REFERENCE

831164   001N

Data Transfer Between a Hewlett-Packard 9845 Computer and a VAX 11/780 Computer

Nancy K. Sulinski
Computer and Information Services Department

REFERENCE ONLY

Naval Underwater Systems Center
Newport, Rhode Island/New London, Connecticut

		Form Approved
	Report Documentation Page	OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **05 DEC 1983**	2. REPORT TYPE **Technical Memo**	3. DATES COVERED **05-12-1983 to 05-12-1983**	
4. TITLE AND SUBTITLE **Data Transfer Between a Hewlett-Packard 9845 Computer and a VAX 11/780 Computer**		5a. CONTRACT NUMBER	
		5b. GRANT NUMBER	
		5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) **Nancy Sulinski**		5d. PROJECT NUMBER **771Y00**	
		5e. TASK NUMBER	
		5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Naval Underwater Systems Center,New London,CT,06320**		8. PERFORMING ORGANIZATION REPORT NUMBER **TM 831164**	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S)	
		11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT **Approved for public release; distribution unlimited**			
13. SUPPLEMENTARY NOTES **NUWC2015**			

14. ABSTRACT
With a variety of computers and software in existence today, all data processing needs for a given task may not be available on one computer system. Thus, it sometimes becomes necessary to transfer data or routines from one system to another. This memoranda provides a method for transporting data and program files between a Hewlett-Packard 9845 computer and a VAX 11/780 computer.

15. SUBJECT TERMS
data transfer; HP 9845; VAX 11/780; Special Projects and Studies

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**33**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

Report Documentation Page

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
05 DEC 1983	**Technical Memo**	**05-12-1983 to 05-12-1983**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Data Transfer Between a Hewlett-Packard 9845 Computer and a VAX 11/780 Computer	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
Nancy Sulinski	**771Y00**
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
Naval Underwater Systems Center,New London,CT,06320	**TM 831164**

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
NUWC2015

14. ABSTRACT

With a variety of computers and software in existence today, all data processing needs for a given task may not be available on one computer system. Thus, it sometimes becomes necessary to transfer data or routines from one system to another. This memoranda provides a method for transporting data and program files between a Hewlett-Packard 9845 computer and a VAX 11/780 computer.

15. SUBJECT TERMS
data transfer; HP 9845; VAX 11/780; Special Projects and Studies

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**33**	

NAVAL UNDERWATER SYSTEMS CENTER
NEW LONDON LABORATORY
NEW LONDON, CONNECTICUT 06320

Technical Memorandum

TRANSFER OF DATA BETWEEN

A HEWLETT-PACKARD 9845 COMPUTER AND A VAX 11/780 COMPUTER

Date: 5 December 1983

Prepared by: *Nancy K. Sulinski*
Nancy K. Sulinski

Computer and Information
Services Department

ABSTRACT

With a variety of computers and software in existence today, all data processing needs for a given task may not be available on one computer system. Thus, it sometimes becomes necessary to transfer data or routines from one system to another. This memoranda provides a method for transporting data and program files between a Hewlett-Packard 9845 computer and a VAX 11/780 computer.

ADMINISTRATIVE INFORMATION

This memorandum was prepared under Job Order No. 771Y00, Special Projects and Studies. The author is located at the Naval Underwater Systems Center in New London, Connecticut.

TABLE OF CONTENTS

APPENDIX--PROGRAM AND DATA LISTINGS

I. INTRODUCTION

Owners of Hewlett-Packard desktop computers have many uses for their machines. The size of the equipment has allowed these computers to be taken aboard submarines and surface ships so that various types of data could be collected. However, even with the versatility of the desktop models, a need occasionally arises for which the desktop computer is not quite suited or software for processing data exists on a different computer. Essentially, the data has been collected and stored by a Hewlett-Packard computer, but the processing needs for the data reside on some other computer system.

Additionally, the reverse situation may be true. Data may have to be transported from a host computer to the Hewlett-Packard desktop computer. This may be necessary so that generated data may be used on board ship or the reason may be simply one of economics--the user may find it cheaper to transfer applications to the desktop computer.

This memorandum sets forth procedures that can be used to transfer data between a Hewlett-Packard desktop computer and any computer with time share capabilities. The examples presented will specifically address the interfacing of a Hewlett-Packard 9845B computer with a VAX 11/780 computer. The outlined procedures can be adapted to interface any Hewlett-Packard model with a terminal emulator program to any host computer meeting the requirements of the emulator program. Procedures to transfer data in the two directions between a host computer and a Hewlett-Packard system are presented.

II. OVERVIEW

Transferring data to or from the Hewlett-Packard computer requires that the desktop computer be equipped with a terminal emulator program that will allow the HP computer to perform data communications with a host computer. Hewlett Packard markets such software and the necessary hardware. In the specific case of the HP 9845 computer, the software is called the High Speed Asynchronous Terminal Emulator and requires the HP 98046 Datacomm Interface and HP 98417A Datacomm ROM.

The High Speed Asynchronous Terminal Emulator Program has various escape codes that will allow the host computer to access the local mass storage devices of the Hewlett-Packard

system, such as the cassette drives and floppy disc drives. The data being transferred from the HP computer to the host computer must be in ASCII string format rather than floating point format. A program written for the host computer can execute the necessary escape codes, read the data from HP mass storage (provided it is stored as ASCII strings), and then write it to a data file on the host computer's mass storage system. These escape codes may be executed via the keyboard and predefined special function keys of the Hewlett-Packard Emulator Program.

Two means of data transfer are presented in this memorandum. One method involves the use the special function keys defined by the Terminal Emulator Program. The other method involves the use of routines written for execution on the host computer. The keyboard/special function key means of data transfer may be suitable for the transfer of some data files. However, the use of a program residing on the host computer gives the user greater control over the data format and reduces the number of keyboard steps that must be performed. Additionally, if several files need to be transported between machines, it becomes quite convenient to place the program in a command file (.COM file on the VAX) thus reducing the repetition of typing the sequences and, if nothing else, it reduces the tedious typing that could occur along with frustrating typing errors.

The bulk of this memorandum concerns itself with the creation of programs for the host computer to perform the tasks for transferring data between the Hewlett-Packard desktop computer and the host computer. The keyboard/special function key method is addressed briefly in its own section.

