Alt2color .....	10-19
Errorcolor .....	10-20
Helpcolor .....	10-20
Help .....	10-21
File Directories .....	10-21
Bootdir .....	10-22
Loaddir .....	10-22
Upldir .....	10-23
Dnldir .....	10-23
Miscellaneous Parameters .....	10-23
Dataeor .....	10-24
Datarem .....	10-24
Dataeof .....	10-24
Plneof .....	10-25
Debuglevel .....	10-25
Loglevel .....	10-26
Background .....	10-26

Restart .....	10-27
Tcomquery .....	10-27
Default Phone Parameters .....	10-27
Phoneretries .....	10-27
Phonewait .....	10-28
Nohalt .....	10-28
Errlog .....	10-29
Netflag .....	10-29
Netchkcnt .....	10-29
Sample System Control File .....	10-30
Using System Control File Menu .....	10-32
Definitions .....	10-32
Procedure .....	10-33

### SECTION 11

<b>Understanding the Menu Control File .....</b>	<b>11-1</b>
Commands .....	11-2
Menu .....	11-2
Name .....	11-3
Label .....	11-3
Entry .....	11-3
Prompt .....	11-3
Program .....	11-4
Submenu .....	11-5
Menu Definition .....	11-6
Menu Option Commands Order .....	11-6
Type Examples .....	11-7
Adding a Submenu .....	11-8
Sample Menu Control File .....	11-10

### SECTION 12

<b>Understanding the Session Control File .....</b>	<b>12-1</b>
Record Format .....	12-1
Record Types .....	12-2
Record Descriptions .....	12-3
Comment (#) Records .....	12-4
Mode (M) Records .....	12-4

Parameters .....	12-4
ZONe .....	12-5
VERsion .....	12-6
Request Records .....	12-6
Terminal Identifier (T) Records .....	12-6
Parameters .....	12-7
ZONe .....	12-7
VERsion .....	12-7
STAtus .....	12-7
/D .....	12-8
Network Communications Controller Identifier (N) Records ..	12-8
Parameters .....	12-9
STAtus .....	12-9
INIt .....	12-9
ZONe .....	12-9
DIR .....	12-10
TYPE .....	12-10
/D .....	12-10
File Set Identifier (S) Records .....	12-10
Parameters .....	12-11
/D .....	12-11
Load Boot Set (L) Records .....	12-11
Parameters .....	12-12
/D .....	12-13
Files and Broadcast Records .....	12-13
File Identifier (F) Records .....	12-13
Parameters .....	12-14
TYPe .....	12-14
CREate .....	12-15
STAtus .....	12-15
/D .....	12-15
/E .....	12-15
Broadcast File Identifier (B) Record .....	12-16
Parameters .....	12-16
TYPe .....	12-16
CREate .....	12-17
STAtus .....	12-17
/C .....	12-17
/D .....	12-17
/E .....	12-18

---

Data (D) Records .....	12-18
Parameters .....	12-19
Activation Records .....	12-19
Port Configuration (P) Records .....	12-19
PORt .....	12-20
DIAI .....	12-20
BUcket .....	12-20
Autoanswer Activate (A) Records .....	12-21
PORt .....	12-21
PROtocol .....	12-22
BEGin .....	12-22
End .....	12-22
SPEed .....	12-23
PARity .....	12-23
BITs .....	12-23
SINGle .....	12-23
LOW address .....	12-24
HIGH address .....	12-24
/D .....	12-24
Autocall Activate (C) Records .....	12-24
PHOne .....	12-25
PROtocol .....	12-25
BUcket .....	12-26
BEGin .....	12-26
End .....	12-27
RETriEs .....	12-27
WAIt .....	12-27
EXPEcted .....	12-27
SPEed .....	12-28
PARity .....	12-28
BITs .....	12-28
SINGle .....	12-28
LOW address .....	12-28
HIGH address .....	12-29
/D .....	12-29
Sample Session Control File .....	12-30
Using Request Database Menu .....	12-32
Definitions .....	12-33
Procedure .....	12-34
Using Activation Database Menu .....	12-34



Definitions .....	12-35
Procedure .....	12-36
Command Line Shortcuts .....	12-36
Retrieving a Menu .....	12-36
Using Command Line Switches .....	12-37
Usage Lines .....	12-37
Usage Line Notation .....	12-38
Maintaining Request Database from Command Line .....	12-38
Maintaining Activation Database from Command Line .....	12-39

### SECTION 13

<b>Understanding the Host Download File .....</b>	<b>13-1</b>
Record Format .....	13-2
Record Types .....	13-2
Comment (#) Records .....	13-3
Request Records .....	13-3
Upload (*U) Records .....	13-3
Parameters .....	13-3
Download (*D) Records .....	13-4
Parameters .....	13-4
Include File Records .....	13-4
File Identifier (*I) Records .....	13-5
Parameters .....	13-5
IDL Include (*L) Records .....	13-5
Parameters .....	13-5
Broadcast File Identifier (*B) Record .....	13-6
Parameters .....	13-6
Data Records .....	13-6
Activation Records .....	13-7
Autocall Activate (*P) Records .....	13-7
Parameters .....	13-7
Sample Host Download File .....	13-9

### SECTION 14

<b>Understanding the Upload Format Control File .....</b>	<b>14-1</b>
PL/N Upload File .....	14-2
Rectype .....	14-2

Format .....	14-2
Sample Upload Format Control File .....	14-3
DOS Files .....	14-4

*SECTION 15*

<b>Understanding the Host Upload File .....</b>	<b>15-1</b>
Upload Data Processing Options .....	15-1
Include Uploads from Bad Sessions Parameter .....	15-2
Include Duplicates Parameter .....	15-2
Host Upload File Record Types .....	15-4
Host.UPL File Format .....	15-4
Begin (B) Record .....	15-4
Data (D) Record .....	15-5
End (E) Record .....	15-5
Sample Host Upload File .....	15-5
Sample Logically Formatted Transaction File .....	15-6

*TABLES*

Table 10-1 Foreground Colors .....	10-18
------------------------------------	-------

*INDEX*

# Section 8

## Interpreting Error Messages

.....

This section lists the Models 4920 and 4921 Series of PC Telecommunication Packages error messages you may see on your screen. You may also see some self-explanatory and other messages which are not explained here. In this case, refer to your operating system's and personal computer's documentation. For hand-held computer (HHC) error messages, refer to the HHC's applications program.

---

### Error Message Format

Most error messages consist of two parts that contain descriptive information to help you solve the problem on your own. A typical format is as follows:

**<Error message>.**

**<Error number> on <line number> in <file name>.**

FIRST PART:

**ERROR MESSAGE:** A description of the error that occurred. This may include line numbers, file names, and other descriptive information so that you can open the specified file and resolve the error.

SECOND PART:

**ERROR NUMBER:** The number assigned to the error message. Errors are listed by number so that you can quickly find additional information about an error and the actions you can take to resolve it.

" NOTE:

*If an error number's explanation indicates a program error occurred, or if the error number is not listed in this section, call Product Support at 1-800-221-9236.*

The abbreviation 498X has been used throughout this section to indicate a 4980 Network Communications Controller or a 4985 Network Communications Controller.

---

## Errors

" NOTE: *If the error number is not listed, call 1-800-221-9236 and ask for Product Support.*

<b>Error</b>	<b>Message</b>	<b>Meaning</b>	<b>Solution</b>
0002	<b>No such file or directory. File name is &lt;name&gt;.</b>	The file or directory specified does not exist.	Make sure you typed the file name correctly and that it is in the current or specified directory.
0004		A "Ctrl-C" was pressed to interrupt the program.	
000C	<b>Not enough memory.</b>	The program ran out of memory.	Call Product Support.
000D	<b>Permission denied. File name is &lt;name&gt;.</b>	An attempt was made to write to a file marked read-only.	Remove the read-only attribute from the file. See "ATTRIB" in your operating system's reference manual for assistance.
0011	<b>File exists. File name is &lt;name&gt;.</b>	An attempt was made to open a file which already exists.	Use a different file name.
0018	<b>Too many open files. File name is &lt;name&gt;.</b>	The maximum number of files has been exceeded.	Make sure the value for the FILES configuration command in the CONFIG.SYS file is 50 or 60. This applies to DOS only. See "CONFIG.SYS" in your operating system's reference manual for assistance with changing the value.
001C	<b>No space left on device.</b>	The hard disk is full.	Erase unwanted files or move some files to a disk. See "ERASE" and "COPY" in your operating system's reference manual for assistance.

<b>Error</b>	<b>Message</b>	<b>Meaning</b>	<b>Solution</b>
0100 to 01FF	<b>Request database error.</b>	A database file is corrupted.	Delete all *.IDX and corresponding *.DAT files, <i>except for the IERR.DAT and IERR.IDX files.</i>
0201	<b>End of terminal/request data.</b>	The normal error code returned from the request database interface when an end-of-data condition is reached.	No corrective action necessary.
0202	<b>Terminal/request not found.</b>	The terminal or request specified does not exist in the request database.	Add the terminal or request to the database.
0203	<b>Data file error.</b>	An error was encountered while opening or reading a user data file.	<p>" Normally found in the LOG.DAT file, this would be an EDAT at the far left of the LOG.DAT field.</p> <p>" Make sure the file exists and is in the current download data directory specified by the SYSTEM.CTL file.</p> <p>" Verify that the CONFIG.SYS file has FILES=30 as a minimum. You may need to increase this value by 10 and retry.</p> <p>" If Norand is to look into this problem, provide the following information: CONFIG.SYS, SYSTEM.CTL, DOWNLOAD FILE (HOST.DNL or SESSION.CTL).</p>
0204	<b>Request record is locked.</b> (OS/2 only)	OS/2 Error only. Record is locked when another process is attempting to end and retry operation.	Wait for process to end and retry operation.

<b>Error Message</b>	<b>Meaning</b>	<b>Solution</b>
0205 <b>Request database not open.</b>	An attempted request database operation failed as the database was not successfully opened.	A database open operation may have failed because one or more database files have become corrupted. Attempt this action again. If the problem persists, try rebuilding or recreating the database. If the problem still persists, contact Norand.
0206 <b>Duplicate request record.</b>	The key used to add a request database record already exists in the database.	The record already in the database will be updated with information in the new record. This error can be ignored.
0207 <b>Terminal does not exist.</b>	An attempt was made to add a data request to a terminal which did not exist in the request database.	Add the terminal to the database and retry the operation.
0209 <b>Maximum open request iterators.</b>	The maximum number of open data iterators has been exceeded. A data iterator is opened to read the data for a terminal, such as download data.	Reduce the level of concurrent access to the data for terminals.
0208, 0210 to 0213 <b>Invalid request parameter.</b>	A terminal request parameter is invalid.	<p>" Check the parameters for the T, F, S, and B records in the session control file for errors or omissions. Correct the parameter and retry the parameter.</p> <p>" This could be a lost cluster, so attempt a SCANDISK /F. Run a disk maintenance program periodically to mark bad sectors, such as Norton Disk Doctor.</p> <p>" You may have to look at the download file, you will need to see the SESSION.CTL or HOST.DNL, SYSTEM.CTL, and possibly COMM.CTL files.</p>

<b>Error</b>	<b>Message</b>	<b>Meaning</b>	<b>Solution</b>
020A	<b>Request is disabled.</b>	A disabled request (/D switch) was encountered while the data requests for a terminal were being read.	Enable or delete the request, or ignore this error. See Section 12, "Understanding the Session Control File," for assistance.
020C	<b>Data packing error.</b>	Problem with download data, either PL/N header is incorrect: (file type, filename, record count, field description, record description, prefix and suffix characters) and is not properly formatted; OR  Incorrect data in a PL/N field, such as an Alpha in a Numeric field.	" Problem will occur when you run PLMFMT.EXE, HOSTDNF.EXE, or SESSFMT.EXE. " Look at the last file being processed to determine the problem area, usually you need to look at a printout of the file. " If Norand Support needs to look at this problem, you need to provide the following: SYSTEM.CTL and DOWNLOAD DATA.
020D	<b>Request delete error.</b>	An error was encountered while a request database record was being deleted.	Retry the command. If the problem persists, call Product Support.
020E	<b>Unable to rebuild terminal file.</b>	An error was encountered while the terminal files in the request database were rebuilding.	Make sure the output disk is neither full nor corrupted. See "SCANDISK" in your operating system's reference manual for assistance.
020F	<b>Unable to rebuild request file.</b>	An error was encountered while the request files in the request database were rebuilding.	Make sure the output disk is neither full nor corrupted. See "SCANDISK" in your operating system's reference manual for assistance.
0300 to 03FF	<b>Request open error.</b>	A database open operation failed, possibly because one (or more) database file is corrupted.  " If you are running SESSFMT.EXE, the HOSTDNF.EXE problem is probably a file limitation issue.	" Verify all minimum requirements.  Increase the number of files in CONFIG.SYS.

Error Message	Meaning	Solution
	" If you are running NCP492x.EXE, the problem is probably a FREE RAM limitation.	Run SCANDISK and verify that the Free RAM is above 520K for NCP492x.EXE and above 300K for LITE series.
	" You might have cross linked files.	Run SCANDISK/F.
	" You may have corrupted database files.	Delete all *.IDX and corresponding *.DAT files in the 4920 directory, <i>except IERR.DAT and IERR.IDX.</i>
		" You may need to look at the following files: COMM.CTL, SESSION.CTL, or HOST.DNL, SYSTEM.CTL, CONFIG.SYS, and SCANDISK information.
0400 to 04FF	<b>Activate database error.</b>	Usually indicates that a database file is corrupted.
	" If you are running SESSFMT.EXE, the HOSTDNF.EXE problem is probably a file limitation issue.	Increase the number of files in CONFIG.SYS.
	" If you are running NCP492x.EXE, the problem is probably a FREE RAM limitation.	Run SCANDISK and verify that the Free RAM is above 520K for NCP492x.EXE and above 300K for LITE series.
	" You might have cross linked files.	Run SCANDISK/F. Run a disk maintenance program periodically to mark bad sectors, such as Norton Disk Doctor.
	" You may have corrupted database files.	Delete all *.IDX and corresponding *.DAT files in the 4920 directory, <i>except IERR.DAT and IERR.IDX.</i>
0501	<b>End of activate data.</b>	The normal error code returned from the activate database interface when an end-of-data condition is reached. No corrective action is necessary.



<b>Error</b>	<b>Message</b>	<b>Meaning</b>	<b>Solution</b>
0502	<b>Activate record not found.</b>	The activate record specified does not exist in the activate database.	The activate record can be added to the database.
0503	<b>Activate record is locked. (OS/2 only)</b>	OS/2 Error only. The record is locked when another process is attempting to update.	Wait for the process to end and retry the operation.
0504	<b>Invalid activate parameter.</b>	An activate record parameter is invalid.	<ul style="list-style-type: none"> <li>" Verify all minimum requirements.</li> <li>" Check parameters for the F, A, and C records in SESSION.CTL file for errors or omissions. Correct the parameter and try command again.</li> <li>" You may have to look at the download file. Look at SESSION.CTL or HOST.DNL, SYSTEM.CTL, and possibly COMM.CTL.</li> </ul>
0505	<b>Activate database not open.</b>	An attempted activate database operation failed because the database was not successfully opened.	A database open operation may have failed because one or more database files have become corrupted. Attempt the action again. If the problem persists, try rebuilding or recreating the database. If the problem still persists, contact Norand.
0506	<b>Duplicate activate record.</b>	The key used to add an activate database record already exists in the database.	The record already in the database will be updated with information in the new record. This error can be ignored.
0507	<b>Activate delete error.</b>	An error was encountered while activate data were being deleted.	Retry the command. If the problem persists, call Product Support.

<b>Error</b>	<b>Message</b>	<b>Meaning</b>	<b>Solution</b>
0508	<b>Unable to create activate file.</b>	An error was encountered while the activation database was being created.	
0509		" A required port, autocall, or autocall parameter was not specified.	Check for the F, A, C, or P records in the session control file for errors or omissions. Correct the parameter and retry the command.
		" You may have a lost cluster.	Attempt a SCANDISK/F. Run a disk maintenance program periodically to mark bad sectors, such as Norton Disk Doctor.
		" You may have corrupted database files.	Delete all *.IDX and corresponding *.DAT files in the 4920 directory, <i>except for IERR.DAT and IERR.IDX.</i>
			" You may have to look at the download file. Look at SESSION.CTL or HOST.DNL, SYSTEM.CTL, and possibly COMM.CTL.
050A or 050B	<b>Required activate parameter missing.</b>	" A required port, autocall, or autoanswer parameter was not specified.	Check for the F, A, C, or P records in the session control file for errors or omissions. Correct the parameter and retry the command.
		" There could be a lost cluster.	Attempt a SCANDISK/F. Run a disk maintenance program periodically to mark bad sectors, such as Norton Disk Doctor.
		" You may have to look at the download file.	Look at SESSION.CTL or HOST.DNL, SYSTEM.CTL, or possibly COMM.CTL.

<b>Error Message</b>	<b>Meaning</b>	<b>Solution</b>
0600 to 06FF <b>Activate open error.</b>	A database open operation failed, possibly because one (or more) database file is corrupted.	" Verify all minimum requirements.
	" If you are running SESSFMT.EXE, the HOSTDNF.EXE problem is probably a file limitation issue.	Increase the number of files in CONFIG.SYS.
	" If you are running NCP492x.EXE, the problem is probably a FREE RAM limitation.	Run SCANDISK and verify that the Free RAM is above 520K for NCP492x.EXE and above 300K for LITE series.
	" You might have cross linked files.	Run SCANDISK/F. Run a disk maintenance program periodically to mark bad sectors, such as Norton Disk Doctor.
	" You may have corrupted database files.	Delete all *.IDX and corresponding *.DAT files in the 4920 directory, <i>except IERR.DAT and IERR.IDX.</i>
0700 to 07FF <b>Session control error.</b>	Error codes 0700 to 07FF are reserved for internal session control processing errors that do not usually occur.	Call Product Support.
0801 <b>End of file.</b>	The normal error returned when an end-of-file condition is reached while reading the session control file.	This error is not passed to the user and can be ignored.

<b>Error</b>	<b>Message</b>	<b>Meaning</b>	<b>Solution</b>
0802	<b>Session control file error.</b> <b>Line &lt;number&gt; in file</b> <b>&lt;name&gt;.</b>	Problem is in SESSION.CTL file.	<p>" Find the line number in the SESSION.CTL file and look for abnormalities.</p> <p>" Verify that the SESSION.CTL file is not empty.</p> <p>" Verify that there are no lines in the SESSION.CTL file that are over 255 characters long.</p> <p>" You may need to look at these 4920 files: COMM.CTL, SESSION.CTL or HOST.DNL, SYSTEM.CTL, CONFIG.SYS, and SCANDISK information.</p>
0803	<b>Invalid parameter. Line</b> <b>&lt;number&gt; in file &lt;name&gt;.</b>	A session control record has an invalid parameter.	<p>" Find the line &lt;number&gt; in the SESSION.CTL file and check for invalid parameters. Correct and retry command.</p> <p>" You may need to look at these 4920 files: COMM.CTL, SESSION.CTL or HOST.DNL, SYSTEM.CTL, CONFIG.SYS, and SCANDISK information.</p>
0804	<b>Invalid record type.</b>	A record type in the session control file is invalid.	<p>" Check the File Format parameters in the SYSTEM.CTL file or if passed on the command line for correctness (fixed length records, record length, etc.). Correct and retry the command.</p> <p>" Verify that files and directories exist, especially directories specified in the SYSTEM.CTL file. Download and boot the directory.</p> <p>" You may need to look at these 4920 files: COMM.CTL, SESSION.CTL or HOST.DNL, SYSTEM.CTL, CONFIG.SYS, and SCANDISK information.</p>

<b>Error</b>	<b>Message</b>	<b>Meaning</b>	<b>Solution</b>
0805	<b>Invalid record length. Record length-&lt;length&gt;.</b>	Record length specified in SYSTEM.CTL file or on the command line are incorrect.	Valid values are 2 through 255. " You may need to look at these 4920 files: COMM.CTL, SESSION.CTL or HOST.DNL, SYSTEM.CTL, CONFIG.SYS, and SCANDISK information.
0806	<b>Maximum broadcast files. Line &lt;number&gt; in file &lt;name&gt;.</b>	The maximum number of active broadcast files has been exceeded. Up to 40 broadcast files may be active concurrently	Check SESSION.CTL for proper format and build. " You may need to look at these 4920 files: COMM.CTL, SESSION.CTL or HOST.DNL, SYSTEM.CTL, CONFIG.SYS, and SCANDISK information.
0807	<b>Input file open error. File name is &lt;name&gt;.</b>	Download data format could not open the file specified.	" Verify that the file name and directory exist. " Verify that the input file exists. " If the HOST.DNL or SESSION.CTL file creates include files, verify that the file name is a proper DOS file and that the directory exists. " You may need to look at these 4920 files: COMM.CTL, SESSION.CTL or HOST.DNL, SYSTEM.CTL, CONFIG.SYS, and SCANDISK information. " Verify all minimum requirements.
0808	<b>Write error. Line &lt;number&gt; in file &lt;name&gt;.</b>	An error occurred while data was being written.	Make sure the output disk is neither full nor corrupted. See "SCANDISK" in your operating system's reference manual for assistance.
0900 to 09FF	<b>Error database error.</b>	Error codes 0900 to 09FF are reserved for internal error message database errors that do not usually occur.	Call Product Support.

<b>Error</b>	<b>Message</b>	<b>Meaning</b>	<b>Solution</b>
0A01	<b>Error file not found.</b>	An error file was not found.	Copy the IERR.IDX and IERR.DAT files from the release diskettes. See Section 3, "Installation," for assistance.
0B00	<b>Error opening error data base.</b>	An error was encountered while the error database was being opened.	Copy the IERR.IDX and IERR.DAT files from the release diskettes. See Section 3, "Installation," for assistance. If the problem persists, call Product Support.
0C00 to 0CFF	<b>NCP4920 error.</b>	Error codes 0C00 to 0CFF are reserved for internal communication program errors that do not usually occur.	Call Product Support.
0D01	<b>Invalid bus message type.</b>	An internal communication program error occurred.	Call Product Support if the problem persists.
0D02	<b>Invalid channel.</b>	An internal communication program error occurred.	<ul style="list-style-type: none"> <li>" Verify all Minimum Requirements.</li> <li>" 4920, verify shared memory is excluded in CONFIG.SYS, such as EMM386 NOEMS X=D000-DFFF. Memory Exclusion must match MPLD.CTL shared memory set-up.</li> <li>" Version 1.11 and 1.12 of 4920 would cause this problem if ports were configured for modems.</li> <li>" Bad Memory Chips on PC.</li> <li>" For Norand to analyze the situation, you need the following files and information: CONFIG.SYS, AUTOEXEC.BAT, SCANDISK statistics, 4920 Version, SYSTEM.CTL, COMM.CTL, and MPLD.CTL.</li> </ul>
0D03	<b>Invalid record type.</b>	An internal communication program error occurred.	Call Product Support if the problem persists.

<b>Error</b>	<b>Message</b>	<b>Meaning</b>	<b>Solution</b>
0D04	<b>Invalid secondary record type.</b>	An internal communication program error occurred.	Call Product Support if the problem persists.
0D05	<b>Invalid communications type.</b>	The protocol type for the host port (port 0) in the COMM.CTL file is invalid. Valid types are: 1 = asynchronous 4980 interface, 2 = PC card bus interface.	Correct the protocol type in the COMM.CTL file. See Section 9, "Understanding the Communication Control File," for assistance.
0D06	<b>Status file open error.</b>	An error occurred while the communication status file was being opened.	<p>" Make sure the value for the FILES configuration command in the CONFIG.SYS file is 50 or 60. This applies to DOS only. See "CONFIG.SYS" in your operating system's reference manual for assistance.</p> <p>" Retry the command. If the problem persists and the communication program was being run in RESTART mode (-ry switch), it may be necessary to run in NEW mode (-rn switch).</p> <p><i>Note: Statistical information for the current communication session will not be retained. See Section 10, "Understanding the System Control File," and Section 7, "Communication with the HHCs," for assistance.</i></p> <p>" If the problem persists, call Product Support.</p>

<b>Error</b>	<b>Message</b>	<b>Meaning</b>	<b>Solution</b>
0D07	<b>Log open error.</b>	A general system error occurred while the communication log file was being opened.	<p>" Make sure the hard disk is neither full nor corrupted. See "SCANDISK" in your operating system's reference manual for assistance.</p> <p>" Make sure the value for the FILES configuration command in the CONFIG.SYS file is 50 or 60. This applies to DOS only. See "CONFIG.SYS" in your operating system's reference manual for assistance.</p> <p>" If the problem persists, call Product Support.</p>
0D08	<b>NCP4920 initialization error.</b>	A general system error occurred during the communication program system initialization phase.	Make sure the hard disk is neither full nor corrupted. See "SCANDISK" in your operating system's reference manual for assistance. If this does not work, call Product Support.
0F01	<b>Dwnlrq - out of space.</b>	The level of concurrent download request activity has exhausted the space allocated for download request list processing.	Retry the download request when the level of activity has subsided.
0F02	<b>Maximum dwnlrq requests.</b>	The maximum number of pending download requests has been exceeded.	Retry the download request when the level of activity has subsided.
0F03	<b>Maximum dwnlrq requests active.</b>	The maximum number of active download requests has been exceeded.	Retry the download request when the level of activity has subsided.
0F04	<b>Invalid dwnlrq handle.</b>	This is an internal download request processing error that does not usually occur.	Call Product Support.
0F05	<b>Dwnlrq file error.</b>	An error occurred while a file specified in a download request was being opened or read.	Ensure that the file specified is accessible.



<b>Error</b>	<b>Message</b>	<b>Meaning</b>	<b>Solution</b>
0F06	<b>Dwnlrq file set error.</b>	An error occurred while a download request file set was being processed.	" Check for an invalid filename in the file set. " If the problem persists, rebuild or recreate the request database. " See Section 12, "Understanding the Session Control File," for assistance.
0F07	<b>End of dwnlrq file.</b>	This is the normal internal error code returned when an end-of-file condition is reached on a download request or set.	Ignore this message. It is not passed to the user.
1001	<b>End of communications status file.</b>	This is the normal internal error code returned when an end-of-file condition is reached on the communication status file.	Ignore this message. It is not passed to the user.
1002	<b>Status record not found.</b>	This is the normal internal error code returned when the status for a terminal does not exist in the status file.	Ignore this error. It is not passed to the user.
1100 to 11FF	<b>Communications status file error.</b>	Error codes 1100 to 11FF are reserved for internal communication status file errors that do not usually occur.	Call Product Support.

<b>Error</b>	<b>Message</b>	<b>Meaning</b>	<b>Solution</b>
1200	<b>PL/N data error. Line &lt;number&gt; in file &lt;name&gt;.</b>	<i>During Download Data Processing:</i> usually indicates a problem with the format of the data.	<p>" Validate the data at the line number and the file name to verify that it matches the PL/N record descriptor.</p> <p>" Look at the output file and see what the last record was before this error occurred.</p> <p>" Look at the download data files: SYSTEM.CTL and download data files for any discrepancies.</p>
		<i>During Upload Data Processing:</i> indicates a problem with the format of the data.	<p>" Validate the UPLFMT.CTL file with the application specification.</p> <p>" Look at the output file and see what the last record was before the error occurred.</p>
		Could be a mixed upload if the customer does not have the proper 49XX and 4XXX programs. (4000 SYS4000 Version 1.02 or above; 4920 Version 1.14 or above; 4980 V1.22 or above; and 4985 1.04 or above.	" See the 4920 upload files: UPLOAD.DAT, UPLFMT.CTL, and SYSTEM.CTL.

