Unisys e-@ction ClearPath Enterprise Servers

File Attributes

Programming Reference Manual

ClearPath MCP Release 7.0 SSP1

March 2002

Printed in USA
8600 0064–508

Unisys e-@ction ClearPath Enterprise Servers

File Attributes
Programming Reference Manual

UNISYS

ClearPath MCP Release 7.0 SSP1

Printed in USA
March 2002
8600 0064–508

NO WARRANTIES OF ANY NATURE ARE EXTENDED BY THIS DOCUMENT. Any product or related information described herein is only furnished pursuant and subject to the terms and conditions of a duly executed agreement to purchase or lease equipment or to license software. The only warranties made by Unisys, if any, with respect to the products described in this document are set forth in such agreement. Unisys cannot accept any financial or other responsibility that may be the result of your use of the information in this document or software material, including direct, special, or consequential damages.

You should be very careful to ensure that the use of this information and/or software material complies with the laws, rules, and regulations of the jurisdictions with respect to which it is used.

The information contained herein is subject to change without notice. Revisions may be issued to advise of such changes and/or additions.

Notice to Government End Users: This is commercial computer software or hardware documentation developed at private expense. Use, reproduction, or disclosure by the Government is subject to the terms of Unisys standard commercial license for the products, and where applicable, the restricted/limited rights provisions of the contract data rights clauses.

Correspondence regarding this publication can be e-mailed to **doc@unisys.com**.

Unisys e-@ction
ClearPath Enterprise
Servers

File Attributes

**Programming
Reference Manual**

**ClearPath MCP
Release 7.0 SSP1**

8600 0064–508

Unisys e-@ction
ClearPath
Enterprise
Servers

File Attributes

**Programming
Reference
Manual**

**ClearPath MCP
Release 7.0
SSP1**

8600 0064–508

Bend here, peel upwards and apply to spine.

Contents

Section 1. **Understanding File Attributes**

Section 2. **General File Attributes A through D**

Section 5. General File Attributes T through Z

Section 6. User-Defined Disk File Attributes (UDDFAs)

Section 7. Direct I/O Buffer Attributes

Section 1
Understanding File Attributes

About This Guide

Purpose

The purpose of this manual is to describe the functionality of each file attribute and direct I/O buffer attribute, and to explain how to define a disk file attribute for your own use.

Audience

This manual is intended for programmers and operations personnel who need to understand the functionality of a given file attribute.

This section explains the fundamentals of file attributes and the notation used in this manual to describe them.

What is a File Attribute?

A file attribute is an element that describes a characteristic of a file and provides information the system needs to handle the file. Examples of file attributes are the file title, record size, number of areas, and date of creation. For disk files, permanent file attribute values are stored in the disk file header. For tape files, the attribute values are stored in the tape label.

A file attribute enables you to interrogate or modify the I/O subsystem information that defines your file. You can access the I/O subsystem file information only through file attributes.

Each file attribute defines a characteristic of the file. Changing the value of a single attribute affects the file defined by that attribute and can cause the values of other attributes to change as well.

You can access the logical file in a program in several ways:

- By modifying or interrogating an attribute
- By opening the file
- By invoking an I/O statement

When you first access a logical file, the following actions occur in the order indicated:

1. The logical file is set up, and all attributes are set to their system default values.
2. The attributes mentioned in the file declaration are set to the values specified in the file declaration. The INTNAME and INTMODE attributes are always implicitly placed in the file declaration by the compilers.
3. If any compile-time file equation exists for the logical file—that is, if the value of the INTNAME attribute matches a name referenced in the file equation—the file-equated attributes are set to the values specified.
4. If any run-time file equation exists for the logical file, which is determined by matching the INTNAME attribute as described in step 3, the file-equated attributes are set to the values specified.
5. The action that first accessed the logical file is performed. For example, if the action was a statement to modify an attribute, the attribute is set.

When a program specifies the value of a file attribute, it is specifying the value for a logical file. Those file attributes associated with a logical file are known as *logical file attributes*. When a physical file is created, some of the logical attributes used to create it must be kept with the file so that the file can be properly used later. Thus, a physical file also has attributes referred to as *physical file attributes*. A logical attribute occasionally changes the physical file attribute value of an existing file.