For the host program method, this memorandum supplies sample programs illustrating the various steps necessary to transfer data. Sample data is created on the HP 9845 and the data is traced through to its final destination on the VAX 11/780. This data is then transported back to the HP 9845 to illustrate the two way transportation of data. The reader can verify the procedures by typing in the supplied programs and running through them from start to finish. An attempt was made to make the sample programs as general as possible so that few modifications would be necessary to tailor the routines to the specific needs of the user's actual data.

A convention is used in this memorandum to distinguish between computer programs written for the HP 9845 and programs written for the VAX 11/780. Hewlett-Packard programs are enclosed in quotation marks (e.g. "NUM2ST"), while VAX programs are written in uppercase letters with no quotation marks (e.g. VAX2HP).

2

The memorandum assumes that the reader has an understanding of BASIC (specifically HP BASIC and VAX BASIC), the HP 9845 system, and the host computer (the VAX).

III. OPERATION OF THE TERMINAL EMULATOR PROGRAM

Hewlett Packard's High Speed Terminal Emulator Program works rather well in converting the HP 9845 desktop computer into a terminal for a host computer. Proper operation is based on the proper hardware connections and software settings within the program. As mentioned earlier, the 9845 must be equipped with a Datacomm ROM and the Datacomm Interface card. The interface card can be connected to either a modem or a direct line to the host computer.

The High Speed Terminal Emulator Program is really a collection of three computer programs that can be used to convert the desktop computer to a terminal. The character mode program has been used to transfer data from the HP 9845 to the VAX 11/780. After ensuring the proper hardware connections and that the mass storage media containing the emulator program is available for use, the user should type, LOAD "CHRMOD", on the HP keyboard and press the EXECUTE key. Once the program has been loaded, press the RUN key. The Emulator Program will perform some initialization operations. If a modem is being used, a "Waiting for Connection" message will appear on the screen of the HP computer until a phone connection is made. When all proper connections have been made, a "Terminal Ready" message will appear. For more detailed information, refer to Hewlett-Packard's High Speed Asynchronous Terminal Emulator manual for the HP 9845B/C.

The Emulator Program will not work unless the proper software settings are made by the program. The proper settings concern the characteristics of the host computer and the interface between the desktop and host. By pressing the Edit special function key (k_{13}), the bottom of the CRT screen will display the default configuration. The displayed values are altered by positioning the cursor beneath the specification and pressing the STEP key until the figure below the specification changes to the desired value. When using a VAX computer as the host, the number of bits per character is changed to 8, the number of stop bits must be changed to 2, and the parity set to none. There is no handshake between the VAX computer and terminals. If the modem is being used, the setting beneath that specification is on. If a direct line is used, the modem setting is off.

The speed specification must be changed to reflect the baud rate being used. The following set up is used for a 2400 baud hard line to a VAX computer.

Edit mode: _____ <--=left -->=right STEP=select   STORE=exit

Hardcopy	HPIB	Screen	Bits/char	Stop bits	Parity	Speed	Echo	Handshk	Modem
0	off	16	8	2	none	2400	on	none	off

One note of caution is included here. The Emulator Program can have difficulty with large amounts of information coming in from the host computer. The Emulator checks each character as it is received from the host. At high baud rates, it is unable to keep pace with the rate at which information is being sent. An error message will occur when such a data overrun occurs and the Emulator Program will stop. Merely re-run the Emulator Program and the connection back to the host and the current job will be made. If a phone line is being used, this connection may also have to be re-established. An easy way to check the possiblility of data loss from a VAX computer to the HP computer is to print a large file or directory to the screen after logging on. Enough information is sent so that data loss would be apparent. Lowering the baud rate of the connection between the two computers will prevent the loss of data. It has been found that 300 baud is a "safe" speed and will not cause overruns, though rates of 1200 and 2400 baud have worked successfully.

Once the proper settings (parity, baud rate, etc.) have been entered, press the STORE key to store all the values. A new set of options will appear at the bottom of the screen. If the host program method of transferring data is to be used, these settings can be ignored. The computer program can make the proper adjustments and the default values can be left as they are by pressing the STORE key a second time.

If data transfer is to be accomplished by using the special function keys defined by the Emulator Program, some of the settings of the second portion of the edit mode will have to be altered as appropriate. The first specification at the far left of the screen indicates the Hewlett-Packard mass storage file and device it resides on that is to be used in the data transfer operation. This data file will be used to transport data to (or from) the host computer when the special function key method of data transfer is to be used.

The file name would be modified accordingly. Size is the specification that shows the number of records that will be defined when a new data file is created ·by the Emulator Program. Once the settings have been altered, they are stored by pressing the STORE key.

Once the proper settings have been made in the Edit mode and are stored (the STORE key has been pressed twice), the HP 9845 is ready to act as a terminal. In the case of the VAX 11/780, pressing the STORE key a third time should give an indication that a connection has been completed to the host computer, such as a request for the Username.

For the program method, the transfer programs (as supplied in the Appendix) can now be executed to move the data from one machine to the other. For the keyboard/special function method, the appropriate keys and commands can now be entered.

IV. USE·OF SPECIAL FUNCTION KEYS TO TRANSFER DATA

Details for using the keyboard/special function keys can be found in the Hewlett-Packard High Speed Terminal Emulator Program manual. Briefly, the method is outlined here.

On the upper right of the HP 9845 keyboard are the various special function keys that have been defined by the Emulator Program, such as Remote (k_0), Create (shift k_{12}), and Record (shift k_{15}). These keys, in combination, can execute the commands to transfer data.

The Remote key is a toggle key which will switch the terminal between local and remote mode. The HP is connected to the host computer while in remote mode. While in remote mode, anything typed in via the keyboard is taken as a command to the host computer. This is the default. In local mode, the HP will respond to escape code sequences entered via the keyboard or via special function keys related to mass storage operations. The escape code sequences can accomplish data transfer operations and other mass storage operations. These sequences are summarized in one of the Appendices of the High Speed Asynchronous Terminal Emulator manual.

