Version 1.3

USER'S GUIDE

1500 DATASHARE
DS1500

Document No. 50482

March, 1983

1500 DATASHARE
DS1500

User's Guide

Version 1.3

March 1983

Document No. 50482

PREFACE

This User's Guide describes the run-time characteristics of the Datapoint 1500 series processor's DATASHARE Interpreter, DS1500. The material is presented as a reference for operating a DATASHARE system. 1500 DATASHARE executes on any 1500/1550/1560 processor with at least 64K bytes of memory. The program permits simultaneous execution of up to four DATABUS programs, each interfacing with its own terminal. DATABUS programming for batch and interactive data processing is described in the DBCMP15 User's Guide.

All references herein to the 1500 processor are valid for the 1550/1560/2150 processor as well.

TABLE OF CONTENTS

CHAPTER 1. DIFFERENCES BETWEEN DIFFERENT VERSIONS OF DS1500

1.1 Differences between DS1500 1.2 and DS1500 1.1

1.1.1 Interpreter Replacement

The DS1500 version 1.2 Interpreter replaces the DS1500 1.1 Interpreter in every aspect of 1500 DATASHARE operation.

1.1.2 Supported Features

DS1500 version 1.2 fully supports all features and configurations of DS1500 1.1, with the exception of the changes to the COMWAIT verb.

1.1.3 Changes to Existing Features

The following features of DATASHARE have been enhanced in this version:

Concurrency Support

> Concurrency is now supported on a 64K 1500. This configuration is valid only when there is a maximum of 1 port configured into the DATASHARE system.

Port One on the Console

> Port one on the console now supports the repeat key feature. This feature performs the same as on all 8200's.

PREPAREing ISAM FILES

> DS1500 PREPARE verb now allows the creation of a null /ISI file and an associated null /TXT file.

1.1.4 New Features

The following features have been added to DS1500 version 1.2:

Ports on the MFC Communications Channel

The 1560 processor with ROM version 31 or greater has an optional second internal communications channel. This allows a user to configure four DATASHARE ports without configuring the FPCA (Four Port Communications Adapter). The MFC (Multifunction Communications) channel can run two DATASHARE ports, the internal communications channel (ICA) can run one port and port 1 can be configured to be on the console. The maximum number of ports on a system with an FPCA remains at four.

Internal Communications

DS1500 supports the full range of internal communications. The RECEIVE verb was added to complete the necessary link to have useful internal communications. DS1500 internal communications now functions as documented in the DBCMP15 User's Guide (Model code 50302). The internal communications no longer function as formerly documented in the DS1500 version 1.1 User's Guide. The COMWAIT verb and the addition of the RECEIVE verb are the most significant changes.

Multilink Support for External Communications

DS1500 supports the Multilink facility to enable a DATABUS program to interface with an external communications device. Multilink is run as an independent task with the ability to start multiple tasks of its own. Multilink requires a 1500 with at least 64K of memory. In this 64K configuration Multilink has the same restriction as a concurrent job, which is a maximum of 1 port configured. In a 1550/1560 with greater than 64K, the Multilink will execute with a maximum of 4 DATASHARE ports. Multilink and Concurreny will NOT execute together because they both use the same region from 0150000-0167777 (8K of memory).

ACALL

ACALL (assembler language "CALL") is supported in a 1550/1560 with 96K or more of memory. The ACALL overlay uses an extended memory sector for its execution. The ACALL overlay is opened and checked for validity. The extended memory is mapped in and the ACALL is loaded. DS1500 will only support the dynamic ACALL feature. That is, only one ACALL overlay is used, then it is only loaded once and is simply "mapped" into logical space as needed. The ACALL is restricted to fit into 4K of RAM or the ACALL overlay will not be loaded.

DATABUS Library Support

In DS1500 the user may now keep DATABUS programs in a library. Using the Datapoint library utility program LIBSYS15, one may create library files of DATABUS object code (/DBC programs). Such libraries are accessed and utilized by DATASHARE in much the same manner as DOS uses the UTILITY/SYS file.

WRITE (Physical and Logical) List Controls

a) *LL - Write the logical length of the variable
b) *PL - Write the physical length of the variable

KEYIN List Controls

a) *T<n> - Keyin timeout of <n> seconds.

TRAP

DS1500 will now support the trap on character function. This function is handled in exactly the same manner as a trap on a function key(s) and the interrupt sequence.

New Verbs

RECEIVE -- The RECV instruction is used to specify a list of variables which serve as a destination for data from a source.

1.2 Differences between DS1500 1.3 and DS1500 1.2

1.2.1 Interpreter Replacement

The DS1500 version 1.3 Interpreter replaces the DS1500 1.2 Interpreter in every aspect of 1500 DATASHARE operation. DS1500 1.3 will not support any configuration file from any previous Interpreter.

1.2.2 Supported Features

The DS1500 version 1.3 fully supports all features and configurations of DS1500 1.2.

1.2.3 Changes to existing features

The following features of DS1500 have been enhanced:

Purge /ISI Sectors When a CLOSE is Executed on an ISAM File

DS1500 will write all /ISI sectors pertaining to that file when a CLOSE is invoked only if the option is configured at configuration time. The /ISI sectors are not normally written until a ROLLOUT/SHUTDOWN is invoked or when the /ISI buffer is the "least recently used buffer" and must be flushed for other use. This is a configurable option and must be selected at configuration time.

Enqueue/Dequeue /ISI Sectors during ISAM Operation

DS1500 will enqueue the /ISI sectors upon an update to an ISAM file in an ARC environment. This is a configurable option and must be selected at configuration time.

64K DATABUS Program Support

DS1500 will now allow a DATABUS program to be 64K bytes or less. The previous restriction for DATASHARE was 32K.

12K UDA (User Data Area) Support

DS1500 will now allow a DATABUS program to use up to 12K UDA. When SPOOLING is configured this limit is 11.75K because SPOOL uses the first page of UDA.

FILEPI and PI Enhancements

When ARC is configured the FILEPI and PI verbs will perform an enqueue to support shared files in a network environment. (See section 8.8 for more information.)

Keyin List Controls

The list controls *HON and *HOFF are now supported on all terminals that support highlighting.

Display List Controls

The list controls *HON and *HOFF are now supported on all terminals that support highlighting.

New TRAP Options

a) TRAP on SPOOL -- This event occurs when an error occurs while printer output is being SPOOLed to a disk file.

b) TRAP GIVING -- If a trap occurs with this option the interpreter places the error message in the character string specified after the GIVING clause.

c) TRAP NORESET -- If a trap occurs with this option the trap is taken but the trap address is not cleared so the trap does not have to be set again.

(See DBCMP15 3.3 User's Guide, Model Code # 50302.)

Shift Inversion Limits

The shift inversion limits, normally from 0101 ("A") to 0132 ("Z"), can now be set during configuration mode. This allows international character sets to be used with DS1500.

1.2.4 New features

The following features have been added to DS1500 version 1.3:

SCAN Verb

The SCAN instruction allows pattern searches within a string variable. (See DBCMP15 3.3 User's Guide, Model Code # 50302.)

PACK Verb

The PACK instruction allows the contents of several character and numeric string variables to be concatenated together into one character string variable. (See DBCMP15 3.3 User's Guide, Model Code # 50302.)

UNPACK Verb

The UNPACK instruction allows the contents of a character or numeric string variable to split apart into several character string variables. (See DBCMP15 3.3 User's Guide, Model Code # 50302.)

NFORMAT Verb

The NFORMAT instruction allows dynamic creation of a DATABUS numeric string variable. (See DBCMP15 3.3 User's Guide, Model Code # 50302.)

SFORMAT Verb

The SFORMAT instruction allows dynamic creation of a DATABUS character string variable. (See DBCMP15 3.3 User's Guide, Model Code # 50302.)

SPLOPEN Verb

The SPLOPEN instruction allows the DATABUS program to direct printer output to a disk file instead of directly to the printer. (See DBCMP15 3.3 User's Guide, Model Code # 50302.)

SPLCLOSE Verb

The SPLCLOSE instruction is used to turn off print spooling. (See DBCMP15 3.3 User's Guide, Model Code # 50302.)

PHANTOM Port Support

DATASHARE will now allow a port to run without a terminal being attached. If a DISPLAY or KEYIN verb is invoked, the port is taken off line. (See section 8.15 for more information.)

Keyin List Controls

a) *RD - Roll down the screen -- not supported on the console.
b) *DION - Double intensity on -- not supported on the console.

Display List Controls

a) *RD - Roll down the screen -- not supported on the console.
b) *DION - Double intensity on -- not supported on the console.

DS1500 Configurator

DS1500 1.3 now has a new configurator to make configuring the DATASHARE system very easy. The user may change one question without having to run through the entire configuration. For more information see chapter 3.

CHAPTER 2. INTRODUCTION

DS1500, the multi-terminal business processing system for the Datapoint 1500 series processors, allows the simultaneous execution of up to four DATABUS programs each dealing with its own terminal. The Interpreter is configurable for the host 1500/1550/1560's physical characteristics and the processing tasks to be performed.

The DS1500 Interpreter executes under the 1500 Disk Operating System, DOS.H, Version 2.6 or later, and takes advantage of all of the DOS's file handling routines. The Interpreter services a high-speed line printer and provides indexed-sequential as well as random and sequential file accessing methods, providing a powerful data entry and processing facility. The several possible configurations allow a flexible mix of remote-batch and interactive processing all under the control of one high-level language program. This enables the user to configure the system to suit particular data processing needs. In addition, the DOS, with its variety of utility and higher-level language systems may be used alternately with DS1500, enabling processing of tasks not appropriate to the multiple-terminal environment.

DS1500 requires a minimum of 64K bytes of memory in the host processor, and is configurable for up to 128K of memory above this minimum. Any extended memory (greater than 64K bytes) available is dynamically used by DS1500's memory manager. DS1500 is Concurrent Job compatible with certain restrictions. A 1500 series processor with 64K bytes of memory can support concurrency only if one port is configured into the DATASHARE system. A 1500 series processors with more than 64K bytes of memory can support concurrency with the maximum port configuration (4 ports) in the DATASHARE system. If the user has a 32K machine in which Concurrency is active, and a single DATABUS program is to be executed on only one port, then the 1500 DATABUS Interpreter DB15 should be used, as it may execute in this configuration.

If the system hardware configuration includes a 9320 cartridge disk system, which is a 9310 cartridge disk with an integral Four Port Communications Adapter (FPCA), up to four terminals may be connected for use. If the system does not include an FPCA, one terminal can be supported on the Internal Communications Adapter (ICA). Two ports can be supported on the optional MFC (Multifunction Communications) channel (ROM version 31 or greater). One port may also be configured on the processor

console, allowing four ports on a system without an FPCA. In a configuration with an FPCA, the total number of configured ports may not exceed four. The terminals attached to the system may be a mixture of both 3600s and 8200s, for either remote or local operation.

Using virtual memory techniques, DS1500 provides each program with a 64K byte area for executable statements. This enables the user to create and use very large DATABUS programs. To provide rapid program execution, the data area for each user program (UDA) is maintained in main memory and is not swapped from disk. Up to 49,152 bytes of main memory may be allocated for the combined data area of all ports depending on system configuration. When the system is configured to run with four ports, the data area may be variably partitioned among them. The data area of any one port can be configured to be from 256 bytes to 12288 bytes. Any memory not used for UDA will be reclaimed by the memory manager.

Any of the Datapoint serial printers may be connected to the DS1500 system, and printing may be controlled from any of the configured ports. If the system printer is busy, a port trying to execute a PRINT statement will wait until the printer is available for use. The RELEASE verb may be used to check for printer availability.

All program execution in DS1500 occurs in the DATABUS language. Terminal command interpretation is handled in special ANSWER and MASTER programs (typically unique for each port); these programs may also handle system security. The MASTER and ANSWER programs are not provided with the system; they are to be created and compiled like any other DATABUS program, enabling the user to define his own terminal command and security system.

Program generation is performed under DOS.H using the general-purpose DOS EDITor and the DOS DATABUS Compiler. (DBCMP15 3.3 User's Guide, Model Code # 50302, DOS.H 2.8 User's Guide, Model Code # 50308).

CHAPTER 3. SYSTEM GENERATION

3.1 Preparing the DS1500 Environment

The DS1500 absolute and relocatable Interpreter files are contained on a disk which should be duplicated as soon after its delivery as is possible to avoid the loss of the Interpreter files in case of a disk error.

The DS1500 Interpreter files are DS1500/CMD (the absolute library containing the System Loader, Interpreter, and Configurator modules), DS1500/CHN (the Configurator CHAIN file), and DS1500/REL (the relocatable library which contains the modules necessary to create any allowable configuration of the Interpreter). The command file (DS1500/CMD) may be renamed to any legal DOS file name, but the Interpreter relocatable library and the Configurator CHAIN file may not be renamed.

3.2 Introduction to the 1500 DATASHARE Configurator

The DS1500 configurator is a user friendly program for custom configuring a DATASHARE Interpreter. The Configurator allows the user to select which options of DS1500 are to be used in any particular Interpreter, and to select the port type and the amount of UDA (User Data Area) available to each of the ports.

Earlier versions of DS1500 forced the user to re-configure the entire system even if only one question needed to be changed. The DS1500 1.3 configurator will allow the user to easily change ANY part of the current configuration without having to answer all of the configuration questions.

Associated with the DS1500 command file is the Configuration file, <Interpreter-name>/CFG, which must exist on the same drive as the command file, and must have the same name as the command file. This file contains the configuration information used by the Interpreter during system initialization.

System configuration is accomplished by specifying the desired answers to the questions that are displayed on the screen. The operator keys in:

DS1500;C

The program displays the first configuration screen and if there is a current configuration the current answers will be displayed as well. If there is a current configuration, there will be a highlighted message on the bottom two lines of the screen. The message:

DSP = RELINK CURRENT CONFIGURATION
F1 = CHANGE CURRENT CONFIGURATION

will be displayed. This means if the user wishes to just relink the current configuration, then the DSP key should be depressed. If the user wishes to change the current configuration, then the F1 key should be depressed. If there is no current configuration the previous message will not be displayed and the first configuration screen will look like this:

```
    DS1500  1.3     - ddmmmyy - SCREEN 1

MACHINE SIZE                    64K     96K     128K
PORT ONE ON THE CONSOLE         YES     NO
PORT COMMUNICATIONS
   FPCA                         YES     NO
   ICA                          YES     NO
   MFC CHANNEL A                YES     NO
   MFC CHANNEL B                YES     NO
NUMBER OF PORTS                 1               2               3               4
PHANTOM PORTS           YES     NO  | YES     NO  | YES     NO  | YES     NO
BAUD RATE: 50 - 9600
   PORT 1                       Rx =            Tx =
   PORT 2                       Rx =            Tx =
   PORT 3                       Rx =            Tx =
   PORT 4                       Rx =            Tx =
EXTERNAL COMMUNICATIONS         YES     NO  NUMBER OF EXTENDED SECTORS =
INTERNAL COMMUNICATIONS         YES     NO
CONCURRENT JOB SUPPORT          YES     NO
ARC SUPPORT                     YES     NO
SHARED FILE SUPPORT             YES     NO
ACALL                           YES     NO
SHIFT INVERSION LIMITS          YES     NO  UPPER =             LOWER =
```

If DS1500 has a current configuration the current configuration answers will contain a bracket around the configured answer. For example, if the current configuration memory size is 96K, the machine size question will appear as follows:

```
MACHINE SIZE                    64K    [96K]    128K
```

After the last question on the first screen has been answered the configurator will automatically display screen two with its current configuration. The second (default) configuration screen will look like this:

```
    DS1500   1.3   - ddmmmyy - SCREEN 2

ROLLOUT                         YES      NO
EDIT                            YES      NO
SCAN                            YES      NO
PACK/UNPACK                     YES      NO
NFORMAT/SFORMAT                 YES      NO
SPOOL OPEN/CLOSE                YES      NO
CHECK10/11                      YES      NO
SEARCH                          YES      NO
ISAM                            YES      NO
ISAM RECORD COMPRESSION         YES      NO
PURGE /ISI SECTORS              YES      NO
FORCE WRITE /ISI SECTORS        YES      NO
ENQUEUE /ISI SECTORS            YES      NO
FORMFEED AFTER RELEASE          YES      NO
REAR TRACTOR SUPPORT            YES      NO
UDA SIZE 256 - 12288
   PORT 1                       UDA =
   PORT 2                       UDA =
   PORT 3                       UDA =
   PORT 4                       UDA =
```

3.2.1 Configurator Operating Keys

The DS1500 configurator has a few special keys to allow the user to operate the configurator. The special keys are listed below as well as a description of what the key will do for the user.