Logical File States

A logical file can be in one of four basic states:

- Open and assigned
- Open and unassigned
- Closed and assigned
- Closed and unassigned

The process of opening a logical file involves either searching for a device or a place on a device on which to create a new physical file, or searching for an existing physical file with file attributes compatible with those of the logical file. When such a search is completed successfully, the logical file is said to be assigned to the physical file. Data transfer can occur only when the logical file is assigned to a physical file. A port file has no associated physical file. Instead, each subfile is mapped onto a dialogue.

You can use the DUMMYFILE or OPTIONAL attribute to open a logical file without assigning it to a physical file.

Closing a logical file has several purposes. A CLOSE operation is most often used to unassign a logical file from a physical file. It is also used to specify the action that is to be taken on the physical file after the association is broken. For example, a disk file can be discarded from or entered into a directory. When a file is entered into a directory, it is referred to as a *permanent file*. For permanent files, a CLOSE operation also updates the end-of-file (EOF) pointer. For a tape file, a CLOSE operation can request that the tape be rewound or unloaded.

It is possible for a logical file to be closed but still assigned to a physical file. This technique is useful for creating or accessing multiple files on a multifile tape, for example. The ALGOL statements CLOSE(F,*) and CLOSE(F,REWIND) and the COBOL74 statement CLOSE FILEID leave the logical file assigned to the physical file.

When a logical file is assigned to a physical file, the attributes that are not relevant to the physical file are ignored. If the file is opened and any attribute of the logical file is found to be invalid, even if the attribute is correctly set, one of two actions can occur:

- The value is changed by the I/O subsystem and a nonfatal, run-time attribute error is issued.
- If the attribute is an event-valued attribute, a fatal error is issued.

The assignment of a logical file to the physical file can cause some attributes of the logical file to be changed. Changed attributes can be the result of the following factors:

- An operator intervened during file assignment.
- Attribute settings conflicted.
- Logical and physical files were mismapped.
- The DEPENDENTSPECS and DEPENDENTINTMODE attribute values were set to TRUE, or the FILETYPE value was set to 7 or 8.

The values also can be affected by system options and by certain task attributes, such as by the FAMILY task attribute. The values returned by attributes while a physical file is assigned to a logical file are the values in use by the subsystem. For example, whenever a conflict exists between the logical and physical frame sizes used, the values returned by the attributes are the frame size of the logical file.

Changing the value of a logical file attribute can result in a change in the corresponding physical file attribute. For details, refer to the discussions of the various attributes.

UTC Time

To avoid the ambiguity of many different local time zones, international applications specify dates and times either as an absolute time value or relative to an absolute time value, called Coordinated Universal Time (UTC - the initial letters come from the French version of this, Universal Time Coordinated). UTC time was also known as Greenwich Mean Time (GMT).

To facilitate writing international applications, file attributes are available that return date and time values relative to UTC time. These attributes have names that end in "UT". Examples of UTC time file attributes are CREATIONDATEUT and CREATIONTIMEUT.

Disk File Headers

Each physical disk file is defined by a collection of file attributes. Examples of these attributes include FILENAME, FILEKIND, CREATIONDATE, CREATIONTIME, and BLOCKSTRUCTURE.

The values of the attributes for a particular file are stored in a data structure called a disk file header. The disk file header for a permanent file (one that has been entered into the directory via a file close operation specifying the lock variant or other, similar operation), is stored in the system directory file of the disk family on which the file resides.

Disk File Header Versions

Prior to the SSR 42.1 release, the system used disk file headers that were in version 6 format. In the SSR 42.1 release, a new version 7 disk file header format was introduced.

Version 7 disk file headers contain some additional information that is not contained in version 6 headers. This information includes

- Six new date valued attributes:
  - ATTMODIFYDATE
  - BACKUPDATE
  - COPYDESTDATE
  - COPYSOURCEDATE
  - EXECUTEDATE
  - READDATE
- Six new time valued attributes:
  - ATTMODIFYTIME
  - BACKUPTIME
  - COPYDESTTIME
  - COPYSOURCETIME

  - EXECUTETIME
  - READTIME
- Time zone valued attributes for the nine date and time attributes:
  - ACCESSTZ
  - ALTERTZ
  - ATTMODIFYTZ
  - BACKUPTZ
  - COPYDESTTZ
  - COPYSOURCETZ
  - CREATIONTZ
  - EXECUTETZ
  - READTZ
- GROUP security attributes

See Appendix C, "Disk File Headers," in the *I/O Subsystem Programming Guide* for a complete description of version 6 and version 7 disk file headers.