Error	Message	Meaning	Solution
1201	<b>PL/N file header was expected. Line &lt;number&gt; in file &lt;name&gt;.</b>	<i>During Download Data Processing:</i> indicates a problem with the format of the PL/N header.	<p>" Validate the data at the line number and the file name to verify that it contains a valid PL/N header (file type, file name, record count, field descriptor., record descriptor).</p> <p>" Look at the output file and see what the last record was before this error occurred.</p> <p>" Look at the download data files: SYSTEM.CTL and download data files for any discrepancies.</p>
		<i>During Upload Data Processing:</i> indicates a problem with the format of the data.	<p>" Validate the UPLFMT.CTL file with the application specification.</p> <p>" Look at the output file and see what the last record was before the error occurred.</p>
		Could be a mixed upload if the customer does not have the proper 49XX and 4XXX programs. (4000 SYS4000 Version 1.02 or above; 4920 Version 1.14 or above; 4980 V1.22 or above; and 4985 1.04 or above.	<p>" See the 4920 upload files: UPLOAD.DAT, UPLFMT.CTL, and SYSTEM.CTL.</p>

Error	Message	Meaning	Solution
1202	<b>Invalid PL/N file header.</b> <b>Line &lt;number&gt; in file</b> <b>&lt;name&gt;.</b>	<i>During Download Data Processing:</i> indicates a problem with the format of the PL/N header.	<p>" Validate the data at the line number and the file name to verify that it contains a valid PL/N header (file type, file name, record count, field descriptor., record descriptor).</p> <p>" Look at the output file and see what the last record was before this error occurred.</p> <p>" Look at the download data files: SYSTEM.CTL and download data files for any discrepancies.</p>
		<i>During Upload Data Processing:</i> indicates a problem with the format of the data.	<p>" Validate the UPLFMT.CTL file with the application specification.</p> <p>" Look at the output file and see what the last record was before the error occurred.</p>
		Could be a mixed upload if the customer does not have the proper 49XX and 4XXX programs. (4000 SYS4000 Version 1.02 or above; 4920 Version 1.14 or above; 4980 V1.22 or above; and 4985 1.04 or above.	<p>" See the 4920 upload files: UPLOAD.DAT, UPLFMT.CTL, and SYSTEM.CTL.</p>

Error	Message	Meaning	Solution
1203	<b>Invalid file type in a PL/N header. Line &lt;number&gt; in file &lt;name&gt;.</b>	<i>During Download Data Processing:</i> indicates a problem with the format of the PL/N header.	<p>" Validate the data at the line number and the file name to verify that it contains a valid PL/N header (Check the file type: the letter in the first position after the "&lt;" prefix). File type should be a D, B, E, or P.</p> <p>" Look at the output file and see what the last record was before this error occurred.</p> <p>" Look at the download data files: SYSTEM.CTL and download data files for any discrepancies.</p>
		<i>During Upload Data Processing:</i> indicates a problem with the format of the data.	<p>" Validate the UPLFMT.CTL file with the application specification.</p> <p>" Look at the output file and see what the last record was before the error occurred.</p>
		Could be a mixed upload if the customer does not have the proper 49XX and 4XXX programs. (4000 SYS4000 Version 1.02 or above; 4920 Version 1.14 or above; 4980 V1.22 or above; and 4985 1.04 or above.	" See the 4920 upload files: UPLOAD.DAT, UPLFMT.CTL, and SYSTEM.CTL.

<b>Error</b>	<b>Message</b>	<b>Meaning</b>	<b>Solution</b>
1204	<b>Invalid field in a PL/N header. Line &lt;number&gt; in file &lt;name&gt;.</b>	<i>During Download Data Processing:</i> indicates a problem with the format of the PL/N header.	<p>" Validate the data at the line number and the file name to verify that it contains a valid PL/N header. A field descriptor in the PL/N file header is invalid. Check each field descriptor in the header. Valid descriptors are X, N, B, W, A, or 9.</p> <p>" Look at the output file and see what the last record was before this error occurred.</p> <p>" Look at the download data files: SYSTEM.CTL and download data files for any discrepancies.</p>
		<i>During Upload Data Processing:</i> indicates a problem with the format of the data.	<p>" Validate the UPLFMT.CTL file with the application specification.</p> <p>" Look at the output file and see what the last record was before the error occurred.</p>
		Could be a mixed upload if the customer does not have the proper 49XX and 4XXX programs. (4000 SYS4000 Version 1.02 or above; 4920 Version 1.14 or above; 4980 V1.22 or above; and 4985 1.04 or above.	" See the 4920 upload files: UPLOAD.DAT, UPLFMT.CTL, and SYSTEM.CTL.

<b>Error</b>	<b>Message</b>	<b>Meaning</b>	<b>Solution</b>
1205	<b>Invalid record type in a PL/N file. Line &lt;number&gt; in file &lt;name&gt;.</b>	<i>During Upload Data Processing:</i> indicates a problem with the format of the data.	<p>" Validate the UPLFMT.CTL file with the application specification. The value for "rectype" in the UPLFMT.CTL file is missing or does not match the data in the file. Ensure that the "rectype" value in the UPLFMT.CTL file matches the PL/N file data.</p> <p>" Look at the output file and see what the last record was before the error occurred.</p>
		Could be a mixed upload if the customer does not have the proper 49XX and 4XXX programs. (4000 SYS4000 Version 1.02 or above; 4920 Version 1.14 or above; 4980 V1.22 or above; and 4985 1.04 or above.	" See the 4920 upload files: UPLOAD.DAT, UPLFMT.CTL, and SYSTEM.CTL.

<b>Error</b>	<b>Message</b>	<b>Meaning</b>	<b>Solution</b>
1206	<b>Unmatched "(" or ")" in a PL/N file header. Line &lt;number&gt; in file &lt;name&gt;.</b>	<i>During Download Data Processing:</i> indicates a problem with the format of the PL/N header.	<p>" Validate the data at the line number and the file name to verify that it contains a valid PL/N header. A left or right parenthesis for an array structure is missing in the PL/N file header. Enter the proper parenthesis to enclose the array.</p> <p>" Look at the output file and see what the last record was before this error occurred.</p> <p>" Look at the download data files: SYSTEM.CTL and download data files for any discrepancies.</p>
		<i>During Upload Data Processing:</i> indicates a problem with the format of the data.	<p>" Validate the UPLFMT.CTL file with the application specification.</p> <p>" Look at the output file and see what the last record was before the error occurred.</p>
		Could be a mixed upload if the customer does not have the proper 49XX and 4XXX programs. (4000 SYS4000 Version 1.02 or above; 4920 Version 1.14 or above; 4980 V1.22 or above; and 4985 1.04 or above.	" See the 4920 upload files: UPLOAD.DAT, UPLFMT.CTL, and SYSTEM.CTL.



Error	Message	Meaning	Solution
1207	<b>Data in PL/N file does not match field type in header. Line &lt;number&gt; in file &lt;name&gt;.</b>	<i>During Download Data Processing:</i> indicates a problem with the format of the PL/N header.	<p>" Validate the data at the line number and the file name to verify that it contains a valid PL/N header. A field descriptor specified in the PL/N file header does not match the structure of the data in the file. Make sure the field descriptor correctly identifies the type of data in the file.</p> <p>" Look at the output file and see what the last record was before this error occurred.</p> <p>" Look at the download data files: SYSTEM.CTL and download data files for any discrepancies.</p>
		<i>During Upload Data Processing:</i> indicates a problem with the format of the data.	<p>" Validate the UPLFMT.CTL file with the application specification.</p> <p>" Look at the output file and see what the last record was before the error occurred.</p>
		Could be a mixed upload if the customer does not have the proper 49XX and 4XXX programs. (4000 SYS4000 Version 1.02 or above; 4920 Version 1.14 or above; 4980 V1.22 or above; and 4985 1.04 or above.	" See the 4920 upload files: UPLOAD.DAT, UPLFMT.CTL, and SYSTEM.CTL.

Error	Message	Meaning	Solution
1208	<b>Internal buffer overflow.</b> <b>Line &lt;number&gt; in file</b> <b>&lt;name&gt;.</b>	<i>During Download Data Processing:</i> indicates a problem with the format of the PL/N header.	<p>" Validate the data at the line number and the file name to verify that it contains a valid PL/N header. The internal buffer used to store the PL/N data to be formatted is full. This usually means the values for the parameters used to read the file (such as fixed length records, record length, and truncate trailing spaces) do not match the data being formatted. Set the values for these parameters to match the data in the file.</p> <p>" Look at the output file and see what the last record was before this error occurred.</p> <p>" Look at the download data files: SYSTEM.CTL and download data files for any discrepancies.</p>
		<i>During Upload Data Processing:</i> indicates a problem with the format of the data.	<p>" Validate the UPLFMT.CTL file with the application specification. In DEX upload processing where the HHC is uploading DEX audit trails, but the UPLFMT.CTL file has not defined the DEX PL/N file. See DEX in upload processing.</p> <p>" Look at the output file and see what the last record was before the error occurred.</p>
		Could be a mixed upload if the customer does not have the proper 49XX and 4XXX programs. (4000 SYS4000 Version 1.02 or above; 4920 Version 1.14 or above; 4980 V1.22 or above; and 4985 1.04 or above.	" See the 4920 upload files: UPLOAD.DAT, UPLFMT.CTL, and SYSTEM.CTL.

<b>Error</b>	<b>Message</b>	<b>Meaning</b>	<b>Solution</b>
1209	<b>Internal format table overflow.</b>	<i>During Upload Data Processing:</i> the internal table used to store information about variable-length files is full. Too many variable-length files have been specified in the UPLFMT.CTL file (maximum is 15).	" Reduce the number of variable-length files, the number of FILE= in the UPLFMT.CTL file must not exceed 15. " Look at the 4920 upload files: UPLOAD.DAT, UPLFMT.CTL, and SYSTEM.CTL for any discrepancies.
120A	<b>Header exceeded size limit.</b>	The PL/N header exceeded the maximum allowable size.	This error would indicate a possible corruption in the UPLOAD data. The SETUSTAT.EXE program can be used to mark the individual terminal BAD.
1300	<b>Data file error. File name is &lt;name&gt;.</b>	A program error occurred while a data file was being processed.	Call Product Support.
1301	<b>Data file read error. File name is &lt;name&gt;.</b>	An error occurred while a file was being read.	" Verify that the file exists, is in the current directory, and is not corrupted. " Make sure the hard disk is not corrupted.  See "SCANDISK" in your operating system's reference manual for assistance.
1302	<b>Data file write error. File name is &lt;name&gt;.</b>	An error occurred while a file was being written.	" Verify that the file exists, is in the current directory, and is not corrupted. " Make sure the hard disk is not corrupted.  See "SCANDISK" in your operating system's reference manual for assistance.

<b>Error</b>	<b>Message</b>	<b>Meaning</b>	<b>Solution</b>
1303	<b>A line is too long in a data file. File name is &lt;name&gt;.</b>	A line is too long in a data file. The internal buffer used to store the PL/N data to be formatted is full. This usually means the parameters used to read the file (such as fixed-length records, record length, and truncate trailing spaces) do not match the data being formatted and the end of the file cannot be found.	" Make sure the values for these parameters match the data in the file. See Section 10, "Understanding the System Control File," Section 5, "Download Data Format," or Section 6, "Upload Data Format," whichever is appropriate. " Verify that there are 1000 characters or fewer per line.
1304	<b>Error reading a file size.</b>	This error can only occur when trying to format a "DOS" file into an include file. The PC cannot determine the correct size of the indicated file.	Call Product Support.
1400	<b>Program error.</b>	A program error occurred.	Call Product Support.
1401	<b>Error initializing the help system. On-line help will not be available.</b>	The "help" parameter in the SYSTEM.CTL file is set to "Y" and the help file for the program being run cannot be opened.	" Check to see if the help file exists and is in the current directory. " Set the "help" parameter in the SYSTEM.CTL file to "N" if you do <i>not</i> want on-line help. " Verify all minimum requirements.
		A lost cluster.	Attempt a SCANDISK/F. Run a disk maintenance program periodically to mark bad sectors, such as Norton Disk Doctor.
1403	<b>An input file name must be entered.</b>	The name of the file to be formatted was not entered.	Type the name of the file to be formatted. The file must already exist and be in the current directory.
1404	<b>An output file name must be entered.</b>	The name of the file that receives the results of formatting was not entered.	Type the name of the output file to be created from the input file. The file must already exist and be in the current directory.

<b>Error</b>	<b>Message</b>	<b>Meaning</b>	<b>Solution</b>
1405	<b>A record length must be specified.</b>	The "Fixed-length records" parameter is set "Yes" or the "Logical records" parameter is set "No," but a record length was not specified.	Type a number for the record length parameter. Valid values are 1 through 256.
1406	<b>Display initialization error.</b>	The program cannot initialize the screen to display a menu.	" If you are using a clone, it may <i>not</i> be IBM-compatible. " Call Product Support.
1407	<b>Communications control file open error.</b>	The communication control file cannot be found.	" Make sure the file is named COMM.CTL " Verify that the file is in the same directory as the program.
1408	<b>System control file open error.</b>	The system control file cannot be found.	" Make sure the file is named SYSTEM.CTL. " Verify that the file is in the same directory as the program.
1409	<b>Menu control file open error.</b>	The menu control file cannot be found.	" Make sure the file is named MENU.CTL. " Verify that the file is in the same directory as the program.
140A	<b>Upload format control file open error.</b>	The upload format control file cannot be found.	" Make sure the UPLFMT.CTL exists and is in the same directory as the programs. " Verify all minimum requirements.
140B	<b>Size of TERMID doesn't match parameter.</b>	The size of the terminal ID found in the SESSION.CTL or HOST.DNL file does not match the size specified in the TERMIDLEN parameter located in the SYSTEM.CTL file.	Either the parameter is incorrect or there are additional trailing characters (possible spaces) on the terminal specifier line and the option was selected during format NOT to strip trailing spaces.
1500	<b>Communications error.</b>	A communication program error occurred.	Call Product Support.

<b>Error Message</b>	<b>Meaning</b>	<b>Solution</b>
1501 <b>Communications port open error.</b>	The communication program could not open the device specified in the "device" parameter in the COMM.CTL file.	<ul style="list-style-type: none"> <li>" Verify all minimum requirements.</li> <li>" Change the MPLD.CTL Shared Memory Segment. There may be a conflict, change the MPLD.CTL file to A000 through E000 (Change CONFIG.SYS exclude memory region to match MPLD.CTL) and reboot PC.</li> <li>" Attempt to disable the PC system: Cache, Shadow RAM.</li> </ul>
	<i>PS/2 running OS/2</i>	<ul style="list-style-type: none"> <li>" Verify that the correct host port is in COMM.CTL file using your EDITCOMM.EXE program. Board one should be addressed as MPLD1, board two MPLD2, etc.</li> <li>" Verify that the MPLD.SYS device driver is loaded correctly during power up. Verify the input parameters of the device driver (Shared memory segment, extended memory segment, and interrupt).</li> <li>" Verify that the board is securely seated in the PC Bus.</li> </ul>
	You have a bad board.	" Replace this board.
1502 <b>Communications IOCTL error.</b>	The program received an invalid command.	A program error occurred. Call Product Support.
1503 <b>Communications read error.</b>	The program could not read from the device for the host port specified in the "device" parameter in the COMM.CTL file.	Make sure the CPC board is installed correctly. See Section 3, "Installation," for assistance.

Error	Message	Meaning	Solution
1504	<b>Communications write error.</b>	The program could not write to the device specified in the "device" parameter in the COMM.CTL file.	<ul style="list-style-type: none"> <li>" Verify all minimum requirements.</li> <li>" Change the MPLD.CTL Shared Memory Segment. There may be a conflict, change the MPLD.CTL file to A000 through E000 (Change CONFIG.SYS exclude memory region to match MPLD.CTL) and reboot PC.</li> <li>" Attempt to disable the PC system: Cache, Shadow RAM.</li> </ul>
		<i>PS/2 running OS/2</i>	<ul style="list-style-type: none"> <li>" Verify that the correct host port is in COMM.CTL file using your EDITCOMM.EXE program. Board one should be addressed as MPLD1, board two MPLD2, etc.</li> <li>" Verify that the MPLD.SYS device driver is loaded correctly during power up. Verify the input parameters of the device driver (Shared memory segment, extended memory segment, and interrupt).</li> <li>" Verify that the board is securely seated in the PC Bus.</li> </ul>
		You have a bad board.	" Replace this board.

<b>Error</b>	<b>Message</b>	<b>Meaning</b>	<b>Solution</b>
1505	<b>Communications read time-out error.</b>	The program could not read data from the communication device in a predetermined amount of time.	<ul style="list-style-type: none"> <li>" Verify all minimum requirements.</li> <li>" Change the MPLD.CTL Shared Memory Segment. There may be a conflict, change the MPLD.CTL file to A000 through E000 (Change CONFIG.SYS exclude memory region to match MPLD.CTL) and reboot PC.</li> <li>" Attempt to disable the PC system: Cache, Shadow RAM.</li> </ul>
		<i>PS/2 running OS/2</i>	<ul style="list-style-type: none"> <li>" Verify that the correct host port is in COMM.CTL file using your EDITCOMM.EXE program. Board one should be addressed as MPLD1, board two MPLD2, etc.</li> <li>" Verify that the MPLD.SYS device driver is loaded correctly during power up. Verify the input parameters of the device driver (Shared memory segment, extended memory segment, and interrupt).</li> <li>" Verify that the board is securely seated in the PC Bus.</li> </ul>
		Possibly a PC resource issue, especially when running a background network or host emulation.	" Versions lower than 2.00, call Product Support. 4920 and 4921 version 2.00 or higher will recover from this error.
		You have a bad board.	" Replace this board.



Error	Message	Meaning	Solution
1506	<b>Communications message error.</b>	The program received invalid data from the controller.	<ul style="list-style-type: none"> <li>" Verify all minimum requirements.</li> <li>" Change the MPLD.CTL Shared Memory Segment. There may be a conflict, change the MPLD.CTL file to A000 through E000 (Change CONFIG.SYS exclude memory region to match MPLD.CTL) and reboot PC.</li> <li>" Attempt to disable the PC system: Cache, Shadow RAM.</li> </ul>
		<i>PS/2 running OS/2</i>	<ul style="list-style-type: none"> <li>" Verify that the correct host port is in COMM.CTL file using your EDITCOMM.EXE program. Board one should be addressed as MPLD1, board two MPLD2, etc.</li> <li>" Verify that the MPLD.SYS device driver is loaded correctly during power up. Verify the input parameters of the device driver (Shared memory segment, extended memory segment, and interrupt).</li> <li>" Verify that the board is securely seated in the PC Bus.</li> </ul>
		Possibly a PC resource issue, especially when running a background network or host emulation.	" Versions lower than 2.00, call Product Support. 4920 and 4921 version 2.00 or higher will recover from this error.
		You have a bad board.	" Replace this board.

<b>Error</b>	<b>Message</b>	<b>Meaning</b>	<b>Solution</b>
1507	<b>Communications read message error.</b>	The program could not read from the device specified in the "device" parameter in the COMM.CTL file.	<ul style="list-style-type: none"> <li>" Verify all minimum requirements.</li> <li>" Change the MPLD.CTL Shared Memory Segment. There may be a conflict, change the MPLD.CTL file to A000 through E000 (Change CONFIG.SYS exclude memory region to match MPLD.CTL) and reboot PC.</li> <li>" Attempt to disable the PC system: Cache, Shadow RAM.</li> </ul>
		<i>PS/2 running OS/2</i>	<ul style="list-style-type: none"> <li>" Verify that the correct host port is in COMM.CTL file using your EDITCOMM.EXE program. Board one should be addressed as MPLD1, board two MPLD2, etc.</li> <li>" Verify that the MPLD.SYS device driver is loaded correctly during power up. Verify the input parameters of the device driver (Shared memory segment, extended memory segment, and interrupt).</li> <li>" Verify that the board is securely seated in the PC Bus.</li> </ul>
		Possibly a PC resource issue, especially when running a background network or host emulation.	" Versions lower than 2.00, call Product Support. 4920 and 4921 version 2.00 or higher will recover from this error.
		You have a bad board.	" Replace this board.
1508	<b>Communications write message error.</b>	The program could not write to the device specified in the "device" parameter in the COMM.CTL file.	Make sure the CPC board is installed correctly. See Section 3, "Installation," for assistance.
1509	<b>Communications configuration file error. Configuration record is &lt;record&gt;.</b>	The program was not able to send configuration information to the controller.	Use EDITCOMM.EXE to verify that all HHC port parameters in the COMM.CTL file are correct.

<b>Error</b>	<b>Message</b>	<b>Meaning</b>	<b>Solution</b>
150A	<b>Communications program load error. File name is &lt;name&gt;.</b>	The 4920 load directory is not defined correctly in the SYS-TEM.CTL file.	" Verify that there are 16 BIN files in the 4920 load directory. " Verify all minimum requirements.
		A board may have no firmware or an incorrect firmware version (a customer upgraded from NI925 to 4920).	" Check the firmware location and name on the 4920 card. " Attempt to disable the PC system: Cache, Shadow RAM.
		You have a bad board.	" Replace this board.
150B	<b>Communications program run error. File name is &lt;name&gt;.</b>	The program could not run a program on the controller.	" Verify all minimum requirements.
		A board may have no firmware or an incorrect firmware version (a customer upgraded from NI925 to 4920).	" Check the firmware location and name on the 4920 card. " Attempt to disable the PC system: Cache, Shadow RAM
		PCI Bus or Pentium machine are not supported.	" See minimum requirements.
		You have a bad board.	" Replace this board.
1600	<b>Communications error.</b>	A communication program error occurred.	Call Product Support.
1701	<b>Async port open error.</b> (4921 only)	4921 System Error, cannot open the serial port specified in the COMM.CTL file.	Parameter must be COM1 or COM2. Verify that the serial port exists and the card is securely seated on the PC Bus.

<b>Error</b>	<b>Message</b>	<b>Meaning</b>	<b>Solution</b>
1702	<b>Communications read timeout. (4921 only)</b>	The program could not read data from the communication device in a predetermined amount of time.	<ul style="list-style-type: none"> <li>" Verify that the cabling is securely connected from the PC to the 4980 or 4985 controller (connected to proper PC serial port, and 4980 physical port). The cable number for a 25-Pin Serial Port is 216-680-00X; a 9-Pin Serial port NPN: 216-639-00X.</li> <li>" Verify that the 4921's COMM.CTL host port speed matches the 4980 speed configuration or the 4985's dip switch selection.</li> </ul>
		Possibly a bad 4980 or 4985 serial port or cable.	" Replace the components to isolate the problem.
		Possibly a PC resource issue, especially when running a background network or host emulation.	" Versions lower than 2.00, call Product Support. 4920 and 4921 version 2.00 or higher will recover from this error.
1703	<b>Upload data CRC error. (4921 only)</b>	The program received invalid data from the 4980 or 4985 controller.	<ul style="list-style-type: none"> <li>" Restart the communication program.</li> <li>" Make sure the PC and the 4980 Network Communications Controller are not too far apart.</li> <li>" Check cables and connections.</li> <li>" Lower the communication speed (such as from 19.2 kilobits per second to 9.6 kilobits per second).</li> </ul>
		Possibly a PC resource issue, especially when running a background network or host emulation.	" Versions lower than 2.00, call Product Support. 4920 and 4921 version 2.00 or higher will recover from this error.
1704	<b>Invalid read state. (4921 only)</b>	A program error occurred	Call Product Support.

<b>Error</b>	<b>Message</b>	<b>Meaning</b>	<b>Solution</b>
1705	<b>Communications read error.</b> (4921 only)	The program could not read from the driver specified in the "device" parameter in the COMM.CTL file.	" Verify that the cabling is securely connected from the PC to the 4980 or 4985 controller (connected to proper PC serial port, and 4980 physical port). The cable number for a 25-Pin Serial Port is 216-680-00X; a 9-Pin Serial port NPN: 216-639-00X. " Verify that the 4921's COMM.CTL host port speed matches the 4980 speed configuration or the 4985's dip switch selection.
		Possibly a bad 4980 or 4985 serial port or cable.	" Replace the components to isolate the problem.
		Possibly a PC resource issue, especially when running a background network or host emulation.	" Versions lower than 2.00, call Product Support. 4920 and 4921 version 2.00 or higher will recover from this error.
1706	<b>Communications write error.</b>	The program could not write to the device specified in the "device" parameter in the COMM.CTL file. This error applies only to Model 4921.	Make sure the cable to the 4980 Network Communications Controller is still connected.
1707	<b>Communications IOCTL error.</b> (4921 only)	The program received an invalid command. A program error occurred.	Call Product Support.
1800	<b>Upload file error. File name is &lt;name&gt;.</b>	A program error occurred while the upload file was being processed.	Call Product Support.

<b>Error</b>	<b>Message</b>	<b>Meaning</b>	<b>Solution</b>
1901	<b>Upload file open error. File name is &lt;name&gt;.</b>	The upload file could not be found, or the operating system returned an error while opening the file.	<p>" Make sure the upload file exists, is in the current directory, and has the correct name.</p> <p>" Make sure the value for the FILES configuration command in the CONFIG.SYS file is 50 or 60. This applies to DOS only. See "CONFIG.SYS" in your operating system's reference manual for assistance.</p>
1902	<b>Upload file write error. File name is &lt;name&gt;.</b>	The program could not write to the upload file because the disk is full or corrupted.	Erase unwanted files or move some files to a diskette. See "ERASE" and "COPY" in your operating system's reference manual for assistance. See also "SCANDISK" for assistance with a corrupted disk.
1903	<b>Upload file read error. File name is &lt;name&gt;.</b>	The program could not read the upload file.	<p>" Verify that the upload file exists, is in the current directory, and is not corrupted.</p> <p>" Make sure the hard disk is not corrupted.</p> <p>See "SCANDISK" in your operating system's reference manual for assistance.</p>
1904	<b>Invalid record type in the upload file. File name is &lt;name&gt;.</b>	The upload file's record type is invalid. The value for "rectype" in the UPLFMT.CTL file is missing or does not match the data in the file.	<p>" Make sure the value for "rectype" in the UPLFMT.CTL file matches the data in the upload file, and ensure the upload file is valid. See Section 14, "Understanding the Upload Format Control File," for assistance.</p> <p>" Verify that the upload file is not corrupted. See "SCANDISK" in your operating system's reference manual for assistance.</p>

<b>Error</b>	<b>Message</b>	<b>Meaning</b>	<b>Solution</b>
1A01	<b>Only one host port can be enabled.</b>	More than one port has been configured with a host protocol (Bisync, Async, Secondary ADCCP, or Bus). Exactly one port must be configured with a host protocol.	Make sure only one port is configured with a host protocol, and all others are configured with a HHC protocol. See Section 9, "Understanding the Communication Control File," for assistance.
1A02	<b>Only one NPCP port can be enabled.</b>	More than one port has been configured with NPCP. Only one NPCP port is supported on the 4920.	Make sure only one port is configured with NPCP. See Section 9, "Understanding the Communication Control File," for assistance.
1A03	<b>NPCP can only be enabled on port A.</b>	NPCP has been enabled on a port other than A. On a 4980, only port A is capable of running NPCP.	Change the configuration of the port. See Section 9, "Understanding the Communication Control File," for assistance.
1A04	<b>An alternate port must be disabled.</b>	A primary port is disabled, but the alternate part is enabled.	Disable the alternate port, or make the alternate configuration the primary configuration. See Section 9, "Understanding the Communication Control File," for assistance.
1A05	<b>Only ADCCP and TTY can alternate on a port.</b>	An alternate port was configured with an invalid protocol. Only ADCCP and TTY can be alternated on a port.	Change the configuration of the port. See Section 9, "Understanding the Communication Control File," for assistance.
1A06	<b>A host port must be enabled.</b>	No ports were configured with a host protocol (Bisync, Async, Secondary ADCCP, or Bus). Exactly one port must be configured with a host protocol.	Make sure only one port is configured with a host protocol, and all others are configured with a HHC protocol. See Section 9, "Understanding the Communication Control File," for assistance.