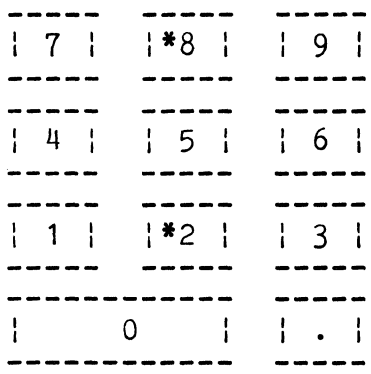
1) Space bar -- This key is used to move through any configuration question that does not require user keyin.

2) Enter key -- This key is used to complete the answer to the current question. The answer to the question will be the one that has the solid cursor at the time the Enter key was depressed. The answer will then acquire brackets as in the MACHINE SIZE example.

3) 8 key on the numeric pad -- This key is used to return the solid cursor back to the previous question. If the solid cursor is on the first question no action will be taken. When referring to this key this document will call it the previous question cursor key.

4) 2 key on the numeric pad -- This key is used to advance the solid cursor to the next question. The only time this key will work is if the current question has an answer. If the current question does not have an answer no action will be taken. When referring to this key this document will call it the next question cursor key.

Special Note: All the keys are repeatable in the DS1500 configurator. To move thecursor through the configuration questions to get to the desired question, just hold down the next question cursor key until the desired question is reached. Here is a diagram of the standard numeric pad so the exact position of the next question cursor key and the previous question cursor key is known. These keys are marked in the diagram with an "*".



3.3 Configuring a DS1500 Interpreter

The user should now be ready to configure a DS1500 Interpreter. The next sub-sections will describe the questions in the order as they appear on the screens. An invalid configuration cannot be configured because the configurator will only allow legal entries. As the user goes through the configuration some questions will be answered by the configurator. This is because the configurator looks at the previous answers and realizes that for some questions there is only one correct answer. This prevents the user from answering a question incorrectly and causing an invalid configuration.

3.3.1 Machine Size

This question tells the configurator what the target machine of the Interpreter will be. A 64K configuration will run on any 1500 series processor with 64K or more of memory. A 96K configuration will run on any 1500 series processor with 96K or more of memory. A 128K configuration will only run on a 1500 series processor with 128K or more of memory. The machine size affects the amount of UDA the user can configure and also affects the number of Virtual Storage buffers for Interpreter use.

3.3.2 Port One on the Console

This question allows the user to tell the configurator whether the console will be used as a port.

3.3.3 Port Communications

This question allows the user to tell the configurator what type of port I/O will be used on the target machine. If the FPCA (Four Port Communications Adapter) is configured the next three responses will be bypassed. DS1500 will only support a maximum of four ports. Since the FPCA can handle up to four ports, the configurator cannot allow a second type of port I/O. If the FPCA is not configured, the user may configure any other type of port communications.

There are three I/O ports on the 1560 series processor that can be used for terminal hook-up, the ICA, MFC (Multifunction Communications) Channel A and MFC Channel B (the MFC is optional). The 1500/1550 series processors have only one I/O channel capable of driving a port, the ICA (Internal Communications Adapter).

3.3.4 Number of Ports

This question will be answered for the user if the FPCA is NOT configured. The only time the user has control over the number of ports configured is when the FPCA is configured. If any other type of I/O is specified the configurator knows exactly how many ports are being configured. Each type of configurable I/O port other than the FPCA can only handle one port, therefore each I/O port configured adds to the total number of ports by one.

3.3.5 PHANTOM Ports

This question allows the user to tell the configurator which of the ports configured are to be PHANTOM ports. A PHANTOM port is not connected to any terminal, therefore NO port I/O can be done to a PHANTOM port. Port I/O is defined to be any KEYIN, DISPLAY, or BEEP instruction. Execution of a KEYIN, DISPLAY, or BEEP instruction from a PHANTOM port, will result in an untrapable error. If the port was in ANSWER or MASTER, the port is permanently disabled and, if the system console is configured (port 1 is NOT on the console), a message is displayed on the console screen indicating that the port has been disabled. If the port was not in ANSWER or MASTER, the port will abort and chain

back to the MASTER program.

3.3.6 Baud Rates

The user will enter the baud rates for the current port configuration. If there is only one port configured and that port is on the console, then the baud rate questions will be bypassed. If port one is on the console and there is more than one port configured, then only the first baud rate question will be bypassed. If any invalid baud rate is entered the configurator will beep and wait for another entry. If the same baud rate is desired for all ports then depress the "^" after the first baud rate has been entered. For a list of the valid baud rates, see chapter 7.

3.3.7 External Communications

This question will allow the user to tell the Interpreter to load and execute a Multilink driver in the concurrent job region of memory (150000-167777). If a 64K machine is configured with more than one port, the configurator will automatically install a "NO" answer and skip the number of extended memory sectors question. If a 64K machine is configured with only one port, the configurator will allow the user to configure external communications, but will skip the extended memory sectors question.

If external communications is configured on a 64K machine 10K is taken away from the total UDA (user data area) leaving only 6K for UDA.

On an extended memory machine (96K or 128K), the user may configure extended memory sectors for the Multilink driver. Whenever extended memory sectors are used by external communications the total amount of UDA is decreased. The maximum number of extended memory sectors is 6 for any Multilink driver.

3.3.8 Internal Communications

This question will allow the user to tell the Interpreter to allow internal communications between ports using the SEND and RECEIVE verbs. If external communications is configured, then the configurator will automatically give this question a YES response and go to the next question.

3.3.9 Concurrent Job Support

This question allows the user to tell the configurator whether or not the Interpreter will be loaded with a concurrent job running. If a concurrent job is running and the user has not configured it, the Interpreter will abort at initialization time. If the user configures the memory size to be 64K with more than one port, then the configurator will automatically give the question a NO response. If external communications is configured the configurator will automatically give the question a NO response, because external communications and the concurrent job use the same logical space in memory (150000-167777).

If concurrent job support is configured on a 64K machine, 8K is taken away from the UDA leaving only 8K for UDA.

3.3.10 ARC Support

This question allows the user to tell the configurator whether the Interpreter will be run in an ARC environment or a stand-alone environment. In an ARC environment FILEPI and PI will enqueue the file or entire remote volume(s) for the number instructions desired. (See Chapter 10 and Section 8.8 for more details on the ARC environment) In a stand-alone environment FILEPI and PI act exactly the same and just suspend other ports for the number of instructions desired.

3.3.11 Shared File Support

This question will allow the user to tell the configurator whether the user wants shared files in this ARC environment. At execution time this will force all accesses through random and ISAM to access the disk rather than a page from virtual storage manager. This question should be answered with a "NO" only if no file sharing is to occur on the system. This question will automatically be answered "NO" if ARC support is not desired.

3.3.12 ACALL Support

This question allows the user to tell the configurator to reserve one extended memory sector (4K of memory) for ACALL usage. This question will automatically be answered "NO" if a 64K environment is configured.

3.3.13 Shift Inversion Limits

This question allows the user to tell the configurator to ask for shift inversion limits that differ from the standard. The standard shift inversion limits are 0101 ("A") to 0132 ("Z"). The new limits must be entered in octal. The range of the shift inversion limits is 0100 to 0136, and the lower limit must be less than or equal to the upper limit. This question should be answered with a "NO" if it is desired to use the standard shift inversion limits.

3.3.14 ROLLOUT

This question allows the user to tell the configurator to create the ROLLOUT files and set the ROLLOUT flag so the Interpreter allows a ROLLOUT. The user saves the state of the machine for DATASHARE in order to perform a DOS function. (See chapter 8 for more on ROLLOUT).

3.3.15 EDIT

This question allows the user to tell the configurator to configure the EDIT verb. The EDIT instruction provides a powerful tool for formatting of variables. (See DBCMP15 3.3 User's Guide, Model Code # 50302.)

3.3.16 SCAN

This question allows the user to tell the configurator to configure the SCAN verb. The SCAN instruction allows pattern searches within string variable. (See DBCMP15 3.3 User's Guide, Model Code # 50302.)

3.3.17 PACK/UNPACK

This question allows the user to tell the configurator to configure the PACK/UNPACK verbs. The PACK instruction allows the contents of several character and numeric string variables to be concatenated together into one character string variable. The UNPACK instruction allows the contents of a character or numeric string variable to split apart into several character string variables. (See DBCMP15 3.3 User's Guide, Model Code # 50302.)

3.3.18 NFORMAT/SFORMAT

This question allows the user to tell the configurator to configure the NFORMAT/SFORMAT verbs. The NFORMAT instruction allows dynamic creation of DATABUS numeric string variable. The SFORMAT instruction allows dynamic creation of DATABUS character string variable. (See DBCMP15 3.3 User's Guide, Model Code # 50302.)

3.3.19 SPOOL OPEN/CLOSE

This question allows the user to tell the configurator to configure the SPLOPEN/SPLCLOSE verbs. The SPLOPEN instruction allows the DATABUS program to direct printer output to a disk file instead of directly to the printer. The SPLCLOSE instruction is used to turn off print spooling. The configuration of SPOOLING requires an extra page (256 bytes) of UDA be reserved for each port. This 256 bytes is used SPOOLing verbs and is not reserved for each port. This page is used for SPOOL data during the SPOOLing operation. The maximum UDA allowed for any one DATABUS program will be 11.75K when SPOOLing is configured. In a 64K environment SPOOLing will take away 1K of UDA along with the 1 page of UDA for each configured port. (See DBCMP15 3.3 User's Guide, Model Code # 50302.)

3.3.20 CHECK10/11

This question allows the user to tell the configurator to configure the CHECK10/CHECK11 verbs. The CHECK10 instruction performs a modulo 10 check digit calculation on two string variables. The CHECK11 instruction performs a modulo 11 check digit calculation on two string variables. (See DBCMP15 3.3 User's Guide, Model Code # 50302.)

3.3.21 SEARCH

This question allows the user to tell the configurator to configure the SEARCH verb. The SEARCH instruction compares a variable (key) to a list of variables and yields an index which indicates which variable in the list matched. (See DBCMP15 3.3 User's Guide, Model Code # 50302.)

3.3.22 ISAM

This question allows the user to tell the configurator to configure all Disk I/O instructions that apply to accessing indexed sequential records. (See DBCMP15 3.3 User's Guide, Model Code # 50302.)

3.3.23 ISAM Record Compression

This question will allow the user to tell the Interpreter to compress multiple ISAM records into one 256 byte page. If the option is not set then each ISAM record will use up to one 256 byte page or more depending on the length of the record. This question will automatically be answered "NO" by the configurator if ISAM is not configured.

3.3.24 Purge /ISI Sectors

This question will allow the user to tell the Interpreter to purge (write back to disk) all /ISI sectors pertaining to the drive the /ISI file is on, when a CLOSE is done to the /ISI file. The /ISI sectors are not normally written until a ROLLOUT/SHUTDOWN is invoked or the /ISI buffer is the "least recently used buffer" and therefore must be flushed for other use. This question will automatically be answered "NO" by the configurator if ISAM is not configured.

3.3.25 Force Write /ISI Sectors

This question will allow the user to tell the Interpreter to write to disk all /ISI sectors whenever the /ISI sectors are updated. This question will automatically be answered "NO" by the configurator if ISAM is not configured.

3.3.26 Enqueue /ISI Sectors

This question will allow the user to tell the Interpreter to automatically set level 2 enqueues of the /ISI sectors. A level 2 enqueue will lock out the currently accessed /ISI sector from any AP wanting to use that sector. This enqueue will expire at the completetion of the ISAM instruction. This is to prevent destruction of the index due to simultaneous modification by two application processors and to prevent invalid results if one application processor reads the index while it is being modified.

In a shared file environment this question should be answered with "NO" only if the user intends to implement their own user-written lockout procedure. This question will automatically be answered "NO" by the configurator if ISAM or ARC are not configured.

3.3.27 Formfeed After RELEASE

This question will allow the user to tell the Interpreter to issue a formfeed after a RELEASE only if the port issuing the RELEASE has control of the printer.

3.3.28 UDA (User Data Area) Size

This question allows the user to tell the configurator the size of the UDA for each configured port. There will be a highlighted message in the lower left hand corner of the screen telling the user exactly how many bytes are left for UDA allocation. The message will look like this:

UDA BYTES REMAINING = nnnnn

where nnnnn is a decimal number representing the number of bytes left for UDA allocation.

Here is a table of the maximum UDA allocation the ports may have for each memory size configuration. The more verbs and features added the less total UDA the ports will have.

64K - 16384 bytes for UDA.

96K - 28672 bytes for UDA.

128K - 49152 bytes for UDA.

3.4 Configuration Completion

After the last configured port's UDA has been entered the highlighted message

PRESS DISPLAY KEY TO COMPLETE CONFIGURATION.

will be displayed. The previous question cursor key will still bump the cursor back to the last configured port's UDA entry, so the user may change any part of the configuration at this time. When the DISPLAY key is depressed the screen will be cleared and

the following actions will be taken. At this time the user may not change any part of the configuration.

3.4.1 Creating the Rollfiles

After DS1500 has been configured the rollfiles (ROLLDS15/SYS & ROLLDS15/UDA) will be created if ROLLOUT is configured. The size of the rollfiles are determined by the configuration. If the disk(ette) does not have the needed file space for either of the rollfiles the following highlighted message is displayed:

> <rollfile-name> CANNOT BE CREATED ON DRIVE <drive number>. PLEASE ENTER A DRIVE SPECIFICATION WHERE <rollfile-name> CAN BE CREATED.

The user can enter a volid or a drive number in the form, :<volid> or :D<number>. If the volid or the drive number is off-line, the configurator will beep and the user must enter a valid on-line drive specification.

3.4.2 User LINKing of the Interpreter

The 1500 DATASHARE Interpreter must be LINKed if it has never been configured, or if the target machine memory size or the mode of port communication (FPCA, ICA, or MFC A/B) has been changed from the previous configuration. If none of these conditions hold, the following question is asked:

> FORCE LINK OF NEW CONFIGURATION?

If a positive response is given to this prompt the new configuration will be re-linked. If a negative response is given the new /CFG file is written to disk and the Configurator returns to DOS.

Note: If the port communication was ICA and the user changed the communication to MFC Channels A/B then a re-link will NOT be required.

If one or more of the re-link conditions hold, the message

> CONFIGURATION CHANGE REQUIRES RE-LINK OF <Interpreter-name>/CMD.

will be displayed. At this point all DOS programs necessary to

perform the LINK are searched for. The required standalone command files are: CHAIN/CMD, CHAIN/OV1, LIBSYS15/CMD, and LINK15/CMD. Required files that may be either standalone or members of UTILITY/SYS are: ABTONOFF/CMD and KILL/CMD. These files may be on any online drive. The two DS1500 files necessary for the LINK, i.e. DS1500/REL and DS1500/CHN, are also searched for; these must be on the same drive as the Interpreter. If all these files are found, the question

SHOULD A PRINT FILE OF THE LINK BE PRODUCED ?

will be asked. Responding YES to this question causes SORT/CMD to be searched for on any online drive, and a print file of the LINK to be produced with the name <Interpreter-name>/PRT. If the response to this question is NO, SORT/CMD will not be searched for and no print file of the LINK will be produced. If any of the required files are not found, the message

ONE OR MORE OF THE FOLLOWING FILES REQUIRED FOR RE-LINKING ARE MISSING OR WERE NOT FOUND ON THE SAME DRIVE AS THE INTERPRETER:
DS1500/REL, DS1500/CHN, LINK15/CMD, LIBSYS15/CMD, CHAIN/CMD CHAIN/OV1, KILL/CMD, ABTONOFF/CMD [, SORT/CMD]
CONFIGURATION ABORTED - CONFIGURATION FILE UNCHANGED.

will be displayed, and the /CFG file will not be written. In this case, the missing files should be placed online and the Configurator invoked again. If all necessary files are found, the Configurator will write the new /CFG file to disk, invoke the DS1500/CHN CHAIN file to perform the LINK, and place the created Interpreter module in the DS1500 command library.

3.5 Aborting From the Configurator

Whenever the user wishes to exit the configurator the KBD key or an '*' may be depressed in response to any question. The current configuration file is unchanged and the message

CONFIGURATION ABORTED - CONFIGURATION FILE UNCHANGED.

will be displayed. The configurator will then return to DOS.

