Burroughs Corporation
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

PLANT IDENT | 0 1 4

Product Specification

PAGE 1 OF 1

CONTROL DOC.	REV
None	

DOCUMENT TRANSMITTAL

SYSTEM/PRODUCT B1800/B1700 DISTRIBUTION TYPE

UNIT NAME PROJ. NO.

COORD. CHKR. J Hale RECORDS ISSUE DATE [illegible]

DOC PFX OR DWG SIZE	DOCUMENT NUMBER	REV OR ISSUE	DOCUMENT DESCRIPTION	ENG. RECORDS ONLY				REMARKS
				QTY REV PAGE	COPIES EACH PAGE	BOND	VELLUM	
P.S.	2222 2996	A	SYSTEM FILE INITIALIZER	17	64			

DOCUMENT TRANSMITTAL

Burroughs Corporation

COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

P.S. 2222 2996

SYSTEM FILE INITIALIZER

PRODUCT SPECIFICATION

REV LTR	REVISION ISSUE DATE	APPROVED BY	REVISIONS
A	9/13/79	J Hale	Original issue - MARK 9.0 Release

"THE INFORMATION CONTAINED IN THIS DOCUMENT IS CONFIDENTIAL AND PROPRIETARY TO BURROUGHS CORPORATION AND IS NOT TO BE DISCLOSED TO ANYONE OUTSIDE OF BURROUGHS CORPORATION WITHOUT THE PRIOR WRITTEN RELEASE FROM THE PATENT DIVISION OF BURROUGHS CORPORATION."

TABLE OF CONTENTS

SECTION I

INTRODUCTION TO

SYSTEM FILE INITIALIZER

INTRODUCTION

This specification describes the format and the initialization of the B1800/B1700 implementation of relative files, as defined for ANSI '74 COBOL.

FUNCTION

SYSTEM FILE INITIALIZER (SYSTEM/FILE.INIT), in some instances, allocates the area used for a relative file and initializes that area. This is described in more detail in the paragraphs that follow.

FEATURES

RELATIVE FILES

A relative file consists of records identified by relative record numbers or "KEYs". The file may be thought of as composed of a serial string of areas, each capable of holding a logical record. Each of these areas is addressed by a relative record number. For example, the tenth record is the one addressed by the relative record number ten. It is in the tenth record area, regardless of whether records have been written in the first through the ninth record areas.

These record slots are of fixed length, the value of which is specified by the file attribute MAXRECSIZE. Each record slot may contain one record. A record slot is "empty" if it contains no valid record. Full record slots may be emptied by deleting the record they contain, making the contents inaccessable through the normal mechanism. (A DELETE verb is implemented in the operating system).

The record slots are grouped into blocks of one or more records and, like the record size, the number of records per block may not be changed. Appended to and at the beginning of the block, is a field known as the Block Control Information (BCI). BCI consists of one bit per record slot, plus enough filler bits to make the size of the BCI modulo eight. The bits in the BCI field are used to indicate the presence or absence of a valid record in the associated record slot. A value of one indicates that a record is present in the associated slot. SYSTEM/FILE.INIT allocates the area and initializes all of the necessary presence bits, setting them to a value of zero. As a consequence of this design, the size of a block in a relative file is equal to the number of records per block times the record size plus the BCI.

INITIALIZATION

The execution of this program is one means of accomplishing initialization; however, ultimate responsibility for ensuring that areas are initialized lies with the Master Control Program (MCP).

The manner in which the file is accessed determines the method of initialization the MCP uses. If the access mode of the file is sequential, the MCP allocates the area and the logical I/O routines initialize each block before access. In this case, use of SYSTEM/FILE.INIT is not required. If the access mode is random or dynamic, and if a new disk area is allocated to the file, the MCP automatically executes SYSTEM/FILE.INIT to ensure that the BCI is initialized. If this is the case, the user program which caused the area to be allocated is stopped and SYSTEM/FILE.INIT is executed at the same priority as the requesting program. If a relative file being accessed in the sequential mode is closed with the end-of-file (EOF) pointer not at the end of an area, the MCP causes the execution of SYSTEM/FILE.INIT to initialize the remainder of that area before the CLOSE operation is completed.

If SYSTEM/FILE.INIT is called by the MCP, as described above, the requesting program is not allowed to execute until SYSTEM/FILE.INIT has completed the initialization of the area. The mechanism used for calling the relative initialization program is similar to that used for a program sort. The program attempting the write (which requires the use of SYSTEM/FILE.INIT) is hung on its dummy event with the next queue being the S-communicate queue. Upon successful completion of the initialization program, the writing program's dummy event is "caused" and the program goes to the SMCP. The SMCP updates the original program's FIB (for sequential communicates) before passing the communicate to the MMCP for completion of the write. If an irrecoverable error is detected during initialization, the program is reinstated with standard file status values set in the FIB of the file, upon EOJ or end of task of SYSTEM/FILE.INIT. A list of these values is contained in the MCPII product specification. Any USE procedures declared for the file will be executed at this time.