Disk Families

Each disk family has a family header version, which controls the format of the disk file headers written to that family. The MCP currently supports only families with a family header version of 7.

Note: *Under some circumstances, the system operator can use the CONVERTHEADERS system utility to change the family header version of a family. Refer to "Version 7 Disk File Headers" in Section 2 of the* MCP Environment Migration Guide *for more information about using the CONVERTHEADERS system utility.*

Library Maintenance Tapes

The system can create library maintenance tapes that contain either version 6 or version 7 disk file headers. The version of the disk file headers written to a specific tape is version 7 by default. Use SW6 to create version 6 disk file headers.

When a disk file header is written to a library maintenance tape in version 6 format, the values of the new version 7 header attributes are discarded. Consequently, the values returned by these attributes may not be reliable for files that have been copied from library maintenance tapes that have version 6 disk file headers.

Examples of File Attributes

The following example demonstrates the use of file attributes. In this example, the WFL deck compiles and runs the ALGOL program and uses file equation in the process.

The ALGOL program symbolic whose file name on disk is FILE/PROGRAM is as follows:

```
BEGIN
FILE
    F(KIND=DISK,MAXRECSIZE=90,BLOCKSIZE=180,NEWFILE=TRUE,            (1)
      FILENAME="TEST.",FRAMESIZE=8,AREALENGTH=3600);
ARRAY
    A[0:14];

F.KIND := VALUE(DISK);                                               (4)
OPEN(F);
    (The entry of an FA (File Attribute) system command by the
     operator.)                                                      (5)
REPLACE POINTER(A) BY " " FOR 90;
REPLACE POINTER(A) BY "THIS FILE'S FILENAME IS:  ", F.FILENAME;
F.SYNCHRONIZE :=VALUE(OUT);                                          (6)
WRITE(F,90,A);
CLOSE(F,CRUNCH);
END.
```

The WFL job deck is as follows:

```
?BEGIN JOB FILE/EXAMPLE;
 COMPILE OBJECT/FILE/PROGRAM ALGOL LIBRARY;
   COMPILER FILE CARD(KIND=DISK,FILENAME=FILE/PROGRAM);
     FILE F(BLOCKSIZE=180,FILENAME=TEST2);                           (2)
 RUN OBJECT/FILE/PROGRAM;
   FILE F(BLOCKSIZE=2520,NEWFILE=FALSE,FILENAME=TEST1);              (3)
 ?END JOB
```

The following file attribute actions are taken in the preceding example code:

Notation Number	Explanation
(1)	The values assigned in this statement are stored in the file declaration of the code file.
(2)	The values assigned at compilation time are stored in the code file as a compile-time file equation.
(3)	The values assigned when the program is run are stored with the job information.
(4)	Because this statement touches file F for the first time, the file attribute values stored at (1), (2), and (3) are applied to the file, with the values assigned in (2) superseding the values of BLOCKSIZE and FILENAME assigned in (1) and the values of BLOCKSIZE, FILENAME, and NEWFILE assigned in (3) superseding the values assigned in (1) and (2). Finally, the value of DISK is assigned to the KIND file attribute.
(5)	If an operator enters an FA system command because no file is available with the correct name, the value or values of any attributes included in the FA command override any values already assigned to those attributes at that point.
(6)	The SYNCHRONIZE file attribute value is changed to OUT while the program is running.

After the attribute assignment statement *F.KIND:= VALUE(DISK)* is executed, the following attributes have the indicated values:

- The KIND attribute has a value of DISK.
- The MAXRECSIZE attribute has a value of 90 characters.
- The BLOCKSIZE attribute has a value of 2520 when the file is opened.
- The NEWFILE attribute has a value of FALSE.
- The FILENAME attribute has a value of TEST1.
- The FRAMESIZE attribute has a value of 8.

During execution of the OPEN statement, if the permanent physical file with the file name of TEST1 does not exist on disk when the assignment to the physical file is attempted, the program waits for an answer to the NO FILE message. In other words, the program displays the message “NO FILE TEST1” and waits for a response from either an operator or a programmer, or waits for the file with the file name TEST1 to be created on or copied to the appropriate disk device.

If the *<mix number> FA FILENAME = TEST/FILE* system command is entered, all the file attributes listed previously now have the values indicated in the previous list, except that the FILENAME attribute value is now TEST/FILE. When a physical file with a file name of TEST/FILE exists on disk, the OPEN operation proceeds normally, assigning the logical file F to the physical file TEST/FILE on disk, setting up the logical file, and marking the logical file as open.