3.6 Necessary Programs

Before the DS1500 Interpreter can be used, the ANSWER and MASTER programs must exist. These programs perform the tasks of logging on when an operator initially signs onto the system, and interfacing with the operator when no DATASHARE program is running. Since program execution in the DATASHARE system occurs in the high-level DATABUS language, the user writes his own ANSWER and MASTER programs, and therefore determines the structure of his system command language. ANSWER and MASTER programming concepts are dealt with more fully in Chapter 5.

If ANSWER and MASTER programs do not exist for a port, the port will never become active even if it is configured into the system. The ANSWER and MASTER programs may have the object names "ANSWERn" and "MASTERn" (where n is the number of the port serviced by the ANSWER and MASTER programs; n = 1 through the number of ports configured).

DS1500 initialization will look for the files ANSWERn/DBC and MASTERn/DBC. If initialization cannot find an ANSWER program with a name of ANSWERn, it will then search for a program with the name ANSWER. Similarly, the "second choice" MASTER program is named MASTER. In this manner all ports may share a single ANSWER or MASTER program if desired.

If a multi-drive system is being utilized, it is generally a good idea to keep all necessary system utilities, the DS1500 Interpreter, the /CFG file, and the ANSWER and MASTER program object files on the booted drive and not to remove the disk from this drive during normal system operation.

Other programs which should be available include the INDEX, REFORMAT, and SORT utilities if the generation of ISAM index files are desired.

3.7 Summary of Configuration Restrictions

This is a summary of possible invalid configurations that may occur. The configurator will check the configuration except for the extended memory allocated to a concurrent job.

3.7.1 Restrictions for any 1500 Processor

Here are some restrictions when configuring DS1500 on any size processor.

A. EXTERNAL COMMUNICATIONS and CONCURRENCY can not be configured simultaneously.

B. Maximum possible UDA for one port is 12K (11.75K if SPLOPEN/SPLCLOSE is configured).

3.7.2 Restrictions On a 64K Processor

Here are the restrictions when configuring DS1500 on a 64K processor.

A. Only ONE port may be configured if EXTERNAL COMMUNICATIONS or CONCURRENCY is configured.

B. Maximum possible total UDA is 16K (16384 bytes). This maximum is decreased as verbs and features are configured.

C. Maximum possible total UDA is decremented by 10K if EXTERNAL COMMUNICATIONS is configured.

D. Maximum possible total UDA is decremented by 8K if CONCURRENCY is configured.

E. Maximum possible total UDA is decremented by 1K, plus .25K for each port, if SPLOPEN/SPLCLOSE is configured.

3.7.3 Restrictions for 96K and 128K Processors

Here are the restrictions when configuring DS1500 on an extended memory machine.

A. Maximum possible total UDA on 96K machine is 28K (28672 bytes). This maximum is decreased as additional verbs and features are configured.

B. Maximum possible total UDA on 128K machine is 48K (49152 bytes). This maximum is decreased as additional verbs and features are configured.

C. The maximum possible total UDA on 96K and 128K processors is decreased by

4K for each extended memory sector that is allocated to the concurren region.

D. The maximum possible number of extended memory sectors (4K) that can be allocated to EXTERNAL COMMUNICATIONS is 6.

E. The maximum possible number of extended memory sectors (4K) that can be allocated to ACALL is 1.

F. If SPLOPEN/SPLCLOSE is configured, .25K of UDA must be reserved for a DS1500 SPOOLing work area for each port.

3.7.4 Memory usage of Configurable Verbs and Features

The following information is provided so that the user may be aware of the memory cost of various verbs and features. If a verb is not configured, the memory which that verb would have used will not necessarily be available for potential use as UDA. Any unallocated memory will be used for VS buffers. All of the sizes provided for the verbs are subject to change and should be treated as estimates.

A. EXTERNAL COMMUNICATIONS -- 8K between location 150000-167777

B. CONCURRENCY -- 8K between location 150000-167777

C. EXTERNAL COMMUNICATIONS routines -- 1.5 K

D. Extended memory sectors for EXTERNAL COMMUNICATIONS, CONCURRENCY, or ACALL -- 4K each

E. INTERNAL COMMUNICATIONS verb group -- 1.1 K

F. ISAM verb group -- 1.7 K

The following group of configurable verbs will be placed into an overlay if they are configured. If the configuration is for a 64K machine each verb will be placed in a separate disk overlay. The size of the overlay area will only be as large as the largest overlay verb that is configured. As each verb will be loaded from disk when it is to be executed (unless the verb was the last verb in the overlay area), repeated usage of two or more of the verbs could result in noticeable performance degradation.

If the configuration is for a 96K or larger machine, and one or more of the following verbs is configured, then the maximum possible UDA for the configuration is decreased by 4K. On a 96K or larger machine, the accessing of these verbs should not be reflected in any noticeable performance degradation (other than any degradation due to a reduced number of VS buffers). The structure of the DS1500 overlay mechanism as well as the size of the verb overlays are subject to change.

A. SPLOPEN/SPLCLOSE -- 1.2 K

B. EDIT -- .6 K

C. CHECK10/CHECK11 -- .2 K

D. NFORMAT/SFORMAT -- .2 K

E. PACK/UNPACK -- .2 K

F. SEARCH -- .4 K

G. SCAN -- .2 K

On extended memory machines (96K or greater), the memory is divided into 4K sectors. When memory is allocated to a port for UDA, only a maximum of 3 partial or full 4K sectors may be used for that port. As a consequence, it is possible that when more than 8K of UDA is allocated to a port, some memory that could have been used for UDA is not useable. For example, if only eleven 4K sectors were available for UDA allocation, a user could not configure 11K of UDA for four ports, as three 4K sectors would be needed for each port. The configuration of 11K for a port, left 1K of memory that was unuseable as UDA for any port needing more than 9K of UDA.

CHAPTER 4. SYSTEM OPERATION

In the following discussion, it is assumed that the 1500 DATASHARE Interpreter is named "DS1500", although any legal DOS filename may be used.

4.1 Command Line Parameters

The syntax of DS1500's command line is:

```
<Interpreter-name> [<special answer program>],[<system DATABUS library>]
                   [<alternate Multilink-driver name>];options
```

The command line options available are "T" to set the time, "D" to set the date, "C" to verify or change the configuration, and "R" to restart the Interpreter from a ROLLOUT. The "C" option is discussed in the "Verifying the Configuration" section of this chapter, and the "R" option is discussed in the "Restarting After a ROLLOUT" section of this chapter.

4.1.1 Time and Date Initialization

The time and date may be initialized from the DS1500 command line by use of the following options:

```
DS1500 ; T=hhmm,D=ddd/yy
```

In the above options, the time (T) is assumed to be in standard 24-hour format where hh = the hour and mm = the minutes. The date (D) is assumed to be in standard Julian format, where yy = the last two digits of the year, and ddd = the Julian date. Specifying the time and date will set the processor clock to the values entered, and override the reading of the processor clock upon bringing up the system.

4.1.2 Verifying the Configuration

The current configuration of the Interpreter may be examined by entering the command line:

```
DS1500 ; C
```

This will invoke the Configurator, and the configuration may be inspected. If the configuration is to be changed, this may be accomplished as outlined in the chapter on SYSTEM GENERATION.

4.1.3 Special Port One on Console Answer Program Feature

If the system is configured to run port 1 on the console, an alternate ANSWER program name for port one may be specified by entering the command line:

```
DS1500 <program>
```

where <program> is the name of a DATABUS object code file on an online drive. If this action is taken, the file <program>/DBC will be used for the ANSWER program for port one instead of ANSWER1/DBC (or ANSWER/DBC). (The MASTER program for port one is still MASTER1/DBC, or MASTER/DBC). When the system console is used for port 1, the console screen is blanked just before execution begins. If the program specified on the command line cannot be found, the message

```
MASTER OR TARGET PROGRAM MISSING
```

will be displayed and the system returns to DOS.

4.1.4 Naming Convention of the System DATABUS Library

DATASHAR/DBL is the default name for the system DATABUS library. If a name other than DATASHAR/DBL is desired for the system DATABUS library, then the name desired can be entered on the command line as the second parameter:

```
DS1500 ,<Library-name>/<ext>
```

<Library-name>/<ext> can be any legal DOS name.

During initialization DS1500 will scan all on-line drives, if not drive directed, for either the library specified on the command line or the default DATABUS library. If the library

cannot be found the highlighted message:

DATASHARE LIBRARY NOT AVAILABLE!

will be displayed and initialization will continue.

4.1.5 Naming Convention of the Multilink Driver

If the system is configured to execute a Multilink driver in the Concurrent partition, then the name of the Multilink driver can be derived two different ways. The default name of the Multilink driver is the <Interpreter-name>/COM. The other name of the Multilink driver can be given on the command line as the third parameter:

DS1500 ,,<Multilink-name>/<ext>

<Multilink-name>/<ext> can be any legal DOS name.

If Multilink is configured, DS1500 will scan all on-line drives, if not drive directed, for <Multilink-name>/<ext> and attempt to load the driver. If the Multilink driver is not loadable the error message:

MULTILINK DRIVER -- <Multilink-name>/<ext> WILL NOT LOAD.
INITIALIZATION ABORTED!

will be displayed. DS1500 will then return to DOS.

4.2 Bringing Up the System

After the DS1500 Interpreter has been configured and LINKed, it may be executed by entering its name on the DOS command line:

```
DS1500
```

This initiates the DS1500 System Loader, which checks the processor environment for the minimum acceptable standards to load the Interpreter. This initiates a series of operations, the first of which is to display the signon message:

```
DS1500 v.r - <date> -
```

where v is the version and r is the revision of the particular DATASHARE release. Since the Interpreter must determine its name from the DOS command line, AUTO-executing DS1500 is illegal. If DATASHARE has been AUTOed, the message

```
CANNOT AUTO-EXECUTE; USE "AUTOKEY/CMD".
```

will be displayed, informing the user of the correct way to automatically bring up DS1500. Next, the DOS.H version and revision are checked for compatibility with the Interpreter. If the Loader finds that the DOS.H version is not compatible, the message

```
EXECUTION REQUIRES DOS.H v.r OR LATER.
```

is displayed, where v is the version and r is the revision of the minimum acceptable DOS.H. If the DOS is acceptable, the machine memory size is checked against the minimum memory size acceptable for any configuration of the Interpreter. If there is not enough memory to load the Interpreter, the message

```
INSUFFICIENT MEMORY AVAILABLE FOR EXECUTION.
```

will be displayed. If the environment is acceptable to the Loader, the Interpreter module is loaded from the command library. If the Interpreter module is missing or unloadable, the message

```
MODULE <module-name> MISSING OR UNLOADABLE; EXECUTION
ABORTED.
```

is displayed. If the Interpreter module was successfully loaded, the loader will transfer control to it. This initiates a series of

operations. If the DS1500 Interpreter module has not been LINKed, the message

THE INTERPRETER HAS NOT BEEN LINKED.
USE THE "C" OPTION TO CONFIGURE AND LINK THE INTERPRETER.

is displayed. If the Interpreter has been LINKed, the command line is scanned for any options. If an illegal option is found, the command line is re-displayed with the first unrecognizable option letter highlighted, followed by the message

INVALID OPTION ON COMMAND LINE; INITIALIZATION ABORTED.

After the command line has been scanned, the configuration file is opened. If the configuration file cannot be found on the drive on which DS1500/CMD is located, the message:

CONFIGURATION FILE MISSING ON :DRnn

is displayed, where nn is the drive number 0 through 15.

Once the /CFG file has been found, the Interpreter checks the available memory of the processor. If the available memory size does not match the memory size that the Interpreter was configured and LINKed for, the message

INCORRECT MEMORY SIZE CONFIGURATION.

will be displayed. If the memory size is correct the configuration options format is checked for validity. If one or more options are found to be invalid,

DS1500/CFG IS INVALID.

will be displayed, indicating an old or damaged configuration file.

If the configuration options format is valid, then the Interpreter will check all options for the current environment. Here are a list of the error messages that may come up due to an illegal configuration option.

CONCURRENCY IS ACTIVE. CONCURRENT JOB SUPPORT IS NOT CONFIGURED

This informs the operator that the Interpreter was loaded with concurrency active and the user did not configure concurrent job support at configuration time.

ARC IS ACTIVE. ARC SUPPORT IS NOT CONFIGURED

This informs the operator that the Interpreter was loaded with ARC active and the user did not configure ARC Support at configuration time.

ARC IS NOT ACTIVE. ARC SUPPORT IS CONFIGURED

This informs the operator that the Interpreter was loaded with ARC not active and the user did configure ARC Support at configuration time.

After all the configuration options have been checked for the current environment, the initialization of the DS1500 system tables will begin. If DS1500 is configured to run on a machine with 96K or more of memory and there are not enough extended memory sectors to bring up DS1500, then the message

NOT ENOUGH EXTENDED MEMORY SECTORS FOR CURRENT CONFIGURATION

will be displayed. This indicates the user has configured too many extened memory sectors to the concurrent partition.

If there are enough extended memory sectors to satisfy all the options, then the virtual storage manager initializes and displays the following message:

VIRTUAL STORAGE INITIALIZATION COMPLETE - xxx BUFFERS ALLOCATED.

where xxx is the number of 256-byte buffers initialized.

If a port has been configured to run on the COMM channel and the channel is in use by the Concurrent partition or Multilink, the message:

COMMUNICATIONS CHANNEL NOT AVAILABLE.

will be displayed.

If a port has been configured to run on the MFC channel A and the channel is in use by the Concurrent partition or Multilink, the message:

MFC CHANNEL A IS NOT AVAILABLE.

will be displayed.

If a port has been configured to run on the MFC channel B and the channel is in use by the Concurrent partition or Multilink, the message:

MFC CHANNEL B IS NOT AVAILABLE.

will be displayed.

Note: Any of the above errors will cause an immediate return to DOS.

If the initialization is completed successfully, the system then checks the date and time options on the command line. If these options were not present, the processor clock is checked. If the clock is not set, the system displays the highlighted message:

OPERATOR, PLEASE DEPRESS THE KEYBOARD OR DISPLAY KEY.

Depressing the Keyboard or Display key verifies that an operator is present. This mechanism allows the time and date to be initialized by an operator as the system is being brought up.

If the KEYBOARD or DISPLAY key is not depressed within 10 seconds after the message is displayed, the machine produces a series of warbles in an effort to attract the attention of an operator. If the KEYBOARD or DISPLAY key is not depressed after 20 seconds of warbling, the system assumes that it is being operated in the unattended mode and initializes the time to 00:00:00 and the date to 000/00. This allows the system to bring itself up in case of power failure and/or unattended operation.

If the time and date are to be initialized, the operator depresses either the KEYBOARD or DISPLAY key. Upon doing this, the screen is initialized with a message indicating the version number of the DS1500 system being used, and the number of ports configured for the system. Also displayed is a column of digits on the left side of the console screen. Each digit denotes one message line which is allocated to its corresponding port. The CHAIN instruction displays the name of any program it invokes on the executing port's message line. Each port's message line may be used to display program status or operator messages through the use of the CONSOLE verb. An asterisk just to the right of the port number on the console screen will be displayed (if port one on the console is not configured) whenever that port is present and online.

To initialize the time and date, the system displays the

message TIME: in the upper right part of the screen. The operator responds to this with a four digit number indicating the current clock value in hours and minutes (HHMM). No colons are entered and a valid 24-hour clock value is required. If the value is not valid, the TIME: message is repeated.

If the time entry is valid, the system displays the message DATE: to the right of the time value just entered. The operator responds to this with a three digit number, followed by a slash, followed by a two digit number (ex. 001/80). The first number is the current Julian date (a number between 1 and 365 or, on leap years, 366) and the second number is the last two digits of the current year. This format mentioned must always be followed, with leading zeros used if necessary. If the Julian date is not valid, the DATE: message is repeated.

After the user enters the time and date, DS1500 sets the internal processor clock with the values given by the user. After the first initialization, DS1500 will not prompt the user for the time and date unless the processor is powered off.

Note that the time and date may be set either on the DS1500 command line, or interactively after prompting by DS1500 (only if the time and date have not previously been set), or by use of the DOS.H CLOCK utility.

After the time has been set, the system looks up all of the ANSWER and MASTER program names in the DOS directory and stores their physical file numbers in a table. If the ANSWER and MASTER programs are in a DATABUS library the LRN (logical record number) of the library member will be stored instead of the physical file number. Ports requesting connection during this time are connected, but the system makes no response until all the programs have been initialized. An asterisk just to the right of the port number at the left side of the console screen is displayed when the Carrier Detect signal for that port is present. This indicates that the terminal is physically turned on.

NOTE: If port one is configured to run on the console, the above console display is not present.