<b>Error</b>	<b>Message</b>	<b>Meaning</b>	<b>Solution</b>
1A07	<b>At least two ports must be enabled.</b>	Only one port was configured. At least one host port and one HHC port must be configured.	Make sure only one port is configured with a host protocol, and at least one is configured with a HHC protocol. See Section 9, "Understanding the Communication Control File," for assistance.
1A08	<b>The port must be enabled to change protocol options.</b>	An attempt was made to change protocol options on a disabled port.	Enable the port, then change protocol options.
1A09	<b>The terminal must be saved before changing file options.</b>	An attempt was made to change file options for a terminal which has not been saved. An added terminal must be saved before its file options can be changed.	Save the terminal, then change file options.
1A0A	<b>The set must be saved before changing file options.</b>	An attempt was made to change file options for a set which had not been saved. An added set must be saved before its file options can be changed.	Save the set, then change file options.
1A0B	<b>The terminal must be enabled to change file options.</b>	An attempt was made to change file options for a disabled terminal.	Enable the terminal, then change file options.
1A0C	<b>The set must be enabled to change file options.</b>	An attempt was made to change file options for a disabled set.	Enable the set, then change file options.
1A0D	<b>A terminal ID must be entered.</b>	A terminal ID was not entered for an added terminal.	Enter a terminal ID.
1A0E	<b>A set name must be entered.</b>	A set name was not entered for an added set.	Enter a set name.
1A0F	<b>The terminal ID entered already exists.</b>	An attempt was made to add a terminal which already exists in the request database.	Use a different terminal ID, or change or delete the existing terminal.
1A10	<b>The set name entered already exists.</b>	An attempt was made to add a set which already exists in the request database.	Use a different set name, or change or delete the existing set name.
1A11	<b>A file name must be entered.</b>	A file name was not entered for an added file.	Enter the file name.



<b>Error Message</b>	<b>Meaning</b>	<b>Solution</b>
1A12 <b>The file name and type entered already exist.</b>	An attempt was made to add a file which already exists in the request database for this terminal or set.	Use a different file name, or change or delete the existing file name.
1A13 <b>A terminal ID cannot contain spaces.</b>	The terminal ID entered contains spaces. A terminal ID cannot contain spaces. Use an underscore ( _ ) or dash ( - ) to separate words or characters.	Reenter the terminal ID.
1A14 <b>A set name cannot contain spaces.</b>	The set name entered contains spaces. A set name cannot contain spaces. Use an underscore ( _ ) or dash ( - ) to separate words or characters.	Reenter the set name.
1A15 <b>An invalid port was entered.</b>	The port letter that was entered is invalid. Only ports A through D are valid.	Reenter the port letter. See Section 9, "Understanding the Communication Control File," for assistance.
1A16 <b>The port entered already exists.</b>	An attempt was made to add a port which already exists in the activation database.	Use a different port letter, or change or delete the existing port.
1A17 <b>A port can use a bucket only once.</b>	A bucket was entered more than once for the same port. Each bucket can be specified only once for a port.	Change the buckets for the port. See Section 12, "Understanding the Session Control File," for assistance.
1A18 <b>A phone number must be entered.</b>	A phone number was not entered for an added call activation.	Enter the phone number. See Section 12, "Understanding the Session Control File," for assistance.
1A19 <b>The phone number entered already exists.</b>	An attempt was made to add a call activation which already exists in the activation database.	Use a different phone number, or change or delete the existing call activation. See Section 12, "Understanding the Session Control File," for assistance.

<b>Error</b>	<b>Message</b>	<b>Meaning</b>	<b>Solution</b>
1A1A	<b>The port and begin time entered already exist.</b>	An attempt was made to add an answer activation which already exists in the activation database.	Enter a different port or begin time, or change or delete the existing answer activation. See Section 12, "Understanding the Session Control File," for assistance.
1B00 1B01 1B02	<b>CSCAPE errors.</b>	An error was encountered while trying to access the display.	Call Product Support.
1C00 to 1CFF	<b>Device database error.</b>	Codes 1C00 to 1CFF are reserved for internal device database errors that do not usually occur. They usually indicate that a database file is corrupted. "Device" is the name of a remote 498X.	Try rebuilding or recreating the database file. If the problem persists, call Product Support.
1D01	<b>End of device/file data.</b>	This is the normal internal error code returned when an end-of-file condition is reached on a device file. "Device" is the name of a remote 498X.	Ignore this message. It is not passed to the user.
1D02	<b>Device/file not found.</b>	A device or file was not found. "Device" is the name of a remote 498X.	Make sure you typed the device or file name correctly and that it is in the current or specified directory.
1D03	<b>Data file error.</b>	An error was encountered while opening or reading a user data file.	" Make sure the file exists. " The file can be specified by a full DOS pathname, and the file can be in the BOOTDIR or DNLDIR directory specified in the SYSTEM.CTL file.
1D04	<b>Device record is locked.</b> (OS/2 only)	OS/2 Error only. Record is locked when another process is attempting to update.	Wait for process to end and retry operation.

<b>Error</b>	<b>Message</b>	<b>Meaning</b>	<b>Solution</b>
1D05	<b>Device database not open.</b>	The device database was not opened before an access was attempted. "Device" is the name of a remote 498X.	Make sure the value for the FILES configuration command in the CONFIG.SYS file is 50 or 60. This applies to DOS only. See "CONFIG.SYS" in your operating system's reference manual for assistance.
1D06	<b>Duplicate device/file record.</b>	An attempt was made to add a device or file which already exists. This error is not passed on to the user in "sessfmt." "Device" is the name of a remote 498X.	Use a different device name.
1D07	<b>Device does not exist.</b>	The device specified does not exist. "Device" is the name of a remote 498X.	Make sure you typed the device name correctly and that it is in the current or specified directory.
1D08	<b>Invalid device/file parameter.</b>	A device or file parameter is invalid. "Device" is the name of a remote 498X.	Correct the parameter in the SESSION.CTL file. See Session 12, "Understanding the Session Control File," for assistance.
1D09	<b>Maximum open iterators exceeded.</b>	The maximum number of open data iterators has been exceeded. A data iterator is opened to read the data (e.g., download data) for a device. This error may indicate that too many devices are active concurrently. "Device" is the name of a remote 498X.	Call Product Support.
1D0A	<b>Device/file is disabled.</b>	This error simply indicates that a device or a device file has been disabled. "Device" is the name of a remote 498X.	Enable or delete the device or file, or both, or ignore this error.
1D0B	<b>Bad iterator handle.</b>	An internal communication program error occurred.	Call Product Support.

<b>Error</b>	<b>Message</b>	<b>Meaning</b>	<b>Solution</b>
1D0D	<b>Device/file delete error.</b>	A database error occurred. "Device" is the name of a remote 498X	Try rebuilding or recreating the database. Note that a delete error will <i>not</i> occur when the target record does not exist. If the problem persists, call Product Support.
1D0E	<b>Unable to rebuild device file.</b>	An error was encountered while the files in the device database were rebuilding. "Device" is the name of a remote 498X.	Make sure the output disk is neither full nor corrupted. See "SCANDISK" in your operating system's reference manual for assistance. If the rebuild error persists, recreate the database and reload it. <i>Note that this assumes the database has been backed up.</i>
1D0F	<b>Unable to rebuild device_file file.</b>	An error was encountered while the device files in the device database were rebuilding. "Device" is the name of a remote 498X.	Make sure the output disk is neither full nor corrupted. See "SCANDISK" in your operating system's reference manual for assistance. If the rebuild error persists, recreate the database and reload it. <i>Note that this assumes the database has been backed up.</i>
1D10	<b>Unable to create device file.</b>	An error was encountered while the device files in the device database were being created. "Device" is the name of a remote 498X.	" Make sure the output disk is neither full nor corrupted. See "SCANDISK" in your operating system's reference manual for assistance. " Make sure the value for the FILES configuration command in the CONFIG.SYS file is 50 or 60. This applies to DOS only. See "CONFIG.SYS" in your operating system's reference manual for assistance.

<b>Error</b>	<b>Message</b>	<b>Meaning</b>	<b>Solution</b>
1D11	<b>Unable to create device_file file.</b>	An error was encountered while the device files in the device_file database were being created. "Device" is the name of a remote 498X.	" Make sure the output disk is neither full nor corrupted. See "SCANDISK" in your operating system's reference manual for assistance. " Make sure the value for the FILES configuration command in the CONFIG.SYS file is 50 or 60. This applies to DOS only. See "CONFIG.SYS" in your operating system's reference manual for assistance.
1D12	<b>Maximum device files exceeded.</b>	The maximum number of files which can be scheduled for a device has been exceeded. "Device" is the name of a remote 498X.	Reduce the number of files (such as combine them into a self-extracting archive if they are 4000 Series boot files).
1D13	<b>File create error on remote device.</b>	" The maximum number of files on the remote device may have been reached. " Space on the remote device is exhausted. " An error occurred while the file on the remote device was being written to. "Device is the name of a remote 498X.	" Check the log to determine the number of files and size. " Make sure the value for the FILES configuration command in the CONFIG.SYS file is 50 or 60. This applies to DOS only. See "CONFIG.SYS" in your operating system's reference manual for assistance.
1E00	<b>Device database open error.</b>	A device database open operation failed, possibly because one (or more) database file is corrupted. "Device" is the name of a remote 498X.	" Make sure the value for the FILES configuration command in the CONFIG.SYS file is 50 or 60. This applies to DOS only. See "CONFIG.SYS" in your operating system's reference manual for assistance. " Retry the command. If the problem persists, call Product Support.

Error	Message	Meaning	Solution
WARN- ING	<b>492X Display Status</b>	The 492X error status on the display is normally caused by a file that was prepared for download, or by an include file that does not exist.	" Look in the LOG.DAT file and find a message which starts in the first column that says EDAT. Look at the right side of the LOG.DAT file for the file in error. (LOG level in the SYSTEM.CTL file should be set to 3).
		Could be a FILES= in the CONFIG.SYS that does not have a large enough value.	" Increase the value by 10 and retry the telecommunications sessions. " Verify that the SYSTEM.CTL file has the appropriate download directory, and that the directory exists. " You will need to see LOG.DAT, COMM.CTL, SYSTEM.CTL, and SESSION.CTL or HOST.DNL.
BAD	<b>492X Display Status</b>		" Normally need to look at the HHC Error message to determine the cause of the problem. (LOG level in SYSTEM.CTL file should be set to 3). " If the upload is good and the download is good but the overall session is <i>bad</i> , you need to look in the LOG.DAT file and find the ENDS for the route in question. Look to the far right and find out what the HHC sent up for an End of Session. Use this information in the HHC Error message to determine the problem. " You need to see LOG.DAT, COMM.CTL, SYSTEM.CTL, and SESSION.CTL or HOST.DNL.

## Section 9

# Understanding the Communication Control File

.....

The communication control file (COMM.CTL) defines:

- " number of ports
- " physical devices to communicate through
- " protocol for each port
- " speed of communication
- " 498X (4980 or 4985) file processing

The communication control file must contain the following information *in this order*:

1. total active ports
2. enabled or disabled file processing
3. commands for the Communication Processor Card (Model 4920 only)
4. host port definition
5. hand-held computer (HHC) port definitions

Always define the first port, "port=0," for communication between the PC and your network communication control hardware.

If you are configuring the Models 4920 and 4921 Series of PC Telecommunication Packages, you are using the Communication Processor Card (CPC) as the communication link. If you are configuring a Model 4921 package, a Model 4980 Communication Controller communicates between the PC and HHCs (HHCs). Differences are noted by model number where they occur.

Configure HHC ports depending on their intended use. Use TTY protocol to communicate to all HHCs including 101s, 121s, and 141s. Use ADCCP to communicate with remote 498X devices and 121s and 141s in Multidrop Quad Lockboxes. Use NPCP to communicate to local 4000 Series HHCs attached to the 4920 or 4921. Otherwise communicate with remote 4000 Series HHCs using TTY protocol. Use Ymodem protocol exclusively to remotely boot 4000 Series HHCs.

There are two ways you can edit the communication control file:

- " Use a straight or flat ASCII text editor. Refer to its reference manual for assistance.
- " Use the on-line file editor provided by Norand. This gives you interactive access to COMM.CTL. The menu is an option on the Main Menu, and is explained in Section 4, "Preparing to Use the System" in Volume A of the *Models 4920 and 4921 Series of PC Telecommunication Packages User's Guide NPN: 961-021-011*.

With one exception, command names and options in the communication control file are not case-sensitive, so you can mix capital (uppercase) and small (lowercase) letters. Hayes modems command strings "reset" and "config" (explained later) are the only exceptions. Their command strings (the part after the equal sign) must be in uppercase.

The following sections explain each command in the communication control file. In some cases, a few commands should be omitted. Specifically, the commands "task =" and "driver =" for the Model 4921 and Hayes modem commands "reset =" and "config =" when using modems from Norand Corporation.

To configure your system, use the examples which follow making changes as appropriate when your system differs from the example.

---

## Port Quantity

The "ports =" command defines the maximum number of ports the entire system uses. Count the PC to CPC or Communication Controller as one. The number of ports can be as few as two and as many as five.



With the CPC in the PC (Model 4920), the maximum number of ports is five. For serial communication to Model 4980 Network Communications Controller (Model 4921 PC Telecommunication Package), the maximum number is four.

When you use fewer than the maximum number of ports, the Model 4920 Series of PC Telecommunication Packages uses the first port numbers it encounters in the communication control file.

#### PORTS

Number of ports.

Length/type: one-digit numeric.

Valid values: 2 through 5.

Default: (none). Do not omit.

Example: ports = 4

Use 4 ports (including the host port; such as 3 HHC ports).

---

### *File Processing*

The "files=" command lets you enable or disable file processing on remote 498X (4980 or 4985 Network Communication Controllers) devices.

Length/type: one-letter logical flag.

Valid values: y (yes) or n (no).

Default: n (no).

Example: files = y

Enable 498X file processing.

---

### *Communication Processor Card*

The communication control file contains the information (e.g., task and driver parameters) required to load the board. The sample Model 4920 Communication Control File at the end of this section lists the commands.

## Host Port

The host port can communicate either using the PC bus (Model 4920) or the serial communication connector (Model 4921). The bus method communicates with the CPC using the PC's internal bus. Use the serial connector to communicate to a Model 4980 Communication Controller by Norand.

The following commands are common to both host protocols. Refer to the description following individual examples for commands specific to a protocol.

### Port

Port number.

The logical port number for host communication. This number should always be zero.

Length/type: one-digit numeric.

Valid values: 0

Default: (none).

Example: port = 0

Host port is zero.

### Device

Device driver.

Operating system device driver used to communicate between the host port and the PC.

Names the device driver for communicating to the 4980 Communication Controller or to the host port on the CPC. The appropriate driver depends on where the host port resides.

Choose the appropriate drive from the following table.

#### **Host Port Device Drivers**

<b>Model</b>	<u>4920</u>	<u>4921</u>
	mpld1	com1 or com2

The Model 4920 Series of PC Telecommunication Packages always uses the device driver "mpld1."

The Model 4921 Series of PC Telecommunication Packages uses the device driver provided with the operating system, either "com1" or "com2," depending on the serial connector you use.

The device command is required when you configure all ports. Do not omit it.

Length/type: five-character alphanumeric.

Valid values: mpld1, com1, and com2.

Default: (none).

Example: device = com1

Use your PC's first serial communication connector.

If you have two serial communication ports and want to use the second one, the command is "device = com2."

### *Protocol*

Protocol.

The protocol this port uses to communicate.

You have two options for host protocol:

1 - host asynchronous (to a 4980 Communication Controller)

6 - internal bus (within the system unit)

Host asynchronous protocol communicates exclusively to the Model 4980 Communication Controller using the serial port. An internal bus port communicates to the CPC mounted within the PC.

The protocol command must appear before speed and data descriptions. It defines which commands to expect with it.

Length/type: one-digit numeric.

Valid values: 1 or 6, where:  
 1 = host asynchronous  
 6 = using the internal bus

Default: (none). Do not omit it.

Example: protocol = 6

Use the internal bus for communicating PC to CPC.

---

## Host Asynchronous Port

Use a host asynchronous port to communicate with the Model 4980 Network Communications Controller by Norand using the PC's serial connector (referred to as "com1"). This is a Model 4921 PC Telecommunication Package configuration.

" **NOTE:** *Ensure that the communication control file on the boot disk of the Model 4980 Communication Controller is compatible with the Model 4920 Series of PC Communication Packages communication control file.*

Commands for the host port must appear in the communication control file before commands to configure HHC ports.

A sample host asynchronous communication port follows with an explanation of commands unique to this protocol. Refer to the "Host Port" paragraphs for an explanation of common individual commands.

**# host asynchronous port commands**

```
port=0
device=com1
protocol=1
speed=9600
databits=8
parity=0
stopbits=1
#
```

### *Speed*

Communication speed.

How fast communication should occur. Speed is the communication rate in bits per second. Generally, you should choose the fastest rate your equipment handles.

Length/type: four or five digits numeric.

Valid values: 1200, 2400, 4800, 9600, or 19200.

Default: (none). Do not omit it.

Example: speed = 9600

Communicate at 9600 bits per second.

### *Databits*

Databits.

The number of bits which make up a character in your data.

Length/type: one-digit numeric.

Valid values: 7 or 8.

Default: 8.

Example: databits = 8

The data contains eight-bit characters.

### *Parity*

Parity.

Parity defines which method, if any, your data uses for error detection.

Length/type: one-digit numeric.

Valid values: 0 through 2, where:

0 = none

1 = odd

2 = even

Default: 0.

Example: parity = 0

No parity checking.

### *Stopbits*

Stopbits.

The number of stop bits your data uses.

Length/type: one-digit numeric.

Valid values: 1 through 3, where:

1 = one stop bit

2 = two stop bits

3 = one-and-a-half stop bits

Example: stopbits = 1

The data contains one stop bit.

---

## Bus Port

A bus port communicates with the CPC mounted within the PC's system unit. The following commands must appear in the communication control file before commands to configure HHC ports, in this order:

1. port = 0
2. device = mpld1
3. protocol = 6

An example of bus port commands follows. Refer to the preceding paragraphs for an explanation of individual commands.

```
# configure bus port
port=0
device=mpl d1
protocol=6
#
```

---

## HHC Ports

Read the following examples to learn how to configure HHC ports.

The HHC ports can be configured for the following:

- NPCP, NORAND® Portable Computer Protocol, to communicate with local 4000 Series HHCs
- ADCCP, Advanced Data Communications Control Procedures, to communicate with remote 498X devices and 121 and 141 HHCs

- " TTY, Teletype, for the same HHCs as ADCCP plus 101 and Series 4000 HHCs. The 4920 supports 7- or 8-bit data with or without parity checking
- " Ymodem, for booting remote 4000 Series HHCs

In addition, you can configure a combination TTY/ADCCP port. ADCCP and TTY are the only two protocols which can be combined on a single port.

HHC ports must have these commands in this order:

1. port
2. minor
3. protocol

" **NOTE:**

*Minor letters (such as minor = C) for HHC port configurations using the serial port (Model 4921) refer to connectors on the back of the Model 4980 Network Communications Controller.*

Each protocol requires some different commands. For example, TTY requires databits, parity, and stopbits; but no minaddr or maxaddr. In contrast, ADCCP requires minaddr and maxaddr, but no databits, parity, or stopbits.

The paragraphs which follow explain commands common to all methods of communication. After the common commands is an example for each method. Commands unique to a specific method follow the example.

### *Port*

The logical port number.

Length/type: one-character alphabetic.

Valid values: 0 through 4.

Default: (none).

Example: port = 1

Communicate through logical port 1.

### *Minor*

Physical connector this port should use.

Length/type: one-letter alphabetic.  
Valid values: A through D.  
Default: (none).  
Example: minor = B  
Communicate through physical connector B.

### *Protocol*

Protocol the equipment uses to communicate through this port.

Length/type: one-digit numeric.  
Valid values: 2, 3, 4, or 7, where:  
2 = ADCCP  
3 = TTY  
4 = NPCP  
7 = Ymodem, for remotely booting 4000 Series terminals only.  
Default: (none).  
Example: protocol = 2  
Use ADCCP protocol.

The protocol command must precede "chans." It defines which commands to expect.

### *Chans*

Defines the greatest number of HHC exchanges that can occur simultaneously on a single port of the Models 4920 or 4921 Series of PC Telecommunication Packages. You should set the channels as high as possible to maximize use of all ports.

Since TTY protocol cannot use multiple channels, this parameter can be omitted from TTY configurations.



Length/type: one-digit numeric.  
Valid values: 1 through 3, with these upper limits:  
1 = TTY maximum  
2 = ADCCP maximum  
3 = NPCP maximum.  
Default: (none).  
Example: chans = 2  
Allow two channels simultaneously.

### *Speed*

The bits-per-second communication rate.

Length/type: four-digit numeric.  
Valid values: 1200, 2400, 4800, 9600. Ymodem also supports 300 and 600 bps. ADCCP also supports 0 bps meaning external modem clocking.  
Default: (none).  
Example: speed = 9600  
Communicate at 9600 bits per second.

### *Modemtype*

Modem type.

Modemtype tells the Models 4920 and 4921 Series of PC Telecommunication Packages which type of modem a port uses. Ports communicating ADCCP, TTY, Ymodem, or either ADCCP or TTY, use this command.

Length/type: one-digit numeric.  
Valid values: 0 through 6, where:  
0 = no modem (local)  
1 = NM2400A  
2 = NM2400  
3 = 1200  
4 = 2400  
6 = 9600 MNP on  
7 = 9600 MNP off\*

Default: 0 (no modem).

Example: modemtype = 6

Use a 9600 bits-per-second modem.

\* Only used if autodialing to non-MNP answering machines (TTY only.)

### **Reset**

Command which instructs another company's modems to reset. Refer to the modem's instructions for additional information. Ports using ADCCP or TTY or both require this command.

This is a command string for Hayes-compatible modems.

When using a NORAND modem, you do not need to specify a reset command.

Length/type: up to 10 uppercase alphanumeric.  
Valid values: (refer to modem instructions).  
Default: (none).  
Example: reset = AT&F  
Reset the modem.

### **Config**

Configuration.

A command string which tells a Hayes-compatible modem how to configure itself. Ports communicating ADCCP or TTY or both use this command.

When using a NORAND modem, you do not need to specify a reset command.

Length/type: up to 40 uppercase alphanumeric.

Valid values: (refer to modem instructions).

Default: (none).

Example: **confi g=ATE0VOX0&C1&M1&D2&R1S0=1**

Configure for 9600 bits-per-second ADCCP.

---

### *NPCP Port*

The following example configures a port to communicate with 4000 Series HHCs using NORAND Portable Computer Protocol (NPCP). Refer to preceding paragraphs for an explanation of common commands.

```
# Configure port 1 for NPCP to 4000
# SERIES HHCs
port=1
mi nor=A
protocol =4
chans=3
#
```

The protocol command for NPCP ports is always the same: "protocol = 4." Do not change it.

---

### *ADCCP Port*

Use ADCCP to communicate with remote 498X devices. Use the following configuration to communicate to HHCs using ADCCP. Refer to the preceding paragraphs for an explanation of common commands.

```
# configure port 2 for ADCCP
port=2
mi nor=B
protocol =2
```

```
chans=2
speed=9600
mi naddr=1
maxaddr=64
modemtype=6
reset=ATF
confi g=ATE0V0X0&C1&M1&D2&R1S0=1
#
```

reset = AT can be used to disable reset to factory options.

This command is the same for all ADCCP protocols: "protocol = 2." Do not change it.

Commands "reset" and "config" are required only if you use another company's modem. NORAND modems configure properly using the "modemtype" command alone. Check the instructions for other company's modem to find the appropriate "reset =" and "config =" command strings.

### *Minaddr*

Minimum address.

Indicates the lowest physical address to which ADCCP communication should be attempted. Each Multidrop Quad LockBox (MQL) slot has as its own number beginning with 1 and going as high as 64.

The minimum address command lets you set the minimum address higher than 1 to skip unused slots.

Length/type: one- or two-digit numeric.

Valid values: 1 through 64.

Default: 1.

Example: minaddr = 12

Begin communication with physical address 12.

### *Maxaddr*

Maximum address.

Indicates the highest physical address to which ADCCP communication should be attempted. Each Multidrop Quad Lockbox (MQL) slot has as its own number beginning with 1 and going as high as 64.

The maximum address command lets you set the maximum address lower than 64 to skip unused slots.

Length/type: one- or two-digit numeric.

Valid values: 1 through 64.

Default: 64.

Example: maxaddr = 30

Make 30 the highest physical address to which communication is attempted.

---

## TTY Port

The following commands configure a port to communicate with HHCs using TTY. The example shows all the necessary commands. An explanation of each command follows the example.

```
# configure port 3 for TTY at 9600 bps
port=3
mi nor=C
protocol =3
si ngl e=n
speed=9600
databi ts=8
pari ty=0
stopbi ts=1
modemtype=6
reset=AT&F
confi g=ATE0V0X0&C1&M0&R1S0=1
#
```

This command is the same for all TTY ports: "protocol = 3." Do not change it.

Commands "reset" and "config" are required only if you use another company's modem. NORAND modems configure properly using the "modemtype" command alone. Check the instructions for other company's modem to find the appropriate "reset =" and "config =" command strings.

### *Single*

Determines whether one or more HHC exchanges are allowed before a communication session ends. "Single" might be used in a development laboratory or with remote sights with a salesperson communicating from home.

Length/type: one-letter logical flag.

Valid values: y (yes) or n (no).

Default: n (no).

Example: single = n

Communication should not end after one HHC exchange.

### *Databits*

Data bits.

The number of bits which make up a character in your data.

Length/type: one-digit numeric.

Valid values: 7 or 8.

Example: databits = 8

The data contains eight-bit characters.

If the HHC application is packed data, set the databits value to 8. If unpacked, set the value to 7.

### *Parity*

Parity defines which method, if any, your data uses for error detection.

Length/type: one-digit numeric.  
Valid values: 0 through 2, where:  
0 = none - Pack Data  
1 = odd  
2 = even - Unpack Data  
Default: 0.  
Example: parity = 0  
No parity checking.

If packed data, set the parity value to either 0 or 1. If unpacked data, set the value to 2.

### *Stopbits*

Stop bits.

The number of stop bits used in your data.

Length/type: one-digit numeric.  
Valid values: 1 through 3, where:  
1 = one stop bit  
2 = two stop bits  
3 = one-and-a-half stop bits  
Example: stopbits = 1  
The data contains one stop bit.

---

### *Combination TTY or ADCCP Port*

The example which follows is configured to operate as either a TTY or ADCCP port. The activation commands, which come from the session control file, host download file, or your input at the PC screen (depending on your operation), controls which configuration to use at a particular time. TTY and ADCCP are the only protocols you can combine on one port.

This configuration might be used for an installation which includes HHCs that communicate from the depot and others that communicate from the homes of salespeople. The port could communicate early in the day with the majority of HHCs in the depot which have the faster ADCCP equipment. Later, the port is reconfigured for TTY communication with individual HHCs in salespeople's homes.

In another example, the port could switch to ADCCP to autodial depots and then revert to TTY to answer calls from salespeople's home.

*The port defaults to the first "protocol = " command until external intervention from you or when an autocal or autoanswer activate record is in effect. Regardless of source, the port is reconfigured as instructed. External intervention is the way the secondary protocol becomes active.*

When the secondary protocol becomes inactive (for example, all phone numbers have been successfully called and answered), the port returns to a "ready" state of the primary protocol.

The following example shows one command set required to configure the combination port. Refer to "HHC Ports" and previous simple ADCCP and TTY port examples for individual command explanations.

```
# assign port and minor
port=4
minor=D
#
# configure for TTY communication
protocol=3
single=n
speed=9600
databits=8
parity=0
stopbits=1
#
# set TTY modemtype, reset, config
modemtype=6
reset=AT&F
config=ATE0V0X0&C1&M&R1S0=1
#
# configure for ADCCP communication
```



```
protocol=2
single=n
speed=9600
chans=2
minaddr=1
maxaddr=64
#
# set ADCCP modemtype, reset, config
modemtype=6
reset=AT&F
config=ATE0V0X0&C1&M1&D2&R1S0=1
#
```

---

### *Ymodem Port*

A Ymodem port is used exclusively for booting remote 4000 Series HHCs. The following is an example configuring a Ymodem port. All the commands have been explained previously. Refer to "HHC Ports" for an explanation of individual commands.

```
# Ymodem boot port
port=3
minor=D
protocol=7
speed=9600
modemtype=6
```

The Ymodem boot port is number 3, connector D, Ymodem protocol, 9600 bits per second, on a 9600 modem.