Pressing the Create special function key while in local mode will create a Hewlett-Packard mass storage file according to the specifications in the second portion of the Edit mode mentioned earlier. Pressing the Record key while in local mode will allow data to be recorded on the specified mass storage file. Execution of the Record key will cause

all subsequent information from the host computer to be placed on the specified mass storage medium (when in remote mode). Thus, a data file on the VAX could be written to a HP mass storage device by using the TYPE command (from the VMS operating system) on the VAX. The information normally seen on the screen will be printed to the file. This information would also include the TYPE command itself since the VAX echos the information sent to it. However, the inclusion of the TYPE command can be avoided by entering the TYPE command without the final carriage return. Local mode is entered and the Record key is pressed. Once control has returned to the host computer by pressing the Remote key a second time, the STORE key can be pressed, thus sending the final carriage return to complete the TYPE command. This carriage return will be a part of the Hewlett-Packard data file, but the remainder of the TYPE command will not. At the end of the data transfer from the host computer, the Record key is pressed (while in local mode) to stop any future data transfer to the Hewlett-Packard mass storage file. If the data transfer is not stopped in this manner, subsequent commands to the host will be placed in the data file; all future commands to the VAX will be printed to the HP mass storage file due to echoing.

A Hewlett-Packard mass storage file can be sent to the host computer with the Upload (shift k_{14}) special function key. The data file specified in the second portion of the Edit mode contains the data that will be sent to the host. The host will treat the information being sent from the file as if it were being typed in via the keyboard.

The escape code sequences described in the Terminal Emulator manual can be entered directly via the keyboard instead of using the special function keys and their default values specified in the Edit mode. The escape character is special function key k_{12}. The desired commands are typed in while in local mode and then control is returned to the host computer by pressing the Remote special function key.

There are a few things to note about transferring data by executing the escape sequences via the keyboard or special function keys rather than through the use of a computer program. Various steps do have to be performed for the transfer of data to be done properly. These would include such things as making sure the proper file specifications are set in the second portion of the Edit mode for each data file transfer, and that each command necessary to perform the transfer is typed properly. Additionally, since the VAX computer does echo data, a record of commands typed to the VAX, in addition to the actual data, could occur in a

Hewlett-Packard data file if the user is not careful. With the keyboard/special function key method of data transfer, the user does not have complete control over the contents of the data files and must be more attentive to what is being done.

V. <u>DATA CREATION</u>

The remainder of this memorandum is concerned with the program method of transferring data between the HP 9845 computer and the VAX 11/780. The method is presented as an exercise. Data is created on the Hewlett-Packard computer and then programs supplied in the Appendix are executed using this data. The reader, with access to an HP computer with terminal emulator capabilities, can follow the procedure step by step to verify the method.

Listing A, in the Appendix, contains a short program, "CREATE", written in HP BASIC. It will produce a data file consisting of one string and twenty real numbers. Use this program to produce a trial data file to follow the transfer procedure step by step. The sample program produces a data file called "ORIG" and writes one string, "Original Data", and twenty "random" values to the data file.

Listing B contains an HP BASIC program, "READ", that will simply read and print values from a data file. The "READ" program can handle any Hewlett-Packard data file consisting of a series of string values followed by a series of real values. Often a data file will contain some clarifying information, such as title, date, and time, which is stored as strings. This information is then followed by the actual data. The "READ" program can read and print the data file as long as all the strings are written in the first part of the data file and the real values are at the end. Use the "READ" program to examine "ORIG", the data created by the "CREATE" program mentioned earlier. This output is printed as Listing C in the Appendix.

The "READ" program will ask the user to enter the name of the data file to be read. It then reads string values until the TYP function returns a value other than 2, 8, 9, or 10. TYP is a Hewlett-Packard supplied function that returns a value which indicates what type of data will be accessed next in the specified file. Thus, in the "READ" program, TYP(2) refers to the input file assigned to unit 2. Values of 2, 8, 9, or 10 returned by TYP indicate that a string value is the next data item. When the TYP function indicates that a string is not the next data item in the file, the

program proceeds to read the numeric values. After all data has been read and printed, a summary of the number of string values and numeric values will be printed.

VI. ENSURING ASCII DATA FOR HOST COMPUTER

The High Speed Terminal Emulator Program requires that the data to be transferred to the host computer be in ASCII string format. Often, scientific applications store data in floating point format. Thus, a simple conversion of the data must be performed before it can be transferred to the host computer. Hewlett-Packard BASIC supports the VAL$ function which will convert a numeric value into a string value. A short program utilizing this function is written to translate the data and store it in a new data file that contains only string values.

Listing D contains an HP BASIC program, "NUM2ST" (NUMber to STring), that will convert real values to string values. "NUM2ST" requires that the format of the data file be the same as the format required by the "READ" program. The data must be a series of strings followed by a series of real values. Use the "NUM2ST" program with "ORIG" as the input file to create a data file consisting of strings. Name the new data file "NEWSTR".

The "NUM2ST" program assumes that no string value will be longer than 20 characters. It also assumes that there will be no more than 2000 string values or 2000 real data values. These assumptions can easily be changed by altering the dimensions of the String$ and Data arrays. The "NUM2ST" program firsts solicits the name of the original data file, including the mass storage unit specifier. This is the file consisting of strings and numerics. It secondly solicits the name for the new data file (including the mass storage unit specifier) that will contain only strings.

"NUM2ST" will read in all the string values in the file until the TYP function indicates that the next data item is not a string value. As each string value is read, the number of bytes required for its storage is added to the Bytes variable. String values require one byte of storage for each character and four "overhead" bytes for each defined record it is stored in. Allocation of bytes is based on the string being stored in only one defined record. If extra bytes are needed, they will, in all likelihood, be taken care of later when creating the new file—though, this is not a guarantee as will be discussed later. If the maximum allowable length of a string is increased from the 20 character limit set in

the "NUM2ST" program, the computation of byte requirements for string storage must also reflect the increase.

The "NUM2ST" program will read in the real values after all strings have been read in. Each numeric value will require 18 bytes of storage space when stored as a string value. One byte is required for each of the 12 digits, one byte for the decimal point, and one byte is required for any negative sign. With the four byte overhead requirement for strings, the total is 18 bytes of storage for each real number.