4.3 Taking Down the System

The SHUTDOWN and ROLLOUT instructions are the only correct methods of taking down the DS1500 system and restarting the DOS. Rebooting the processor is an improper way of taking down DATASHARE. If rebooting the processor is done indiscriminately, it can cause data to be lost, /ISI files to be incorrect, unused space to be left allocated to data files, and the termination of any concurrent job that was running.

ROLLOUT and SHUTDOWN are the only DATABUS instructions that ensure that all virtual storage buffers containing /ISI sectors are written to disk. Therefore, if the INDEXED SEQUENTIAL method of disk access has been in use on the system, a manual reboot of the processor must NOT be done, or all the /ISI sectors may not get written to disk. In case of an inadvertent reboot, or power failure, any /ISI files that were being processed by the programs should be re-indexed before being used again.

ROLLOUT and SHUTDOWN will also ensure that all write-pending data buffers are written to disk before the return to DOS. If LOGICAL (SEQUENTIAL) WRITES with write-pending (-2) are used in a program, the data entered could be lost if rebooting of the processor was done before the data buffers were written to disk.

The CLOSE instruction will also ensure that any write-pending data buffers, except /ISI buffers, for the particular file being CLOSEd will be written to disk. In addition, CLOSE is the only instruction which ensures that any unused disk space is deallocated from the data file being CLOSEd. The CLOSE instruction will NOT ensure that any write-pending /ISI buffers are written to disk, unless the purge /ISI option is configured (see section 8.6.1).

The preceding paragraphs outline the main reasons why ROLLOUT and SHUTDOWN are the safest and preferred means of taking down a DS1500 system. Manually rebooting the processor to restart DOS is strongly discouraged.

The ROLLOUT instruction is primarily intended as a means to TEMPORARILY suspend execution of all programs in order to execute a DOS command line. As a general rule, the DS1500 system environment should not be altered between a ROLLOUT and a ROLLOUT return (see the section on Precautions to Ensure a Successful ROLLOUT Return). Of course, the system environment may be altered after a SHUTDOWN. Thus, if it is desired to suspend all programs for a PROLONGED period of time, the SHUTDOWN instruction should be used.

4.4 Precautions to Ensure a Successful ROLLOUT Return

In order to ensure a successful ROLLOUT return, the DS1500 system environment should not be altered between a ROLLOUT and a ROLLOUT return. An exception to this rule is that the status of Concurrency in the machine may be altered.

The ROLLOUT section of the DBCMP15 User's Guide discusses several aspects of the system environment which should not be changed between a ROLLOUT and a ROLLOUT return. These apply to all DATASHARE systems. The following is a list (not necessarily inclusive) of additional aspects of the system environment which should not be altered before a ROLLOUT return. These apply specifically to DS1500.

1). If DS1500 is configured to use the 9320 FPCA, then it is assumed upon ROLLOUT return that the FPCA is initialized and set for the same baud rates as it was upon execution of the ROLLOUT instruction. A ROLLOUT return will fail, and the results will be indeterminate, if the FPCA was powered down or the baud rate settings changed between the ROLLOUT and the ROLLOUT return.

2). If DS1500 is configured to use the COMM channel, then it is assumed upon ROLLOUT return that the COMM channel is initialized and set for the same baud rate as it was upon execution of the ROLLOUT instruction. If the processor was powered off or rebooted, or if the COMM channel baud rate was changed before ROLLOUT return, the return will not be successful. This assumption also holds for both the MFC communication channels A and B.

3). If DS1500 is reconfigured, and the new configuration required the Interpreter to be re-LINKed, the ROLLOUT return will not be successful. The Interpreter should not be reconfigured between ROLLOUT and ROLLOUT return.

4). The files ROLLDS15/SYS and ROLLDS15/UDA may not be renamed.

4.5 Restarting After a ROLLOUT

If ROLLOUT was configured during system configuration, two files were created: ROLLDS15/SYS, which contains the Interpreter memory image, and ROLLDS15/UDA, which contains each port's UDA and the port one on the console screen image (if port one is configured on the console). Whenever a ROLLOUT instruction is performed by DS1500, the information necessary to restart the

Interpreter from a point immediately after the ROLLOUT instruction is written to these two files. If ROLLOUT was not configured, control is returned to the DATABUS program with the OVER flag set. In order to restart DS1500 from these files, the option ";R" must be entered on the command line after the Interpreter's name, i.e.

```
DS1500 ; R
```

This command causes the message

```
DS1500 v.r - <date> - RETURN FROM ROLLOUT PROGRAM
```

to be displayed on the console and the roll files will be read into memory to perform the restart. The time and/or date may be re-initialized by the "T" and "D" options on the command line. If the "T" and "D" options were not on the command line and the processor firmware clock has not been set, the time and date will be requested. After the roll files have been loaded, execution of DS1500 will continue exactly as if the rollout instruction had caused no action to be taken.

4.6 Special Considerations on Write-buffering

Write-buffering may be used to increase the speed of sequential writes by performing logical record writes using a value of "-2" for the record specifier.

If write-buffering is used, the DATABUS disk I/O instruction will not immediately write the sector to disk; typically the sector is merely flagged as "pending" a write to disk. The actual write occurs under one of the following conditions:

a) If any user program performs a "SHUTDOWN" or "ROLLOUT", or the user program performs an explicit "CLOSE" of the file.

b) The virtual storage buffer containing the write-pending sector becomes the least-recently-used buffer and is required for some other purpose.

The consequences of this are as follows:

a) The DATASHARE Interpreter may not discover that a disk has gone off-line until one of the above conditions occurs.

b) The DATASHARE Interpreter may not discover that the disk sector destined to receive the buffer has irrecoverable bad parity until one of the above occurs.

c) Data contained in the write-pending buffers will be lost if the processor is restarted or powered off before the physical write is performed.

Thus, a trap set for a parity or disk off-line condition during a "WRITE" instruction, for example, may not actually be taken until many instructions later. The error message generated may not have a program counter address pointing to the "WRITE" instruction responsible, but to an unrelated instruction much farther along in the program.

If it becomes necessary to restart the DOS, executing a ROLLOUT or SHUTDOWN instruction will ensure that any write-pending buffers are written to disk.

4.7 Executing DS1500 with Concurrency

Concurreny is supported under all DS1500 configurations except when Multilink is configured. In a 64K machine DS1500 is restricted to 1 port when Concurrency is active. On a machine of 96K or greater up to 4 ports can be configured with no restrictions on Concurrency. Running DS1500 with Concurrency active is identical to running any other DOS program with Concurrency active. See the JOB15 User's Guide for more information on Concurrent jobs.

When DS1500 is executing with Concurrency active, the console screen and keyboard (hereafter referred to as the "console screen") are a resource that is shared with the Concurrent job, just as the system printer is a shared resource. That is, DS1500 and the Concurrent job cannot use the console screen at the same time. The partition wishing to use the console screen must wait until the other partition has released control of the resource. When DS1500 must wait to use the console screen, because the Concurrent job currently has control of it, it may appear that the Interpreter has halted. Actually, the Interpreter is still

running, but is just delaying certain functions until it can acquire the console screen.

The following list contains information pertinent to the running of DS1500 with a Concurrent job. This information can assist the user in planning and optimizing any DS1500 programs or configurations that are intended to be run concurrently.

1. Any console screen I/O that has been started by the Interpreter or the Concurrent job must be completed before the Interpreter or Concurrent job can relinquish control of the console screen. For example, if the Concurrent job is waiting on keyin from the operator, any console screen I/O which DS1500 wishes to do will be delayed.

2. All DS1500 console screen I/O is queued so that the Interpreter can do other port's processing while waiting for access to the console screen. Console I/O that is queued include:

    a. KEYIN and DISPLAY statements executed by the port one on the console program.
    b. CONSOLE statements executed when port one is not configured on the console.
    c. Implicit CONSOLE statements caused when a CHAIN or STOP is executed and port one is not configured on the console.

3. The DATABUS instruction which initiated the console screen I/O cannot continue until the console screen I/O has completed. Thus the port which was executing the instruction will have to wait if the console screen is under the control of the Concurrent job.

4. ROLLOUT will wait until all queued console I/O is completed.

5. One of ROLLOUT's final actions before returning to DOS is to save the processor screen if port one is configured on the console. Thus, ROLLOUT cannot complete if the Concurrent job gains control of the screen before the console screen has been saved.

4.8 Fatal Error Conditions

Some error conditions within the DOS cannot be trapped. These errors invoke a DOS overlay called the ABORT overlay which reloads the DOS to ensure the presence of the DSPLY$ routine. An error message in the standard DOS format is then displayed and control is returned to the DOS Command Interpreter. A Concurrent job may also cause irrecoverable DOS errors. See the DOS.H User's Guide for the error messages and their causes.

CHAPTER 5. ANSWER AND MASTER CONCEPTS

There are two DATABUS programs required for each port that will be active in the system. The first is called the ANSWER program. The ANSWER program deals with the user when he initially connects to the system; i.e., dials in on the telephone or turns on his direct-wired terminal. The second program is called the MASTER program. The MASTER program interfaces with the user whenever he is not executing the ANSWER program or an application program. The MASTER is generally written to allow the user to select the next application program he wishes to execute. Note that both ANSWER and MASTER are written in DATABUS, enabling a user to tailor the command aspects of the DATASHARE system to suit his particular needs. Simple examples of ANSWER and MASTER programs are shown in Appendix C.

The ANSWER and MASTER programs may have the object names "ANSWERn" and "MASTERn" (where n is the number of the port serviced by the ANSWER and MASTER programs; n = 1 through the number of ports configured). DS1500 initialization will look for the files ANSWERn/DBC and MASTERn/DBC. If initialization cannot find an ANSWER program with a name of ANSWERn, it will then search for a program with the name ANSWER. A similar search is done for the MASTER program. All ports may share a single ANSWER and/or MASTER program if desired. Note that if no ANSWER and MASTER program can be found for a port, that port will never become active.

5.1 System Security

The ANSWER concept allows the DS1500 system to force users to give some type of identification before they are allowed to use the system. The INTERRUPT key sequence on the terminal is ignored while execution is taking place between the time the system first acknowledges the presence of a user at his port and the first CHAIN instruction executed by the program for that port. In other words, while the user is executing the ANSWER program for his port (after first signing onto the system or later CHAINing to the ANSWER program,) he will not be allowed to escape around any identification request and go directly into the MASTER program by simply striking the INTERRUPT key sequence.

The ANSWER program should set traps for all of the trappable errors (IO, PARITY, RANGE, etc.) to prevent an untrapped error

from allowing the user to escape around the identification request. If an untrapped error should occur during the execution of the ANSWER program, the MASTER program will begin execution.

The ANSWER program may also be structured to enforce file access limitations depending upon the identification of the user.

Execution of a STOP instruction by any DATASHARE program (including ANSWER) brings up and executes the MASTER program for that port.

5.2 System Convenience

The ANSWER program chains (via a STOP instruction) to the MASTER program, which usually requests the name of the program the terminal operator wishes to execute. This name is generated from information supplied by the terminal operator. For example, the operator might enter the number of a data-entry form; the MASTER program would then decide which program to execute for that form number. The DOS directory cannot be directly accessed by the MASTER program, implying that if a directory service or file access limitation is to be implemented, a file must be generated which contains the names of programs and files that are to be accessed. It is the responsibility of the author of the ANSWER and MASTER programs to provide any such convenience facilities to the terminal user.

5.3 ANSWER and MASTER Programs

If the DATASHARE object file for either the ANSWERn or ANSWER, or MASTERn or MASTER programs do not exist, port n will not be activated when the system is brought up.

COMMON data areas used in the MASTER program should be used with caution; the MASTER program may be entered due to a program trap or the INTERRUPT key sequence being struck. If programs which have been CHAINed do not preserve this data area, the MASTER program will become confused as to the contents of the COMMON area.

When the ANSWER program is activated it executes until a KEYIN or DISPLAY statement is encountered (the ANSWER program will be activated whenever a given terminal disconnects from the system, not when it connects to it). If, for example, the time of connection, disconnection and total connection time are being kept in a file, the ANSWER program's first operation might be to note

when the user disconnects from the system and to log the total amount of time he was connected. Then, perhaps, a KEYIN statement requesting a new user identification would be issued and cause execution to cease for that port.

For example: A log-out function is executed when a terminal disconnects from the system. When the terminal again re-connects to the system and the user at the terminal enters an identification code, the previously-mentioned KEYIN statement would be satisfied and the new user would then be logged on with his sign-on time being recorded in the log file.

When a system is initialized, all ports would appear to be logging off (since all ANSWER programs are executed), but no corresponding log-on times will be set. The ANSWER program must therefore handle this special case by allowing for log-offs without corresponding log-on times.

Appendix C contains examples of both ANSWER and MASTER programs.

5.4 Down-line Loading of 8200 FUNCTION Keys

When 8200 terminals are in use on the system, it may be necessary for the ANSWER program to down-line load a character set to the 8200. This is especially important if the 8200 FUNCTION keys are to be recognized by DS1500, since the default 8200 character set (loaded at power-up by the terminal firmware) does not contain the correct FUNCTION key values.

Two files, FUNCTION/TXT and FUNCTION/SYS, are provided with the DS1500 system release to demonstrate how an ANSWER program can down-line load the FUNCTION keys to an 8200.

CHAPTER 6. SYSTEM CHARACTERISTICS AND PERFORMANCE CONSIDERATIONS

6.1 Virtual Storage - An Introduction

During initialization the DS1500 memory manager will allocate all unused memory to the Virtual Storage (V/S) system. The memory is broken up into 256 byte pages (this includes the DOS reserved bytes), referred to as V/S buffers. Disk sectors from /DBC files, data files, and /ISI files are kept in these V/S buffers. Before a disk read is executed, the V/S will check to see if the sector desired already exists in memory. If so, the sector is used without doing a disk read. If the sector is not in memory the V/S will "bump" out the "least recently used" buffer and read in the sector desired. This "least recently used" buffer has now become the "most recently used" buffer. Buffered disk sectors used frequently tend to stay in memory for faster access. DATABUS programs can increase their speed by taking advantage of this virtual storage system.

6.2 Virtual Memory

To achieve a reasonable amount of program space for four simultaneously executing programs, DS1500 employs a virtual memory manager. DATABUS code is very compact, with a few bytes of object code capable of evoking a large amount of processor activity; the rate at which DATABUS program bytes are fetched is therefore very low. This low rate allows the program code bytes to be kept in randomly-accessible memory buffers; sequentially executing program instructions thus have very little effect on program execution speed. Another characteristic of DATABUS object code is re-entrancy; since this code is never modified, it never has to be written to disk. In addition, ports which are executing the same program can share the program code pages.

Program data (UDA) is accessed at a very high rate and must be in main memory to be effectively available to the DS1500 Interpreter. For this reason all program data is kept resident in main memory (which is divided among the ports configured). This has further advantages in the case of port/printer I/O, as will be mentioned later.

In order to facilitate an effective virtual memory accessing

algorithm, the program code is kept on the disk in 250-byte pages. Whenever a byte of DATABUS object code is fetched by the Interpreter, a check is made to determine if the page of code containing the byte is immediately available or if it must be read from the disk. Any time a page boundary is crossed, a new page must be read in if it is not already in the buffer. The time required to read a page of code from disk, in order to fetch a byte, is much greater than the fetch time for a byte in a page already in a buffer. The DATABUS programmer can, in general, increase his code's execution speed by forcing his program to cross as few page boundaries as possible. Actually, in a lightly loaded system, a single program could have a number of pages all resident in memory buffers at once. Thus crossing a page boundary would probably not cause a disk read. However, any significant increase in loading would nullify this ideal condition. The DATABUS programmer should therefore assume that any time his program crosses a page boundary a disk read will occur and cause a delay in the execution of the program. By causing an excessive number of page boundary crossings, a programmer can easily cause his program to execute very slowly.

The instruction TABPAGE exists in the DATABUS language to aid a programmer in making his program's execution speed as high as possible. During compilation, this instruction causes the location counter in the compiler to be incremented until it is at the start of the next page (nothing will be generated if the location counter was already at the start of a page). When this instruction is executed, it causes a GOTO to the start of the next page. By using this instruction, a programmer can cause logical parts of his program to cross as few page boundaries as possible.

Other methods that may increase execution speed are:

a) the use of in-line coding, especially for short operations
b) the grouping of commonly used subroutines in the same code pages
c) the use of TABPAGE prior to often executed loops to reduce the reading of code pages from disk

The effectiveness of these techniques will, of course, depend on the application, system load, and processor memory size.