---

### *Sample Model 4920 Communication Control File*

```
#
# total number of ports and enable file processing
ports=5
files=y
```

```
#
# for Communication Processor Card
load=lan_mgr
load=piox
load=boottask
load=systimer
load=pc_com
run=piox
msgtask=pc_com
comtask=lan_mgr
task0=em311
driver0=async
task1=em315
driver1=bisync
task2=padptask
driver2=adccp
task3=ttytask
driver3=async
task4=lan_task
driver4=npcp
task5=sadptask
driver5=sadccp
task6=em311
driver6=busasc
task7=yboot
driver7=async
#
#      host port (PC using CPC)
#
port=0
device=mdl1
protocol=6
#
#      HHC port 1 (NPCP)
#
port=1
minor=A
```

```
protocol=4
chans=3
#
#       HHC port 2 (ADCCP)
#
port=2
minor=B
protocol=2
#       ADCCP commands
chans=2
speed=9600
minaddr=1
maxaddr=64
#
#       HHC port 3 (TTY)
#
port=3
minor=C
protocol=3
#       TTY commands
single=n
speed=9600
databits=8
parity=0
stopbits=1
#
#       HHCport 4 (TTY/ADCCP)
#
port=4
minor=D
#       primary and inactive but ready protocol
protocol=2
#       ADCCP commands
chans=2
speed=9600
minaddr=1
maxaddr=64
```

```
modemtype=6
#
# command strings for Hayes-compatible modems only
reset=AT&F
config=ATE0VOX0&C1&M1&D2&R1S0=1
#
# secondary protocol
protocol=3
# TTY commands
single=n
speed=9600
databits=8
parity=0
stopbits=1
modemtype=6
#
# command strings for Hayes-compatible modems only
reset=AT&F
config=ATE0VOX0&C1&M0&R1S0=1
```

---

### *Sample Model 4921 Communication Control File*

```
#
# total number of ports and enable file processing
ports=4
#
# host port
# (serial communication to Model 4980)
#
port=0
device=com1
protocol=1
speed=9600
databits=8
parity=0
stopbits=1
```

```
#
#   HHC port 1 (NPCP)
port=1
minor=A
protocol=4
chans=3
#
#   HHC port 2 (ADCCP)
#
port=2
minor=C
protocol=2
chans=2
speed=9600
minaddr=1
maxaddr=64
#
#   HHC port 3 (TTY/ADCCP)
#
port=3
minor=D
#
#   primary and inactive but ready protocol
protocol=3
speed=9600
single=n
databits=8
parity=0
stopbits=1
modemtype=6
#
#   command strings for Hayes-compatible modems only
reset=AT&F
config=ATE0V0X0&C1&M0&R1S0=1
#
#   secondary protocol
protocol=2
```

```

chans=2
speed=9600
mi naddr=1
maxaddr=64
modemtype=6
#
# command strings for Hayes-compatible modems only
reset=AT&F
confi g=ATE0VOX0&C1&M1&D2&R1S0=1

```

---

## *Using Communication Control File Menu*

The "Edit the communication control file" program's menu consists of submenus that provide an easy way for you to edit commands used for communication between the host and HHCs. The menus' parameters are equivalent to commands in the communication control file.

To retrieve the communication control file menu, select "Edit control files" on the top-level Main Menu, and then select "Communication control file." Or, type "editcomm" at the system prompt and press <Enter>. Upon entering the menu, you can select a task, seek help, or exit the program.

### **EXAMPLE: Edit the communication control file**

```

<version> <date>
Copyright (c) Norand Corporation 1990
Host port, bus, mpl d1
Port A NPCP, 3 channels
Port A di sabled
Port B di sabled
Port B di sabled
Port C di sabled
Port C di sabled
Port D di sabled
Port D di sabled

```

### System control

Save Restore Help Exit (ESC)

<message line>

### Definitions

**Host Port** or **Port <letter>**: This task gives you access to a series of menus that let you edit port and protocol commands.

**System Control**: This task's menu lets you enable or disable file processing on 498X devices.

### Procedure

Select, on the "Edit the communication control file" menu, the port whose commands you want to edit and press <Enter>. Then select the parameter whose default value you want to change and choose its new value.

If you make any changes on a menu, remember to select "Save" and press <Enter>. "Save" again before leaving the "Edit the communication control file" menu program. This last action places the new default values into the communication control file.





# Section 10

## Understanding the System Control File

.....

The system control file (SYSTEM.CTL) contains most of the default settings used by the Models 4920 and 4921 Series of PC Telecommunication Packages by Norand. The file:

- sets the method for ending communication
- gives the location and length of hand-held computer (HHC) identification and software version information in the first upload record
- enables or disables initial download requests
- points to the name and location of boot, load, upload, and download directories
- shows how to format upload and download data
- identifies delimiters for the end of records and PL/N data and the beginning of comments
- determines PC display colors
- determines number of autocall retries and length of waits between retry attempts
- disables help messages during unattended operation or similar conditions
- new parameters under miscellaneous files

The Models 4920 and 4921 Series of PC Telecommunication Packages uses this information to determine how it should respond to a variety of conditions and define system defaults. If you edit the system control file, you can skip a parameter. So, you need to know which defaults the system inserts in the absence of a parameter.

There are two ways you can edit the system control file.

- " Use a straight or flat ASCII text editor. Refer to its reference manual for assistance
- " Use the on-line file editor, EDITSYS.EXE, provided by Norand. This gives you interactive access to SYSTEM.CTL. It can be entered from the command line or it can be reached through the menu program. The menu is an option on the Main Menu, and is explained in Section 4, "Preparing to Use the System" in Volume A of the *Models 4920 and 4921 Series of PC Telecommunication Packages User's Guide NPN: 961-021-011*.

Commands are not case-sensitive. You can make entries in any combination of upper and lowercase letters.

The explanation which follows describes each parameter in the system control file. Each explanation includes a "default value," which is the value the system control file inserts when you provide none on the upload, download, and "Controls" menus. For *one time* or *occasional variations* to download format, upload format, or communication with the HHCs, you can temporarily override the settings in the system control file by:

- " *altering the settings* the upload, download, and communications menus present
- " using appropriate *command line switches* after the program name at the operating system prompt
- " placing *switches* after the "program =" command in the MENU.CTL file (because this applies only to certain situations, see Section 11, "Understanding the Menu Control File," for details).

Examples have a descriptive comment to clarify their meaning.

---

## Ending Communication Parameters

Four parameters are available for controlling the communication end:

- " end time
- " end date
- " end requests

" end terminals

Two of these, end time and end date, work in tandem to end communication at a specific time on a specific day. They form the first method.

The other two, end requests and end terminals, provide similar but mutually exclusive methods of ending communication based on the number of completed HHC exchanges.

A communication session does not end without external intervention (for example, you interrupt it), or until one of the ending conditions is met. If none is met, communication continues indefinitely.

Both methods and the two parameters which affect them are described in the following paragraphs.

---

## *Ending Communication Methods*

The time and count methods for ending communication are *not* mutually exclusive. You can combine the time method with either of the count methods. Communication ends as soon as one condition is met.

However, the HHC exchange counting methods are mutually exclusive. If you choose the HHC count based on requests, you cannot choose the arbitrary number. If you choose the arbitrary number, you cannot use the HHC count.

The Models 4920 and 4921 Series of PC Telecommunication Packages always ensures the integrity of exchanges in progress. It does not end communication (unless you so direct) until all active exchanges are complete and all channels are inactive.

### *Time*

You can specify a time of day at which communication should end. In addition, by creating a number that represents an offset from the current day, you can also specify a day on which communication should end.

You can end communication during the current day without specifying an ending day. However, you cannot specify an end day without an end time.

### *HHC Count*

These methods end communication based on the number of successful exchanges. When the expected or the preset number of exchanges is complete, communication ends.

### *HHC Requests*

Before communication begins, the Models 4920 and 4921 Series of PC Telecommunication Packages counts the requests for download. In the batch mode, the session control file counts the number of requested downloads. In the interactive mode, a control program counts the requested downloads you make at the PC keyboard.

### *Predetermined Number*

During communication the program counts successful exchanges and compares the count to the “total count.” When successful exchanges equal the “total count,” communication ends.

You can create a session end by assigning a number to a parameter. The number may range from zero to 65,536. The session will end when the number of HHCs serviced equals the number specified in the parameter.

---

## *HHC Counting Method Parameters*

For additional control over scheduling, you have access to two parameters that can change the rules for counting individual HHC exchanges.

First, some HHCs may be downloading for the first time, an activity called “initial download,” or IDL; you may or may not want to include these HHCs in the session total. The default setting for the associated parameter is “n” (no), which means that the number of HHCs requesting IDL service is not included in the total.

Second, some HHCs receive service in an “unscheduled” manner (HHCs that uploaded but did not have *download data* scheduled for them. Routes will be marked with an “\*” on the STATUS screen.); you may or may not want to include these unscheduled HHCs in the session total. The default setting for the related parameter is “n” (no),

which means the number of unscheduled HHCs is not included in the count.

**EXAMPLE I:** Assume you are using the time and HHC request parameters. You have set date and time to end communication on Thursday at 10:39 PM. The HHC requests total is 48. It is 2:30 Thursday afternoon. The session count has reached 46 and the Models 4920 and 4921 Series of PC Telecommunication Packages is unattended and servicing three more HHCs.

When the next two exchanges end, the HHC count increases to 48. The system compares its expected count (requests) to successes and determines that a criteria for ending communication has been met. Communication ends as soon as the remaining exchange is completed. Since the exchange count criteria was met first, communication ended without regard for the ending time.

**EXAMPLE II:** Changing only the number of completed exchanges and the time on Thursday. Assume instead, 46 HHCs have been serviced and it is 10:45 PM. Since the ending time has passed, communication would have ended, despite the successful exchanges being fewer than expected.

Each parameter is explained in more detail in the following paragraphs.

---

### *Communication End Parameters*

The following paragraphs explain the individual parameters that affect the end of communication with the HHCs. Endtime, Enddays, Endreq, Endterms, Endidl, and Endunsched.

#### *Endtime*

Session end time.

Time at which the communication session is forced to end.

Length/type: HH:MM — two-digit hour value, two-digit minute value.

Valid values: any valid time in HH:MM 24-hour clock format. HH must be 00 through 23; MM must be 00 through 59.

Midnight's minute (00:00) is not a valid end time, since it is used to disable ending communication based on time.

Default: 00:00. Disables communication window based on time.

Example: endtime = 22:40

End time is 10:40 PM.

### *Enddays*

End days.

A number that specifies the day on which the end time parameter takes effect. The number represents a day-offset from the current day.

When cutoff time is disabled (00:00) this parameter has no effect.

Length/type: one-digit numeric.

Valid values: 0 through 7.

Default: 0 (current day).

Example: enddays = 1

Communication should end one day after it begins at end time.

### *Endreq*

Use request count.

Determines whether communication ends when the number of successful HHC exchanges equals the number scheduled.

Length/type: one-letter flag.

Valid values: y (yes) or n (no).

Default: n (no).

Example: endreq = y

End communication when expected count is reached.

### *Endterms*

End terminals count.

Ends a communication session when the number of successful HHC exchanges equals the value stored in this parameter. Endterms affects communication only when end request is turned off (endreq = n).

Length/type: one- to five-digits numeric.

Valid values: 1 through 65,536 and 0, where 0 disables this method of ending communication.

Default: 0.

Example: endterms = 315

End communication after 315 successes.

### *Endidl*

Include IDL terminals.

Determines whether HHCs requesting IDL service should be included in the total for the session end.

Length/type: one-letter flag.

Valid values: y (yes) or n (no).

Default: n (no).

Example: endidl = y

Include IDL exchanges in end count.

### *Endunsched*

End unscheduled HHCs.

Determines whether unscheduled HHCs should be included in the "total count."

Length/type: one-letter flag.

Valid values: y (yes) or n (no).

Default: n (no).

Example: endunsched = y

Include unscheduled exchanges in end count.

### *Tcomretries*

Telecommunication retries.

Determines whether the system should attempt a self-restart when a fatal system error occurs.

Length/type: one to two digits.

Valid values: 1 through 99 and 0, where 0 disables this feature.

Default: 0.

Example: tcomretries = 10

Tcom will try to restart 10 times before aborting.

---

## *Terminal Identifying Parameters*

Terminal identifying parameters tell the Models 4920 and 4921 Series of PC Telecommunication Packages where to find information in the first upload record, how to respond to requests for initial download (IDL), and whether to update HHC application software when it is not current.

The first two parameters, *termidpos* and *termidlen*, tell the system which character begins the terminal identifier and how long it is.

The next two tell whether to honor requests for initial download and which position in the upload record contains the initial download request flag.



The next three parameters tell the system whether to update the HHC's application software if the version is not the most recent, where the version identifier field begins, and how many positions it occupies.

The following paragraphs describe each parameter in more detail.

### *Termidpos*

Terminal identifier position.

Position in the first upload record for the first character of the HHC identifier field.

Length/type: one- to three-digits numeric.

Valid values: 1 through 256.

Default: 33.

Example: termidpos = 36

Terminal identifier begins in position 36.

### *Termidlen*

Terminal identifier length.

Number of digits in the HHC identifier field in the first upload record.

Length/type: one- to two-digits numeric.

Valid values: 1 through 16.

Default: 6.

Example: termidlen = 5

Terminal identifier is 5 characters long.

### *IDLCheck*

Initial download check.

Determines whether HHC requests for initial download are fulfilled.

Length/type: one-letter flag.  
Valid values: y (yes) or n (no).  
Default: n (no).  
Example: idlcheck = y  
Allow initial download.

### *IDLPos*

Initial download flag position.

Position in the first upload record for the flag that an HHC uses to request initial download processing.

Length/type: one- to three-digit numeric.  
Valid values: 1 through 256.  
Default: 60.  
Example: idlpos = 34  
Initial download flag begins in position 34.

### *VersCheck*

Version check.

Determines whether HHC application programs should be updated based on comparison of version numbers.

Length/type: one-letter flag.  
Valid values: y (yes) or n (no).  
Default: n (no).  
Example: verscheck = y  
Update old software versions.

### *VersPos*

Version position.

Position in the first upload record for the first character of the version-identifier field.

Length/type: one- to three-digit numeric.

Valid values: 1 through 256.

Default: 28.

Example: verspos = 24

Version identifier begins in position 24.

### *VersLen*

Version length.

Number of characters in the version-identifier field of the first upload record from each HHC.

Length/type: one- to two-digits numeric.

Valid values: 1 through 16.

Default: 4.

Example: verslen = 3

Version identifier occupies 3 positions.

---

## *Data Format Parameters*

Upload format parameters tell the Models 4920 and 4921 Series of PC Telecommunication Packages whether upload processing runs concurrently with communication and how to format upload data.

Download format parameters tell the Models 4920 and 4921 Series of PC Telecommunication Packages in what format to expect files created by another system.

### *Uplconcur*

Upload concurrently.

Determines whether upload formatting runs concurrently with communication or after communication has completed.

Length/type: one-letter flag.  
Valid values: y (yes) or n (no).  
Default: n (no).  
Example: uplconcur = n  
Upload processing is not concurrent.

### *Uplfixed*

Upload fixed.

Determines whether upload data should be formatted into fixed-length records.

Length/type: one-letter flag.  
Valid values: y (yes) or n (no).  
Default: n (no).  
Example: uplfixed = y  
Format upload into fixed-length records.

### *Uplreclen*

Upload record length.

The meaning of upload record length depends on the setting of the preceding parameter, upload fixed (uplfixed).

When record length is *fixed* (uplfixed = y), all records will have the number of characters specified in this parameter. The software pads spaces onto the end of records to make them the proper length.

When upload record length is *not fixed* (uplfixed = n), this parameter sets the maximum number of characters in a record.

Upload record length has a special meaning when set to zero. A zero indicates *no maximum* record length.

Length/type: one- to three-digit numeric.

Valid values: 1 through 256 and 0, where 0 indicates no maximum.

Default: 0.

Example: uplfixed = y  
uplreclen = 128

Make all records 128 characters long.

### *Upllogical*

Upload logical records.

Determines whether upload data is formatted into logical records. Y (yes) places one logical record per line. N (no) places the number of characters specified in upload record length on each line.

Length/type: one-letter flag.

Valid values: y (yes) or n (no).

Default: n(no).

Example: upllogical = y

Format upload data into logical records.

### *Uplincbad*

Upload include bad.

Determines whether upload data should be processed when the upload phase of the HHC exchange is marked "bad" during communication.

Length/type: one-letter flag.

Valid values: y (yes) or n (no).

Default: n (no).

Example: uplincbad = y

Process upload data for HHC exchanges marked "bad" during the upload phase.

### *Uplincbadsess*

Upload include bad sessions.

Determines whether upload data should be processed when the upload phase of the HHC computer session is marked "good" and the download phase is marked "bad."

Length/type: one-letter flag.

Valid values: y (yes) or n (no)

Default: n (no).

Example: uplincbadsess = y

Process upload data for HHC exchanges when the upload phase is marked "good" and the download phase is marked "bad."

### *Uplincdup*

Upload include duplicates.

Determines whether all upload data from routes marked as duplicate will be formatted.

Length/type: one-letter flag.

Valid values: y (yes) or n (no).

Default: n (no).

Example: uplincdup = y

Process duplicate uploads from same HHC.

### *Upldelimit*

Upload delimit.

Determines whether the end of upload records should be marked with the "dataeof" delimiter (specified later).

Length/type: one-letter flag.  
Valid values: y (yes) or n (no).  
Default: n (no).  
Example: upldelimit = y  
Mark end of records with delimiter.

### *Uplcrlf*

Upload includes carriage return and line feed.

Length/type: one-letter flag.  
Valid values: y (yes) or n (no)  
Default: n (no).  
Example: uplcrlf = y  
Process upload data to add carriage return and line feed to fixed length records.

### *Uploadfname*

Upload include file names.

Determines whether all upload data should include file names.

Length/type: maximum of 260 ASCII characters  
Valid values: standard DOS-naming conventions  
Default: upload.dat.  
Example: uploadfname = upload.dat  
Process the file: upload.dat.

### *Uploadback*

Upload backup.

Determines whether to make a backup of the upload data file.

Length/type: one-letter flag.  
Valid values: y (yes) or n (no).  
Default: n (no).  
Example: uploadback= y  
Backup upload files.

### *DEXfname*

DEX audit trail file.

This file separates the DEX audit trail out of the formatted upload file. All DEX data are placed in this file.

Length/type: maximum 260 ASCII characters  
Valid values: standard DOS-naming conventions  
Default: none.  
Example: dexfname= **UPLOAD\DEXFILE.DEX**

### *Dnlfixed*

Download fixed.

Tells the Models 4920 and 4921 Series of PC Telecommunication Packages whether download data comes in fixed-length records.

Length/type: one-letter flag.  
Valid values: y (yes) or n (no).  
Default: n (no).  
Example: dnlfixed = y  
Download records are fixed-length.

### *Dnlreclen*

Download record length.

This parameter is necessary only when "dnlfixed" has been set to y (yes). It defines how many characters each fixed-length record contains.



Length/type: one- to three-digits numeric.

Valid values: 0 through 256.

Default: 0.

Example: `dnlreclen = 80`

Fixed-length records contain 80 characters each.

### *Dnltrunc*

Download truncated.

Determines whether trailing spaces and the record delimiter should be truncated from each line of the download file.

Length/type: one-letter flag.

Valid values: y (yes) or n (no).

Default: n (no).

Example: `dnltrunc = y`

Truncate trailing spaces and the record delimiter.

### *Dnlpack*

Download packed.

Determines whether download data should be packed. You can use packed data *only* if the PL/N application program for the HHCs supports it.

Length/type: one-letter flag.

Valid values: y (yes) or n (no).

Default: n (no).

Example: `dnlpack = y`

Pack download data.

---

## System-Wide Display Options

The display options which follow set the colors for various screens. These colors are of interest primarily for color monitors since monochrome monitors use only two colors; white (7) and black (0). The other hexadecimal codes have different meanings for monochrome monitors. Refer to Table 10-1 when selecting colors.

Table 10-1  
Foreground Colors

Color	Hex Value	Color	Hex Value
Black	0	Dark Grey	8
Blue	1	Light Blue	9
Green	2	Light Green	A
Cyan	3	Light Cyan	B
Red	4	Light Red	C
Magenta	5	Light Magenta	D
Brown	6	Yellow	E
White	7	Bright White	F

" NOTE: Background Colors are the same as foreground values 0-7.

The left hexadecimal digit sets the background color and the right digit sets the foreground color.

EXAMPLE: A main color of 70 is a white (7) background with a black (0) foreground.

### *Maincolor*

Main color.

The main colors of the PC display.

Length/type: Two-digit hexadecimal. The left digit sets background color and the right digit sets foreground color.

Valid values: 00-FF. Avoid setting the same color for background and foreground, since this makes the screen unreadable.

Default: 07 (black background, white letters).

Example: maincolor = 1F

Set main color to blue background with bright white letters.

### *Alt1color*

Alternate one color.

The colors of the first window level.

Length/type: Two-digit hexadecimal. The left digit sets background color and the right digit sets foreground color.

Valid values: 00-FF. Avoid setting the same color for background and foreground, since this makes the screen unreadable.

Default: 70 (white background, black letters).

Example: alt1color = 4F

First alternate color is red background with bright white letters.

### *Alt2color*

Alternate two color.

The colors of the second window level.

Length/type: Two-character hexadecimal. The left digit sets background color and the right digit sets foreground color.

Valid values: 00-FF. Avoid setting the same color for background and foreground, since this makes the screen unreadable.

Default: 70 (white background, black letters).

Example: alt2color = 74

Second alternate color is white background with red letters.

### *Errorcolor*

Error color.

The colors used for error message windows on the display.

Length/type: Two-character hexadecimal. The left digit sets background color and the right digit sets foreground color.

Valid values: 00-FF. Avoid setting the same color for background and foreground, since this makes the screen unreadable.

Default: 07 (black background, white letters).

Example: errorcolor = 5F

Set error color to magenta background with bright white letters.

### *Helpcolor*

Help color.

The colors used for help windows.

Length/type: Two-character hexadecimal. The left digit sets background color and the right digit sets foreground color.

Valid values: 00-FF. Avoid setting the same color for background and foreground, since this makes the screen unreadable.

Default: 70 (white background, black letters).

Example: helpcolor = 2F

Set help color to green background with bright white letters.

## *Help*

Help.

Indicates whether on-line help is available. When the help screens are not available or needed, set to "N" to suppress error messages indicating on-line help is not available.

### **EXAMPLE:**

You might use this for unattended PC operation, since no one is present to read the messages.

Length/type: one-letter flag.

Valid values: y (yes) or n (no).

Default: n (no).

Example: help = y

On-line help screens are available.

---

## *File Directories*

These parameters determine the defaults for:

- " boot file directory
- " load file directory
- " upload file directory
- " download file directory

File directories tell the Models 4920 and 4921 Series of PC Telecommunication Packages which directories contain files it must download and into which directory to put files uploaded from the HHCs. If a drive letter is not specified in the full path name, the current drive is assumed.

### *Bootdir*

Boot directory name.

The full path name of the subdirectory that holds the boot files for 4000 Series HHCs.

This parameter is necessary for Model 4920 only, since the same function is fulfilled by the boot disk in a Model 4980 Communication Controller for the Model 4921 PC Telecommunication Package.

Length/type: 80 alphanumeric characters.

Valid values: any valid file name characters.

Default: current directory.

Example: bootdir = C: \4920\boot\

“Boot” directory is a subdirectory of “4920” on drive C.

### *Loaddir*

Load directory name.

The full path name of the subdirectory that holds programs for the PC’s communication processor board.

This parameter is necessary for Model 4920 only, since the same function is fulfilled by the boot disk in a Model 4980 Communication Controller for the Model 4921 PC Telecommunication Package.

Length/type: 80 alphanumeric characters.

Valid values: any valid file name characters.

Default: current directory.

Example: loaddir = C: \4920\load\

“Load” directory is a subdirectory of “4920” on drive C.

### *Upldir*

Upload directory name.

The full path name of the subdirectory that holds the formatted upload files. Output files from the "Format upload data by HHC" program are placed here. Used only when "UPLFMT.EXE" is used to format the upload data.

Length/type: 80 alphanumeric characters.

Valid values: any valid file name characters.

Default: current directory.

Example: `upl dir = C: \4920\upl oad\`

Put HHC files in "upload," a subdirectory of "4920," on drive C.

### *Dnldir*

Download directory name.

The full path name of the subdirectory that holds download request files. The path name becomes prefixed to all request file names.

Length/type: 80 alphanumeric characters.

Valid values: any valid file name characters.

Default: current directory.

Example: `dnldir = C: \4920\downl oad\`

Put down-load request in "download" subdirectory of "4920" on drive C.

---

## *Miscellaneous Parameters*

Data format options tell the Models 4920 and 4921 Series of PC Telecommunication Packages the end-of-record character for data, end-of-file character for PL/N file, character to delimit the beginning of remarks in data, and the character to delimit the end of files in data.

Other commands determine the usual:

- " amount of debug information to report, and
- " amount of information sent to the LOG.DAT file.

The RESTART command governs whether counts of HHC exchanges accumulate from one PC-to-HHC communication session to the next.

The BACKGROUND command controls whether multitasking operating systems can run programs for the Model 4920 Series of PC Telecommunication Packages in the background.

The following paragraphs explain each command in more detail.

### *Dataeor*

Data end of record.

Denotes the character which marks the end of data records.

Length/type: one-character delimiter.

Valid values: any printable ASCII character.

Default: ! (exclamation point).

Example: dataeor = !

Use "!" to mark end of records.

### *Datarem*

Data remark.

Denotes the character which marks the beginning of a remark line in a data file. Control files (such as SYSTEM.CTL and COMM.CTL) always use "#" (pound sign) to begin comment records.

Length/type: one-character delimiter.

Valid values: any printable ASCII character.

Default: # (pound or crosshatch).

Example: datarem = #

Use the pound sign to begin remark lines.

### *Dataeof*

Data end of file.



Denotes the characters marking the end of data files. The Models 4920 and 4921 Series of PC Telecommunication Packages discards this character when it appears as the last character in a data file.

Length/type: two-digit data delimiter.

Valid values: any decimal ASCII digits.

Default: (none).

Example: dataeof = 26

Use 26 (ASCII control Z) to mark end of data files.

### *Plneof*

PL/N end of file.

Denotes the character which marks the end of PL/N files. The software marks the ends of variable length PL/N files with two of this end-of-file character for PL/N files.

Length/type: one-character delimiter.

Valid values: any printable ASCII character.

Default: ~ (tilde).

Example: plneof = ~

Use two tildes to mark the end of PL/N data files.

### *Debuglevel*

Debug level.

Determines the amount of diagnostic information displayed during communication. The higher the number, the more information.

Values greater than zero should be used only to troubleshoot the system. Displaying large amounts of diagnostic information during communication degrades system performance.

Length/type: one-digit numeric.  
Valid values: 0 through 3.  
Default: 0.  
Example: debuglevel = 2  
Set debug one step below maximum for diagnosis.

### *Loglevel*

Log level.

Detail of information sent to the communication log file (LOG.DAT). The higher the number, the more information recorded. See Appendix A. "Interpreting LOG.DAT Information" in the *Models 4920 and 4921 Series of PC Telecommunication Packages Reference Guide NPN: 961-021-013* for details.

Length/type: one-digit numeric.  
Valid values: 0 through 3.  
Default: 0.  
Example: loglevel = 2  
Save an intermediate level of detail to the LOG.DAT file.

### *Background*

Tells whether programs can be run in the background.

Used by the menu program. Setting to "yes" creates a new process to run the selected program in. Use this program only under multi-tasking operating systems such as OS/2.