"NUM2ST" creates a new data file using the default defined record length of 256 bytes. The number of records required for storage of the string values and the converted numeric values must be computed. The number of records required will be the total number of bytes required for the data values divided by 256 bytes. The CREATE statement will round the computed number of records to the next larger even integer. Thus, the extra bytes required when storing a string in more than one defined record will probably be taken care of when the computed number of records is rounded upwards. A problem will arise if the estimated number of bytes came very close to an even multiple of 256--the created data file would not be large enough to contain all the data. If such a possibility frightens the user, the "NUM2ST" program can be altered to add bytes to the allocation estimates for each string value and numeric value.

Once the new data file has been created, the strings and converted numerics are written to the new file. The VAL$ function is used to translate the real values to string values. A summary of the number of strings, number of numeric values, and the total number of values in the file is printed out at the end of the program.

Use the "READ" program to verify that a data file created by "NUM2ST" contains only string values. The "READ" program should verify that "NEWSTR" (the translated "ORIG" data file) contains 21 string values and no numerical data.

VII. SENDING DATA FROM THE HEWLETT-PACKARD SYSTEM

Entering the data from the Hewlett-Packard computer system into the host computer's mass storage system requires a program residing on the host computer, in addition to the use of the Terminal Emulator Program. The program on the host computer will execute the escape codes that will signal the Hewlett-Packard desktop computer to send data from its mass storage device. The host computer will receive the data as if it is being entered via the keyboard.

Listing E contains a program, HP2VAX, (Hewlett-Packard to VAX) written in VAX BASIC that will read data from any 9845 mass storage device and store the information in a data file on a VAX computer. HP2VAX requires that the data in the mass storage device be stored in ASCII format and that the program be run via the HP Terminal Emulator Program. The program also requires that the user enter the number of values that must be transferred from the HP desktop computer to the host computer. (This is the number of strings returned by "READ".)

HP2VAX first solicits the name of the data file residing on the Hewlett-Packard system. The second input is the mass storage unit specifier for the HP device, such as :T15 for the right hand cassette drive, :H8,0,0 for the left hand floppy disk drive of the HP 9885, or :C12 for the HP hard disk drive. The HP2VAX program requests the number of values that are to be transferred from the HP system to the host. Lastly, the program requests the name of the data file on the VAX where the HP data will be stored.

The program ensures that the mass storage unit specifier for the HP mass storage device contains a colon as the first character. It opens the VAX data file and assigns unit 3 to the file.

The HP2VAX program executes the necessary escape codes to let the HP computer know that it is to send data from the mass storage device. CHR$ is a VAX supplied function that returns the ASCII string character which corresponds to the numeric expression. CHR$(27) is the escape character and CHR$(13) is a carriage return. The first PRINT statement identifies the mass storage file and the device it resides on with the "ASSIGN SOURCE" command. The second print statement readies the computer for the transfer of data with the "ENABLE SEND" command. Further information on these and other escape code commands to the Hewlett-Packard desktop computer can be found in the HP High Speed Asynchronous Terminal Emulator Program manual.

The data is actually transferred from the HP computer to the VAX within the FOR-NEXT loop. The INPUT statement in the HP2VAX program is the request by the host computer for data. The escape codes have set up the automatic input of data from the HP mass storage file. The data is read and then printed to the VAX data file. This procedure continues until all the data has been transferred. A summary statement appears at the end to indicate the number of values that have been successfully transferred to the host computer.

Use the HP2VAX program on the VAX to transfer the data contained in the "NEWSTR" file created earlier on the Hewlett-Packard system. Put the data in a VAX file called HPDATA.DAT. The data file on the host computer can be examined by using any available text editor, or simply using the VAX command, TYPE, to list the contents of the file.

VIII. USING THE DATA ON THE HOST COMPUTER

The data file created by the HP2VAX program was created by printing BASIC string values. These are simply ASCII characters. This format poses no problem for further analysis of the data. The VAX data file can now be easily read by any other program the user desires. The values can be read into a new program as real numbers even though they were entered into the file as characters. Listing F contains a FORTRAN program that will read the specified number of strings and numeric data. Note that the supplied FORTRAN program uses a double precision variable for reading the "string" data. The extra precision ensures that all the digits of the value in the data file are retained when read in to the FORTRAN program. Use this FORTRAN program to verify that the data file created by the transfer program, HPDATA.DAT, contains one string and twenty real values and that printing the values as strings from a BASIC program and reading them as numeric values in a FORTRAN program really poses no difficulty.

IX. TRANSFERRING A PROGRAM FILE FROM THE HEWLETT-PACKARD

Although the preceding process specifically addressed the transfer of data such as numerics, the transfer of computer programs is also possible. The outlined method can also accomplish this task. The supplied HP2VAX program is run on the host computer through the Emulator Program using the HP program file name as the name of the data file. The only point that needs to be mentioned is that the program to be transported from the Hewlett-Packard must be a SAVE'd file

rather than a STORE'd file. Only program files that have
been stored on mass storage with the HP SAVE command can be
accessed as data. STORE'd files can not be used in this
manner. The SAVE command stores the program as an ASCII
file. Since the HP2VAX program can transport ASCII files to
the host computer, a program file that has been stored with
the HP SAVE command can be transported from the
Hewlett-Packard computer in the same manner as any ASCII data
file. Any STORE'd program file must be re-stored with the
SAVE command before it can be transported. The HP2VAX
routine can handle any number of lines and any number of
characters per line.

Even though a program can be transported in this manner,
it does not mean that the program will compile or execute on
the host computer. For example, the BASIC language used on
Hewlett-Packard systems does not agree entirely with the
BASIC on the VAX systems. One difference is found with the
concatenation operator. On the HP, two strings are joined by
the "&" symbol, whereas the VAX uses a "+" symbol for the
same operation. The file assignment syntax differs on the
two machines. To assign a mass storage data file referred to
by the variable File$ to unit three, the HP 9845 computer
would use the statement, ASSIGN File$ TO #3. On the VAX
computer, this same operation would be performed by,
OPEN File$ FOR OUTPUT AS FILE #3. Thus, only the source code
is transported using the outlined procedure. No allowance
has been made for syntax differences between HP BASIC and the
host computer's BASIC. A transported program may have to be
modified once it is on the host computer.

As an added exercise for the reader, SAVE a program on
an HP mass storage device. Use the "READ" program to verify
that the SAVE'd program file contains only strings. Use the
HP2VAX program to transport the program from the HP computer
to the VAX 11/780. Use the TYPE command of the VAX to list
the program. The transported program looks just as it did on
the HP system. Try to compile the program with the BASIC
compiler on the VAX. Chances are the source code was not
error free. Some work would be needed to get the program to
compile and subsequently run properly on the VAX system. As
an exercise in frustration, when time is available, make
modifications so that the program will execute on the VAX
system.