6.3 Virtual Storage Usage

All memory space not used for some other purpose is used for virtual storage (V/S). DS1500 can use a maximum of 512 pages of virtual storage. A page is 256 bytes and corresponds to one sector (physical record) on the disk.

Whenever a valid image of a disk sector is available in a virtual storage page, that page will be used instead of reading the sector from disk or from the file processor of an ARC system. Using the page in memory is much faster than reading from disk or an FP, this re-utilization of sector images in memory is critical to the performance of a DATASHARE system.

6.3.1 Volatility

Sectors read into virtual storage pages can be either volatile or non-volatile. A non-volatile sector is a sector that can not change. Sectors of DATABUS code are considered non-volatile, as are intermediate level sectors of ISI files, sectors of write-protected text files and all sectors of ISI files associated with write- protected text files. DS1500 will never attempt to write a non-volatile sector to disk. All other sectors are considered volatile: highest level and lowest level ISI sectors, text file sectors, and system table sectors.

On ARC systems it is important to be certain that DS1500's assumptions about non-volatile sectors are not violated. If you INDEX a file or compile a DATABUS program on one applications processor while that file or program is in use at another applications processor, you will have changed sectors that must not change during DS1500 execution and the results will be unpredictable, (read: unpleasant).

6.3.2 The Page Lists

DS1500 uses three lists to keep track of virtual storage pages. The "free" list contains entries for all pages that do not contain a sector image. The "busy" list contains entries for all pages that do currently contain a sector image. The "enqueue" list contains entries for all pages that contain a sector image associated with a file that is currently enqueued.

When a new sector must be read from disk, it must be read into a virtual storage page. The page to be used is chosen first from the free buffers. The page so used is removed from the free

list and is placed on either the busy list or the enqueue list, as appropriate. When an enqueue is terminated, the associated pages on the enqueue list are moved to the busy list. If there are no free pages available, then the least-recently-used (that is, the page which has had the same contents for the longest time without being used by any port) busy page is overwritten. Its previous contents are lost and must be read from the disk again if they are needed later. In the unlikely event that there are no entries on the free list or the busy list, the least-recently-used page on the enqueue list will be used.

The contents of each page on the busy and the enqueue list is identified in the list entry for that page. Most sectors are identified by PDN (Physical Drive Number), PFN (Physical File Number), and LRN (Logical Record Number). Some sectors (such as ISI sectors) are identified by PDN and PDA. When any DATABUS program needs a sector, the busy and enqueue lists are searched for a match on PDN/PFN/LRN or PDN/PDA as appropriate. The search method used builds a hash code from the search argument and then searches a linked list of list entries having that hash code.

Different DATABUS verbs can cause a page to be "purged". When a page is purged it is taken from the enqueue or busy list and placed on the free list. It then no longer contains a sector image at all.

NOTE: The term "purge" should not be confused or used interchangably with the term "flush". "Flush" refers to the process of forcing sectors still in V/S to be written to disk. This does not necessarily imply that they are removed, or "purged" from V/S; in fact, there is a good chance that the sector images will still be in V/S. "Purge" refers to the process of removing the sector image from V/S; (in the process of "purging" sectors they are also flushed.)

6.3.3 Validity

When DS1500 is running in a stand-alone DOS environment, any V/S copy of a sector is always valid, so it may be used any time until it is overwritten. Under ARC operation, the contents of a page is valid if it contains non-volatile data. If it contains volatile data then it may be valid if

1) no enqueue is currently active and the access method of the read is sequential, or

2) an enqueue is currently active and the page is on the

enqueue list.

6.3.4 V/S Utilization

The most important use of virtual storage pages is to store sectors of DATABUS code. DATABUS code is stored on disk in pages that correspond to the capacity of a virtual storage page. In a program compilation listing one can find the page boundaries of the code by noting where the "hundreds" (octal) digits of the addresses on the listing change from 3 to 4, or from 7 to 0. DATABUS object code is reentrant, so several ports can use the same copy of a page of code at the same time. Each time a new instruction is fetched, the DATASHARE interpreter must see if it already has in memory the page in which that instruction resides. If the page is not in memory, it must be read from the disk. Since DATABUS code is non-volatile, any sector image found in V/S is valid.

6.3.5 Verb Effects

Several verbs have special effects on 1500 DATASHARE Virtual Storage.

CLOSE -- When a file is CLOSEd, all text sectors from that file are purged from V/S. Any writes pending are forced out at this time (flushed). The implicit close that occurs when a program ends does not purge sectors or force out pending writes. Due to the much larger number of V/S pages available under DS1500, programs performing "write pending" functions that previously (accidentally) worked may suddenly encounter problems if they do not properly CLOSE their files.

CHAIN -- CHAIN will purge all V/S copies of sectors of the DATABUS program being CHAINed to, if the program file is a separate /DBC file, if it is not write protected, and the Interpreter is in an ARC environment. CHAIN never previously purged sectors, unless the /DBC file is on a diskette, and systems using non-write-protected separate program files may see performance degradation caused by this feature.

ROLLOUT / SHUTDOWN -- When a ROLLOUT or SHUTDOWN is performed, all pending writes are forced from V/S. Upon rollback from a ROLLOUT, no pages are preserved, so all previous contents are lost and V/S starts with all pages free.

UPDATE / DELETE / INSERT -- DS1500 maintains a pointer to the

last record accessed for each indexed file currently open (here "indexed" refers to ISAM). This file specific pointer is called the "local pointer". Each port also maintains a "global pointer" which is the pointer to the last indexed record accessed by that port. Both the local and global pointers are set by any indexed read or write operation (READ, READKS, or WRITE). Random and sequential accesses do not affect either pointer.

An INSERT uses the global pointer to determine what text record is to be inserted in the specific index file. Neither the global or local pointer is modified by this operation. Past versions of Datashare did not maintain the PFN with their global pointer, so programs could erroneously insert text records from the wrong text file into an indexed file. DS1500 "remembers" the PFN of the text file in the global pointer; any attempt to INSERT to an inappropriate index will cause an I/O error.

An UPDATE instruction uses the local pointer for the specified file. Neither the local or global pointer is modified by the update operation.

The DELETE operation uses the local pointer for the specified logical file. The global pointer is not modified on a DELETE but the local pointer is cleared, leaving it in the same state as it was prior to any valid indexed READ or WRITE. Unlike previous Datashares, DS1500 will issue an I/O error when a DELETE instruction is followed by an UPDATE.

*NOTE: See Section 8.6, which gives additional information on Shared Files, and the effects of the CLOSE verb.

6.4 Scheduling

In order to provide optimum response time, DS1500 handles all port and printer I/O using task-driven foreground routines. Data transfer between the devices and the system can therefore occur regardless of the computational task being handled by the background program(s) at any given time. The foreground routines interpret the KEYIN, DISPLAY, PRINT, and CONSOLE instructions while the background interpretive code merely passes these instructions to the foreground via a circular buffer allocated for each port. Conventional systems use such a buffer to hold any characters transferred between the system and the device. DS1500 uses this buffer to hold interpretive code bytes which enables many more bytes of information to be transferred than can be held in the buffer. For example, a DISPLAY statement may contain some text information followed by a variable name. The variable name

would be represented by two bytes, but the contents of the variable might be fifty bytes long, enabling two bytes of buffer space to cause the transfer of fifty bytes to the device. This is made possible by the fact that all program data is resident in main memory and thus is commonly accessible to both foreground and background.

The foreground and background routines for a given user always execute exclusively of each other to prevent conflicts over data values. When the background routine executes a DISPLAY statement, the statement is stored in the buffer for the user. The background routine then relinquishes control to the foreground routine. When foreground has completely executed the I/O statement it causes a high priority interrupt in background, resulting in a task-switch to the program executing the DISPLAY statement.

An important consideration concerning port I/O must be taken into account by the DS1500 programmer: each time an I/O instruction is completed in the foreground the background swaps in the user's program. If this consideration is not kept in mind, a careless programmer might cause the system to spend most of its time swapping between ports instead of doing useful work. For example, the use of one continuous DISPLAY statement is more efficient than the use of many separate DISPLAY statements.

An important consideration concerning printer I/O must also be taken into account: the user may check for printer availability by using the RELEASE instruction. RELEASE will set the over flag if the printer is off-line or not available. The use of RELEASE to check for printer availability does not guarantee that the printer may then be allocated to the port. When the first PRINT instruction is encountered, the Interpreter will attempt to allocate the printer for the use of the port wishing to print. In the interval between the RELEASE and PRINT instructions, the printer may have been allocated to another port, or a Concurrent job may have received exclusive use of the printer. If the printer is not available to a port executing its first PRINT instruction, the port will wait until the printer is available.

The above discussion concerns only port, printer, and console I/O. All disk I/O is performed under the DOS, which is a background only operation. This means that all DOS activities are non-interruptible. Long directory searches (which may take considerable time) will cause poor response to I/O-completion interrupts. Lengthy DOS activities, however, occur infrequently and may be ignored from an average response time standpoint.

When the background process resumes execution due to the completion of a foreground I/O task, it is guaranteed a minimal amount of execution time. This prevents the system from spending all of its time swapping background tasks when foreground I/O-completion rate is high.

6.5 Terminal Devices

DS1500 is capable of driving certain serial terminal devices which use an ASCII character set. Use of devices without cursor-positioning features, however, will restrict the programmer from using the cursor-positioning facility in the KEYIN and DISPLAY statements. If the programmer does not use the cursor-positioning feature he will be able to write a program which is Teletype-machine compatible. The *ES and *EL list controls send control characters that are ignored by a model 35 ASR Teletype. The "Cursor On" character which is sent before each KEYIN variable entry request and the "Cursor Off" which is sent after the ENTER key is struck, are "TAPE ON" and "TAPE OFF" respectively on a model 35 ASR Teletype.

DATASHARE is also capable of handling 9408/9409/9478/9479 communications adapters, 103-type datasets, and full-duplex four wire 202 dataset connections. It handles all of the 103 handshaking involved and needs only the proper cable to work correctly. In fact, the 3600 hard-wire cable is connected in such a way that it makes the 3600 appear to be a 103 data set; power-on causes Ring-Detect and Carrier-Detect to be sent to the DS1500 system. The use of a hard-wire or dataset connection employed at a given terminal cannot be differentiated by the DATABUS programmer. (The chapter on PHYSICAL INSTALLATION contains more information about terminal connections.)

CHAPTER 7. PHYSICAL INSTALLATION

7.1 Main Peripherals

The DS1500 system requires a disk peripheral. Since the system maintains its entire file structure under the DOS, many logical drives may be available. DS1500 is compatible with all disk types supported by DOS.H.

Attempting to change mapped volumes while DS1500 is running or rolled-out may produce indeterminate results. It is not advisable to change the logical drive mapping, or swap mapped volumes.

Under DS1500, the Four Port Communications Adapter (FPCA), the Internal Communications channel (ICA), or the MFC channels A and B may be used to communicate with a DATASHARE port.

The FPCA is capable of driving up to four fully independent, full-duplex, asynchronous lines at speeds ranging from 50 to 9600 baud. These rates are software programmable and are configurable from DS1500. This enables the DATASHARE system to achieve baud rate compatibility with specific terminals. Communications are made through the microbus leading from the processor to the FPCA (the FPCA's microbus address is 4). The microbus contains an 8-bit command bus, an 8-bit data bus, and two command strobes. Since these signals are all Transistor-Transistor Logic (TTL) level, they are not intended to be used over cables more than ten feet in length.

The COMM channel is capable of driving a single DATASHARE port in full-duplex, asynchronous mode at line speeds of 75 to 9600 baud. The baud rates are software programmable and are configurable from DS1500. Although the cable for the COMM port has pin connections similar to the FPCA port, the COMM port cable requires a male connector on the 1500 while the FPCA port cable requires a female connector on the FPCA. The 8200/3600 connection is the same for both cables.

The MFC channels A and B are capable of driving two DATASHARE ports in full-duplex, asynchronous mode at line speeds of 75 to 9600 baud. The baud rates are software programmable and are configurable from DS1500. The cable for the MFC channel A is

identical to the cable for the Internal Communications channel. The cable for the MFC channel B is identical to the FPCA cable.

Note: The MFC is an optional configuration only on a 1560 series processor.

There is a restriction on the Receiving baud rate of the MFC channel B. The baud rate can be either 1/8th of the transmit baud rate or equal to the transmit baud rate. The receive baud rate clock is taken from the transmit baud rate clock of channel B. The 1/8th baud rate of the transmit clock comes in for special modems (9408/9,9478/9) that run at 150 baud transmit and 1200 baud receive. (From the DS1500 side this means 150 baud receive and 1200 baud transmit.) The configurator for DS1500 assures that no incompatible baud rates are entered at configuration time for MFC channel B.

7.2 Terminal Connections

In general, a remote terminal functioning as an external port may be connected to the DS1500 system in one of four ways: 1) direct hard-wire, 2) through a 103-type modem, 3) through a 202-type modem, and 4) through a 9408/9 or 9478/9 communications adapter. There are four 25-pin receptacles on the back of the FPCA card in the 9320 unit, one for each external port. There is also a 25 pin receptacle on the back of the 1500 for one port using the COMM channel and a 25 pin receptacle on the back of the 1560 for two ports using the MFC channels A/B. Pin assignments for the attached connectors are shown in the following table in columns 1 and 2. Columns 3 and 4 show pin numbers and signal names at the other end of the connection at the 8200/3600 terminal, or at a 103/202-type modem, or at a 9408/9409/9478/9479 communications adapter.

PIN	FPCA/COMM/MFC	8200/3600	103,202,9408/9,9478/9
1	PROT GROUND	SIG GROUND	SYS GROUND
2	DATA OUT	DATA OUT	DATA IN
3	DATA IN	SIG GROUND	DATA OUT
4	REQ TO SEND	DATA IN	REQ TO SEND (202)
5	CLR TO SEND	SIG GROUND	CLR TO SEND
6	DATA SET READY	-	DATA SET READY
7	SIG GROUND	-	SIG GROUND
8	CARRIER DET	PRINTER DATA	CARRIER DET
-			
11	-	SIG GROUND	-
12	-	DATA SET READY	-
-			
15	TRANSMIT CLOCK	-	-
-			
17	RECEIVE CLOCK	-	-
-			
20	DATA TERM RDY	-	DATA TERM RDY
-			
22	RING DETECT	-	RING DETECT

NOTE: Pins 18 and 25 are unique to the COMM channel. Pin 18 is an internal transmit clock and pin 25 is an internal receive clock.

The DS1500 system automatically goes through the following handshaking procedure when a connection is established:

1. Clear Data Terminal Ready and Request To Send
2. Wait for Ring Detection
3. Set Data Terminal Ready and Request To Send
4. Wait up to ten seconds for Carrier-Detect
5. Go to step 1 if time-out in step 4
6. Wait one second, and then start the ANSWER program

This procedure works with any of the three types of connections if the proper cable is used.

7.2.1 Direct Connection to Port

The direct connection cable swaps the data wires (pins 2 and 3) and connects Carrier-Detect, Ring-Detect, and CTS on one end to Data Terminal Ready on the other end as shown in the following table:

FPCA/COMM/MFC TO 8200/3600 CABLE CONNECTIONS

FPCA/COMM/MFC	8200/3600
2	4
3	2
7	3
8,22,5	12

NOTE: For the COMM channel ONLY, pin 15 is tied to pin 18, and pin 17 is tied to pin 25.

This arrangement requires only four wires in the cable. If the cable is to be more than several hundred feet long, each of the two signal wires (the ones connected to pins 2, 3, and 4) should be twisted separately with a ground wire; no other shielding is necessary. Direct connections up to one thousand feet may be made if the above precautions are followed. (It should be noted that there are some limitations: a cable length of more then 500 feet will not sustain a 9600-baud line.)

The 8200 sets Data Terminal Ready whenever it is powered on. With the above cable connected, this will cause Ring Detect, Carrier Detect, and CTS to be presented to the FPCA and COMM. This has the effect of causing the ANSWER program to be executed whenever power is applied to the 8200.

7.2.2 103-Type Modem Telephone Interface

The FPCA, COMM, and MFC can be connected to a 103-type modem with a one-to-one cable (e.g., a pin at one end is connected to a pin of the same number at the other end). Only pins 2, 3, 5, 7, 8, 20, and 22 in the connector are used, but the connection works properly even when all the pins are connected. (It may be helpful to know that 103 and 113b modems have similar pin connections.)