Length/type: one-letter flag.  
Valid values: y (yes) or n (no).  
Default: n (no).  
Example: background = n  
Do not create a second process to allow multitasking.

### *Restart*

Restart communication.

Determines whether the next communication session should be treated as new or a resumption of the previous session. A restarted session retains previous status flags.

Setting restart to "yes" makes each session appear to be a resumption of the previous one.

Length/type: one-character logical flag.

Valid values: y (yes) or n (no).

Default: n (no).

Example: restart = n

Treat each communication session as a new one.

### *Tcomquery*

Telecommunications query.

Queries the operator as to whether TCOM should be started.

Length/type: one-character logical flag.

Valid values: y (yes) or n (no).

Default: n (no).

Example: tcomquery = n

---

## *Default Phone Parameters*

These parameters control the number of:

- " times autodialed phone calls to each number are retried
- " minutes to wait between attempts to autodial a phone number

### *Phoneretries*

Phone retries.

Sets the system-wide default for the number of times autodialed phone numbers are redialed.

Length/type: one- or two-digit numeric.  
Valid values: 0 through 99.  
Default: 0.  
Example: phoneretries = 6  
Each autodialed phone number is retried six times.

### *Phonewait*

Phone try wait.

Sets the system-wide default for the number of minutes waited between attempts to autodial a phone number.

Length/type: one- or three-digit numeric.  
Valid values: 0 through 1440.  
Default: 0 (no wait).  
Example: phonewait = 2  
Wait two minutes between attempts to autodial a number.

### *Nohalt*

This parameter tells the system whether to stop the current program when an error occurs (the default is to halt the program) or to continue on.

Sets the system-wide default.

Length/type: one-letter flag.  
Valid values: y (yes) or n (no).  
Default: n (no).  
Example: nohalt = n  
Continue normal operation, halt and display the message.

" **NOTE:** *The logic for this parameter is negative because the default condition is to STOP and display the error message. To avoid halting the program, the parameter needs to be set to "Y".*

### *Errlog*

The Errlog parameter allows the creation of a log file to hold any system error messages. This is useful with the NOHALT parameter when creating an unattended system.

Length/type: maximum of 260 ASCII characters.

Valid values: standard DOS-naming conventions

Default: errlog.err

Example: errlog = errlog.err

Run the error logs.

### *Netflag*

The NETWORK STOP FLAG provides for a convenient method to halt multiple 492X systems that share a common network drive.

Length/type: maximum of 260 ASCII characters

Valid values: standard DOS-naming conventions

Default: ncp4920.stp

Example: netflag = ncp4920.stp

" **NOTE:**

*A plus (+) sign or no entry informs the system to terminate when the stop flag file comes to exist. A minus (-) sign tells the system to terminate when the stop flag file disappears.*

### *Netchkcnt*

Network check count tells the system how often to look for the NETWORK STOP FLAG file.

Length/type: one- or three-digit numeric.

Valid values: 0 through 999.

Default: 0.

Example: netchkcnt = 2

Check the network every two minutes.

---

### Sample System Control File

```
#           Scheduling Parameters
endtime=12:00
enddays=1
endreq=y
endterms=0
endi dl =n
endunsched=n
tcomreties=10
#           Terminal Identifying Parameters
termi dpos=33
termi dl en=6
i dl check=y
i dl pos=60
verscheck=y
verspos=40
versl en=4
#           Data Format Options
upl concur=n
upl fi xed=n
upl recl en=0
upl l ogi cal =y
upl i ncbad=n
upl i ncbadsess=n
upl i ncdup=n
upl del i mi t=y
upl crl f=n
upl oadfname=upl oad. dat
```

```
upl oadback=y
tcomretries=10
dexfname=UPLOAD\DEXFILE.DEX
dnl fixed=n
dnl reclen=0
dnl trunc=y
dnl pack=n
#           System-wide Display Options
maincolor=1F
alt1color=4F
alt2color=74
errorcolor=5F
helpcolor=2F
mouse=n
help=y
#           File Directories
bootdir=c:\4920\boot\
loaddir=c:\4920\load\
upl dir=c:\4920\upload\
dnl dir=c:\4920\download\
#           Miscellaneous Parameters
dataeor=!
datarem=#
dataeof=26
plneof=~
debuglevel=0
loglevel=0
background=n
restart=n
tcomquery=n
#           Default Phone Parameters
phoneretries=99
phonewait=99
nohalt=n
errlog=errlog.err
netflag=ncp4920.stp
netchkcnt=2
```

---

## Using System Control File Menu

The "Edit the system control file" program consists of submenus that provide an easy way for you to change system-wide default values. The menus' parameters are equivalent to the commands in the system control file.

To retrieve the "Edit the system control file" menu, select "Edit control files" on the top-level Main Menu, and then select "System control file." Or, type "editsys" at the system prompt. Upon entering the menu, you can select a task, seek help, or exit the program.

### EXAMPLE:

**Edit the system control file**

<Version> <date>

Copyright (c) Norand Corporation 1990

Session ending

Terminal identification

Download data format

Upload data format

Data format

Screen colors

Miscellaneous

Save Restore Help Exit (ESC)

<message line>

### Definitions

**Session Ending:** This menu lets you control the end of communication. You can choose to end communication based on the time and data or on the number of completed HHC exchanges. Two additional parameters determine whether or not to include initial download and unscheduled terminals in the end count.

**Terminal Identification:** Terminal identifying parameters tell the Models 4920 and 4921 Series of PC Telecommunication Packages where to find information in the first upload record, field lengths, how to respond to requests for initial download, and whether or not to update HHC application software when it is not current.



**Data Format:** The parameters for this menu denote the characters which mark the end of records and files and the beginning of data file remark lines.

**Display-wide Options:** This menu lets you change the foreground and background colors of various screens and windows. Two additional parameters determine whether or not to enable a mouse (if present) and the help screens.

**File Directories:** This lets you enter directory path names.

**Miscellaneous:** The parameters for this menu let you set the diagnostic and log levels, and determine whether or not to run programs in the background.

**Default Phone:** These parameters let you set the number of times autodialed phone numbers should be retried, and the number of minutes between autodial retries.

### *Procedure*

Select the task you want to perform and press <Enter> to retrieve its menu. Then select the parameter whose default value you want to change and type its new value. If you make any changes on a menu, remember to select "Save" and press <Enter>, and "Save" again before leaving the "Edit the system control file" menu program. This last action places the new default values into the system control file.



# Section 11

## Understanding the Menu Control File

.....

The menu control file (MENU.CTL) lets you customize the Models 4920 and 4921 Series of PC Telecommunication Packages menus so you can tailor the system to the special needs of your business. The following are a few ways you can change the system:

- change a menu name
- change the wording of a current prompt
- remove a choice you do not use
- change the wording of a message line
- add a choice for your host communication package

Make these changes by directly changing the menu control file with a straight or flat ASCII text editor.

This section explains how the menu control file works and how you might change it. Turn to the sample file at the end of this section if you want a preview of the menu control file's format.

With knowledge of the commands, an ASCII text editor, and knowledge of your company's special needs, you can tailor the Main Menu and subordinate menus of the Models 4920 and 4921 Series of PC Telecommunication Packages to your business environment.

## Commands

These seven commands make up the menu control file:

- " menu
- " name
- " label
- " program
- " entry
- " prompt
- " submenu

Each command appears on its own line followed by an equal sign, as shown in this example.

### **entry=Edit control files**

In this case, "entry" is the command name and "Edit control files" is the literal text string the program uses.

Each menu begins with the commands *menu* and *name*. Then *label*, *entry*, *prompt*, and *submenu* or *program* follow for each option on the menu. The next *menu* entry or the end of the file indicates the completion of options for a menu.

A more thorough definition of each command follows.

## *Menu*

**Menu number.** This command marks the beginning of information for a single menu. The next menu command also marks the end of the preceding menu's commands.

The menu control file must, at a minimum, contain *menu=1* to denote the Main Menu. Menu numbers *need not* appear in order within the menu control file.

Length/type: one-digit numeric character.

Valid values: 1 to 9. Must have a number 1.

Example: menu = 2

Indicates the end of commands for the previous menu and the beginning of a new menu.

### *Name*

Title appearing on a screen.

Length/type: up to 80 alphanumeric characters.

Valid values: any printable ASCII character.

Example: name = Main menu

"Main menu" appears in the title area of the menu.

### *Label*

A shorthand "tag" for each choice, so you can run its associated program from the command line or within a batch file, bypassing the menu.

Length/type: one- or two-digit alphanumeric character.

Valid values: any printable ASCII character except "/" and "-".  
These characters denote switches in command line instructions.

Example: label = e

You can call the associated program or submenu from the command line by typing "menu e."

### *Entry*

The wording for the menu option on the screen.

Length/type: up to 80 alphanumeric characters.

Valid values: any printable ASCII character.

Example: entry = Download format

Makes the literal "Download format" a choice on the menu.

### *Prompt*

Wording for the choice's message line. The message line appears near the bottom of the screen when you highlight the option with the cursor.

Length/type: up to 80 alphanumeric characters.

Valid values: any printable ASCII characters.

Example: prompt = File where upload is stored.

"File where upload is stored." appears in the message line when you highlight the choice with the cursor.

### *Program*

The Models 4920 and 4921 Series of PC Telecommunication Packages program name linked to a menu choice.

Length/type: up to 80 alphanumeric characters.

Valid values: any ASCII characters.

Example: program = plnfmt

Calls the PL/N format program.

You may supplement a program name with a "switch" or "switches" (exceptions: editsys, editcomm, editreq, and editact programs). The switch follows the program name on the same line and specifies the condition for one or more of the program's parameters. The switches replace the values for a program's parameters when you retrieve a task's menu.

- " through the *Main Menu*

- " by typing "menu <letter>" at the command line prompt

Command line switches may follow the program name. If you *do not* put a switch after the "program = <name>" command, the *SYS TEM.CTL* file inserts its default values for the menus' parameters and the "restart" option.

#### " NOTE:

*Look ahead to the sample file at the end of this section. Notice that the "program = <name>" command for the group beginning with "label = s" (session control format) is supplemented with switches. In this case, the fixed-length records, pack, and truncate switches have been set to "no." Therefore, each time you retrieve the "Format the session control file" menu through the Main Menu or by typing "menu s" at the command line, the menu's parameters will be set to "no" until you change them in the menu control file. Of course, you can manually change the values on the menu.*

**EXAMPLE:**

Notice that the fourth and fifth groups, beginning "label = c" and "label = r," utilize the same program, ncp4920.

As you may conclude from the entry and prompt information, both of these choices call the communication program. The difference is the "-y" and "-rn" switches after the program name. The "r" stands for restart, "y" means yes, and "n" means no.

For the "communicate" option the restart switch is set to "no," which means that counts of Hand-Held Computer (HHC) exchanges should not be retained from the previous communication session, but started anew. Therefore, *each time* you start the communication session through the Main Menu or type "menu c" at the command line, counts of HHC exchanges are not retained from the previous communication session.

For the "Restart communication" option the switch is set to "yes," which means the HHC count is resumed with totals from the last session assumed as beginning counts for the upcoming communication session. Therefore, *each time* you restart the communication session through the Main Menu or type "menu r" at the command line, counts of HHC exchanges are resumed.

Length/type: single character delimiter followed by a single character indicating condition.

Valid values: "-" (hyphen) or "/" (backslash).

Example 1: program = **pl nfmt -pn**

Call the PL/N format program, but do not pack data.

Example 2: program = **pl nfmt /py**

Call the PL/N format program and pack data.

You can place more than one switch after the program name if needed. Simply leave a space before each switch. See the sample menu control file for assistance.

### *Submenu*

A subordinate menu for the Models 4920 and 4921 Series of Telecommunication Packages to call when you select the menu choice.

Length/type: one-digit numeric character.

Valid values: any ASCII numeral.

Example: submenu = 2

Calls the menu named "menu = 2" when you select the choice.

---

## Menu Definition

The first two components of each menu within the menu control file are the commands "menu" and "name." The "menu" command assigns a number to the menu and marks its beginning in the file. The next "menu" command marks the end of the contents for the preceding menu. As mentioned previously, the menu control file must always have a "menu = 1" that denotes the Main Menu.

The "name" command provides a title which appears in the title area of the menu. Consider the following commands.

**menu=1**

**name=Main menu**

"Menu = 1" marks the beginning of a menu titled "Main menu." All commands for individual menu options that appear after these commands, but before the next "menu" command, become part of the first menu.

---

## Menu Option Commands Order

The position of a command within the menu control file is important for two reasons. Its position within a set of commands for a single menu choice must conform to a prescribed order. In addition, the position of sets of menu option commands relative to each other affects where the option appears on the menu.

Consider the two examples which follow. The first example shows the acceptable form for an option that calls a program. The second example shows an option that retrieves a subordinate menu.



**EXAMPLE I:** Command order for program choices  
**l**abel  
**e**ntry  
**p**rompt  
**p**rogram

**EXAMPLE II:** Command order for submenu choices  
**l**abel  
**e**ntry  
**p**rompt  
**s**ubmenu

The command set positions relative to each other affects where prompts appears on the menu.

**EXAMPLE:** The first command set prompt appearing after the "name =" command appears first on the screen, the second set in the file is the second choice on the screen.

Assume the preceding examples appeared in the menu control file in the same order. The option calling a program would appear on the menu above the option calling a subordinate menu.

---

## Type Examples

Command sets like the following examples comprise each choice in the menu control file.

**EXAMPLE I:** Calling another menu  
**l**abel = e  
**e**ntry = Edit control files  
**p**rompt = Edit the control files  
**s**ubmenu = 2

**EXAMPLE II:** Calling a program  
**l**abel = s  
**e**ntry = Session control format  
**p**rompt = Format the session control file.  
**p**rogram = sessfmt

Notice the last line of each set has a different command. The first example builds a choice which calls a subordinate menu. In contrast, the second builds a choice that calls a program. The following explains each example in more detail.

“**Label =**” in the first example names “**e**” as the shorthand notation for directly referring to this menu choice from the command line or a batch file. In this case, the letter “**e**” input on the command line calls the menu pointed to by the submenu. Elsewhere in the file, a menu numbered “**2**” defines the display which you call with this command.

The “**entry =**” command names the literal text string “Edit control files” for this choice. “**Entry**” is the line you position the cursor beside to select the choice.

The “**prompt =**” command provides the literal text string, “**Edit the control files,**” that appears in the message line when you highlight the entry with the cursor.

The “**submenu =**” command gives the number for another menu the Models 4920 and 4921 Series of PC Telecommunication Packages call when you select this choice.

---

### *Adding a Submenu*

You may want to add a submenu to the menu control file. Doing this would let you select a new choice from the Main Menu, then make a choice from a subordinate menu.

Suppose your central office is converting the host system to a new data communication program. For several months during the transition, you must be able to communicate with the host using either program.

To accomplish this, you could add an option to the Main Menu which calls a submenu listing the two communication programs. Assume that each package is called from a batch file. Further assume the new program’s batch file is called **NEWCOMM BAT** and the current program’s batch file is called **HOSTCOMM BAT**.

To place the submenu choice on the Main Menu, you could place a set of commands like the following with the Main Menu commands in the menu control file.

```

label = t
entry = Transition host communication
prompt = Select a host communication method
submenu = 3

```

Where you place these commands, relative to the other menu choices, governs where the new prompt appears on the Main Menu. For example, if you place these commands at the beginning of the Main Menu portion of the menu control file, the new option appears first on the Main Menu. In contrast, if you place them at the end of the Main Menu segment, the new option appears last on the Main Menu.

Now for the new subordinate menu.

Commands like the following would comprise the submenu. They form a new menu called "Host communication" with a choice for both host communication methods. You would place these commands at the end of the menu control file.

```

menu=3
name=Host communication
#
label=o
entry=Communicate using old program
program=hostcomm
#
label=n
entry=Communicate using new program
prompt=Initiate new host communication program
program=newcomm

```

Notice that the new menu begins with menu number 3. This corresponds to the submenu called "**submenu = 3**" in the commands for the Main Menu.

To communicate using the old host program, choose "Transition host choices" from the Main Menu, then select "Communicate using the old program" from the submenu. The Models 4920 and 4921 Series of PC Telecommunication Packages would follow your instructions and initiate the "**host comm**" batch file.

On the other hand, you could communicate to the host using the new host program by selecting “Communicate using new program” from the “Host Communication menu.”

If you want to experiment with such a change, create batch files with the appropriate names that simply write messages to the screen. Then insert the commands from the preceding examples.

You now know how to tailor the Models 4920 and 4921 Series of PC Telecommunication Packages to the needs of your business. By following these instructions, you can change menu names, reword options, revise prompts in the message line, add options to the menu, and create your own subordinate menu.

Look over the sample menu control file which follows. With your new understanding, careful consideration should coalesce your understanding and help you remember what you have learned.

---

### *Sample Menu Control File*

```
#
# Main Menu
#
menu=1
name=Main menu
#
label=2
entry=Session control format
prompt=Format the session control file.
program=sessfmt /fn /tn /pn
#
label=h
entry=Host download format
prompt=Format the host download file.
program=hostdnf
#
label=d
entry=Download format
prompt=Format PL/N data for download.
```

```
program=pl nfmt
#
label=c
entry=Communicate
prompt=Start communication.
program=ncp4920 -rn
#
label=r
entry=Restart communication
prompt=Restart communication.
program=ncp4920 -ry
#
label=u
entry=Upload format
prompt=Format the upload file.
program=hostupf
#program=uplfmt
#
label=e
entry>Edit control files
prompt>Edit the control files.
submenu=2
#
label=m
entry=Maintain databases
prompt=Maintain the databases.
submenu=3
#
# edit menu
menu=2
name>Edit control files menu
#
label=es
entry=System control file
prompt>Edit the system control file
program=editsys
#
```

```
label=ec
entry=Communications control file
prompt=Edit the communications control file.
program=editcomm
#
# maintenance menu
#
menu=3
name=Maintain databases menu
#
label=mr
entry=Request database
prompt=Maintain the request database.
program=editreq
#
label=ma
entry=Activation database
prompt=Maintain the activation database.
program=editact
#
# other program
# this menu is not displayed,
# but its options can be
# selected from the command line
menu=4
name=Other programs
#
label=w
entry=Wait
prompts=Wait
program=wait
```

## Section 12

# Understanding the Session Control File

.....

The default formatting parameters for the session control file (SESSION.CTL) are stored in the system control file (SYSTEM.CTL).

ASCII characters that have a value of hexadecimal 20 (decimal 32) or lower are called “nonprinting” characters. Of principal interest in the following discussion are “space” (SP, 20 hexadecimal), “carriage return” (CR, 0D hexadecimal), “line feed” (LF, 0A hexadecimal), and “tab” (HT, 09 hexadecimal).

Characters with values ranging from 21 to 7E hexadecimal are called “printing” (nonblank) characters.

---

### Record Format

A record in the session control file consists of two fields. The first field identifies the category type the record belongs to. The second field is a data field, the contents of which vary according to the record category.

The format of the record is as follows:

**<record type><data[/C][/D][/E][!]>[spaces][\][CR/LF]**

The fields in angle brackets are mandatory. The characters in square brackets are optional.

[/C], [/D], and [/E], which are called “switches,” are defined later as they appear in various record descriptions. The exclamation mark [!] is used by *data records* only. Comments on the use of the exclamation mark are included in the description of the Data Record.

Spaces between fields and field entries are ignored. However, since parameters in data fields are delimited by spaces, do not insert spaces between a parameter and its associated value. Furthermore, if the parameter value itself is a character string that normally includes spaces, the value must then be enclosed by quotation marks (" . . ").

The maximum size of a record is 264 bytes. For convenience in printing or on-screen viewing, you may wish to break a long record into several lines, with line lengths limited to 80 columns. To continue a record onto succeeding lines, enter a backslash (\) as the last printing (nonblank) character on the preceding line. (Any nonprinting characters that appear to the right of the backslash are ignored.) However, please note that the continuation character is not used in *data records*.

The records in the session control file may be of the same length (fixed-length), or the length may vary from record to record (variable-length). If the file contains variable-length records, each record must be terminated by an ASCII carriage return (CR) and line feed (LF).

---

## Record Types

The following are the record type categories, related group names, and the associated designations:

Description	Group:	Record Type
#		Comment
M		Mode
	REQUEST	
T		Terminal identifier
N		Network communications controller identifier
S		File set identifier
L		Load boot set



Description	Group:	Record Type
	FILE AND BROADCAST	
F		File identifier
B		Broadcast file identifier
D		Data
	ACTIVATION	
P		Port configuration
A		Autoanswer activate
C		Autocall activate

### Record Descriptions

In the following descriptions, parameters are specified by a parameter name, immediately followed by an equal sign (=), immediately followed by a value (no spaces between the name, equal sign, and value). The value can be a character string, a numerical value, a logical value (true/false, yes/no), or an *HHMM* time string (for 24-hour clock time). Parameters enclosed by angle brackets are required parameters.

When parameters are defined, the text uses upper-case letters to specify the minimum number of characters that must be entered for each parameter name; the parameter names are not case-sensitive, however.

**EXAMPLE:** PORT=1, POR=1, port=1, and por=1 all work.

If a parameter value is a character string that normally includes spaces, the value must be enclosed by quotation marks (" . .").

Parameter values that are logical true/false flags are shown in parentheses and separated by a diagonal, or solidus (/), as in y (yes)/n (no), where y and n represent the logical true and false flags.

---

### Comment (#) Records

#<comment>

Use “#” records to insert comments into the session control file. All characters in the record following the # character are ignored by control programs. The line-continuation character (\) mentioned earlier does not apply to comment records.

The comment character can be redefined in the SYSTEM.CTL file.

---

### Mode (M) Records

**M NEW=li st**

**VERsi on=val ue**

**ZONe=val ue**

The mode (M) record, which should be the first record in the session control file, sets the mode to “update” or “new” for records of the T, S, P, A, and C type and for “request” types (request types 0 through 4). You specify a “new” mode by entering the record type designation (T, S, P, A, or C) and the request type designation (0 through 4) in a “NEW list.”

Request types are as follows:

- 0 = regular download
- 1 = initial download
- 2 = clock - sends PC's current date and time to the HHC (optionally adjusted for the HHC time zone)
- 3 = program - sends a new program to the HHC
- 4 = host download status - sends PC's TCOM status to the HHC
- 5 = special download files

#### *Parameters*

**NEW=li st**

The NEW parameter specifies the record and request types that should be purged from the 4920 database before SESSION.CTL is processed. The types are specified in a character string, or list.

The types can be in any order from the group **TSPNLAC01234** or **\***, where:

- NEW = \* deletes all terminals, sets, ports, autoanswer and autocal records, and request types (0-4). The "\*" indicates "global."
- NEW = T deletes all terminals and associated file requests
- NEW = S deletes all sets and associated file requests
- NEW = P deletes all ports
- NEW = N deletes all 498X devices and associated file assignments
- NEW = L deletes load sets
- NEW = A deletes all autoanswer activate records
- NEW = C deletes all autocal activate records
- NEW = <request types 0-4> deletes the request type.

The types can be combined.

**EXAMPLE:**

NEW = A0 deletes all autoanswer activate records (type A) and regular download requests (type 0). NEW=TSPNLAC01234 is equivalent to NEW=\*. Deleting T, S, , N also deletes associated requests.

Valid list: Any characters in any order from the group **TSPNLAC01234\***

Default: "Update" (for all record types).

### **ZONE**

The global time zone offset of all HHCs from the PC. If a "clock" request is scheduled for an HHC, the time zone offset is used to set the time in the HHC. The PC host adjusts its time based on the zone offset and sends this time to the HHC. The zone can be set per HHC in the T record. The T record value overrides the M record, or default value.

Valid values: integers -23 through 23.

Default: 0.

*VERsion*

The version number of the software running on the HHC. You can use the version number to send a new program release to the HHC, if the HHC software supports automatic updates.

"Program" request files are sent to the HHC if (1) the option is enabled and (2) the version number text string in the database is not the same as the version number string stored in the HHC.

Valid values: any 1- to 6-byte text string.

Default: (none).

**EXAMPLE:**

```

1   5   10   15   20   25   30   35   65   70   75   80
|---+---+---+---+---+---+---+---+. . . +---+---+---|
# SET THE MODE TO NEW FOR ALL ACTIVATE RECORDS (TYPE A)
# AND INITIAL DOWNLOAD REQUESTS (TYPE 1).
M NEW=A1

```

This M record specifies that all A (activate) records and all type 1 (initial download) requests should be deleted from the ISAM database before the session control file is processed.

---

## *Request Records*

Request records include records to identify Terminals (T), Files and File Groups (F, S, and B), and Data (D).

*Terminal Identifier (T) Records*

```

T<term ID>
ZONe=va l ue
VERsi on=va l ue
STAtus=(0/1)
/D

```

T records add or change information in the database for the hand-held computers (HHCs). T records also are used to schedule, or "attach,"

data requests to HHCs. A T record becomes current when it is encountered. Any files specified in F records that follow T records are attached to the HHC specified in the T record. B records must precede the T record for their files to be attached to its HHC.

### *Parameters*

#### **<term ID>**

The HHC identifier, <term ID>, which must immediately follow the record type designation, can be up to 16 characters long. (The ID might consist of a district ID joined with an HHC ID unique to the district for example.)

Valid values: any printable ASCII character string.

### *ZONe*

The time zone offset of the HHC from the PC. If a "clock" request is scheduled for an HHC, the time zone offset is used to set the time in the HHC. The PC host adjusts its time based on the zone offset and sends this time to the HHC.

Valid values: integers -23 through 23.

Default: 0.

### *VERsion*

The version number of the software running on the HHC. You can use the version number to send a new program release to the HHC, if the HHC software supports automatic updates.

"Program" request files are sent to the HHC if (1) the option is enabled and (2) the version number text string in the database is not the same as the version number string stored in the HHC.

Valid values: Any 1- to 6-byte text string.

Default: (none).

### *STAtus*

The status of the HHC. A status value of 1 disables all data requests for the HHC.

Valid values: 0 or 1.

Default: 0 for new HHCs, current value for existing HHCs.

**/D**

/D is a “delete” switch. Use /D to delete an HHC when the mode record has specified “update.” If the mode record has specified “new,” a T record that contains the /D switch is ignored and any file identifier (F) records that immediately follow are skipped.

**EXAMPLE:**

```

1   5   10   15   20   25   30   35   65   70   75   80
|---+---+---+---+---+---+---+---+. . . +---+---+---+---|
T 120012 ZONE=- 3 VERSION=02.04 STATUS=0

```

### *Network Communications Controller Identifier (N) Records*

**N <device ID>**

**STAtus=(0/4)**

**INI t=(498X initialization parameters)**

**ZONe=val ue**

**DIR=(D/B)**

**TYPE=(B/N)**

**/D**

The 4920 or 4921 provides the facilities to maintain user files on remote 498X (4980 or 4985) devices. Use the N record to enter remote 498X devices into the device database.

A list of files for the device must be entered in a device file database on the 4920 or 4921 host system. Currently, device files can be entered by associating F records with previous N records in the session control file.

See also Appendix H, “File Processing” in your *Models 4920 and 4921 Series of PC Telecommunication Packages Reference Guide NPN: 961-021-013* for more information on 4920 and 4921 support for 498X file maintenance.

### Parameters

#### <device ID>

The 4980 or 4985 device identifier, which must immediately follow the record type designation, can be 1 to 8 characters long. The device identifier for a 498X connected directly to a PC running the 4921 software package is always named "LOCAL".

Valid values: any printable ASCII character string.

#### STAtus

The status sets or does not set the real-time clock on the remote 4980 device (see zone). *Note that the 4985 does not have a real time clock.*

Valid values: 0 or 4, where:  
0=do not set the real-time clock  
4=set the real-time clock

Default: 4.

#### INIt

4980 or 4985 specific initialization parameters as documented in the appropriate 4980 or 4985 user's guide. Use *caution* when changing any of these parameters.

Valid values: 498X initialization parameters.

Default: none.