X. TRANSFERRING DATA AND PROGRAMS FROM THE VAX 11/780

The reverse procedure, sending data from the VAX 11/780 to the Hewlett-Packard desktop computer is a fairly straightforward task. With the HP Terminal Emulator Program set up as described earlier, a program is executed on the VAX system to send the appropriate escape codes to create a data file on a Hewlett-Packard mass storage device and then send the VAX data to this file. Listing G is the VAX BASIC program, VAX2HP (VAX to Hewlett-Packard), that can be used to transport data or a program file to the Hewlett-Packard computer.

VAX2HP solicits the name of the source file on the VAX, the destination file on the Hewlett-Packard computer, the mass storage device of the HP system, the total number of lines of data, and the number of characters per line, that will be transferred to the HP desktop computer. After ensuring that a colon precedes the mass storage unit specifier and after opening the source file on the VAX, a series of escape codes are executed.

The user has one more input when using the VAX2HP program. The user has the option of adding line numbers to the front of each line of data, or each line of the program being transferred. For most applications, the answer to this option would be "N", so that no line numbers would be added. BASIC programs already have line numbers, and usually data files are to remain as they are and the user would not want additional line numbers placed in the file. The line number option is included mainly for those wishing to transport a program file, such as one written in FORTRAN, that will be converted to BASIC at a later time.

The first escape code sequence, the PRINT statement containing "CREATE FILE", creates a data file on the specified HP mass storage device. The length of this file is determined by multiplying the number of characters per line plus a four byte "overhead" for strings, by the number of lines of data. If an additional line number is to be included, five bytes are added to the estimate for each line--one byte for each of the four digit line numbers and one byte for the space between the line number and the actual line. This is merely a broad estimate of the needed space on the HP mass storage device. The final total number of records will be rounded to the next larger even integer.

If a program file is to be transported with the VAX2HP routine, the Hewlett-Packard file size is easy to estimate. An assumption of 80 characters per line could be made. When

transporting a program file, the number of lines of data
would be the actual number of lines in the program. Again,
the transporting a program file merely transfers the source
code. The presented procedure in no way ensures the program
will execute properly on the Hewlett-Packard computer.

All information transferred from the host computer is
placed in a data file as ASCII strings. There is a slight
difference on the Hewlett-Packard 9845 between a data file
containing actual data and a data file containing a program.
The program file contains line numbers and the data file does
not. This one difference allows the user to GET a program
data file and load it into computer memory. Data files
containing data (and thus, no line numbers) can not be loaded
into memory with the GET command of the HP. Thus, if a
FORTRAN file is being transported from the host computer,
line numbers must be added to the front of each line of code
as the FORTRAN source code does not contain a sequence line
number for each line of code. Without the added line
numbers, the FORTRAN source code would be treated solely as a
data file and could not be accessed with the GET command. A
"Y" response to the aforementioned option will insert the
needed line numbers so the HP program data file can be
accessed from mass storage with the GET command, allowing for
further modifications.

After creating the HP mass storage file, VAX2HP will
execute escape codes to assign the file ("ASSIGN DESTINATION
FILE"), and enable writing onto the file ("ENABLE RECORD").
The FOR-NEXT loop will control the data transfer. Data is
read from the VAX file and printed to the HP mass storage
device. The VAX2HP routine writes the data as if it is
writing to the terminal. The escape codes have notified the
HP computer that the data is to be written to the mass
storage device and not to the CRT screen. The execution of
the "DISABLE RECORD" escape code stops the printing to the
mass storage device. Subsequent messages will be printed on
the CRT screen, such as the file message indicating the total
number of values transferred from the VAX computer to the
Hewlett-Packard machine.

Execution of VAX2HP places data from the VAX computer
onto a Hewlett-Packard mass storage device. Use VAX2HP to
transport HPDATA.DAT to the Hewlett-Packard system. The data
will then reside in a file that can be accessed by a program
on the Hewlett-Packard system. The HP "READ" program will
verify that the data consists of 21 string values.

XI. USING TRANSFERRED DATA ON THE HEWLETT-PACKARD

Data that has been transported from the host computer to Hewlett-Packard mass storage can be accessed with a program written on the HP system. (The emulator program must be stopped and the new program loaded into main memory.) Keep in mind that VAX2HP creates a data file containing ASCII strings. Real values can not be transported from one machine to the other in a floating point format. Thus, data transported from the host must undergo a conversion from a string to a numeric if it is to be used in computations on the HP computer. Listing H, "ST2NUM" (STring to NUMeric), will simply create a new file containing a specified number of strings followed by any number of numeric values. Use "ST2NUM" to convert the data transferred from the VAX HPDATA.DAT file into a file consisting of one string followed by 20 numeric values. If all the steps presented in this memorandum were followed, the execution of the "READ" program on the HP will confirm that this last data file is the same as "ORIG", the first data file that was created and transferred to the VAX.

XII. SUMMARY

Procedures have been presented to transfer data files and programs from the Hewlett-Packard desktop computer to a host computer and from a VAX 11/780 to a HP 9845 desktop computer. A method for using the built in capabilities of the keyboard/special function keys was briefly addressed. This method of data transfer may sometimes be all that is needed to accomplish the task at hand. However, it may also be a too tedious procedure for the size and number of files that must be transported as it requires the user to alter the file specification, and possibly other items, for each file being transported. A second method was explained more fully. The second procedure involved the creation of routines on the host computer to perform the necessary tasks for data transfer. The routines supplied in the Appendix will suffice for program file transfers and many simple data transfers from the HP system. The routines were not designed to be general software tools to cover every need that may arise. Modifications may be necessary to adapt the routines to the specific data requirements of the user. But, regardless of the specific needs, the general procedures remain the same.

A summary of the steps required for the host program method for each direction of data transfer between the desktop computer and the host computer follows. Each step of the procedure is briefly outlined. The name of the routine

supplied in this memorandum that can accomplish the task of
the step is also listed.