FPCA/COMM/MFC TO 103-TYPE MODEM CONNECTIONS

FPCA/COMM/MFC	103-TYPE MODEM
2	2
3	3
5	5
7	7
8	8
20	20
22	22

When calling a 103-type modem over a dial-up network, if automatic answer does not occur by the second ring, the DS1500 system is down (or the wrong number was dialed). When the system answers the call, the originating end of the line hears the carrier signal from the modem connected to the FPCA; at this tone, the calling end activates its modem's data key or makes a coupler connection. DS1500 allows 10 seconds for either of these actions that return the carrier tone. If all the actions are satisfactorily completed, DS1500 allows one more second, then begins execution of the ANSWER program.

If the call is not satisfactorily completed, DS1500 hangs up the telephone at its end and resumes waiting for ringing to occur. Since DS1500 waits up to ten seconds for a satisfactory connection, there must be a 10-second wait before the same telephone is dialed again. Also, DS1500 disconnects as soon as it loses the Carrier-Detect signal from the caller's modem. This means that disconnection occurs if carrier is lost for even a fraction of a second.

There is no Ring Detect status for MFC channel B, therefore auto answer across a phone line is not possible with a 103-type modem with MFC channel B.

7.2.3 202-Type Modem Telephone Interface

DS1500 requires a full-duplex connection to its terminals. A 202-type modem can be used for full duplex only if it is connected through a four-wire circuit, creating two separate signal paths: one for data flow in one direction and another for data flow in the opposite direction. This implies that there must be a point-to-point (non-switched) connection between the modems, since a switched telephone network cannot support four-wire connections.

The 202-type modem must be strapped in a four-wire mode to be used in this application.

The connecting cable between the FPCA/COMM/MFC and 202 modem is similar to the one used for connection to a 103-type modem except that 202s used in point-to-point four-wire service do not use ringing. The carrier-detection signal from the 202 is connected to both the carrier-detection and ring-detection inputs on the FPCA/COMM/MFC (pins 8 and 22).

FPCA/COMM/MFC TO 202 MODEM CONNECTIONS

FPCA/COMM/MFC	202 MODEM
2	2
3	3
4	4
5	5
7	7
8,22	8
20	20

When Data Terminal Ready is supplied by the terminal device to the remote 202 modem, that modem turns on its carrier. The carrier signal causes the modem connected to the FPCA/COMM/MFC to turn on its carrier-detect signal, and presents ring-detection and carrier-detection to the DATASHARE system. The system then proceeds to set its Data Terminal Ready signal, causing the 202 to turn on its carrier and complete the connection. One second later the ANSWER program resumes execution. In this way, operation over a 202 modem telephone connection appears similar to the direct connection operation described in section 7.2.1.

Remote modems are connected to Datapoint 3600 and 8200 terminals via a standard modem cable supplied with the terminal. This cable provides the required Data-Terminal-Ready signal that causes the operational characteristics described above.

7.2.4 9408/9 or 9478/9 Communications Adapter

These devices are somewhat akin to the 202 type modem, except that the Datapoint communication adapters work over switched networks, and are fixed baud rate devices (1200 baud and 150 baud). If the 9408/9 are used, a special DAA (Direct/Data Access Arrangement) is necessary - 1001A or 1001F. If the 9478/9 are used, no DAA is necessary. The pin connections are as follows:

FPCA/COMM/MFC TO 9408/9478 MODEM CONNECTIONS

FPCA/COMM/MFC	9408/9478
2	2
3	3
4	4
5	5
6	6
7	7
8	8
20	20
22	22

7.3 Baud Rate Selection

The FPCA, COMM, and MFC are possible links for communications to the outside ports. These adapters are software programmable, making it possible to have a different baud rate for every port, if desired. The FPCA/COMM/MFC baud rates should be paired up with the 8200/3600 baud rates as follows:

Transmit baud rate (8200/3600) = Receive baud rate (FPCA/COMM/MFC)

Transmit baud rate (FPCA/COMM/MFC) = Receive baud rate (8200/3600)

Baud rates that may be used for the 3600 and 8200 terminals are given in the following table:

BAUD RATE TABLE

FPCA	COMM/MFC
50	-
75	75
110	-
134.5	-
150	150
200	-
300	300
600	600
1200	1200
1800	-
2400	2400
4800	4800
9600	9600

7.3.1 8200 Terminal

The 8200 terminal can be manually configured from the keyboard by putting the 8200 in the configuration mode. The procedure is as follows.

1) Turn the 8200 off; wait 10 seconds and turn it back on.

2) Press the control key and hold it down.

3) With the CONTROL key held down, press the INTerrupt key twice. The screen presents a display that reads: OFFLINE.

4) Keyin: (OPT). The letters must be upper case and the parentheses are required. The screen display then changes to show the available options.

5) Set the configuration as desired. Key in each option and press <ENTER> to fix it in memory. Press <ENTER> to bypass an option .

The 8200 User's Guide, Document No. 50457, contains complete information on the configuration mode.

Note: On Version 2 8200's the terminal does not have to be

powered off in order to enter the configuration mode.

7.3.2 3600 Terminal

The baud rate for the 3600 terminal cannot be set from the keyboard. The port speed is hardware-strapped within the circuitry of the terminal by the Datapoint Customer Service Engineer during initial installation.

7.4 Non-8200/3600 Terminal Devices

Terminals other than the Datapoint 8200/3600 may be connected to the DATASHARE system, although performance is not optimum without Datapoint's cursor-positioning feature which enchances the speed of form displays.

Non-CRT terminals such as the Teletype 33 and 35 KSR or ASR may also be connected with either hard-wired or modem connections. In addition, conventional CRT terminals such as the Datapoint 3300 (for 300 or 1200 BAUD) or Datapoint 3000 (for 300 BAUD only) may be connected. All Datapoint 3000s and 3300s terminals use identical cable configurations for a given type of installation. The cable for a given device must have both Carrier- and Ring-Detect on the FPCA/COMM/MFC connected to a wire that is logically "high" when the connection is to be established and is logically "low" when the connection is to be broken.

CHAPTER 8. PROGRAMMING CONSIDERATIONS

8.1 AUTO-START

The DS1500 interpreter can be forced to automatically execute when the DOS is brought up by the use of the AUTOKEY/CMD program. If AUTOKEY is set to <interpreter-name>/CMD and the AUTOKEY/CMD program is AUTOed, the DS1500 interpreter will automatically start up whenever the DOS is booted. The AUTO program should not be set to execute <interpreter-name>/CMD directly as the interpreter will scan the command line (MCR$) to determine if any options were entered by the user. MCR$ is untouched by the AUTO program and might contain invalid information leftover from another operation.

8.2 ROLLOUT and SHUTDOWN

A ROLLOUT instruction may be executed by any port at any time. A ROLLOUT instruction will not be performed if ROLLOUT was not configured, or if the two roll files do not exist or have been tampered with. A SHUTDOWN instruction will only be honored if all ports, other than the one executing the SHUTDOWN, are idling in their ANSWER or MASTER programs. If the ROLLOUT or SHUTDOWN function cannot be performed, control is returned to the DATABUS program with the OVER flag set. A ROLLOUT or SHUTDOWN request is not queued. A DATABUS program denied a ROLLOUT or SHUTDOWN request must execute another ROLLOUT or SHUTDOWN statement at a later time if DOS activation is to occur.

When utilizing a 9320 internal FPCA, the power to the FPCA MUST remain on while the interpreter is rolled-out. If the power to the FPCA is shut off while the interpreter is rolled-out, the rollback will NOT perform correctly.

8.3 ROLLOUT or SHUTDOWN and CHAIN

It is possible to run DS1500 from the CHAIN utility. To return from DS1500 to the CHAINing process, a ROLLOUT or SHUTDOWN must be performed with the following format:

```
ROLLOUT/SHUTDOWN "CHAIN/OV1"
```

*NOTE: Nested CHAINing (recursive calls to CHAIN) will not work properly after a ROLLOUT if the command:

```
ROLLOUT/SHUTDOWN "CHAIN/OV1"
```

is issued.

8.4 TRAP

If an event occurs and the trap is not set, an error message is displayed. The last 11 characters of the error message are placed in the first 14 bytes (FP, LP, ETX) of the User Data Area as a string (LP = 11, FP = 1). The program then chains back to the MASTER program.

8.5 OPEN

Execution of the OPEN instruction for an IFILE will result in the following search pattern for the index and text file. The index file is opened first using the drive direction given with the file name. If no drive direction is given, an all drive search will be used. Once the index file has been opened, DATASHARE will search for the text file. The search will begin on the drive containing the index file. If the text file cannot be found on the same drive as the index file, an all drive search will be used.

8.6 Shared Files

When running in a ARC environment the /ISI sectors are enqueued during all /ISI I/O if the user configured enqueues on /ISI sectors at configuration time. This does not protect the text file. If the user wants to protect the text file the programmer must issue a FILEPI against that file. ISAM READs that have non-null keys and ISAM READKS statements should not be executed under a FILEPI, because of the number of disk accesses they may generate. Rather if an ISAM record is to be updated, re-READ the record previously read by issuing an ISAM READ with all null keys and then UPDATE under a FILEPI.

8.6.1 CLOSE

Execution of the CLOSE instruction will cause the logical file (FILE/IFILE) to be closed. This causes all pages remaining in the virtual storage buffers which belong to the closed file, except /ISI buffers, to be written to disk. The /ISI buffers are written to disk at CLOSE time if the purge /ISI option is set at configuration time.

In the past, it has been essential that shared files never be CLOSEd, due to the possibility of invalid space deallocation. DS1500 distinguishes between shared and non-shared files based on FILEPI usage in an ARC environment. If the file specified in the CLOSE instruction is mentioned in a FILEPI statement executed anywhere in a program, it is assumed to be shared; if it is not it is non-shared. Thus if the file was specified in a FILEPI instruction while the file was open, DS1500 assumes that the file is a shared file and will not invoke the DOS system CLOSE function for that file. Any space allocated to the file that is not used will not be deallocated. In this manner shared files may be CLOSEd resulting in the purging of all buffers from the DS1500 virtual storage manager without the possibility of losing some of the information written to the file. Unless the DATABUS programmer is using the FILEPI instruction to tell DS1500 that the file is shared, he should not attempt to re-use a logical file entry in a shared file environment since re-using the logical file entry is equivalent to executing a CLOSE instruction for the first file prior to opening the second file. The following pair of examples will result in identical action being taken by DS1500:

```
FILE1       FILE
FNAME1      INIT   "FILE1/TXT"
FNAME2      INIT   "FILE2/TXT"
.
            OPEN    FILE1,FNAME1
            ...
            OPEN    FILE1,FNAME2


FILE1       FILE
FNAME1      INIT "FILE1/TXT"
FNAME2      INIT "FILE2/TXT"
.
            OPEN FILE1,FNAME1
            ...
            CLOSE FILE1
            OPEN FILE1,FNAME2
```

8.7 Code Page Purging Under ARC

In an ARC environment, DS1500 will purge and re-read all object code pages from disk when the program is CHAINed only if the DATABUS object code file is free-standing and not write protected. If the program object file is in a library, or is write protected, the buffers will not be purged. No buffer purging is performed when running under DOS since the DATABUS program can not be changed while the interpreter is running. When running under DOS (stand-alone), DS1500 will purge the V/S buffers of code pages if the program is free-standing and on diskette.

8.8 PI and FILEPI

Under DOS, both the PI and FILEPI instructions are supported identically. That is, a FILEPI is treated as if it were a PI. Although both the PI and FILEPI instructions are supported by 1500 DATASHARE, it is recommended that the FILEPI instruction be used. Use of the FILEPI instruction allows the DATABUS programmer to CLOSE shared files when they are no longer needed (See section 8.6). The PI instruction, since it does not specify the files involved, does not support CLOSEing shared files.

The use of the PI instruction under ARC will result in decreased system performance since a general all-drive enqueue of all MOUNTed volumes from the executing AP must be requested for each PI issued. The FILEPI instruction avoids this decreased performance by requesting an enqueue only for the files specified

in the file list.

If an OPEN/PREP sequence is to be enqueued, the PI instruction MUST be used since the FILEPI instruction requires all files in the file list to be OPEN.

```
FILE     FILE
FNAME    INIT          "FILE/TXT"
.
         TRAP          NOFILE IF IO
         PI            2
         OPEN          FILE,FNAME
         PI            0
         ...
.
NOFILE   PREP          FILE,FNAME
         RETURN
```

The effect of a PI or FILEPI instruction will be cancelled by the execution of any of the following instructions: PI 0, KEYIN, DISPLAY, BEEP, PRINT, CONSOLE, PAUSE, COMWAIT, CHAIN, STOP, SHUTDOWN, or ROLLOUT.

8.9 Null ISAM files

If it is desired to build a null ISAM file for the purpose of creating the data file through DS1500, the index file can be created using the INDEX utility (version 2.6.1 or later) or the ISAM PREP feature in DS1500 (version 1.2 or later). If an older version of INDEX is used to create a null ISAM file, DS1500 will flag the error by posting an I * W error when the file is initially OPENed.

The ISAM PREP feature of DS1500 enables the user to PREP an /ISI file without performing a ROLLOUT/SHUTDOWN. This feature is simply an extension of the former PREPARE verb. If the /ISI and /TXT files did not formerly exist, a PREP of these files will function the same as the PREP of a file to be used for PHYSICAL record accessing. A NULL /ISI file is created with the correct level keys, and its associated NULL /TXT file is created. If the /ISI and /TXT files previously existed, the ISAM PREP will write on top of the existing /ISI and /TXT files. The former /ISI and /TXT files will be changed into NULL /ISI and /TXT files and their previous contents will be destroyed. (See DBCMP15 3.3 User's Guide, Model Code # 50302).

8.10 RELEASE

Execution of the RELEASE instruction will set the OVER condition flag if the system printer is unavailable for use by the port executing the RELEASE, or is off-line. The use of RELEASE does not guarantee that the printer may then be allocated to the port executing the instruction. In the interval between the RELEASE instruction and the first PRINT instruction, the printer may have been allocated to another port or to the Concurrent partition.

8.11 DEBUG

The DEBUG instruction is ignored. That is, execution is continued with no "DEBUG" activity taken.

8.12 CLOCK

The VERSION extension of the CLOCK instruction provides for the retrieval of up to a seven character Interpreter name, although the DS1500 Interpreter may be given up to an eight character name. Thus, users who wish to use the VERSION extension should ensure that any eight character Interpreter name can be uniquely identified by the first seven characters of the name.

8.13 ACALL - Assembler "CALL" Facility

DATASHARE supports the ACALL (assembler language CALL) facility. This facility allows an assembly language overlay to be invoked from a DATABUS program. The implementation structure is that of a user-written overlay which is loaded upon the demand of the DATABUS program (i.e. "dynamic" ACALL). The ACALL overlay uses an extended memory sector for its execution. The ACALL overlay is opened and checked for validity and only then is the extended memory mapped in and the ACALL is loaded. DS1500 will only support the dynamic ACALL feature and if only one ACALL overlay is used, it is only loaded once and is simply "mapped" into logical space as needed. The ACALL is restricted to fit into 4K of RAM or the ACALL overlay will not be loaded. If the ACALL facility is not enabled, all ACALL instructions will be trapped to a bad-opcode ("B") error. ACALL is only supported in a 1550/1560 with 96K or more of memory.

It is important to note that there is no assembler supported by DATAPOINT on the 1500 series processors.

8.14 Multilink - For External Communications

DS1500 supports the Multilink facility. This facility enables a DATABUS program to interface with an external communications device through the external SEND and RECV queues of DS1500. Multilink is run as an independent task with the ability to start multiple tasks of its own. In a 64K environment DS1500 can have a maximum of 1 port configured when Multilink is configured. In a 96K environment, DS1500 can have a maximum of 4 ports. Multilink and Concurrency will NOT execute together because they both use the region from 0150000-0167777 (8K of memory).

8.15 PHANTOM Ports

A PHANTOM port can execute a program without a terminal being attached. NO I/O can be sent to that port. Execution of a KEYIN, DISPLAY, or BEEP instruction from a PHANTOM port, will result in an untrapable error. If the port was in ANSWER or MASTER, the port is permanently disabled and, if the system console is configured, a message is displayed on the console screen indicating that the port has been disabled. If the port was not in ANSWER or MASTER, the port will abort and chain back to the MASTER program.

CHAPTER 9. DATABUS LIBRARY FACILITY OF DS1500

Using the Datapoint library utility program LIBSYS15, one may create library files of DATABUS object code (/DBC programs). Such libraries are accessed and utilized by DS1500 in much the same manner as DOS uses the UTILITY/SYS file. Proper use of DATABUS library programs will result in greater system integrity, more file names available on system disks, and easier backup.