#### ZONe

The time zone offset of a remote 4980 device in hours. You can use this parameter to set the real-time clock on a remote 4980. *Note that the 4985 does not have a real time clock.* If this parameter is nonzero, and STAtus equals 4, the time on the remote 4980 will be set to the time on the 4920 host and adjusted for the time zone offset.

Valid values: integers -23 through 23.

Default: 0.

**DIR**

If the Dnldir/Bootdir parameter has been specified in the system control file, you can use the DIR parameter to indicate that the file specified in trailing F records can be found in the Dnldir/Bootdir path.

**" NOTE:**

*The Dnldir/Bootdir path name is simply prefixed to the path name specified in the F record.*

Valid values: D (download directory)  
B (boot directory)

Default: none.

**TYPE**

Use this parameter to indicate that the files specified in trailing F records are or are not 4000 Series HHC boot files. This creates a file named NETRPL.LST if TYPE=B is specified for one or more files. See Appendix H, "File Processing" in your *Reference Guide* for more detailed information.

Valid values: B or N, where:  
B=boot files  
N=none

Default: N

**/D**

/D is a "delete" switch. Use /D to delete a 498X device when the mode record has specified "update." If the mode record has specified "new," an N record that contains the /D switch is ignored and any file identifier (F) records that immediately follow are skipped.

**EXAMPLE:**

```

1   5   10   15   20   25   30   35   65   70   75   80
|---+---+---+---+---+---+---+---+. . . +---+---+---+---|
N 49850000 STATUS=4 INIT="-S0" ZONE=0 DIR=B TYPE=N

```

**File Set Identifier (S) Records**

**S<set name>**



**/D**

Use the S record to create a set of files. HHCs that support download request processing can request a group of files by the set name. All the files in the set must be of type 0 (regular download). Files specified in the F records that immediately follow the S record are attached to the set named in the S record.

*Parameters*

<set name>

The set name, which must immediately follow the record type designation, can be up to 16 characters long.

Valid values: any printable ASCII character string.

**/D**

/D is a "delete" switch. Use /D to delete a named set when the mode record has specified "update." Any file identifier (F) records that immediately follow are skipped.

**EXAMPLE:**

```

1   5   10   15   20   25   30   35   65   70   75   80
|---+---+---+---+---+---+---+---+. . . +---+---+---|
# CREATE A SET OF FILES THAT HAS A SET NAME OF "PGMPOX"
S PGMPOX
F C: \DATA\PGMPOP
F C: \DATA\PGMPOM
F C: \DATA\PGMPOR

```

*Load Boot Set (L) Records*

**L** <boot set name>

**/D**

L records name a set of boot files specified in trailing (F) records. The L record and its associated F records identify files required to cold boot 4000 Series HHCs. An L record becomes current when it is encountered. Any files specified in F records that follow L records are sent to the HHC when it requests a boot.

If you have only one HHC application, you do not need to bother with an L record. The 4920 simply sends everything in the boot directory identified in the system control file whenever an HHC requests a boot.

If you are running more than one application on the HHCs, you must create multiple L records. Name the first L record, then attach all F records for the first HHC application program immediately following it. Next, name a second L record and attach following that L record all F records for the second HHC application.

When initiating communication with multiple boot sets defined, you need to select the appropriate set name from the boot file menu on the communication screen. In the absence of a choice by you, the 4920 sends the top HHC application appearing on the boot file menu.

Before you attempt a cold boot, HHC application software must be loaded onto the PC hard disk in the location specified by the "bootdir" command in the system control file or in the boot subdirectories specified in the load F records. See Appendix G, "Booting 4000 Series Hand-held Computers" in your *Reference Guide* for more information.

At communication time, the 4920's download request program uses the L record to locate the appropriate F records for the application. It then prefixes the boot directory name to the program names you have specified with F or S records.

If you are viewing debug information, only the literal name specified by the L record appears, not the F record names.

### *Parameters*

#### **<boot set name>**

The load set identifier, <boot set name>, must follow the L record type designation on the same line.

Length/type: One to 40 alphanumeric characters, including "idiots" such as comma, period, asterisk, hyphen, etc.

Valid values: any printable ASCII character string. The sole exception is blank spaces.

**/D**

/D is a "delete" switch. Use /D to delete a boot set when the mode record has specified "update." If the mode record has specified "new," an L record that contains the /D switch is ignored and any file identifier (F) records that immediately follow it are skipped.

**EXAMPLE:**

```

1   5   10   15   20   25   75   80
|---+---+---+---+---+...+---|
L BAKERY
F PBBYMOP.EXE
L BEVERAGE
F PBEVMOP.EXE
L DAIRY
F PBDIMOP.EXE

```

The F record PBBYMOP.EXE comprises the load set BAKERY. Two other load sets, BEVERAGE and DAIRY, have also been prepared.

---

## *Files and Broadcast Records*

Files and broadcast records include File Identifier (F) and Broadcast File Identifier (B) Records.

A File Identifier (F) Record is a single file. A Broadcast File Identifier (B) Record is attached to *all* terminals (HHCs) or devices (T or N records) which follow, until it is canceled.

### *File Identifier (F) Records*

```

F<file name>
TYPe=valu e
STAtus=(0/1)
CREate=(Y/N)
/D
/E

```

Use F records to attach data requests and files to T, S, L, and N records. F records are associated with the current record (if there is a current record). Also use F records to create files. If the create parameter in the F record is set to "Y," the file specified is created and the data records that immediately follow the F record is placed in the file. If the create parameter is set to "N," the data records are ignored.

### Parameters

<file name>

Use the path specification for the file name. If a download prefix has been specified in the system control file, it will become prefixed to the path name in the F record.

" NOTE: Since "TYPE" and "CREATE" are left off, the defaults of TYP=0 and CRE=N are assumed.

### EXAMPLE:

```

1   5   10   15   20   25   30   35   40   45   75   80
|---+---+---+---+---+---+---+---+---+---+...+---|
F C: \DATA\CUSTA. DAT

```

### TYPE

The type parameter specifies the type of request.

Valid values: 0 through 5, where:  
 0=regular download  
 1=initial download  
 2=clock  
 3=program  
 4=host download status  
 5=special download

For F records which follow N records, B and N are also valid types:

Valid values: B=boot file  
 N=not a boot file  
 Default: 0 (regular download).

*CREate*

If create is set to Y, a new file is built or an existing file is overwritten. If create is set to N or if create is not specified, the file must already exist.

Valid values: y (yes) or n (no)

Default: n (no).

*STAtus*

The status of the request. A status of 1 disables the request for each T, S, N, and L record to which it is attached.

Valid values: 0 or 1.

Default: 0.

*/D*

*/D* is a "delete" switch. If the mode record has specified "update" for the request types, you can use */D* to delete the request specified by the F record from the HHC specified in the associated T record.

*/E*

*/E* is a delete and erase switch. It deletes just as the */D* parameter and erases the file from the disk. This next example assumes that the "DNLDR" parameter in the SYSTEM.CTL file is left blank.

**EXAMPLE:**

```

1   5   10   15   20   25   30   35   40   45   75   80
|---+---+---+---+---+---+---+---+---+---+...+---|
# CREATE FILE "PROD1.DAT" IN SUBDIRECTORY NAMED "DATA"
F C: \DATA\PROD1.DAT CREATE=Y
# THE FOLLOWING ARE THE DATA RECORDS FOR PROD1.DAT
D<DPROD 00003X036N005X010>!
DONE- HALF GALLON BUTTERMILK          001. 23  00123312!
DONE GALLON CHOCOLATE ICE-CREAM 004. 89  95883493!
DEIGHT OUNCE LOW-FAT BUTTER          000. 75  93832022!

```

### Broadcast File Identifier (B) Record

**B**<file name>  
**TYPe**=value  
**STAtus**=(0/1)  
**CREate**=(Y/N)  
 /C  
 /D  
 /E

Use the B record to activate or deactivate a broadcast file. Active broadcast files are attached to any T records that follow the B record, until the B record is canceled.

A broadcast file is active at the point at which it is entered in the file; it stays active until it is deactivated. The broadcast file is deactivated by the "/C" (cancel) switch.

In addition, you can use the B record to create a data file. If the create parameter in the B record is set to Y, the file specified is created, and any data records that immediately follow the B record are placed in the file. If the create parameter is set to N, the data records are ignored.

#### Parameters

<file name>

Use the full path specification for the file name only if the "DNLDIR" is not specified.

#### EXAMPLE:

```

1   5   10   15   20   25   30   35   40   45   75   80
|---+---+---+---+---+---+---+---+---+---+...+---|
B C: \DATA\PRICES. DAT

```

*TYPe*

The type parameter identifies the type of request.

Valid values: 0 through 5, where:  
0 = regular download  
1 = initial download  
2 = clock  
3 = program  
4 = host download status  
5 = special download

Default: 0 (regular download).

### *CREate*

If create is set to Y, a new broadcast file is built or an existing file is overwritten. If create is set to N or if create is not specified, the file must already exist.

Valid values: y (yes) or n (no)  
Default: n (no).

### *STAtus*

The status of the request. A status of 1 disables the request for each HHC to which it is "broadcast."

Valid values: 0 or 1.  
Default: 0.

### */C*

*/C* is a "cancel" switch you can use to deactivate a broadcast request specified in a previous B record that contains the same filename and request type. (Refer to the second B record in the example which follows the E switch.)

### */D*

*/D* is a "delete" switch. If the mode record at the beginning of the file has specified "update," you can use the */D* switch to delete the request specified by the type parameter in the B record. This action deletes the requests for the HHCs identified in the T records that follow the B record.

**/E**

**/E** is a delete and erase switch. It deletes just as the **/D** switch *and* erases the file from the disk.

**EXAMPLE:**

```

1   5   10   15   20   25   30   35   40   45   75   80
|---+---+---+---+---+---+---+---+---+---+...+---|
# ACTIVATE THE FILE PRICES.DAT AS A BROADCAST FILE
B PRICES.DAT TYPE=0 CREATE=N
# DEACTIVATE PRICES.DAT AS A BROADCAST FILE
B PRICES.DAT TYPE=0 /C

```

### *Data (D) Records*

**D <data>**

D records contain “business” information, such as product descriptions or price lists. They may follow F or B records. Like other SESSION.CTL records, data records can be fixed-length or variable-length with carriage return/line feed record delimiters. The data filed in the data record may contain trailing spaces. The data must be in character format unless fixed-size records are specified. The format must be consistent throughout the session control file.

The maximum length of the data field of the data record is 256 characters.

You may use an ASCII exclamation mark (21 hexadecimal) to designate the end of the data field in the data record. If the final character in the data group should happen to be an exclamation mark, you simply add another exclamation mark.

You can change the character for the end-of-data-field marker in the system control file. The exclamation mark is the default character, but you may use other characters.

**EXAMPLE:**

```

1   5   10   15   20   25   30   35   40   45   75   80
|---+---+---+---+---+---+---+---+---+---+...+---|
F PROD1.DAT CREATE=Y

```



```

D<DPROP 00003X036N005X010>!
DONE- HALF GALLON BUTTERMILK      001. 23  00123312!
DONE GALLON CHOCOLATE ICE- CREAM  004. 89  95883493!
DEIGHT OUNCE LOW- FAT BUTTER      000. 75  93832022!

```

### Parameters

**<file name>**

Use the full path specification for the file name.

---

## Activation Records

Activation records consist of Port Configuration (P), Autoanswer Activate (A), and Autocall Activate (C) Records.

Criteria for activating a port:

- " *Autocall Enabled*: The communications program attempts to draw a phone record from the buckets. The Port (P) Record attaches the buckets to the port.
- " *Autoanswer Enabled, No Phone Record*: The communications program uses an available Autoanswer Activate (A) Record.
- " *Autoanswer Enabled, No Phone Record, No Autoanswer Activate Record*: The port is activated with the first set of parameters that the communication control file specified for the port.
- " *No Phone Record, Autoanswer Disabled*: The port is temporarily disabled.

If you are going to use activation record windows, you need to set the time zone environment variable for them to work correctly. Setting time zones is discussed in Section 3, "Installation" in Volume A of the *Models 4920 and 4921 Series of PC Telecommunication Packages User's Guide NPN: 961-021-011*.

### Port Configuration (P) Records

**P <PORT=va lue>**

**DIAl =va lue**

**BUCKET=va lue**

Use the P record to specify a physical port and the dial mode, and to designate the “buckets” the port may draw phone numbers from.

#### *PORT*

The port parameter specifies the letter of a physical port.

Valid values: A through D (indicates connector labels).

Default: (none).

#### *DIAL*

The dial parameter specifies the autocal-autoanswer mode for the port.

Valid values: 1 through 4, where:

1 = autocal only

2 = autoanswer only

3 = autocal and autoanswer

4 = autocal followed by autoanswer

Default: 3 (autocal and autoanswer).

#### *BUCKET*

A bucket is a user-defined group of phone records. You may associate up to three buckets with each physical communications port. When a port is activated, phone records are drawn from the buckets with that port.

The phone records are used to activate the port and to dial a phone number. The order in which the phone records are drawn from the buckets is determined by the sequence in which the buckets were defined.

A default bucket is implicitly associated with any port defined as ADCCP/TTY. The designation for the default bucket is 0 (or the value specified in the system control file).

If all buckets are employ, default parameters activate the port in an “autoanswer” mode, unless “autocal only” has been specified.

Valid values: 0 though 5.

Default: (none).

## EXAMPLE:

```

1   5   10   15   20   25   30   35   40   45   75   80
|---+---+---+---+---+---+---+---+---+---+...+---+---|
# DEFINE THE BUCKETS THAT ADCCP/TTY PORTS CAN DRAW FROM
# DEFINE THE DEFAULT ACTIVATE PARAMETERS FOR THE PORTS
P PORT=C DIAL=3 BUCKET=1 BUCKET=2
P PORT=D BUCKET=1 BUCKET=2
P PORT=B DIAL=4 BUCKET=2

```

Notice the absence of a dial parameter (DIAL =) for port D in the preceding example. Port D uses the default, autocall and autoanswer.

### *Autoanswer Activate (A) Records*

```

A <PORT=va l ue>
<PROtocol =va l ue>
BEGi n=va l ue
END=va l ue
SPEed=va l ue
PARi ty=va l ue
BITs=va l ue
SINgl e=(Y/N)
LOW address=va l ue
HI Gh address=va l ue
/D

```

Use the A record to specify autoanswer parameters for activating a communications port. You can also use A records to specify activate defaults for a time window.

## " NOTE:

*There is no difference between autoanswer and "direct connect."*

### *PORT*

Use this parameter to associate autoanswer activate parameters with the physical communications port specified in the Port Record. (If no Autoanswer Activate Records are entered for a port, the first port type definition in the communications control file supplies the default activation parameters.)

Valid values: A through D.

Default: (none).

### *PROtocol*

Use the protocol parameter to define the activate parameters as either ADCCP or TTY or to toggle an active port between ADCCP and TTY protocols. This toggle works only for those ports defined as both ADCCP and TTY (in the communication control file).

Valid values: 2 or 3, where:

2 = ADCCP

3 = TTY

Default: (none).

### *BEGin*

The begin time of the "active time window" for the activate record. If a time window is specified, the activate record is available for use only during the window. If the begin time is greater than the end time, then the time window spans midnight. If either the begin time or end time is less than or equal to 0000, then no time window is in effect, and the activate record can be used at any time.

" **NOTE:** *If there is a begin time, there **must** be an end time.*

Valid values: 0000 or any valid time in HHMM 24-hour format.

Default: 0000 (no time window).

### *End*

The end time of the "active time window" for the activate record. If a time window is specified, the activate record is available for use only during the window. If the end time is less than the begin time, then the time window spans midnight. If either the begin time or the end time is less than or equal to 0000, then no time window is used, and the activate record can be used as any time.

" **NOTE:** *If there is an end time, there **must** be a begin time.*

Valid values: 0000 or any valid time in HHMM 24-hour format.

Default: 0000 (no time window).

### *SPEed*

The communications link speed (bits per second) for ADCCP or TTY protocol.

Valid values: 0, 1200, 2400, 4800, or 9600, where 0 specifies external clocking (by a modem—for ADCCP only).

Default: no change (use current value).

### *PARity*

Specifies parity checking per character for TTY protocol.

Valid values: 0, 1, or 2, where:

0 = no parity

1 = odd parity

2 = even parity

Default: no change (use current value).

### *BITs*

Specifies the number of data bits per character for TTY protocol.

Valid values: 7 or 8, where:

7 = 7 data bits

8 = 8 data bits

Default: no change (use current value).

### *SINgle*

TTY protocol only - single HHC. Shall the communications session end (hang up) when one HHC exchange is complete?

Valid values: y (yes) or n (no).

Default: n (no). Wait for another HHC to connect.

*LOW address*

The lowest ADCCP link address that is polled. Normally, the lowest address should be 1. The LOW address must be less than or equal to the HIGH address.

Valid values: 1-64

Default: no change (use current value).

*HIGH address*

The highest ADCCP link address that is polled. The HIGH address must be greater than or equal to the LOW address.

Valid values: 1-64. (Suggestion: Set at 4 if polling remote 4980 Communication Controllers.)

Default: no change (use current value).

The ADCCP range specified with LOW and HIGH address is the polling range used during an ADCCP communications session. The range must include addresses 1 through 4 for 4920-to-4980 communications. For NT121/141 ADCCP communications, the range must include any addresses used by the 121/141 HHCs. (The 121/141 address is usually determined by the particular slot in the Multidrop Quad Lockbox.)

*/D*

/D is a "delete" switch you can use to delete the activate record defined by the begin time and port number.

**EXAMPLE:**

```

1   5   10   15   20   25   30   35   40   45   75   80
|---+---+---+---+---+---+---+---+---+---+...+---|
# DEFINE THE AUTOANSWER ACTIVATE RECORDS.
A PORT=A PRO=2 BEG=2200 END 0300
A PORT=A PRO=3 BEG=2200 END=0300 SPEED=2400 PARITY=0

```

*Autocall Activate (C) Records*

**C** <PHOne=val ue>

**<PR0tocol =val ue>**  
**BUCKet=val ue**  
**BEGi n=val ue**  
**END=val ue**  
**RETri es=val ue**  
**WAI t=val ue**  
**EXPEcted=val ue**  
**SPEed=val ue**  
**PARi ty=val ue**  
**SINgl e=(Y/N)**  
**BITs=val ue**  
**LOW=val ue**  
**HI Gh=val ue**  
**/D**

Use the C record to specify autocall parameters for activating a communications port. C records must include a phone number and a protocol specifier. C records have priority over A (autoanswer) records when both autoanswer and autocall are enabled.

#### *PHOne*

The phone parameter, which is mandatory for the C record, is the string of characters needed for the dialing sequence; in other words, the phone number to be called. The string is Hayes-compatible.

If the string begins with "AT" the string is sent to the modem unchanged. If the string begins with a "T" or a "P," "ATD" is prefixed to the string. If the string begins with a number, "ATDT" is prefixed to the string.

Valid values: any valid Hayes dial string.

Default: (none).

#### *PROtocol*

Use the protocol parameter to define the activate parameters as either ADCCP or TTY or to toggle an active port between ADCCP and TTY protocols. This toggle works only for those ports defined as both ADCCP and TTY.

Valid values: 2 or 3, where:  
2=ADCCP  
3=TTY  
Default: (none).

### *BUCKET*

This parameter specifies the designation of the container that holds user-defined phone records. When a port is activated, phone records are drawn from the buckets associated with that port.

The phone records are used to activate the port and to dial a phone number. The order in which the phone records are drawn from the buckets is determined by the sequence in which the buckets were defined.

A default bucket is implicitly associated with any port defined as ADCCP/TTY. The designation for the default bucket is 0 (or the value specified in the system control file).

If all buckets are empty, default parameters activate the port in an "autoanswer" mode.

Valid values: 0 through 5.  
Default: 0 (or the value specified in the system control file).

### *BEGIN*

The begin time of the "active time window" for the activate record. If a time window is specified, the activate record is available for use only during the window. If the begin time is greater than the end time, then the time window spans midnight. If either the begin time or end time is less than or equal to 0000, then no time window is in effect, and the activate record can be used at any time.

" **NOTE:** *If there is a begin time, there **must** be an end time.*

Valid values: 0000 or any valid time in HHMM 24-hour format.  
Default: 0000 (no time window).



### *End*

The end time of the "active time window" for the activate record. If a time window is specified, the activate record is available for use only during the window. If the end time is less than the begin time, then the time window spans midnight. If either the begin time or the end time is less than or equal to 0000, then no time window is in effect, and the activate record can be used as any time.

" **NOTE:**

*If there is an end time, there **must** be a begin time.*

Valid values: 0000 or any valid time in HHMM 24-hour format.

Default: 0000 (no time window).

### *RETriEs*

The number of times the autocall record should be retried.

Valid values: 0 through 999.

Default: 99 (or the global value specified in the system control file).

### *WAIt*

The minimum number of minutes to wait before retrying an activate record.

Valid values: 0 through 1440.

Default: 99 (or the global value specified in the system control file).

### *EXPeCted*

The number of HHCs expected at the dial sequence (phone number) specified in the activate record. When the expected count is used (a number greater than 0), the phone number is not called after the expected number of HHCs have succeeded. The criteria that define a successful call are determined at a global level in the system control file.

Valid values: 0 through 9999.

Default: 0 (not used).

### *SPEed*

The communications link speed (bits per second).

Valid values: 0, 1200, 2400, 4800, or 9600, where 0 specifies external clocking (by a modem—for ADCCP only).

Default: no change (use current value).

### *PARity*

Specifies parity checking per character for TTY protocol.

Valid values: 0, 1, or 2, where:

0 = no parity

1 = odd parity

2 = even parity

Default: no change (use current value).

### *BITs*

Specifies the number of data bits per character for TTY protocol.

Valid values: 7 or 8, where:

7 = 7 data bits

8 = 8 data bits

Default: no change (use current value).

### *SINgle*

TTY protocol only - single HHC. Shall the communications session end (hang up) when one HHC has been serviced?

Valid values: y (yes) or n (no).

Default: n (no).

### *LOW address*

The lowest ADCCP link address that is polled. Normally, the lowest address should be 1. The LOW address must be less than or equal to the HIGH address.

Valid values: 1-64  
 Default: no change (use current value).

### *HIGH address*

The highest ADCCP link address that is polled. The HIGH address must be greater than or equal to the LOW address.

Valid values: 1-64.  
 Default: no change (use current value).

The ADCCP range specified with LOW and HIGH address is the polling range used during an ADCCP communications session. The range must include addresses 1 through 4 for 4920-to-4980 communications. For NT121/141 ADCCP communications, the range must include any addresses used by the 121/141 HHCs. (The 121/141 address is usually determined by the particular slot in the Multidrop Quad Lockbox.)

### */D*

/D is a "delete" switch you can use to delete the activate record defined by the dial sequence.

#### EXAMPLE:

```

1   5   10   15   20   25   30   35   40   45   50   80
|---+---+---+---+---+---+---+---+---+---+---+...|
# DEFINE THE AUTOCALL ACTIVATE RECORDS.
C BUCKET=1 PRO=1 PHONE=9W13128885555 BEG=2200 \
  END=0300 RET=5 WAIT=30
C BUCKET=2 PRO=3 PHONE=9W13128786666 BEG=2200 \
  END=0300 RET=5 WAIT=30 SPEED=2400
C BUCKET=2 PRO=3 PHONE=9W13128786644 BEG=2200 \
  END=0300 RET=5 WAIT=30 SPEED=2400

```

---

## Sample Session Control File

" NOTE: *Parameter names can be in either upper or lower case and that only the first three characters in the parameter name are required.*

*This sample assumes that a download data path prefix has not been specified for the DNLDIR parameter in the system control file. Normally, a download path prefix would already be set.*

```

1   5   10   15   20   25   30   35   40   45   50   80
|---+---+---+---+---+---+---+---+---+---+---+...|
# SET THE MODE TO "NEW" FOR ALL ACTIVATE RECORDS (TYPE
# A) AND REGULAR DOWNLOAD REQUESTS (TYPE O).
M NEW=A0
#
# NAME BOOT SETS TO LOAD 4000S FOR BAKERY, BEVERAGE,
# AND DAIRY.
# SPECIFY FILE NAMES FOR EACH BOOT SET.
L BAKERY
F PBBYMOP. EXE
L BEVERAGE
F PBEVMOP. EXE
L DAIRY
F PBDIMOP. EXE
#
# DEFINE THE BUCKETS THAT ADCCP/TTY PORTS CAN DRAW FROM
# AND DEFINE THE DEFAULT ACTIVATE PARAMETERS FOR THE
# PORTS
P PORT=B DIAL=3 BUCKET=1 BUCKET=2
P PORT=C BUCKET=1 BUCKET=2
P PORT=D DIAL=4 BUCKET=2
#
# DEFINE THE AUTOANSWER ACTIVATE RECORDS.
A PORT=A PRO=2 BEG=2200 END 0300
A PORT=A PRO=3 BEG=0301 END=2159 SPEED=2400 \
PARITY=0 BITS=8
#

```

```
# DEFINE THE AUTOCALL ACTIVATE RECORDS.
C BUCKET=1 PRO=2 PHONE=9W13128885555 BEG=2200 \
  END=0300 RET=5 WAIT=30
C BUCKET=2 PRO=3 PHONE=9W13128786666 BEG=2200 \
  END=0300 RET=5 WAIT=30 SPEED=2400
C BUCKET=2 PRO=3 PHONE=9W13128786644 BEG=2200 \
  END=0300 RET=5 WAIT=30 SPEED=2400
#
# CREATE A SET OF FILES THAT HAS A SET NAME OF "PGMPOX"
S PGMPOX
F PGMPOP TYPE=0 CREATE=N
F PGMPOM TYPE=0 CREATE=N
F PGMPOR TYPE=0 CREATE=N
#
# CREATE A FILE NAMED "PROD1.DAT" IN THE SUBDIRECTORY
# NAMED "DATA"
F PROD1.DAT CREATE=Y
# THE FOLLOWING ARE THE DATA RECORDS FOR PROD1.DAT
D<DPROD 00003X036N005X010>!
DONE- HALF GALLON BUTTERMILK      001. 23  00123312!
DONE GALLON CHOCOLATE ICE-CREAM 004. 89  95883493!
DEIGHT OUNCE LOW-FAT BUTTER      000. 75  93832022!
#
# ACTIVATE THE FILE "PRICES.DAT" AS A BROADCAST FILE
B PRICES.DAT TYPE=0 CREATE=N
#
# SCHEDULE HHC #120012 TO RECEIVE REGULAR DOWNLOAD FILES
T 120012
F PROD1.DAT TYPE=0 CREATE=N
F CUST1.DAT TYPE=0 CREATE=N
F PROMD1.DAT TYPE=0 CREATE=N
#
# ACTIVATE THE FILE "BRDCAST.DAT" AS A BROADCAST FILE
B BRDCAST.DAT TYPE=0 CREATE=N
#
```

```
# SCHEDULE HHC #100955 TO RECEIVE DOWNLOAD FILES
T 100955
F PROD2.DAT TYPE=0
F CUST2.DAT TYPE=0
F PROMD2.DAT TYPE=0
#
# DEACTIVATE THE BROADCAST FILE "PRICES.DAT"
B PRICES.DAT TYPE=0 /C
#
# SCHEDULE HHC #101584 TO RESET ITS CLOCK FOR 3
# TIME ZONES WEST OF THE PC, UPDATE ITS APPLICATION
# PROGRAM (IF OLD), AND GET REGULAR DOWNLOAD.
T 101584 ZONE=- 3 VERSION=02.04 STATUS=0
F PROD2.DAT TYPE=0
F CUST2.DAT TYPE=0
F PROMD2.DAT TYPE=0
```

---

### *Using Request Database Menu*

The "Maintain the request database" program is composed of submenus you can use to edit commands used to maintain the request database. This includes:

- " adding HHCs
- " establishing program defaults for new HHCs
- " editing records that add or change information in the database for existing HHCs
- " attaching data requests to existing HHCs
- " editing download request and boot sets

You can also purge, rebuild, or dump (in session.ctl format) the request database.