To transport information from the Hewlett-Packard system
to the VAX 11/780,

1) All data must be in ASCII string format. For program
files, this means the file must be stored with the HP SAVE
command. For data, all numerics must be converted to
strings. "NUM2ST" (Listing D) can perform the conversion.
"READ" (Listing B) can confirm the contents of a file as
being all strings.

2) The Hewlett-Packard system must be equipped with a
terminal emulator and the appropriate hardware. (See the
OVERVIEW section for details.)

3) A program written for the host computer is executed
through the use of the emulator program. The program will
execute appropriate escape codes to send the data from the
Hewlett-Packard mass storage device to the VAX. HP2VAX
(Listing E) is a program that can perform such feats.

4) A transferred data file can be used on the VAX with any
program deemed appropriate. A transported program file
usually needs modifications so that it will run correctly.

To transport information from the VAX 11/780 system to
the Hewlett-Packard desktop computer,

1) The Hewlett-Packard system must be equipped with a
terminal emulator and the appropriate hardware.

2) A program written for the host computer is executed
through the use of the emulator program. The program will
execute appropriate escape codes to send the data from the
VAX to a Hewlett-Packard mass storage device. VAX2HP
(Listing G) will perform these operations.

3) A transferred data file will need to undergo a conversion
from strings to numerics if it is to be used in
computations. "ST2NUM" (Listing H) can perform these
conversions. Transferred program files usually need
modifications so that they will run properly on the
Hewlett-Packard system.

Data transfer between the Hewlett-Packard desktop
computer and a host computer can be a simple task as long as
the rules of the emulator program are abided. A careful
approach to the task is essential to ensuring that the
transfer is completed with the minimum amount of frustration.

APPENDIX

PROGRAM AND DATA LISTINGS

```
 10    ! ***********************************************************************
*************************************************************************
 20    !
 30    !     This program ("CREATE") will create a data file consisting of
 40    !         one string value followed by twenty random numbers.
 50    !
 60    ! ***********************************************************************
*************************************************************************
 70    !
 80    !     VARIABLES
 90    !
100    !     Data         Numeric value
110    !
120    !     FUNCTIONS
121    !
130    !     RND          Produces a pseudo-random number
140    !     EXP          Gives the value of e raised to the power of the argument
150    !
160    ! ***********************************************************************
*************************************************************************
170    !
180    CREATE "ORIG",2                      ! Create data file
190    ASSIGN #1 TO "ORIG"                  ! Assign unit number to data file
200    INTEGER I
210    PRINT #1;"Original Data"             ! Write title into data file
220    FOR I=-9 TO 10
230       Data=EXP(I)-RND                   ! Make a random data value
240       PRINT #1;Data                     ! Write value to data file
250    NEXT I
260    PRINT "ALL DONE"                     ! Signify end of data creation
270    END
280    !
290    ! ***********************************************************************
*************************************************************************
```

```
10    ! ********************************************************************
********************************************************************
20    !
30    !     This program ("READ") will read a data file consisting of any
40    !         number of string values followed by any number of numeric values.
50    !         The program will print out each data value and the total number
60    !         of string and numeric values.
70    !
80    ! ********************************************************************
********************************************************************
90    !
100   !     VARIABLES
110   !
120   !     Data          Numeric data value
130   !     File$         String containing the name of the data file to read
140   !     I             Index for array
150   !     String$       String data value
160   !     Strings       Number of string values
170   !
180   ! ********************************************************************
********************************************************************
190   !
200   INTEGER I
210   DIM String$[160]
220   LINPUT "Which data file would you like to read?",File$
230   ASSIGN #2 TO File$                    ! Assign unit number to data file
240   PRINTER IS 0                          ! Set hardcopy printer
250   Strings=0                             ! Initialize number of strings
260   !
270   Read_strings:                         !
280   IF (TYP(2)<>2) AND (TYP(2)<7) THEN GOTO Real_data! If value not string
290   Strings=Strings+1
300   READ #2;String$                       ! Read in string value
310   PRINT String$                         ! Print out string value
320   GOTO Read_strings                     ! Get next value
330   !
340   Real_data:                            !
350   Number=0                              ! Initialize counter
360   ON END #2 GOTO Finish                 ! When end of file found. . .
370   Read_data:                            !
380   Number=Number+1                       ! Increment counter
390   READ #2;Data                          ! Read in value
400   PRINT Number,Data                     ! Write out counter and value
410   GOTO Read_data                        ! Get next data value
420   !
430   Finish:                               !
440   PRINT LIN(2);"Total number of string values = ";Strings
450   PRINT "Total number of numeric values = ";Number-1
460   PRINTER IS 16                         ! Return default printer to screen
470   END
480   !
490   ! ********************************************************************
********************************************************************
```

Original Data
```
 1                      -.678095599541
 2                      -.381851396423
 3                      -.418236580623
 4                      -.501378285728
 5                      -.737030929855
 6                      -.491309796409
 7                      -.545204016069
 8                      -.252166729101
 9                      -.521312937984
10                       .18920912752
11                       1.8431580733
12                       6.40997803057
13                       19.6808815754
14                       54.2830027347
15                       148.374280049
16                       402.731510716
17                       1096.2183403
18                       2980.09592926
19                       8102.09335273
20                       22026.3923315
```