9.1 The System DATABUS Library

The <system DATABUS library> file must be a LIBSYS15-created grouping of DATABUS object code programs. It is searched by DS1500 during CHAIN operations for DATABUS object programs. During CHAINing operations, DS1500 will search any on-line drive to find a free-standing DATABUS program name which matches the program specification operand given in the CHAIN instruction. If this search is unsuccessful, DATASHARE will then search the <system DATABUS library>. Failure to then locate the program in the library will result in a CFAIL condition.

9.2 The CHAIN program specification

DS1500 has the ability to accept both drive numbers and volume names as components of a file specification; the CHAIN facility has now been extended to allow specification of member names within a library. The new program specification syntax of CHAIN is:

<program name>/<extension>:<drive # or VOLID>.<library member name>

No intervening blanks are allowed in the character string or literal used as a program name, and the member name must consist of eight characters or less.

If a <library member name> is used in a program specification, the <program name> is presumed to be a DATABUS program library file. Failure to locate either the library or the proper member within the library will result in a CFAIL; if the <program name> is not a DATABUS library file a CFAIL will also result. If a <member name> alone is specified, a search of the <system DATABUS library> will be performed; no free-standing program search will occur. The extension of the DATABUS program

library file must be specified.

9.3 Examples of CHAIN specifications

MYPROG

This specification would cause DS1500 to attempt to find the file MYPROG/DBC on any drives on-line. Failure to locate the file would cause a search of the <system DATABUS library> file for a member with the name MYPROG. Note that the extension could have been specified if the file had an extension of other than "/DBC".

.MYPROG

This specification would cause DS1500 to attempt to locate the member MYPROG in the <system DATABUS library>. No attempt to find any free-standing file would be made; absence of a <system DATABUS library> would cause a CHAIN failure.

SYSLIB/DBL.MYPROG

This specification would cause DS1500 to locate the file SYSLIB/DBL and interrogate the file for the member MYPROG.

SYSLIB/DBL:DAILY.JOBA

This specification would cause DS1500 to locate the file SYSLIB/DBL on any on-line drive with a volume name of DAILY. The member JOBA would then be found and executed if present.

9.4 Possible Uses of DATABUS Libraries

The programs most frequently used by a DATASHARE system are the ANSWER and MASTER programs for the various ports in use. All ANSWER and MASTER programs may now be kept resident in a <system DATABUS library> instead of existing free-standing. This will allow more DOS directory names to remain available to the user as only the <system DATABUS library> name will be entered in the directory.

In typical business environments, most application programs belong to a certain class of processing, such as payroll or accounts receivable. Using DATABUS libraries, the organization, testing, and everyday use of specific-class programs may be

greatly simplified. For example, a typical office might create the following libraries:

```
PAYROLL/DBL   containing all payroll programs
ACCTSRCV/DBL  containing all accounts receivable programs
ACCTSPAY/DBL  containing all accounts payable programs
TEST/DBL      containing new programs in the testing phase
```

CHAPTER 10. 1500 DATASHARE UNDER ARC

10.1 Minimum Considerations

Only one modification to DATABUS programs should be necessary to run them under DS1500 in an ARC network: The DATABUS "PI" verb must be changed to the new "FILEPI" verb. This modification will allow programs running under DS1500 to execute with or without ARC. The change into the FILEPI verb requires that a programmer inform DS1500 which files must be locked out during a particular sequence of instructions. The only additional consideration is that the specified files must have been OPENed prior to the FILEPI instruction. (See section 8.8)

10.2 Use of Volume Identification

The standard DATABUS file specification used by OPEN and CHAIN allow the use of volume names (VOLID's). This is useful under ARC systems because ARC is a volume-oriented network. The logical drive numbers (assigned at execution time) may vary from user to user, especially if the same DATABUS program is used by more than one applications processor, but the volume names typically do not change. The use of volume names instead of hard-coded drive numbers therefore permits greater DATABUS program flexibility under ARC.

10.3 Write-Protected Data Files

Access time to data files, which are not modified, can be decreased by write-protecting the file. This enables DS1500 to re-use buffer pages from the file which are found in memory.

10.4 ROLLOUT under ARC

All remote volumes are controlled through sub-directories (specified at MOUNT15 time) in an ARC system. The use of sub-directories allows many users to share access to a single logical volume, but certain precautions must be taken in order to prevent contention problems.

When DS1500 is configured, certain files are created to allow the system to rollout (ROLLDS15/SYS and ROLLDS15/UDA). Under ARC these files will be created in the sub-directory currently active on the disk containing the DS1500 object library. If, however, the ROLLOUT file already exists in the SYSTEM sub-directory, it may accidentally be overwritten by another DS1500 system performing a ROLLOUT. Therefore, if two applications processors roll out and the roll files are in the SYSTEM sub-directory, then the second ROLLOUT will overwrite the first and cause the first system to roll in improperly. This will also occur if two applications processors "sign on" into the same sub-directory. In general, if one applications processor should ROLLOUT into the same file as another, erroneous results should be expected.

Great care must be taken to ensure that the location and number of all logical volumes associated with an applications processor prior to a ROLLOUT be identical with their configuration at roll-back. Failure to guarantee this will probably result in severely damaged files on disk.

Conversely, if two applications processors are identical in type and memory size and have the same logical volumes MOUNTed in the same positions and are "signed on" into the same sub-directories on all disks, DS1500 may be rolled out on one processor and rolled back into another. This capability may possibly be useful but should be used with extreme caution.

When running under ARC, remote volumes that have been MOUNTed remain mounted until they are removed via the MOUNT15 command or by restarting the system. Terminating DS1500 in the normal manner is accomplished by doing a ROLLOUT, if the state of DS1500 is to be saved or by doing a SHUTDOWN which will not save the state of the system.

APPENDIX A. INSTRUCTION SUMMARY

SYNTACTIC DEFINITIONS

<aclist>	Any combination of numeric or character string variables, FILEs, IFILEs, or COMLSTs separated by commas. The list may be continued on more than one line by placing a colon (:) after the last operand on the line to be continued.
<blist>	The name assigned to the first of a set of physically contiguous numeric string or character string variables.
<brlist>	A list of execution labels separated by commas. The list may be continued on more than one line by placing a colon (:) after the last label on the line to be continued.
<char>	Any single character of the form "<string>" where string is of length one (1).
<cmlist>	A name assigned to a statement defining a COMLST data declaration.
<dlist>	Any combination of <slit> and <occ> separated by commas. The list may be continued on more than one line by placing a colon (:) after the last variable on the line to be continued.
<dnum>	A decimal number between 0 and 255.
<dnum1>	A decimal number indicating the number of digits that should precede the decimal point.
<dnum2>	A decimal number indicating the number of digits that should follow the decimal point.

<dnum3>	A decimal number between 1 and 20 inclusive.
<dnum4>	A decimal number between 1 and 64 inclusive.
<dnum5>	A decimal number between 0 and 20 inclusive.
<dnum6>	A decimal number between -128 and 127 inclusive.
<dnum7>	A decimal number between 1 and 255 inclusive.
<dnvar>	A name assigned to a statement defining a destination numeric string variable. This variable is generally changed as a result of the instruction.
<DOS file spec>	A DOS compatible file specification (see DOS user's guide).
<dsvar>	A name assigned to a statement defining a destination character string variable. This variable is generally changed as a result of the instruction.
<equ>	A name assigned to an EQUATE statement.
<event>	The occurrence of a program trap: PARITY, RANGE, FORMAT, CFAIL, IO, INTERRUPT, INT, F1, F2, F3, F4, F5.
<event1>	The occurrence of one of the following program traps: PARITY, RANGE, FORMAT, CFAIL, IO.
<file>	A name assigned to a FILE declaration.
<file list>	A list of one or more FILE or IFILE names separated by commas.

<flag>	One of the following flags: OVER, LESS, ZERO, or EOS (EQUAL and ZERO are two names for the same flag). These flags are used to indicate the result of certain DATABUS operations.
<fflag>	One of the following flags: F1, F2, F3, F4, or F5. These flags are used to indicate the status of the console's function keys, (if the function key feature is available on the processor), and are used with the GOTO instruction.
<ifile>	A name assigned to an IFILE declaration.
<index>	A numeric variable used in connection with list accessing.
<key>	A non-null string variable used as a key to indexed I/O accesses.
<label>	A letter, followed by any combination of up to seven (7) additional letters and digits.
<list>	Any combination of <slit>, <occ>, <list controls>, <nvar> and <svar> separated by commas. The list may be continued on more than one line by placing a colon (:) after the last variable on the line to be continued.
<nlist>	A list of numeric variables each pair of which is separated by a comma (,). The list may be continued on more than one line by placing a colon (:) after the last variable on the line to be continued.
<nlit>	A literal of the form "<string>" where string is a valid numeric string.
<nslist>	Any combination of numeric and character string variables separated by commas. The list may be continued on more than one line by placing a colon (:) after the last variable on the line to be continued.

<null>	A null string variable used as a key to an indexed read.
<nvar>	A name assigned to a statement defining a numeric string variable.
<occ>	An octal control character (000 to 0377 inclusive).
<occ1>	An octal control character between 0 and 0177 inclusive.
<pdnum>	A positive decimal number between 0 and 127 inclusive.
<pdnum1>	A positive decimal number between 1 and 127 inclusive.
<prep>	A comma (,) or a valid preposition: BY, FROM, IN, INTO, OF, TO, USING, and WITH. (A preposition is allowed for source code readability only, but any preposition may be used even if it does not make sense in English in the context of the particular verb.)
<rn>	A numeric variable which contains a positive record number (greater than or equal to zero) used to randomly READ or WRITE a file.
<route>	A character string variable used for routing.
<seq>	A numeric variable which contains a negative number (less than zero) used to READ or WRITE a file sequentially.
<skey>	A numeric or character string variable used with SEARCH.
<slist>	A list of character string variables, each pair of which is separated by a comma (,). The list may be continued on more than one line by placing a colon (:) after the last variable on the line to be continued.

<slit>	A literal of the form "<string>".
<snvar>	A name assigned to a statement defining a source numeric string variable. This variable is unchanged as a result of the instruction.
<ssvar>	A name assigned to a statement defining a source character string variable. This variable is unchanged as a result of the instruction.
<string>	Any sequence of characters with the exception of the forcing character (#) which itself will not become part of the string literal's character sequence. The character following the # will become part of the character sequence (e.g. another # or ").
<svar>	A name assigned to a statement defining a character string variable.

FOR THE FOLLOWING SUMMARY:

Items enclosed in brackets [ ] are optional.

Items separated by the ¦ symbol are mutually exclusive (one or the other but not both must be used).

COMPILER DIRECTIVES

```
<label>  EQU       <dnum¦occ>
<label>  EQUATE    <dnum¦occ>
<label>  IFC       <equ¦dnum¦occ>
<label>  IFEQ      <equ¦dnum¦occ>,<equ¦dnum¦occ>
<label>  IFGE      <equ¦dnum¦occ>,<equ¦dnum¦occ>
<label>  IFGT      <equ¦dnum¦occ>,<equ¦dnum¦occ>
<label>  IFLE      <equ¦dnum¦occ>,<equ¦dnum¦occ>
<label>  IFLT      <equ¦dnum¦occ>,<equ¦dnum¦occ>
<label>  IFNE      <equ¦dnum¦occ>,<equ¦dnum¦occ>
<label>  IFNG      <equ¦dnum¦occ>,<equ¦dnum¦occ>
<label>  IFNL      <equ¦dnum¦occ>,<equ¦dnum¦occ>
<label>  IFNZ      <equ¦dnum¦occ>
<label>  IFS       <equ¦dnum¦occ>
<label>  IFZ       <equ¦dnum¦occ>
<label>  INC       <DOS file spec>
<label>  INCLUDE   <DOS file spec>
<label>  LISTOFF
<label>  LISTON
```

FILE DECLARATIONS

```
<label>  FILE
<label>  IFILE
```

DATA DEFINITIONS

```
<label>  FORM      <dnum1>.<dnum2>
<label>  FORM      <dnum1>.
<label>  FORM      .<dnum2>
<label>  FORM      <dnum1>
<label>  FORM      <nlit>
<label>  DIM       <pdnum1>
<label>  INIT      <slit>
<label>  INIT      <dlist>
<label>  FORM      *<dnum1>.<dnum2>
<label>  FORM      *<dnum1>.
<label>  FORM      *.<dnum2>
<label>  FORM      *<dnum1>
<label>  FORM      *<nlit>
```

```
<label>  DIM       *<pdnum1>
<label>  INIT      *<slit>
<label>  INIT      *<dlist>
<label>  COMLST    <dnum4>

CONTROL

         BRANCH    <index><prep><brlist>
         CALL      <label>
         CALL      <label> IF <flag>
         CALL      <label> IF NOT <flag>
         CHAIN     <svar|slit>
         FILEPI    <dnum3>;<file list>
         GOTO      <label>
         GOTO      <label> IF <flag>
         GOTO      <label> IF NOT <flag>
         GOTO      <label> IF <fflag>
         GOTO      <label> IF NOT <fflag>
         NORETURN
         PAUSE     <nvar|nlit>
         PI        <dnum5>
         RETURN
         RETURN    IF <flag>
         RETURN    IF NOT <flag>
         ROLLOUT   <svar|slit>
         SHUTDOWN  <svar|slit>
         STOP
         STOP      IF <flag>
         STOP      IF NOT <flag>
         TABPAGE
         TRAP      <label> IF <event>
         TRAP      <label> GIVING <svar> IF <event1>
         TRAP      <label> NORESET IF <event>
         TRAP      <label> GIVING <svar> NORESET IF <event1>
         TRAPCLR   <event>
```

ARITHMETIC

ADD	<snvar|nlit><prep><dnvar>
CHECK10	<svar><prep><svar|slit>
CHECK11	<svar><prep><svar|slit>
CK10	<svar><prep><svar|slit>
CK11	<svar><prep><svar|slit>
COMPARE	<nvar|nlit><prep><nvar>
DIV	<snvar|nlit><prep><dnvar>
DIVIDE	<snvar|nlit><prep><dnvar>
LOAD	<dnvar><prep><index><prep><nlist>
MOVE	<snvar|nlit><prep><dnvar>
MULT	<snvar|nlit><prep><dnvar>
MULTIPLY	<snvar|nlit><prep><dnvar>
STORE	<snvar|nlit><prep><index><prep><nlist>
SUB	<snvar|nlit><prep><dnvar>
SUBTRACT	<snvar|nlit><prep><dnvar>

CHARACTER STRING HANDLING

APPEND	<ssvar|slit|snvar><prep><dsvar>
BUMP	<dsvar>
BUMP	<dsvar><prep><dnum6|snvar>
CLEAR	<dsvar>
CLOCK	TIME<prep><dsvar>
CLOCK	DAY<prep><dsvar>
CLOCK	YEAR<prep><dsvar>
CLOCK	VERSION<prep><dsvar>
CLOCK	PORT<prep><dsvar>
CMATCH	<ssvar|char|occ1><prep><dsvar>
CMATCH	<ssvar><prep><char|occ1>
CMOVE	<ssvar|char|occ1><prep><dsvar>
EDIT	<ssvar|snvar><prep><dsvar>
ENDSET	<dsvar>
EXTEND	<dsvar>
LENSET	<dsvar>
LOAD	<dsvar><prep><index><prep><slist>
MATCH	<svar|slit><prep><svar>
MOVE	<ssvar|snvar|slit|nlit><prep><dsvar>
MOVE	<ssvar|snvar|nlit><prep><dnvar>
MOVEFPTR	<ssvar><prep><dnvar>
MOVELPTR	<ssvar><prep><dnvar>
NFORMAT	<dnvar><prep><dnum1|nvar><prep><dnum2|nvar>
PACK	<dsvar><prep><slist|nlist>
REP	<ssvar|slit><prep><dsvar>
REPLACE	<ssvar|slit><prep><dsvar>
RESET	<dsvar>
RESET	<dsvar><prep><char|pdnum|snvar|ssvar>
SCAN	<ssvar|slit|occ1><prep><dsvar>
SEARCH	<skey><prep><dlist><prep><nvar><prep><dnvar>
SETLPTR	<dsvar>
SETLPTR	<dsvar><prep><char|pdnum1|snvar|ssvar>
SFORMAT	<dsvar><prep><pdnum1|snvar>
STORE	<ssvar|slit><prep><index><prep><slist>
TYPE	<svar>
UNPACK	<ssvar|snvar><prep><slist>

INPUT/OUTPUT

```
BEEP
CLOSE      <file|ifile>
COMCLR     <cmlist>
COMTST     <cmlist>
COMWAIT
CONSOLE    <list>[;]
DELETE     <ifile>,<key>
DELETEK    <ifile>,<key>
DISPLAY    <list>[;]
FPOSIT     <file|ifile>,<dnvar>,<dnvar>
INSERT     <ifile>,<key>
KEYIN      <list>[;]
OPEN       <file|ifile>,<svar|slit>
PREP       <file|ifile>,<svar|slit>
PREPARE    <file|ifile>,<svar|slit>
PRINT      <list>[;]
READ       <file>,<rn|seq>;<;|<list>[;]>
READ       <ifile>,<rn|seq|key|null>;<;|<list>[;]>
READKS     <ifile>;<;|<list>[;]>
RECV       <cmlist>,<route>;<slist>
RELEASE
SEND       <cmlist>,<route>;<nslist>
SPLCLOSE
SPLOPEN    <svar|slit>[,<svar>|<slit>]
UPDATE     <ifile>;<;|<list>[;]>
WEOF       <file|ifile>,<rn|seq>
WRITAB     <file>,<rn|seq>;<;|<list>[;]>
WRITE      <file>,<rn|seq>;<;|<list>[;]>
WRITE      <ifile>,<rn|seq|key>;<;|<list>[;]>
```

APPENDIX B. INPUT/OUTPUT LIST CONTROLS

In the table below, the following abbreviations are used in the USED IN column to indicate which DATABUS instructions the list controls can be used in: C=CONSOLE, D=DISPLAY, K=KEYIN, P=PRINT, R=READ, W=WRITE.