**EXAMPLE:**

**Maintain the request database**

**<version> <date>**

**Copyright (c) Norand Corporation 1990**

**Program defaults**

**Terminal s**  
**Boot sets**  
**Download request sets**  
**Maintenance**  
     **Save   Restore   Exit (ESC)**

<message line>

### *Definitions*

**Program Defaults:** Use this task's menu to set the status of each HHC you are adding to the system, the time zone offset of the HHC from the PC, the file status (enabled or disabled), the type of request, and the program version of the software running on the HHC.

**Terminals:** This task's menu lists by identification number all HHCs and their current status (enabled or disabled). Use it to edit an HHC's file information and add an HHC to the system.

**Boot Sets:** A boot set contains boot files to be sent to 4000 Series HHCs when the Models 4920 and 4921 Series of PC Telecommunication Packages receives a boot request. This task's menu lists boot set names and their current status (enabled or disabled). Use this task to change the status of a boot set, add a boot set, add a file to a boot set, delete a file from a boot set, change the status of a boot file, and delete a boot set.

**Download Request Sets:** A download request set contains download request files to be sent to HHCs that support download request processing. This task displays download request set names and their current status (enabled or disabled). Use this task to change the status of a download request set, add a download request set, add a file to a download request set, delete a file from a download request set, change the status of a download request file, and delete a download request set.

**Maintenance:** Use this task's menu to

- " rebuild the request database if files became corrupted
- " purge all information from the request database
- " dump the request database to a file

**Save:** This command saves the changes you typed on the menus and places the new values in the request database. You *must* select this option and press <Enter> before you exit the program if you want to save the changes you made, in addition to saving them for individual menus. If you exit before you do this, the new values are *not* retained.

**Restore:** This command reverts the values *on every menu* within the "Maintain the request database" program to what they were when last saved.

**Exit (ESC):** Exits the "Maintain the request database" menu.

### *Procedure*

Select the task you want to perform and press <Enter> to retrieve its menu. Then select the parameter whose value you want to change and type its new value. If you make any changes on a menu, remember to select "Save" and press <Enter>, and "Save" again before leaving the "Maintain the request database" menu program. This last action places the new values into the request database.

---

## *Using Activation Database Menu*

The "Maintain the activation database" program is composed of submenus that provide an easy way for you to edit commands used to maintain the port activate database. This includes:

- " adding HHC ports
- " establishing program defaults for new ports
- " editing information associated with autocall and autoanswer phone records

You can also purge, rebuild, and dump (in session control format) the port activate database.

**EXAMPLE:**

```
Maintain the activation database  
<version> <date>  
Copyright (c) Norand Corporation 1990  
Program defaults  
Ports
```



**Call activations**  
**Answer activations**  
**Maintenance**  
     Save   Restore   Exit (ESC)  
 <message line>

### Definitions

**Program Defaults:** Use this task to set the parameters for each port you are adding to the system. This includes port mode and call parameters, and ADCCP and TTY default protocol options.

**Ports:** This task lists all physical HHC ports, their dial mode, and the buckets from which the port draws phone numbers. Use it to set mode and bucket information for a port and add a port to the system.

**Call Activations:** An autocal activate record specifies autocal parameters for activating an HHC communications port. The record contains phone numbers and protocol specifiers. This task lists autocal activate phone numbers and the protocol for each phone number (or activation). Use it to edit information associated with a phone number or delete a phone number.

**Answer Activations:** An autoanswer activate record specifies autoanswer parameters for activating an HHC communications port. The record contains time window specifiers. This task lists autoanswer activate ports and their begin and end time. Use this task to add a port.

**Maintenance:** Use this task to

- " rebuild the activation database if files became corrupted
- " purge all information from the activation database
- " dump the activation database

**Save:** This command saves the changes you typed on the menus and places the new values in the activation database. You *must* select this option and press <Enter> before you exit the program if you want to save the changes you made, in addition to saving them for individual menus. If you exit before you do this, the new values are *not* retained.

**Restore:** This command reverts the values *on every menu* within the "Maintain the activation database" to what they were when last saved.

**Exit (ESC):** Exits the "Maintain the activation database" menu.

### *Procedure*

Select the task you want to perform and press <Enter> to retrieve its menu. Then select the parameter whose value you want to change and type its new value. If you make any changes on a menu, remember to select "Save" and press <Enter>, and "Save" again before leaving the "Maintain the activation database" menu program. This last action places the new values into the activation database.

---

## *Command Line Shortcuts*

As you become more experienced with the Models 4920 and 4921 Series of PC Telecommunication Packages by Norand, you can use shortcuts at the operating system command line. The first shortcut involves bypassing the Main Menu to retrieve a menu. The second shortcut involves using switches to bypass the entire menu system and purge, rebuild, or dump the request or activation database from the command line.

### *Retrieving a Menu*

Two ways exist to bypass the Main Menu and retrieve a menu from the operating system command line.

- " Assign labels to Main Menu tasks in the MENU.CTL file. The commands associated with each download format option are stored in MENU.CTL and designated "label = <letter>." To retrieve a menu from the command line, type "menu <letter>" at the prompt and press <Enter>. To retrieve menus consecutively, type "menu <letter> <letter>."

**EXAMPLE I:** You could type "menu mr" to retrieve the "Maintain the request database" menu.

- " Type the name of the program at the prompt.

**EXAMPLE II:**

type...	to retrieve the...
editreq	"Maintain the request database" menu
editact	"Maintain the activation database" menu

and press <Enter>.

" **NOTE:** For information on putting commands into a batch file, see Appendix F, "Operating Unattended."

### *Using Command Line Switches*

You can purge, rebuild, or dump the request or activation database into a file whose format is identical to that of a session control file.

Each task is explained below.

**Rebuild:** You would normally use this task when files become corrupted. It deletes all index files and rebuilds them from the data files, thus making them synchronous again.

**Purge:** This task removes the contents of the request or activation database by removing data from files. It is not fatal.

**Dump:** Use this task to periodically back up the request or activation database. Two options exist.

- " You can create an output file and unload the contents of the current request or activation database to the file you specify
- " You can add (append) the contents of the current request or activation database to the file you specify

### *Usage Lines*

The information you type at the command line must follow a prescribed format and include certain symbols. Usage lines are available to assist you.

The usage line for a program shows the program's parameters (designated by a letter and equivalent to the associated menu's parameters), their acceptable values, and the correct format.

**EXAMPLE:** If you want to rebuild, purge, or dump the contents of the request database from the command line and need assistance, you would type

```
C: \4920>EDITREQ /h
```

(where “h” means “help”) and press <Enter>. The request database usage line appears:

```
editreq: usage: editreq [/r | /p | /ddumpfile  
[/a{y|n}]]
```

The following list explains usage line notation.

### *Usage Line Notation*

- The program name (editreq or editact) and information enclosed within braces { } is required. If you use the “/a” switch, you *must* type its value (Y or N).
- Use either “/” or “-” (hyphen) to designate a switch.
- The “/” symbol indicates that you must type the value to either the immediate left *or* right of it (such as “Y” or “N”).

“ **NOTE:**

*For information on putting commands into a batch file, see Appendix F, “Operating Unattended.”*

The remainder of this section lists each database program’s usage line parameters and gives examples.

---

## *Maintaining Request Database from Command Line*

**Editreq:** Names the program that rebuilds, purges, or dumps the contents of the request database. Its equivalent submenu options are “Maintain the request database” and “Dump the request database.” The usage line looks like this:

```
editreq: usage: editreq [/r | /p | /ddumpfile [/a{y|n}]]
```

\* editreq = program name

/r = rebuild the request database

/p = purge the request database

/d = dump the request database

dumpfile = the output file that will be receiving the contents of the request database. The format of the file is similar to that of a session con-

trol file. If dumping the database, you must indicate the value for the "/a" switch.

/a = append the contents of the request database (Y/N). Y = Add the contents of the current request database to an existing file. N = Create the file before dumping to it. The default value is N.

\* required information.

**EXAMPLE I:** Rebuilding the request database  
**C: \4920>edi treq /r**

The program is called "editreq." When you press <Enter>, the request database rebuilds *immediately*.

**EXAMPLE II:** Purging the request database  
**C: =4920>edi treq /p**

The program is called "editreq." When you press <Enter>, the request database is purged *immediately*, without warning.

**EXAMPLE III:** Dumping the request database  
**C: \4920>edi treq /dsave1 /ay**

The program is called "editreq." When you press <Enter>, the contents of the request database are appended to the end of the file named SAVE1.

**C: \4920>edi treq /dsave2 /an**

The program is called "editreq." When you press <Enter>, a file named SAVE2 is created, and the contents of the request database are dumped to it.

---

## Maintaining Activation Database from Command Line

**Editact:** Names the program that rebuilds, purges, or dumps the contents of the activation database. Its equivalent submenu options are "Maintain the activation database" and "Dump the activation database." The usage line looks like this:

**edi tact: usage: edi tact [/r | /p |**

**/ddumpfile** [/a{y|n}]]

\* editact = program name

/r = rebuild the activation database

/p = purge the activation database

/d = dump the activation database

dumpfile = the output file that will be receiving the contents of the activation database. The format of the file is similar to that of a session control file. If dumping the database, you must indicate the value for the “/a” switch.

/a = append the contents of the activation database (Y/N). Y = Add the contents of the current activation database to an existing file. N = Create the file before dumping to it. The default value is N.

\* required information.

**EXAMPLE I:** Rebuilding the activation database

**C: \4920>editact /r**

The program is called “editact.” When you press <Enter>, the activation database rebuilds *immediately*.

**EXAMPLE II:** Purging the activation database

**C: =4920>editact /p**

The program is called “editact.” When you press <Enter>, the activation database is purged *immediately*, without warning.

**EXAMPLE III:** Dumping the activation database.

**C: \4920>editact /dsave1 /ay**

The program is called “editact.” When you press <Enter>, the contents of the activation database are appended to the end of the file named SAVE1.

**C: \4920>editact /dsave2 /an**

The program is called “editact.” When you press <Enter>, a file named SAVE2 is created, and the contents of the activation database are dumped to it.

## Section 13

# Understanding the Host Download File

" **NOTE:**

*This form of the download scheduling file is made available primarily for existing Norand customers that have already been using the HOST.DNL with one of the older NORAND<sup>®</sup> telecommunication handlers. It is not the preferred method since many of the new features supported in the SESSION.CTL file are lost. All new customers should start out using the SESSION.CTL file.*

The host download file (HOST.DNL) is the host input to the program (HOSTDNF.EXE) that does download formatting for the hand-held computers (HHCs). The host download format program generates a database used by the communication program when sending data to the HHCs.

The host download file is similar to the session control file (SESSION.CTL), but provides far fewer of the features available with the Model 4920 or 4921. Use it only when you want compatibility with previous NORAND Tcom Handlers.

The formatting parameters for the host download file are stored in the system control file (SYSTEM.CTL).

ASCII characters that have a value of hexadecimal 20 (decimal 32) or lower are called "nonprinting" characters. Of principal interest in the following discussion are "space" (SP, 20 hexadecimal), "carriage return" (CR, 0D hexadecimal), "line feed" (LF, 0A hexadecimal), and "tab" (HT, 09 hexadecimal).

Characters with values ranging from 21 to 7E hexadecimal are called "printing" (nonblank) characters.

---

## Record Format

A record in the host download file consists of two fields. The first field identifies the category (type) the record belongs to. The second field is a data field, the contents of which vary according to the record category. The format of the record is as follows:

**\*<record type><data>[spaces][CR/LF]**

The "\*" is an ASCII character which marks control records.

The fields in angle brackets are mandatory. The characters in square brackets are optional.

The maximum size of a record is 264 bytes. For convenience in printing or on-screen viewing, you may wish to break a long record into several lines, with line lengths limited to 80 columns.

The records in the host download file may be of the same length (fixed-length), or the length may vary from record to record (variable-length). If the file contains variable-length records, each record must be terminated by an ASCII carriage return (CR) and line feed (LF).

---

## Record Types

The following are the record type categories, related group names, and the associated designations:

<b>Description</b>	<b>Group:</b>	<b>Record Type</b>
#		Comment
	REQUEST	
*U		Upload
*D		Download



Description	Group:	Record Type
	INCLUDE	
*I		File identifier
*L		IDL include file
*B		Broadcast file identifier
	ACTIVATION	
*P		Autocall activate

---

### Comment (#) Records

Use “#” records to insert comments into the host download file. All characters in the record following the # character are ignored by control programs.

---

### Request Records

Request records include records to identify terminals for upload only (\*U) and for download (\*D).

#### Upload (\*U) Records

**\*U <term ID>**

With \*U records, only uploads are expected from the HHC (HHC). Data is not downloaded to it.

#### Parameters

**<term ID>**

The HHC identifier, <term ID>, which must immediately follow the record type designation, can be up to 16 characters long.

#### EXAMPLE:

The ID might consist of a district ID joined with an HHC ID unique to the district.

Valid values: any printable ASCII character string.

## EXAMPLE:

```

1   5   10  15  20  25  30  35  65  70  75  80
|---+---+---+---+---+---+---+. . . +---+---+---|
*U120012

```

*Download (\*D) Records***\*D** <term ID>

\*D records are also used to schedule, or attach, data requests to HHCs. A \*D record becomes current when it is encountered. Any data, \*I, or \*L records that follow are attached to the HHCs. \*B records must precede the \*D record for their files to be attached to its HHC.

*Parameters*

&lt;term ID&gt;

The HHC identifier, <term ID>, which must immediately follow the record type designation, can be up to 16 characters long.

## EXAMPLE:

The ID might consist of a district ID joined with an HHC ID unique to the district.

Valid values: any printable ASCII character string.

## EXAMPLE:

```

1   5   10  15  20  25  30  35  65  70  75  80
|---+---+---+---+---+---+---+. . . +---+---+---|
*D120012

```

*Include File Records*

Include file records consist of File Identifier (\*I), IDL Include File Identifier (\*L), and Broadcast File Identifier (\*B) Records.

A File Identifier (\*I and \*L) Record makes up a single file. A Broadcast File Identifier (\*B) Record is attached to *all* terminals (HHCs) which follow.

*File Identifier (\*I) Records***\*I <file name>**

Use \*I records to attach data requests and files to HHCs. \*I records are associated with the current \*D records. Data specified by \*I records are sent to the HHC when it requests a regular (not initial) download.

*Parameters***<file name>**

Name of the file on disk.

**EXAMPLE:**

```

1   5   10   15   20   25   30   35   40   45   75   80
|---+---+---+---+---+---+---+---+---+---+. . . +---|
*D120012
*I PROD. DAT

```

*IDL Include (\*L) Records***\*L <file name>**

Use \*L records to attach data requests to HHCs. \*L records are associated with current \*D records. Data specified by \*L records is sent to the HHC when it requests an initial download (IDL).

*Parameters***<file name>**

Name of the file on disk.

**EXAMPLE:**

```

1   5   10   15   20   25   30   35   40   45   75   80
|---+---+---+---+---+---+---+---+---+---+. . . +---|
*D120012
*L PROD. DAT

```

Sent file PROD.DAT to HHC number 120012 as data.

### Broadcast File Identifier (\*B) Record

**\*B <file name> 0**

Use the \*B record to activate or deactivate a broadcast file. Active broadcast files are attached to the HHCs identified in any \*D records that follow the \*B record, until the \*B record is canceled.

A broadcast file is active where it is entered in the file; it stays active until it is deactivated. Only one broadcast file can be active at a time, since the occurrence of a second \*B record terminates the previous broadcast file. The broadcast file is deactivated by another \*B record or \*B without a file name attached.

In addition, you can use the \*B record to create a data file.

#### Parameters

**<file name>**

Name of a file on disk. Place the file name immediately following \*B.

**0 (the letter o)**

The “O” stands for old. If the 19th position in the record *does not* contain a “O,” the file specified is created, and any data records immediately following the “B record are placed in the file.

**EXAMPLE:**           Activating then deactivating a broadcast file.

```

1   5   10   15   20   25   30   35   40   45   75   80
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----|
# ACTIVATE THE FILE PRICES. DAT AS A BROADCAST FILE
*BPRICES. DAT           0
# DACTIVATE PRICES. DAT AS A BROADCAST FILE
*B!
```

**" NOTE:**           *“!” represents an empty broadcast file.*

### Data Records

**[space] <data>**

Data records contain “business” information, such as product descriptions or price lists. They may follow \*D or \*B records. Like other HOST.DNL records, data records can be fixed-length or

variable-length with carriage return/line feed record delimiters. The data field in the data record may contain trailing spaces. The data must be in character format unless fixed-size records are specified. The format must be consistent throughout the host download file.

The maximum length of the data field of the data record is 256 characters.

You may use an ASCII exclamation mark (21 hexadecimal) to designate the end of the data field in the data record. If the final character in the data group should happen to be an exclamation mark, you simply add another exclamation mark.

You can change the character for the end-of-data-field marker in the system control file. The exclamation mark is the default character, but you may use other characters.

**EXAMPLE:**           Containing three data records.

```

1   5   10   15   20   25   30   35   40   45   75   80
|---+---+---+---+---+---+---+---+---+---+. . . +---|
*BPROD. DAT
D<DPROP 00003X036N005X010>!
ONE- HALF GALLON BUTTERMILK      001. 23  00123312!
ONE GALLON CHOCOLATE ICE- CREAM  004. 89  95883493!
EI GHT OUNCE LOW- FAT BUTTER     000. 75  93832022!

```

---

## Activation Records

Activation records consist of phone Autocall Activate (\*P) Records.

### *Autocall Activate (\*P) Records*

**\*P**

Use the \*P record to specify autocall parameters for activating a communications port.

#### *Parameters*

**<phone number>**

The phone parameter, which is mandatory for the \*P record, is the string of characters needed for the dialing sequence; in other words, the phone number to be called. The string is Hayes-compatible.

If the string begins with "AT," it is sent to the modem unchanged. If the string begins with "T" (tone) or "P" (pulse), "ATD" is prefixed to it. If the string begins with a number, "ATDT" is prefixed to it. A "W" in the string indicates a short wait.

Valid values: any valid Hayes dial string.

Default: (none).

#### <port>

The <port> represents the "logical" port, *not* the physical port. Run EDITCOMM.EXE to determine the logical port number and note the order of the physical ports on the screen. The port to dial phone numbers on, or 0 (zero) for the next available port.

Valid values: 1 through 4, or 0

Default: (none).

#### <retries>

The number of times the autocal record should be retried.

Valid values: 0 through 999.

Default: 0 or the global value specified in the system control file.

#### <wait>

The minimum number of minutes to wait before retrying an activate record.

Valid values: 0 through 1440.

Default: 0 or the global value specified in the system control file.

**<expected>**

The number of HHCs expected at the phone number specified in the activate record. When the expected count is used (greater than 0), the phone number is not called after the expected number of HHCs have succeeded. The criteria that defines a successful call is determine at a global level in the system control file.

Valid values: 0 through 9999.

Default: 0 (not used).

**EXAMPLE:**

```

1   5   10   15   20   25   30   35   40   45   50   80
|---+---+---+---+---+---+---+---+---+---+---+---+. . . |
*P9W131933699999, 1, 3, 30, 10
*P9W13193698888, 0, , 30

```

**Sample Host Download File**

" **NOTE:** *Parameter names can be in either upper or lower case and only the first three characters in the parameter name are required. The ends of data records have been truncated so the example fits on the page.*

```

1   5   10   15   20   25   30   35   40   45   50   80
|---+---+---+---+---+---+---+---+---+---+---+---+. . . |
# PHONE NUMBER TO DIAL ON PORT 1
*P9W1319369999, 1, 3, 30, 10
# PHONE NUMBER TO DIAL ON ANY PORT
*P9W13193698888, 0, , , 20
#
# CREATE PROD. DAT AS A BROADCAST FILE
*BPROD. DAT
#
# FOLLOWING ARE THE DATA FILES FOR PROD. DAT
<DBYPRD 0000N004X016N010N006N004N008X014>!
0001ASSORTED MUFFI#11111111110231000225000222024101111
0002CORN MUFFINS #21122222220116500115500222012602221
0003HONEY GRAM MUF#31133333330062000060000222007203331

```

```

0004SMALL LEM COOK#411444444440034000032000222004404441
0005BUTTER COOKIES#511555555550020000017000222003005551
#
#ACTIVATE PRICES.DAT AS A BROADCAST FILE
*BPRICES.DAT      0
#
# SCHEDULE HHC #120012 TO RECEIVE DOWNLOAD DATA
*D120012
<DBYMSG 00000X001N006N004X032X001>!
40000000000*****0!
40000000000      BOB' S BAKERY      *0!
40000000000      550 SECOND STREET SE  *0!
40000000000      CEDAR RAPIDS, IA 52401 *0!
40000000000      (319) 369- 3100      *0!
40000000000*****0!
*I PROD. DAT
*I CUST. DAT
*PROMD. DAT
*LIDLPROD. DAT
#
# DEACTIVATE PRICES.DAT AS A BROADCAST FILE
*B !
# ACTIVATE THE FILE "BRDCAST.DAT" AS A BROADCAST FILE
*BRDCAST.DAT      0
#
# SCHEDULE HHC #100955 TO RECEIVE DOWNLOAD DATA
*D100955
*PROD1. DAT
*LIDLPROD1. DAT
*I CUST1. DAT
*LPRMOD1. DAT
*LPROIDL. DAT
*LCUSTIDL. DAT
#
# DEACTIVATE THE BROADCAST FILE "BRDCAST.DAT"
*B !
#

```



## Section 14

### *Understanding the Upload Format Control File*

---

The upload format control file (UPLFMT.CTL) must contain a description of the upload file for the application program running on the handheld computers (HHCs). The file contains the PL/N upload file name and a series for record types (categories). Each record type is immediately followed by a format description of the data field or fields for this record type.

This file enables the upload formatting program to format the data received from the HHCs for communication to a host computer. The executable program is either UPLFMT.EXE or HOSTUPF.EXE. UPLFMT.EXE generates a separate file for each route, while HOSTUPF.EXE generates a single file.

Each upload file of the form:

```
<PL/N file name> (file)
<record type A> (rectype)
<type A format> (format)
<record type B>
<type B format>
.
.
.
<record type Z>
<type Z format>
```

If you do not have the necessary upload file, you should find the required information in the PL/N application program specification.

You can use a text editor to edit UPLFMT.CTL. Refer to the editor's reference manual for assistance.

---

## *PL/N Upload File*

Name of the PL/N upload file.

Length/type: 1- to 80-character alphanumeric.

Valid values: any printable ASCII character string.

Example: file=BYTRXN

PL/N transaction file for bakery application.

## *Rectype*

Record type.

The label for a data company.

Length/type: one-character alphanumeric.

Valid values: any printable ASCII characters\*.

Example: rectype=A

Record type for this category is A.

\* *rectype* = DEX uses DEX/UCS formatting for a file, which contains only carriage return and line feed record delimiters.

## *Format*

Format of a record type.

Format can contain one or more 4-character, alphanumeric descriptors for data contained in a record type. Each descriptor represents one field within the category. The descriptor begins with a letter indicating the data type. Appended to the letter is a three-digit decimal number telling how many characters the field contains.

Length/type: one-character alphabetic prefixed to a three-digit decimal number.

Valid values: any valid PL/N field descriptor as defined by the application program

Example: format=N012X016N004

Expect 12 binary coded decimal (BCD) followed by 16 alphanumeric ASCII characters and 4 BCDs.

---

### Sample Upload Format Control File

```
#
# This is the upload file for base bakery , PBBYMOP.
#
file=BYTRXN
rectype=A
# record type A has a 4-digit BCD field followed by a
# 5-digit BCD field.
format=N004N005
rectype=B
format=N004N005
rectype=C
format=N004N005
rectype=D
format=N004N005
rectype=E
# 4-digit BCD field, 6-digit BCD field, 6-digit BCD
# field, and 1 character
format=N004N006N006X001
rectype=F
format=X006N010
rectype=G
format=N016
rectype=H
format=X035
rectype=I
# 4-digit BCD field, and 16 by 5 BCD matrix
```

**format=N004(016N005) 000**  
**rectype=J**  
**format=X017**  
**rectype=K**  
**format=N006N008**  
**rectype=M**  
**format=X007N004N007N007N007N007N007N007N007N007N007**  
**rectype=N**  
**format=X001N004N007N006N007**  
**rectype=0**  
**# 10 BCD digits followed by 4 characters & 9 characters**  
**format=N010X004X009**  
**rectype=P**  
**format=N004N005X002**  
**rectype=Q**  
**format=N006N009**  
**rectype=R**  
**format=N010X011N006N009N009X001**  
**rectype=S**  
**format=N010X003N014X008N045N045N009N009N009X002**  
**rectype=T**  
**format=X004N012X001**  
**rectype=U**  
**format=X001N004N005N006N004X001**  
**rectype=V**  
**format=X016X001N004X001N004X004**  
**rectype=W**  
**format=X014N003X001X006**  
**rectype=Y**  
**format=X021**  
**rectype=Z**  
**format=X010**

---

## *DOS Files*

This feature allows the system to send and receive DOS files.  
PLNFMT.EXE prefixes a special "DOS" PL/N header to any file so that

the file can be sent intact to the HHC without changing the actual data within the file. The HHC must have the ability to extract this DOS file back into its original state. HOSTUPE.EXE and UPLFMT.EXE have the ability to extract these DOS files into the the current subdirectory. The UPLFMT.CTL file must contain the appropriate file name information:

**FILE=DOSFIL**

**RECTYPE=DOS**

The UPLOAD.DAT file name and path can be specified in the SYSTEM.CTL file under the parameter: UPLOADFNAME. The default is UPLOAD.DAT in the default subdirectory. When telecommunication starts, the UPLOAD.DAT file can be saved as a file with the extension of .BKK. The parameter to specify this function is in the SYSTEM.CTL file as: UPLOADBACK=y.



## Section 15

### *Understanding the Host Upload File*

.....

The host upload file is the output of an upload format file. The conversion takes place after communication with the HHCs has finished. It starts when you choose the “upload format” command on the “Format upload data into host upload file” menu and press <Enter>, or when you use switches to format the file from the command line.

The outcome of upload conversion is a single PC file we will call HOST.UPL, but which you can name as you wish. The executable program is named HOSTUPF.EXE.

HOST.UPL is created so that it can be sent to the host computer. It is created with the exact record size required by your host; or, it may contain carriage return/line feed characters to delimit each record. You enter the file name for the OUTPUT FILE NAME parameter on the “Format upload data into host upload file” menu.

Data unloaded from an HHC is usually packed as binary. The formatter will “unpack” the data into logical records. If you are uploading fixed-length records, each PL/N file header describes the logical records and fields within the file’s records.

---

#### *Upload Data Processing Options*

The Models 4920 and 4921 Series of PC Telecommunication Packages keeps a separate status for the upload and the entire communications session. The Models 4920 and 4921 communications programs mark the upload as good as soon as the download session begins.

Whether or not upload data is processed into HOST.UPL depends on how the "Include uploads from bad sessions" and "Include duplicates" parameters are set in the SYSTEM.CTL file or on the "Format upload data into host upload file" menu. (See Section 6, "Upload Data Format" in Volume A of the *Models 4920 and 4921 Series of PC Telecommunication Packages Users' Guide NPN: 961-021-011* for information on how to set the menu's parameters, how to format upload data from the operating system command line, and how the SYSTEM.CTL file inserts default values.)

The following paragraphs explain the two parameters in more detail.

### *Include Uploads from Bad Sessions Parameter*

HHC upload data marked "bad" following communication is processed into HOST.UPL if the "Include uploads from bad sessions" parameter is set to "Yes." Upload data marked "bad" following communication is *not* processed if the parameter is set "No."

" **NOTE:** *Normally, you would not want to process bad data.*

### *Include Duplicates Parameter*

If the "Include uploads from bad sessions" parameter is set to "No," then an uploaded duplicate is processed from any terminal with more than one good session.

If the "Include uploads from bad sessions" parameter is set to "Yes," then an uploaded duplicate is processed from a terminal with more than one good upload (but it may have had a bad session).