Total number of string values =   1
Total number of numeric values =   20

```
10    ! *************************************************************************
**************************************************************************
20    !
30    !     This program ("NUM2ST") will read a data file consisting of any
40    !         number of strings followed by any number of numeric values
50    !         (limited only by the dimension of the string and data arrays).
60    !         The numerics will be converted to ASCII string values and
70    !         written into a newly created data file along with the original
80    !         string values.
90    !
100   ! *************************************************************************
**************************************************************************
110   !
120   !     VARIABLES
130   !
140   !     Bytes                           ! Bytes needed for data storage
150   !     Data(*)                         ! Array of numeric values
160   !     File$                           ! Name of data file
170   !     I                               ! Index for arrays
180   !     Number                          ! Total number of numeric values
190   !     Number_string                   ! Total number of string values
200   !     String$(*)                      ! Array of string values
210   !
220   !
230   !     FUNCTIONS
240   !
250   !     VAL$                            ! Converts numeric to ASCII string
260   !     TYP                             ! Gives type of next value to read
270   !
280   ! *************************************************************************
**************************************************************************
290   !
300   OPTION BASE 1                         ! Start array with index 1
310   INTEGER I,Number,Bytes,Number_string
320   DIM Data(2000),String$(2000)[20]      ! Dimension arrays for 2000 values
330   !
340   LINPUT "What is the name of the original data file?",File$
350   ASSIGN #2 TO File$                    ! Assign unit to file with numerics
360   LINPUT "Enter name of the new data file to contain string values.",File$
370   Number_string=0
380   !
390 Read_strings:                           !
400   IF TYP(2)<>2 THEN GOTO Real_data      ! If next data item is not a string
410   Number_string=Number_string+1
420   READ #2;String$(Number_string)        ! Read string value from original
430   Bytes=24+Bytes                        ! Count bytes needed for strings
440   GOTO Read_strings                     ! Get next value
450   !
```

```
450    !
460 Real_data:                              !
470   Number=0                              ! Initialize counter for data array
480    !
490   ON END #2 GOTO New_file                ! When end of file found. . .
500 Read_data:                              !
510   Number=Number+1                        ! Increment index
520   READ #2;Data(Number)                   ! Read in numeric values
530   GOTO Read_data                         ! Go get next value
540    !
550 New_file:                               !
560   Number=Number-1                        ! Save total number of numeric values
570   Bytes=Number*18+Bytes                  ! Add bytes needed for numerics
580   CREATE File$,Bytes/256                 ! Create data file of appropriate size
590   ASSIGN #3 TO File$                     ! Assign unit to file for string value
600    !
610  -FOR I=1 TO Number_string
620     PRINT #3;String$(I)                  ! Write string to data file
630   NEXT I
640   FOR I=1 TO Number
650     PRINT #3;VAL$(Data(I))               ! Change numeric to string and print
660   NEXT I
670    !
680   PRINTER IS 0
690   PRINT LIN(1);Number_string;" strings written to ";TRIM$(File$);"."
700   PRINT Number;" numerics converted to strings and written to same file."
710   PRINTER IS 16
720   END
730    !
740    ! *********************************************************************
*********************************************************************
```

```
10    !********************************************************************
20    !
30    !    This program (HP2VAX), when executed on a VAX via the Hewlett
40    !        Packard Terminal Emulator Program, will read data from the
50    !        9845 System and write it to a data file on the VAX computer.
60    !        The program will transfer the specified number of string
70    !        values to the VAX mass storage file.
80    !
90    !********************************************************************
100   !
110   !****** Solicit file names and attributes *************************
120   !
130       LINPUT " WHAT IS THE NAME OF THE SOURCE FILE ON THE HP";IN_FILE$
140       LINPUT " WHAT HP DEVICE IS THE FILE ON (E.G. :T15, :H8,0,0)";DEV$
150       INPUT " HOW MANY VALUES (LINES) ARE IN THE FILE"; NUMBER
160       LINPUT " WHAT IS THE NAME OF DESTINATION FILE ON THE VAX";     &
              OUT_FILE$
170   !
180   !****** Ensure colon in the device name ***************************
190   !
200       IF POS(DEV$,":",1) = 0 THEN DEV$ = ":" + DEV$
210   !
220   !****** Ready unit three as the output file **********************
230   !
240       OPEN OUT_FILE$ FOR OUTPUT AS FILE #3
250   !
260   !****** Execute escape codes to identify and send the HP data ******
270   !
280       PRINT CHR$(27)+",c ASSIGN SOURCE FILE "+IN_FILE$+" DEVICE "+&
DEV$+CHR$(13)
290       PRINT CHR$(27)+",c ENABLE SEND"+CHR$(13)
310   !
320   !****** Transfer strings from HP to VAX **************************
330   !
340       ON ERROR GO TO 400
350       FOR I=1 TO NUMBER
360           INPUT A$
370           PRINT #3;A$
380       NEXT I
390   !
400       IF I<NUMBER THEN NUMBER=I-1
410       RESUME 430
420   !
430       PRINT " TOTAL NUMBER OF VALUES TRANSFERRED TO VAX =";NUMBER
440       END
```

```
C*********************************************************************
C
C        This program (READ) is set up to read an ASCII file that contains
C           one string value and any quantity of real values
C
C*********************************************************************
C
         IMPLICIT DOUBLE PRECISION (A-H,O-Z)
         CHARACTER*20 TITLE, DATFILE
C
C    Solicit name of data file and make proper file assignment
C
         PRINT *, 'WHAT IS THE NAME OF THE DATA FILE?'
         READ 5, DATFILE
         OPEN (UNIT=3, FILE=DATFILE, TYPE='OLD')
C
C    Read in and print character string
C
         READ (3,5), TITLE
         PRINT 5, ' ', TITLE
   5     FORMAT (A,A)
C
C    Read and print real values
C
         DO 10 I = 1, 20
             READ (3,15) DATA
             PRINT 20, I, DATA
  10     CONTINUE
C
  15     FORMAT(BZF30.12)
  20     FORMAT (1X,I3,F30.12)
         END
```

```
10   !*******************************************************************
20   !
30   !    This program (VAX2HP), when executed on a VAX computer via
40   !         the Hewlett-Packard Terminal Emulator Program, will read
50   !         data from a VAX computer and write the data on a
60   !         Hewlett-Packard 9845 mass storage device.
80   !
90   !*******************************************************************
100  !
110  !*********** Solicit file names and attributes *********************
120  !
130     INPUT " WHAT IS THE NAME OF THE SOURCE FILE ON THE VAX";IN_FILE$
140     LINPUT " WHAT IS THE NAME OF THE DESTINATION FILE ON THE HP";    &
            OUT_FILE$
150     LINPUT " WHAT HP DEVICE IS THE FILE ON (E.G. :T15)";DEV$
160     INPUT " HOW MANY LINES OF DATA WILL YOU TRANSFER";NUMBER
170     INPUT " HOW MANY CHARACTERS PER LINE";CHARACTERS
180     INPUT " DO YOU WANT LINE NUMBERS ADDED (Y OR N)";ANSWER$
190  !
200  !*********** Ensure colon in the HP device name *******************
210  !
220     IF POS(DEV$,":",1) = 0 THEN DEV$ = ":" + DEV$
230  !
240  !*********** Ready unit two as the input file *********************
250  !
260     OPEN IN_FILE$ FOR INPUT AS FILE #2
270  !
280  !*********** Determine Hewlett-Packard file size *****************
290  !
300    IF ANSWER$="Y" THEN FILE_SIZE% = NUMBER * 5
310    FILE_SIZE% = INT((((NUMBER * (CHARACTERS + 4) + FILE_SIZE)/256) + 1)&
            / 2) * 2
330  !
340  !*********** Execute escape codes to send VAX data ***************
360  !
370     PRINT CHR$(27)+",c CREATE FILE "+OUT_FILE$+" DEVICE "+DEV$       &
            +" LENGTH "+NUM1$(FILE_SIZE%)+CHR$(13)
380     PRINT CHR$(27)+",c ASSIGN DESTINATION FILE "+OUT_FILE$+" DEVICE " &
            +DEV$+CHR$(13)
390     PRINT CHR$(27)+",c ENABLE RECORD"+CHR$(13)
400  !
```

```
410   !*********** Transfer ASCII chracters from VAX to HP ****************
420   !
430      ON ERROR GO TO 570
440      IF ANSWER$="Y" THEN GO TO 510
450      FOR I%=1 TO NUMBER
460           LINPUT #2;A$
470           PRINT A$
480      NEXT I%
490      GO TO 570
500   !
510      FOR I%=1 TO NUMBER
520           LINPUT #2;A$
530           A$=NUM1$(I%)+" "+A$
540           PRINT A$
550      NEXT I%
560   !
570      IF I%<NUMBER THEN NUMBER=I%-1
580      RESUME 600
590   !
600      PRINT CHR$(27)+",c DISABLE RECORD"+CHR$(13)
610      PRINT " TOTAL NUMBER OF VALUES TRANSFERRED TO HP =";NUMBER
620      END
```

```
10    ! ************************************************************************
**********************************************************************************
20    !
30    !     This program ("ST2NUM") will read a data file consisting of
40    !         of strings followed by any number of numeric values
50    !         (limited by the dimension of the string and data arrays).
60    !         The specified ASCII strings will be converted to numeric
70    !         values and all values will be written to a new data file.
80    !
90    ! ************************************************************************
**********************************************************************************
100   !
110   !     VARIABLES
120   !
121   !     Bytes                           ! Bytes needed for data storage
122   !     Data$(*)                        ! Array for numeric values
123   !     File$                           ! Name of data file
124   !     I                               ! Index for arrays
125   !     Number                          ! Total number of numeric values
126   !     Number_string                   ! Total number of string values
127   !     String$(*)                      ! Array of string values
128   !
129   !     FUNCTIONS
130   !
131   !     VAL                             ! Converts string to number
180   !
190   ! ************************************************************************
**********************************************************************************
200   !
210   OPTION BASE 1
220   INTEGER I,Number,Bytes,Number_string
230   DIM Data$(2000),String$(2000)[20]
240   !
250   LINPUT "What is the name of the original data file (all strings)?",File$
260   ASSIGN #2 TO File$
270   LINPUT "Enter the name of the new data file (converted strings).",File$
280   INPUT "Enter number of strings to remain strings.",Number_string
290   Bytes=0
300   !
310 Read_strings:                          !
320   FOR I=1 TO Number_string
330       READ #2;String$(I)              ! Read in string values
340       Bytes=24+Bytes                  ! Allocate bytes for each string
350   NEXT I
360   !
370 Real_data:                            !
380   Number=0
390   !
400   ON END #2 GOTO New_file              ! Read data until EOF
410 Read_data:                            !
420   Number=Number+1                     ! Count numerics
430   READ #2;Data$(Number)               ! Read in numbers as strings
440   GOTO Read_data
450   !
```