CONTROL	USED IN	FUNCTION
;	KDP	Suppress a new line function when occurring at the end of a list.
;	R	Suppress scanning for logical end of record.
;	W	Suppress writing logical end of record.
*▮	KDP	Turn on Keyin Continuous for KEYIN or suppression of space insertion after the logical length of a variable for DISPLAY, and PRINT.
*▮	W	Turn on space compression during WRITE.
*-	KDP	Turn off Keyin Continuous or allow insertion of spaces into a variable after its logical length for DISPLAY and PRINT.
*-	W	Turn off space compression during WRITE.
*<n>	P	Causes a horizontal tab on the printer to the column indicated by the number <n>.
*<n>	RW	Tab specification for READ, WRITAB, or UPDATE operations.
*<nvar>	P	Causes a horizontal tab on the printer to the column indicated by the value of <nvar>.
*<nvar>	RW	The logical file pointers are moved to that character position relative to the current record.

*B	KD	Emit an audible BEEP at the terminal.
*C	KDP	Causes a carriage return to be generated.
*CL	K	Clear the port's key-ahead buffer.
*DE	K	Restrict string input to digits (0-9) only.
*DION	KDP	Turns on the double-intensity video mode. This list control is not supported on the console.
*DV	K	Display a variable's value during KEYIN without performing a KEYIN operation on it.
*EF	KDC	Causes the screen to be erased from the current cursor position to the bottom of the display.
*EL	KDC	Causes the line to be erased from the current cursor position.
*EOFF	K	Prevents character echo to the display during keyboard input operations.
*EON	K	Causes character echo to the display during keyboard input operations.
*ES	KDC	Causes the cursor to be positioned at horizontal position 1 of the top row of the display and the entire display to be erased.
*F	P	Causes the printer to be positioned to the top of form.
*HOFF	KD	Turn off highlighting mode (display characters normally).
*HON	KD	Turn on highlighting mode (display inverted image of all characters displayed).

*IN	KD	Clear Text-inversion mode.
*IT	KD	Set Text-inversion mode.
*JL	K	Left justify numeric variable and zero fill at right if there is no decimal point. Left justify string variable and blank-fill (or zero-fill if *ZF option is given) to end of string.
*JR	K	Right justify string variable and blank (or zero if *ZF option is given) fill at left.
*L	KDP	Causes a line feed to be generated.
*LL	W	Instructs the interpreter to write the logical length of the variable, the characters between the formpointer and the logical length pointer.
*MP	W	Convert data in a numeric variable to minus overpunch format on disk.
*N	KDP	Causes the cursor or printer to be positioned in column 1 of the next line.
*P<h>:<v>	KD	Causes the cursor to be positioned horizontally and vertically to the column and line indicated by the numbers <h> (horizontal 1-80) and <v> (vertical 1-24). These numbers may either be literals or numeric variables.
*P<h>:<v>	C	Causes the cursor to be positioned horizontally to the column indicated by <h> inside the area on the console reserved for terminal to operator communications (the <v> vertical position of the list control is ignored).
*PL	W	This list control instructs the interpreter to revert back to normal mode of disk I/O (clears *LL list control).

*PON	KD	Send a "printer on" character to a terminal.
*POFF	KD	Send a "printer off" character to a terminal.
*R	KD	Roll up the screen one line.
*RD	KD	Roll down the screen one line. This list control is not supported on the console.
*RV	K	Retain the variable value if a keyin of ENTER only is received. Also enable the LESS flag to be set if the KEYIN is terminated by a (*T) timeout, the OVER flag if it is terminated by the NEW LINE key or function keys, and the EOS flag if it is terminated by a null entry.
*T	K	Time out after 2 seconds have elapsed between successively entered characters for KEYIN statement.
*T<n>	K	Time out after <n> seconds.
*W	KD	Pause for one second.
*W<n>	KD	Pause for <n> seconds.
*ZF	K	Zero fill instead of blank fill string variable.
*ZF	PW	Left zero fill numeric variable.

APPENDIX C. PROGRAM EXAMPLES

The following is an example of a "universal" ANSWER program; in other words, an ANSWER program that is independent of the logical port number on which it is run. This technique of using CLOCK PORT to obtain the port's number may be adopted througnout the DATASHARE environment as opposed to the older method of compiling-in the port's number into an ANSWERxx program.

The ANSWER program below displays the port number on the terminal (the ANSWER's program name will automatically be displayed on the console). The program tnen requests an identification and checks it for validity using a very simple rule: the identification given must be exactly the word DATAPOINT. If the word matches (note the use of both the NOT EQUAL and LESS conditions for checking for an exact match), a STOP statement is executed which causes a CHAIN to the MASTER program. Otherwise, an indication is given that the proper identification was not entered and another request for identification is made.

The MASTER program merely requests the name of a program to be initiated and a CHAIN is executed to the name given. If a CHAIN failure occurs, an indication is given that the name does not exist in the DOS directory and another request for a program name is made. Note that neither the ANSWER or MASTER programs are written using cursor-positioning in the KEYIN or DISPLAY statements to aid in a Teletype terminal compatibility.

Simple ANSWER Program

```
.
. SIMPLE ANSWER PROGRAM
.
PORTN   DIM     2
IDCODE  DIM     9
ID      INIT    "DATAPOINT"
.
        CLOCK   PORT TO PORTN
        DISPLAY *ES,"D A T A S H A R E PORT ",PORTN:
                "ON LINE"
LOOP    KEYIN   "ID: ",IDCODE
        MATCH   ID TO IDCODE
        GOTO    BADID IF NOT EQUAL
        MATCH   IDCODE TO ID
        GOTO    BADID IF LESS
        STOP
BADID   DISPLAY "*** INVALID ID ***"
        GOTO    LOOP
```

Simple MASTER Program

```
.
. SIMPLE MASTER PROGRAM
.
FILNAM  DIM     8
.
LOOP    KEYIN   *N,*EL,"PROGRAM NAME: ",FILNAM
        TRAP    NONAME IF CFAIL
        CHAIN   FILNAM
NONAME  DISPLAY "*** NO SUCH PROGRAM ***"
        NORETURN
        GOTO    LOOP
```

For a more complex example of ANSWER and MASTER programs see the DATABUS Compiler User's Guide.

APPENDIX D. ERROR CODES

If an event occurs and the trap corresponding to that event has not been set, the message:

```
* ERROR * LLLLL X *        or
* ERROR * LLLLL X * Q
```

will appear on the screen display. The first form appears for all untrapped errors except I/O, SPOOL, and CHAIN traps. In the event of one of these traps, a qualification letter is given where a "Q" is shown in the example (explained below). The LLLLL is the current value of the program counter and the X is an error letter. In most cases LLLLL points to the instruction following the one that caused the problem. However, in certain I/O errors, the program counter will point after the list item where the problem occurred. The following error letters can appear:

A - Interrupts already prevented
B - Illegal operation code
C - Chain failure (See Appendix G for explanation)
F - Record format error
I - I/O error (See Appendix E for explanation)
P - Parity failure
R - Record number out of range
S - Spooling I/O error (see Appendix F for explanation)
U - Call stack underflow or overflow

The "A", "B", and "U" errors cannot be trapped. An "A" error will be displayed if a PI or FILEPI instruction is executed and interrupts are already prevented. A "B" error will appear only if an invalid object file is executed or if the system is failing. The "U" error will occur if a programmer executes a RETURN instruction without a corresponding CALL having been previously executed, or if CALLs are nested more than eight levels deep.

The events that may be trapped are shown below (the capitalized name shown is the name used for the event in the TRAP statement):

CFAIL - The specified program was not in the DOS directory, or in the specified DATABUS library, the library specified was not a DATABUS library, loading of a DATABUS program with an oversize user data area was attempted, or a program containing compile-time errors was CHAINed. A CFAIL can also occur whenever a ROLLOUT was attempted and failed.

FORMAT - Data being read into a numeric variable was not all digits and/or decimal point and minus sign, or a decimal point in the input did not agree with a decimal point in a FORM, or data input from disk had a negative multi-punch but no room was available for a minus sign in the FORM, or a WRITE-specified "multi-punch" and the last item of the field was a decimal point. The operation terminates with the item in error in an indeterminate state and the statement is aborted.

IO - Error during I/O statement. (Either a programming error or disk failure may cause this TRAP. See Appendix E for I/O error codes).

PARITY - Disk CRC (hardware) error during READ or disk CRC error during write-verification. (DOS retries the operation several times before indicating this failure.)

RANGE - Record number out of range. (An access was attempted beyond the physical end of the file, a record was read which was never written, or a WRITAB was used on a record which was never written.)

SPOOL - I/O error during write to spool file. (A write to the open spool file resulted in a parity or CRC failure, a disk-space-full trap, or some other abnormal abort of a disk sector write command.) The secondary trap letter (S * <letter>) is explained within the same appendix (Appendix E) as the I * <letter> errors.

The following TRAPs are activated by keyed-in characters. These characters may be entered and used without actually executing a KEYIN instruction in the user's DATABUS program. None of these TRAP conditions will be taken unless the corresponding TRAP is set; e.g. an F1 key trap will not be activated if "TRAP <label> IF F1" is not set and the F1 key is struck.

INT - The INT (interrupt) key sequence was keyed in while the "TRAP <label> IF INT" trap was set.

F1 - The F1 key was keyed in while the "TRAP <label> IF F1" trap was set.

F2 - The F2 key was keyed in while the "TRAP <label> IF F2" trap was set.

F3 - The F3 key was keyed in while the "TRAP <label> IF F3" trap was set.

F4 - The F4 key was keyed in while the "TRAP <label> IF F4" trap was set.

F5 - The F5 key was keyed in while the "TRAP <label> IF F5" trap was set.

<char> - The <character> given in the "TRAP <label> IF <character>" trap was keyed in.

APPENDIX E. INTERPRETER I/O TRAP CODES

All of the following codes apply to I/O trap letters; however, only certain of the codes are used in SPOOLing traps, the others being inapplicable to SPOOL. The errors that can occur during SPOOLing are: B, C, F, M, N and P.

A - An access with a null key or sequentially by key was attempted before any indexed sequential access was made using the logical file.

B - The READ mechanism ran off the end of a sector without encountering a physical end-of-record character (003).

C - An operation on a closed logical file was attempted.

D - A non-READ, non-DELETE indexed sequential operation was attempted where the specified key already existed in the index.

E - An invalid EOF mark was encountered.

F - Insufficient file space available.

G - An attempt was made to SEND an already active COMLST.

H - Invalid routing variable in COMLST or number of variables specified by COMLST exceeded during SEND.

I - The index file specified in an OPEN statement does not exist on the specified drive(s).

J - The index file found by the OPEN statement does not reside in the correct physical location on the disk (index files may never be moved, they must always be re-created by re-indexing).

K - A null key was supplied in an operation where the key may not be null.

M - The data file specified in the OPEN statement does not exist on the specified drive(s), or the specified drive was offline. The I*M error can also occur on a READ/WRITE statement where the drive of the file went off-line.

N - The data file name specified in the OPEN or PREPARE statement was null.

O - The index file name specified in the OPEN/PREP statement was null.

P - The file specified in the PREPARE statement was DOS write protected or an attempt was made to delete a file with delete protection.

Q - A dynamic ACALL overlay either could not be read due to disk errors or would not load into the configured ACALL area.

R - Unexpected EOF mark was encountered (before end of current logical record).

T - The tab value in the READ or WRITAB statement was off the end of the sector.

U - An EOF mark was encountered while a record was being deleted in the indexed sequential file.

W - The root sector of the ISI tree was not an 012 sector. This is generally caused by using an old version of the INDEX utility to create a null ISI file. Re-INDEX the file.

X - The ISI file contains a circular link (is partially destroyed). Re-INDEX the file.

Y - An ISI tree sector could not be read. Usually indicates a partially destroyed ISI tree. Re-INDEX the file.

Z - An illegal INSERT was attempted. Either there was no valid READ or WRITE done to set up the pointers to the text file that INSERT uses, or these pointers indicate a different text file than the one associated with the file specified in the INSERT instruction.

NOTE: WXY errors can be caused by parity errors, the drive being switched off line, or the disk being swapped with another while an operation is taking place.

APPENDIX F. INTERPRETER CHAIN TRAP CODES

The format of the CHAIN trap is:

nnnnnn C * x

where nnnnnn is the address of the point of failure in the user's DATABUS program, and x is one of the following letters:

B - Bad object code file

C - Program contains instructions not configured in the current Interpreter

D - A DOS trap occured during a CHAIN

L - Library not found

M - Member of library not found

N - Name of program was bad or program could not be found

R - ROLLOUT failure

U - User data area (UDA) too large

APPENDIX G. FREEDOM PRINTER CONSIDERATIONS

If the DATASHARE system utilizes a dual-tractor FREEDOM printer and wishes to direct some or all of its PRINT output to the rear tractor, the following method should be used:

All rear-tractor-directed output is prefaced by a user-defined variable with the following format:

```
PREAMBLE INIT      05,"xx"
```

where xx is a two-byte ASCII string whose value is the rear-tractor's left-most character position (margin) expressed in hexadecimal (i.e., for a margin of 90 (decimal) characters, the string "5A" would be used). Since the FREEDOM printer is a 132-column device, the maximum usable value of this string is "83". This variable is used as the first item of any PRINT statement the user desires to be directed to the rear tractor. For example,

```
        PRINT     PREAMBLE,"LINE 1 ON REAR TRACTOR",*N:
                  "LINE 2 ON REAR TRACTOR",*C,"____"
```

produces the following output on the rear tractor:

```
LINE 1 ON REAR TRACTOR
LINE 2 ON REAR TRACTOR
```

In this special variable-prefaced PRINT instruction, the *N and *C controls affect the rear tractor and not the front tractor as normal. Any and all PRINT list controls may be used on the FREEDOM printer rear tractor with the same effect as in a normal PRINT statement, except that tabbing and carriage return are performed relative to the left margin of the rear tractor.

Paper-out sensing is also controlled by the presence or absence of the rear-tractor preamble characters. If the front tractor is out of paper but printing is being directed to the rear tractor, then printing will not be stopped (assuming the rear tractor is supplied with paper) until a front-tractor-directed PRINT statement is issued.

CUT ALONG LINE FOR MAILING

Manual Name _____

Manual Number _____

READER'S COMMENTS

Did you find errors in this manual? If so, specify by page.

_____

_____

_____

_____

_____

Did you find this manual understandable, usable, and well-organized? Please make suggestions for improvement.

_____

_____

_____

_____

_____

_____

Name _____ Date _____

Organization _____

Street _____

City _____ State _____ Zip Code _____

All comments and suggestions become the property of Datapoint.

Fold Here

Fold Here and Staple

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 7225 SAN ANTONIO, TEXAS

POSTAGE WILL BE PAID BY ADDRESSEE:

DATAPOINT CORPORATION
DIRECTOR OF SOFTWARE SUPPORT
8550 Datapoint Drive, Mail Station # N60
San Antonio, Texas 78284