SECTION II

OPERATING PROCEDURES

OPERATING PROCEDURES

Operational instructions are not required by the program when it is executed automatically by the MCP. Only when initializing the file prior to an actual production run is it necessary to provide certain information to SYSTEM/FILE.INIT. Users may wish to do the initialization prior to the time it is actually required to ensure that the required disk space is available and to minimize the amount of processing that is actually required when a record is added to a new disk area. This latter consideration might be important in an online environment where it is desirable to minimize response time.

USER EXECUTION

Users are responsible for providing file attributes for each file they want initialized. Attributes may come from one of three sources determined by Program Switch 0. These attributes are described in Table 2.1. Keywords should be used for card or operator display terminal (ODT) input. File attributes are set by entering keyword-value pairs of the form <keyword> = <value>. Input is as follows: a TITLE specification followed by its attributes, followed by another TITLE specification and its attributes. Input is basically free-form, the only restriction being that keyword-value pairs may not be split across record boundaries. See Figure 2.1 for a sample execution.

TABLE 2.1 FILE ATTRIBUTES

Attribute	Keyword	Default
File name	TITLE	none
Record size	RECSIZE, RECORD.SIZE, or RSZ	175
Records per block	RECORDS.BLOCK or R.B	1
Blocks per area	BLOCKS.AREA or B.A	100
Number of areas	AREAS or ARE	25
File kind	FILEKIND or FILE.TYPE	none

```
?EX SYSTEM/FILE.INIT;SWO 1;
?DATA CARDS
TITLE=MY/FILE2 RECSIZE=80 BLOCKSIZE=2
TITLE=MY/FILE3 TITLE=MY/FILE4 AREAS=105
TITLE=PACKID/MY/FILE5
RECSIZE=80
BLOCKSIZE=2
AREASIZE=4
AREAS=105
FILEKIND=RELATIVE
?END
```

Figure 2.1 Sample of User Execution

If only one file is to be initialized, a file-equate is prooably the easiest means of executing the program. Figure 2.2 depicts a sample execution. The defaults shown in Table 2.1 apply to any attributes not specified in the FILE cards. SWO is defined under SWITCH SETTINGS.

```
?EX SYSTEM/FILE.INIT; SWO 0;
?FI RELFILE NAME TEST/F1 RECORD.SIZE 80 RECORDS.BLOCK 19
?BLOCKS.AREA 10 FILE.TYPE=RELATIVE; % AREAS DEFAULT TO 25
```

Figure 2.2 Sample Execution for a Single File

MCP INTERNAL USE

When the MCP automatically executes the program, it passes three pieces of information to the initialization program via a file equation to RELFILE. The string that performs the call (see section on intitialization) is:

```
?EX SYSTEM/FILE.INIT SW9 1;FI RELFILE NAM <file name> AREAS <area #>;
```

The three pieces of information are:

- the file name
- the area to be initialized
- the MCP request that only one area is to be initialized (indicated by Program Switch 9 set to a nonzero integer as shown in the string above).

SWITCH SETTINGS

Program Switch 0 determines the source of file attributes.

SW0 0 indicates the source is a file equation to RELFILE.

SW0 1 indicates the source is card input.

SW0 2-15 indicates the source is ODT input.

Program Switch 9, set to any nonzero number, specifies that the program was initiated automatically by the MCP; that only one area is to be initialized and that the file is a relative file. This switch is reserved for use by the MCP and users are prohibited from setting it.

APPENDIX A

ERROR MESSAGES

ERROR MESSAGES

When the program is executed automatically by the MCP, error messages are not displayed; however, three conditions might be returned to the calling program.

1. The file is not present on disk.

2. The file is locked (by some other program).

3. An irrecoverable I/O error occurred on initialization.

These conditions are stored in the file status variable for the appropriate file. The calling program may take action based on these conditions. The first two conditions are unlikely to occur by virtue of the fact that the calling program is either creating the file (in the case of a new file), or accessing an existing file. The third condition is more likely to occur, in which case, the calling program should be prepared to handle the difference or be discontinued.

When the program is executed from the ODT or from the card reader, the following error messages are subject to display.

INVALID <token name> --<value>
The name of the token entered was a valid attribute name but the value associated with it exceeded the MCP's limits on that attribute. The values allowed for the relevant attributes are presented in the Software Operational Guide.

INVALID TOKEN --<name>
The attribute specified by <name> is not recognizable by the program.

EQUAL SIGN EXPECTED
The equal sign between the attribute and its value is mandatory.

VALUE EXPECTED AFTER =
No value was found in the input string after the equal sign in an attribute specification. When the input is from cards, the attribute value must be contained on the same card as the attribute name.

UNEXPECTED TOKEN --<name> --TITLE EXPECTED
The program was looking for a TITLE attribute and encountered a different attribute name.

<file name> NOT INITIALIZED FILE MISSING
<file name> NOT INITIALIZED FILE LOCKED
<file name> NOT INITIALIZED EXCEPTION ON WRITE

The file was not initialized due to the cause displayed. The first two messages cannot occur when the program is executed from the ODT or from the card reader because in these cases, the program always creates a new file; thus, the program cannot be used to initialize one area of an existing file.

APPENDIX B

INTERNAL FILES

INTERNAL FILES

FILE NAME	FUNCTION
RELFILE	Disk file
CARDS	Card file

INDEX