A-10

```
450    !
460 New_file:                              !
470    Number=Number-1
480    Bytes=Number*8+Bytes               ! Allocate bytes for numerics
490    CREATE File$,Bytes/256             ! Create new data file
500    ASSIGN #3 TO File$
510    !
520    FOR I=1 TO Number_string
530       PRINT #3;String$(I)             ! Write strings to file
540    NEXT I
550    FOR I=1 TO Number
560       PRINT #3;VAL(Data$(I))          ! Convert string to number
570    NEXT I                             !   and write to file
580    !
590    PRINTER IS 0                       ! Set thermal printer
600    PRINT LIN(1);Number_string;" strings written to ";TRIM$(File$);"."
610    PRINT Number;" strings converted to numerics and written to same file."
620    PRINTER IS 16                       ! Set printer to CRT screen
630    END
640    !
650    ! **********************************************************************
       ***********************************************************************
```

DISTRIBUTION

Internal:

3202	G. Leibiger		341	D. Cheng
3202	E. Siborg		3411	G. Majewski
3202	E. Smith		3411	E. Wolkoff
3202	N. Stahmer		3421	A. Bruno
3211	J. Fay		3421	D. Portofee
3212	P. Conlon		3421	G. Majewski
3212	R. Cox		3422	J. Ball
322	J. Geary		3431	D. Dixon
3221	J. Conrad		3431	R. Hall
3221	F. Camlet		3431	S. Sherman
3222	M. Gill		3432	M. Spaziante
3222	R. Matis		3492	C. Floyd
3222	M. Sansevero		413	D. Leary
323	W. Strawderman		61	B. Almeida
3232	D. Abraham		70	C. Devoe
3232	LTJG Wilson		71	G. Elias
3234	F. Alatalo		711	M. Goldstein
3234	P. Kindl		711	D. Aker
3234	J. Lindberg		711	J. Sikorski
3234	T. Mapes		711	N. Sulinski (15)
3253	D. Nudelman		0211	
3261	N. Dull		021311	New London
3302	A. Nuttal		021312	Newport
3302	P. Cable			
3303	J. Bairstow		81 internal copies	
3312	J. Cooke			
3312	R. Hayford			
3313	C. Fenick			
3313	W. Haigh			
33201	S. Rupinski			
3321	D. Armstrong			
3321	W. Goldman			
3321	P. Seaman			
3322	R. Dionne			
3324	N. Toplosky			
3331	P. Stahl			
3331	R. Tremblay			
3332	E. Harris			
3333	R. MacDonald			
3342	G. Brown			
3343	R. OConner			
3351	R. Wilkinson			
3352	K. Hansen			
3353	H. Szankovics			