Why can the upload file include duplicate data?

- " If an HHC has a good upload but a bad session, it will attempt to retransmit and the upload data can be duplicated
- " If an HHC has good uploads and bad sessions repeatedly, it may save the upload transactions and send them with the next day's transactions

" **NOTE:** *The Models 4920 and 4921 can only detect if an HHC has more than one set of upload data in the same upload file (UPLOAD.DAT).*



For a host application which supports duplicate uploads, you should set the "Include duplicates" and "Include uploads from bad sessions" parameters to "Yes."

For a host application which does *not* support duplicate uploads, you should set "Include uploads from bad sessions" to "Yes" and "Include duplicate uploads" to "No." You can use this option if the HHC application marks its upload session "good" as soon as it starts its download session. If the HHC never gets an entire good session, the HHC operator should be notified.

If the host application does *not* support duplicate uploads, and the HHC application saves the upload for retransmission until the entire session is good, you should set the "Include uploads from bad sessions" and "Include duplicates" parameters to "No."

**B CAUTION:** *If the HHC session status is good but the host session is bad, then upload can be lost.*

**EXAMPLE:** Data can be lost if a file is missing on the host and the DNLSTT option is disabled.

The following table shows acceptable combinations of the "Include uploads from bad sessions" and "Include duplicates" parameters, and whether or not upload data is processed into the HOST.UPL file (or the file name you choose).

**EXAMPLE:** Suppose "Include uploads from bad sessions" is "Yes" and "Include duplicates" is "No," and the result of the first tcom is a good upload and a bad session. In this case upload data is processed into HOST.UPL, because the application was instructed to process uploads from bad sessions. Upload data *is not* processed during the second tcom, because the application was instructed to *not* process duplicate uploads.

parameter and value		Upload processed into HOST.UPL?			
		bad session good upload (1st tcom)	bad session good upload (2nd tcom)	good session good upload (1st tcom)	good session good upload (2nd tcom)
IUBS	Yes				
ID	No	Yes	No	Yes	No
IUBS	No				
ID	No	No	No	Yes	No
IUBS	Yes				
ID	Yes	Yes	Yes	Yes	Yes
IUBS	No				
ID	Yes	No	No	Yes	Yes

### Host Upload File Record Types

The HOST.UPL file consists of Begin (B), Data (D), and End (E) records. One B record exists for each terminal being converted. Following the B record is a set of one or more D records that contain the upload data being converted. The next B record indicates the end of the previous terminal's data and the start of a new terminal's data. The last record in the HOST.UPL file is the E record.

### Host.UPL File Format

The format of the HOST.UPL file is as follows.

#### Begin (B) Record

Position	Length	Field
01	01	B = Begin record.
02	16	Hand-held computer identifier.
18	08	Date in MMDDYY format.
27	08	Time in HHMMSS, 24-hour clock format.
36	06	Number of fixed-length data records for this terminal. This field is blank if data records are in logical format.

Position	Length	Field
43	01	Duplicate (good) upload flag (Y/N). Y = duplicate, N = not a duplicate (first good upload only).
45	01	Session type. 0 = regular, 1 = initial download.
47	01	Upload status. G = good, B = bad.
49	01	Duplicate (good) session flag (Y/N). Y = duplicate, N = not a duplicate (first good upload only).
51	01	Session status. G = good, HHC received all of its data; B = bad.

### Data (D) Record

Position	Length	Field
01	01	D = Data record.
02		Upload data for the terminal ID specified in the previous B record.

### End (E) Record

Position	Length	Field
01	01	E = End record.

---

### Sample Host Upload File

Following is a sample host upload file with logically formatted PL/N records.

```

B006      04-06-91 12:49:59    N  1 G N G
D<DSCNTRL00001X042>!
DPSTEL0P6  1.00 006          910406123932!
D<DCTMSG 00008X014>!
D10000168706150!
D00020100000210!

```

```

D00019900000990!
D00020100000210!
D10001787061600!
.
.
.
B003      04-06-91 12:53:32      N 1 G N G
D<DSCNTRL00001X038>!
DPSTEL0P6 1.00 003      910406123956!
D<DBTMSG 00008X014>!
D10000168706150!
D00020100000210!
D00020200000220!
D10000268706150!
D10000368706150!
D00030500000340!
.
.
.
B002      04-06-91 12:41:07      N 1 G N G
D<DSCNTRL00001X038>!
DPSTEL0P6 1.00 002      910406124017!
D<DSCNTRL00001X042>!
DPSTEL0P6 1.00 002      910406124017!
E!

```

---

### *Sample Logically Formatted Transaction File*

Following is a sample logically formatted transaction file.

```

B0001      08-01-95 13:55:28      N  G N G!
D<DSCNTRL00001X110>!
DPBQWMDP  V1.1100000001000001051295165400000. . .
D<DCRTRXN00000X001>!
DJ051295+0000000+000000+0217985+0501855+00000. . .
DH00000000100000100000000000000000!
DH00000000100000100000000000000000!

```





# INDEX

This index covers all topics. Sections 1 through 7 are in the User's Guide, Volume A. Sections 8 through 15 are in the User's Guide, Volume B. Letters refer to the Reference Guide.

Numbers in italics are figures.

## NUMBERS

4000 Series. *See* HHCs

492X. *See* Models

498X. *See* Models

## A

Absolute base address, 3-14

Adaptor-descriptor file, 2-7

ADCCP, 1-1, 3-10

- combination port, 9-17
- communication errors, C-1
- default bucket, 12-20
- direct connect, 3-20, 3-31
- HHC ports, 9-2, 9-8
- modem channels, 1-4
- parameters
  - BUCKET, 12-26
  - HIGH address, 12-24, 12-29
  - LOW address, 12-24, 12-28
  - Maxaddr, 9-14
  - Minaddr, 9-14
  - PROTOCOL, 12-22, 12-25
  - SPEEd, 12-23

ports, 7-8, 9-13

Tcom handlers, E-1

Advanced Data Communication Control Procedures. *See* ADCCP

APU, 1-3

ATcomm4 controller, 3-29

ATcomm4 Intelligent Communication Controller, 3-26

ATcomm4 serial interface jumpers as shipped, 3-33

- channel 0
  - direct connect ADCCP, 3-36
  - modem ADCCP, 3-35
- channel 2 direct connect ADCCP, 3-34
- locations, 3-32

Autoanswer activate records, 12-21

- /D, 12-24
- BEGIN, 12-22
- BITs, 12-23
- End, 12-22
- HIGH address, 12-24
- LOW address, 12-24
- PARity, 12-23
- PORT, 12-21
- PROTOCOL, 12-22
- SINGLE, 12-23
- SPEEd, 12-23

Autocall activate records

- host download, 13-7
- parameters, 13-7

session control, 12-24

- /D, 12-29
- BEGIN, 12-26
- BITs, 12-28
- BUCKET, 12-26
- End, 12-27
- EXPECTed, 12-27
- HIGH address, 12-29
- LOW address, 12-28
- PARity, 12-28
- PHONE, 12-25
- PROTOCOL, 12-25
- RETRies, 12-27
- SINGLE, 12-28
- SPEEd, 12-28
- WAIT, 12-27

Auxiliary Power Unit. *See* APU

## B

Batch files, F-2

Boot sets

- downloading, G-3
- loading, G-2
- menu, 7-5, 7-10

Booting

- 4000 Series HHCs, G-1
- system, 3-25

Broadcast records

- host download, 13-6
  - parameters, 13-6
- session control, 12-13
  - /C, 12-17
  - /D, 12-17
  - /E, 12-18

- CREate, 12-17
  - parameters, 12-16
  - STAtus, 12-17
  - TYPe, 12-16
- C
- Cables, 2-2, 3-40, 3-41
- Changing values, 4-9
- Channel 0
  - ADCCP modem, 3-22, 3-37
  - direct connect ADCCP, 3-23, 3-38
- Channel 2 direct connect ADCCP, 3-37
- Checking
  - 4921 to 4980, 4-4
  - communication control, 4-3
  - HHC ports, 4-4
  - system control, 4-6
- Command line shortcuts
  - formatting from command line
    - activation database, 12-39
    - communications, 7-16
    - host upload file, 6-14
    - PL/N files, 5-18
    - request database, 12-38
    - session control file, 5-17
    - upload data by HHC, 6-15
  - formatting from control line,
    - host download file, 5-18
  - retrieving a menu
    - communication status, 7-13
    - download data format, 5-14
    - session control, 12-36
    - upload data format, 6-11
  - usage lines
    - communications, 7-15
    - download data format, 5-16
    - session control, 12-37
    - upload data format, 6-13
- using switches
  - batch files, F-4
  - communications, 7-14
  - download data format, 5-15
  - menu control file, 11-4
  - session control, 12-37
  - upload data format, 6-12
- Communication control file, 9-1
- CPC, 9-3
- ports
  - ADCCP, 9-13
  - bus, 9-8
  - combination ADCCP, 9-17
  - combination TTY, 9-17
  - HHC, 9-8
  - host, 9-4
  - host asynchronous, 9-6
  - NPCP, 9-13
  - quantity, 9-2
  - TTY, 9-15
  - Ymodem, 9-19
- preparation, 4-3
- processing, 9-3
- sample file
  - model 4920, 9-19
  - model 4921, 9-22
- using menu, 9-24
- Communication end parameters
  - Enddays, 10-6
  - Endidl, 10-7
  - Endreq, 10-6
  - Endterms, 10-7
  - Endtime, 10-5
  - Endunsched, 10-7
  - Tcomretries, 10-8
- Communication equipment, 2-2
- Communication Processor Card.
  - See CPC
- Communications
  - boot sets menu, 7-5
  - command line shortcuts, 7-13
  - controls menu, 7-3, 7-9
  - HHCs, 7-1
  - menus, 7-1
  - options, 7-3
  - screens, 7-6
    - data monitor, 7-6, 7-11
    - exit, 7-9, 7-12
    - port status, 7-8, 7-12
    - statistics, 7-6, 7-11
  - status screen, 7-2
- Compatibility with Tcom handlers, E-1
- Connecting cables, 3-39
- Control files, 2-6
- Controllers, 3-11
- Converting HHC application programs, J-8
- CPC, 1-3, 2-2, 3-6, 3-12
  - PS/2, 3-13
  - PS/2 installation, 3-25
- Creating
  - batch files, F-2
  - program command files, J-5
- D
- Data files, 2-6
- Data flow, 1-6, 1-7
- Data format parameters
  - DEXfname, 10-16
  - Dnlfixed, 10-16
  - Dnlpack, 10-17
  - Dnlreclen, 10-16
  - Dnltrunc, 10-17
  - Uplconcur, 10-11
  - UplCrLf, 10-15
  - Upldelimit, 10-14
  - Uplfixed, 10-12
  - Uplincbad, 10-13
  - Uplincbadsess, 10-14
  - Uplincdup, 10-14
  - Upllogical, 10-13



- Uploadback, 10-15
- Uploadfname, 10-15
- Updreclen, 10-12
- Data packing terms, B-1
- Data records, 12-18, 13-6
  - parameters, 12-19
- Definitions
  - communication control file, 9-25
  - communications, 7-3
  - HHC upload data, 6-7
  - host download file, 5-8
  - host upload data file, 6-2
  - main menu, 4-10
  - PL/N download files, 5-11
  - session control
    - activation database, 12-35
    - download files, 5-6
    - request database, 12-33
  - system control file, 10-32
  - wait utility, F-6
- Determining protocols, 3-8
- Disk drive size, 2-2
- Diskettes, 2-4, 3-42
- Display options
  - Alt1color, 10-19
  - Alt2color, 10-19
  - Errorcolor, 10-20
  - Help, 10-21
  - Helpcolor, 10-20
  - Maincolor, 10-18
- DOS. *See* Operating systems
- DOS files, 14-4, K-20
  - HHC changes, K-20
- Download data format, 1-7, 5-1
  - command line shortcuts, 5-14
  - first host input file, 5-2
  - host download file
    - building, 5-3
    - formatting, 5-7
  - PL/N files, 5-10
  - session control file
    - building, 5-2
    - formatting, 5-5
- Download data preparation, 1-9
- Download records, 13-4
  - parameters, 13-4
- Downloading boot sets, G-3
- Dual-ported RAM address, 3-27
- E
  - Ending communications
    - HHC count, 10-4
    - predetermined number, 10-4
    - requests, 10-4
  - methods, 10-3
  - parameters, 10-2, 10-5
  - time, 10-3
- Errors
  - communication
    - ADCCP, C-1
    - TTY, D-1
  - example
    - ADCCP communication, C-4
    - TTY communication, D-4
  - message format, 8-1
  - messages
    - ADCCP, C-2
    - TTY, D-2
  - nonprotocol
    - ADCCP, C-1
    - TTY, D-1
  - numbers and messages, 8-2
  - protocol
    - ADCCP, C-1, C-3
    - TTY, D-1, D-3
  - requirements, 2-8
  - types
    - ADCCP, C-2
    - TTY, D-2
- Executable
  - files, 2-6
  - programs, F-1
- F
  - File directories
    - Bootdir, 10-22
    - Dnldir, 10-23
    - Loaddir, 10-22
    - Upldir, 10-23
  - File identifier records, 13-5
    - parameters, 13-5
    - session control
      - /D, 12-15
      - /E, 12-15
      - CREate, 12-15
      - parameters, 12-14
      - STAtus, 12-15
      - TYPe, 12-14
  - File processing, H-1
    - booting 4000 Series HHCs, H-4
    - formatting records
      - F, H-2
      - N, H-2
    - general operation, H-3
    - to enable, J-4
  - File records, 12-10
    - /D, 12-11
    - parameters, 12-11
    - session control, 12-13
  - Files
    - Adaptor-descriptor, 2-7
    - control, 2-6, 4-1
    - data, 2-6
    - executable, 2-6
    - help, 2-7
    - load, 2-8
  - Formatting records
    - F, H-2
    - N, H-2

## G

General file operation, H-3

Getting help, 4-9

## H

Hand-held computers. *See* HHCs;

Models

Hard drive size, 2-2

Hardware requirements, 2-1

personal computer, 2-1

Help files, 2-7

HHCs, 2-4

4000 Series

boot files, H-4

booting, G-1

booting with Ymodem, 3-10

cold boot capability, 1-5

multiple applications, G-2

single application, G-2

Tcom handlers, E-1

type parameters, 12-10

warm start, J-1

warm start implementation,

J-2

counting methods, 10-4

port parameters

Chans, 9-10

Config, 9-12

Minor, 9-9

Modemtype, 9-11

Port, 9-9

Protocol, 9-10

Reset, 9-12

Speed, 9-11

status parameters, 12-7

upload. *See* Upload, data

Host asynchronous ports

Databits, 9-7

Parity, 9-7

Speed, 9-6

Stopbits, 9-7

Host Computers, 2-4

Host download file, 13-1

activation records, 13-7

building, 5-3

data files, 5-1

formatting, 5-7

parameters

activation records, 13-7

broadcast records, 13-6

data records, 13-6

download records, 13-4

file identifier, 13-5

IDL, 13-5

request records, 13-3

records

comment (#), 13-3

format, 13-2

include file, 13-4

request, 13-3

types, 13-2

sample, 13-9

Host port

Device, 9-4

Port, 9-4

Protocol, 9-5

Host upload file, 15-1

data processing options, 15-1

format, 15-4

include parameters

bad session uploads, 15-2

duplicates, 15-2

record types, 15-4

records

Begin, 15-4

Data, 15-5

End, 15-5

sample file, 15-5

## I

IBM

PC AT. *See* PCs

PS/2. *See* PCs

IBM Personal System/2 (IBM

PS/2). *See* PCs

IDL, 1-5

parameters, 10-9

IDL include records, parameters,

13-5

Initial Download. *See* IDL

Installation, 3-1, 3-2, 3-12

channel 0

ADCCP modem, 3-22, 3-37

direct connect ADCCP, 3-23,  
3-38

channel 2 direct connect

ADCCP, 3-37

network configurations, K-1

preparation, 3-3

summary, 3-1

Interface jumpers, 3-16-3-27

Interrupt level, 3-14

ISA Bus (AT Bus), 3-26

## L

Lite, functionality, 1-4, K-9

Load

boot set parameters, 12-11

boot sets, G-2

files, 2-8

Load boot records

/D, 12-13

parameters, 12-12

Log file, A-1

communications, 7-1

LOG.DAT contents, A-2

parameters, 10-26

update status, A-14

## Log keys

BEGA, A-3  
 BEGD, A-11  
 BEGS, A-5  
 BEGT, A-3  
 DDIR, A-12  
 DVFS, A-13  
 DVFU, A-12  
 EDAT, A-7  
 ENDA, A-4  
 ENDS, A-8  
 ENDT, A-9  
 FILE, A-6  
 LMSG, A-8

## M

Main menu, 4-10

## Menu commands

Entry, 11-3  
 Label, 11-3  
 Menu, 11-2  
 Name, 11-3  
 Program, 11-4  
 Prompt, 11-3  
 Submenu, 11-5

Menu control file, 11-1

example command sets, 11-7  
 menu definition, 11-6  
 operating unattended, F-3  
 option commands, 11-6  
 parameters, 11-2  
 preparation, 4-7  
 sample file, 11-10  
 submenus, 11-8

Micro Channel card installation,  
 K-17

## Mode records

VERsion, 12-6  
 ZONe, 12-5

## Models

4920, 1-3  
 additional new parameters,  
 K-11  
 cables, 2-2  
 configuration, 1-3  
 diskettes, 2-5, 3-3  
 DWNLRQ file servers, J-5  
 hardware, 2-2  
 HHCs, 2-4  
 load files, 2-8  
 new parameters, K-10  
 updating with new applica-  
 tion, J-7  
 4920 multitasking, K-1  
 4921, 1-4, 4-4  
 cables, 2-3  
 configuration, 1-3  
 diskettes, 2-6  
 hardware, 2-2  
 492X lite functionality, 1-4  
 4970 APUs, 1-3  
 4980, 1-4, 2-2, 4-4  
 controller Id, J-2  
 DWNLRQ file servers, J-5  
 file maintenance list, J-2  
 program support, J-8  
 4985  
 controller Id, J-2  
 DWNLRQ file servers, J-5  
 file maintenance list, J-2  
 498X, file maintenance, 12-8

Modems, 2-3, 3-11

Moving around screens, 4-8

MPLD.SYS Device Driver, K-17

MQL, 3-10, 9-2

Multidrop quad lockbox. *See* MQL

Multiple 4920 Micro Channel  
 boards, K-18

Multiterminal products, 3-12

## N

Network Communications Con-  
 troller. *See* Models, 4980

Network configuration, 3-15, 3-27

Network records, 12-8

/D, 12-10  
 DIR, 12-10  
 INIt, 12-9  
 STATus, 12-9  
 TYPE, 12-10  
 ZONe, 12-9

## Networking

configurations, K-1  
 installation, K-1  
 memory requirements, K-9  
 options, K-2

Norand Portable Computer Proto-  
 col. *See* NPCP

NPCP, 1-2, 3-10

HHC ports, 9-2, 9-8  
 jumpers, 3-19, 3-36  
 ports, 7-8, 9-13  
 Tcom handlers, E-1

## O

Operating systems, 2-4, 3-2  
 installation, 3-3

Operating unattended, F-1

OS/2. *See* Operating systems

## P

## Parameters

Background, 10-26  
 broadcast records, 13-6  
 data format, 10-11  
 Dataeof, 10-24  
 Dataeor, 10-24  
 Datarem, 10-24  
 Debuglevel, 10-25

- default phone, 10-27
  - download records, 13-4
  - ending communications, 10-5
  - file directories, 10-21
  - file identifier records, 13-5
  - host download file records
    - activation, 13-7
    - request, 13-3
  - IDL records, 13-5
  - Loglevel, 10-26
  - menu control file, 11-2
  - miscellaneous, 10-23
  - PL/N, 10-25
  - PLNeof, 10-25
  - Restart, 10-27
  - session control file records
    - autoanswer activate (A), 12-21
    - autocall activate (C), 12-24
    - broadcast file (B), 12-16
    - data (D), 12-19
    - file (F), 12-14
    - file set (S), 12-11
    - load boot set (L), 12-12
    - mode (M), 12-4
    - network communication controller (N), 12-9
    - port configuration (P), 12-19
    - terminal identifier (T), 12-7
  - system-wide display options, 10-18
  - Tcomquery, 10-27
  - terminal identifiers, 10-8
- PCs, 2-1, 3-2
- IBM PC AT
    - cables, 3-39
    - CPC, 3-26
    - installation, 3-27
  - IBM PS/2
    - adaptor-descriptor file, 2-7
    - configuration, 3-15
    - CPC installation, 3-25
    - CPC running DOS, 3-14
    - CPC running OS/2, 3-13
    - installation, 3-12
    - interface jumpers, 3-16-3-27
  - ISA PC
    - configuration, 3-30
    - CPC installation, 3-38
    - interface jumpers, 3-31-3-40
- Personal Computers. *See* PCs
- Phone parameters
- Errlog, 10-29
  - Netchkcnt, 10-29
  - Netflag, 10-29
  - Nohalt, 10-28
  - Phoneretrieys, 10-27
  - Phonewait, 10-28
- PL/N files
- ADCCP errors, C-2
  - categories, B-4
    - fixed files-fixed records, B-4
    - variable file-fixed records, B-5
    - variable files-variable records, B-6
  - data conversion, B-8
  - data packing
    - terms, B-1
    - use of, B-7
  - download, 5-10
  - download data files, 5-1
  - headers, B-1
    - arrays, B-7
    - format, B-2
  - parameters, 10-25
    - format, 14-2
    - Rectype, 14-2
  - TTY errors, D-2
  - upload, 14-2
  - upload data conversion, B-9
  - upload file name, 14-1
- Port configuration records, 12-19
- BUCKET, 12-20
  - DIAL, 12-20
  - PORT, 12-20
- Ports
- ADCCP, 9-13
  - bus, 9-8
  - combination ADCCP, 9-17
  - combination TTY, 9-17
  - HHC, 9-8
  - host, 9-4
  - host asynchronous, 9-6
  - NPCP, 9-13
  - quantity, 9-2
  - TTY, 9-15
  - Ymodem, 9-19
- Preparation, 3-2
- changing
    - control file values, 4-1
    - FILES values, 3-3
  - control files
    - communication, 4-3
    - menu, 4-7
    - system, 4-5
    - upload format, 4-7
  - menus
    - main, 4-10
    - using the, 4-8
    - wait, 4-13
  - user manuals, 3-3
- Procedures
- communication control file, 9-25
  - communications, 7-9
  - files
    - host download, 5-9
    - host upload data, 6-5
    - PL/N download, 5-13
    - system control, 10-33
  - HHC upload data, 6-9
  - main menu, 4-12
  - session control
    - activation database, 12-36
    - download files, 5-7
    - request database, 12-34
  - wait utility, F-6

- Processing files. *See* File processing
  - Programs, F-1
    - command file, J-5
    - command lines, F-2
    - menu control commands, F-3
    - support, J-8
  - Protocol characteristics, 3-10
  - Protocols, 3-8
  - PS/2 serial interface jumpers
    - as shipped, 3-18
    - channel 2 direct connect
      - ADCCP, 3-21
    - standard locations, 3-17
  - PScomm4 Intelligent Communication Controller, 3-13
  - PScomm4 processing engine, 3-19
  - PScomm4 serial interface jumpers, channel 0
    - direct connect ADCCP, 3-24
    - modem ADCCP, 3-23
- R**
- Recovery requirements, 2-8
  - Request records
    - /D, 12-8
    - STatus, 12-7
    - terminal identifier, 12-6
    - VERsion, 12-7
    - ZONe, 12-7
  - Requirements
    - 492X, 2-8
    - errors, 2-8
    - hardware, 2-1
    - host computer, 2-10
    - memory, 2-9, 2-10
    - networking memory, K-9
    - personal computer, 2-1
    - publications, 2-11
  - recovery, 2-8
- S**
- Sample files
    - communication control
      - model 4920, 9-19
      - model 4921, 9-22
    - control
      - menu, 11-10
      - session, 12-30
      - system, 10-30
      - upload format, 14-3
    - host, upload, 15-5
    - host download, 13-9
  - Session control file, 12-1
    - building, 5-2
    - command line shortcuts, 12-36
    - download data files, 5-1
    - formatting, 5-5
    - operating unattended, F-2
    - record parameters
      - autoanswer activate (A), 12-21
      - autocall activate (C), 12-24
      - broadcast file (B), 12-16
      - data (D), 12-19
      - file (F), 12-14
      - file set (S), 12-11
      - load boot set (L), 12-12
      - mode (M), 12-4
      - network communications controller (N), 12-9
      - port configuration (P), 12-19
      - terminal identifier (T), 12-7
  - records
    - activation, 12-19
    - broadcast, 12-16
    - comment (#), 12-4
    - description, 12-3
    - files, 12-13
    - format, 12-1
    - mode (M), 12-4
  - request, 12-6
    - types, 12-2
  - sample, 12-30
    - using database menu
      - activation, 12-34
      - request, 12-32
  - Setting
    - I/O base address, 3-28
    - version number, J-7
  - Software requirements, 2-4
  - Supporting programs, J-8
  - System control file, 10-1
    - ending communication methods, 10-3
      - parameters, 10-2, 10-5
    - formatting parameters, 12-1
    - parameters
      - background, 10-26
      - data format, 10-11
      - default phone, 10-27
      - ending communications, 10-2
      - file directories, 10-21
      - HHC counting methods, 10-4
      - levels, 10-25
      - miscellaneous, 10-23
      - restart communications, 10-27
      - system-wide display options, 10-18
      - terminal identifiers, 10-8
    - preparation, 4-5
    - sample, 10-30
    - using menu, 10-32
- T**
- Tcom handlers, E-1
  - Telecommunications network. *See* Networking
  - Teletype. *See* TTY
  - Terminal identifiers
    - IDLCheck, 10-9

- IDLPos, 10-10
- Termidlen, 10-9
- Termidpos, 10-9
- VersCheck, 10-10
- VersLen, 10-11
- VersPos, 10-10
- Time zone. *See* TZ
- Transaction file, 15-6
- TTY, 1-1, 3-10
  - combination port, 9-17
  - communication errors, D-1
  - default bucket, 12-20
  - HHC ports, 9-2, 9-8
  - modem channels, 1-4
  - parameters
    - BITs, 12-23, 12-28
    - BUCKet, 12-26
    - PARity, 12-23, 12-28
    - PROtocol, 12-22, 12-25
    - SINGle, 12-23, 12-28
    - SPEed, 12-23
  - ports, 7-8, 9-15
  - Tcom handlers, E-1
- TTY ports
  - Databits, 9-16
  - Parity, 9-16
  - Single, 9-16
  - Stopbits, 9-17
- TZ, 3-4
- U**
- Unattended operation, F-1
- Upload data, 1-6
  - PL/N conversion, B-9
- Upload data format, 1-10
  - command line shortcuts, 6-11
  - HHC, 6-6
  - host file, 6-1
- Upload format, K-20
  - concurrent, K-11
  - memory management, K-16
  - OS/2 installation, K-17
  - Windows 3.X, K-16
- control file, 14-1
  - DOS, 14-4
  - PL/N, 14-2
  - sample, 14-3
- Upload format control file, control file, preparation, 4-7
- Upload from diskette to PC, 3-42
- Upload processing, 1-10
- Upload records, 13-3
  - parameters, 13-3
- Usage line notation
  - data
    - download, 5-16
    - upload, 6-13
  - HHC, 7-15
  - session control, 12-38
- Usage lines
  - data
    - download, 5-16
    - upload, 6-13
  - HHC, 7-15
  - session control, 12-37
- Using
  - menus, 4-8
  - message line, 4-10
- W**
- Wait utility, F-4
  - menu, F-6
- Warm start, J-1
  - controller Id, J-2
  - implementation, J-2
- Y**
- Ymodem, 1-2, 3-10
  - HHC ports, 9-2, 9-8
  - modem channels, 1-4
  - ports, 7-8, 9-19
  - Tcom handlers, E-1