When a program interrogates a file attribute, certain state-related file attribute values, such as LASTSUBFILE, STATE, and CURRENTRECORDLENGTH, change with every I/O operation performed on the logical file. If the logical file is being accessed by more than one task, there is no guarantee that the value of that file attribute when a given task interrogates the attribute is the value resulting from the I/O operation that the task invoked. Refer to the *I/O Subsystem Programming Guide* for further information about interrogating file attributes.

Attribute Characteristics

The following characteristics are identified for each attribute in this manual:

Characteristic	Meaning
Kinds	Indicates the kind of physical file that can use the attribute.
Interrogate	Indicates whether the value of the attribute can be read and, if so, under what restrictions.
Modify	Indicates whether the value of the attribute can be modified and, if so, under what restrictions.
Type	Indicates the data type of the attribute.
Range	Indicates a list of possible values associated with the attribute. These values vary according to the type of the data for the attribute.
Default	Indicates the value that the system uses when the user does not specify a value for the attribute.
Abstract value	Indicates the values that can be used with the ALGOL *SETABSTRACTVALUE* statement.
Stored Permanently	Indicates whether the value of the attribute is stored with the permanent file. If the value is stored, it can be used again when the file is reopened.
Parameters	Indicates the number of parameters associated with the attribute and whether they are optional or required. When no parameters are required, the value is none.
FTAM File Access	Indicates whether Open Systems Interconnection (OSI) File Transfer, Access, and Management (FTAM) file access supports the attribute.
Heritage Host Services Logical I/O	Indicates whether Heritage Host Services logical I/O supports the attribute.
Port Services	Indicates which of the SERVICE attribute values specific to port files can support the attribute.

The following tables describe the values that are associated with the preceding characteristics:

KIND Value	Meaning
All	Relevant to all KIND values.
CD	Relevant to CD-ROM
Direct I/O disk	Relevant to direct I/O disk files.
Disk (DISK and PACK)	Relevant to a disk file or a POSIX special file.
ODT	Relevant to an operator display terminal (ODT) file.
Port subfile	Relevant to a port subfile.
Port	Relevant for a port file. An attribute that is specified for port files is not a port subfile attribute or vice versa, unless specifically stated. Refer to Appendix B for a list of port file and port subfile attributes.
Printer	Relevant to a printer file.
Reader	Relevant to a card reader file.
Remote	Relevant to a remote file.
Tape	Relevant to a tape file.
Virtual	Relevant to a virtual file.

Note: *Although POSIX special files do not reside on disk, they do have directory entries.*

INTERROGATE Value	Meaning
Never	The attribute can never be interrogated. It is a write-only attribute.
When closed	The file must be closed in order for the attribute to be interrogated.
When open	The file must be open in order for the attribute to be interrogated.
When assigned	The logical file must be assigned to a physical file in order for the attribute to be interrogated.
When unassigned	The logical file must not be assigned to any physical file in order for the attribute to be interrogated.
Anytime	The attribute can be interrogated under any circumstances.

MODIFY Value	Meaning
Never	The attribute can never be modified. It is a read-only attribute.
When closed	The file must be closed in order for the attribute to be modified.
When open	The file must be open in order for the attribute to be modified.

MODIFY Value	Meaning
When assigned	The logical file must be assigned to a physical file in order for the attribute to be modified.
When unassigned	The logical file must not be assigned to any physical file in order for the attribute to be modified.
When not in directory	The attribute can be modified when the file is not entered in the disk directory.
When FILESTATE is...	The attribute can be modified when the value of the FILESTATE attribute is equal to the specified value.
Anytime	The attribute can be modified under any circumstances.

TYPE Value	Meaning
Boolean	The attribute value is a Boolean. A Boolean variable or the result of a Boolean expression is TRUE if the least significant bit is 1 and FALSE if the least significant bit is 0. All other bits are ignored for the purpose of determining truth values.
Event	The attribute is event-valued.
Integer	The attribute value is an integer.
Mnemonic	The attribute value has mnemonics associated with the integer values.
Character string	The attribute value contains a string of characters. In ALGOL, this is a pointer-valued attribute. Some languages, including ALGOL, require that the value be terminated by a period (.). For example, TITLE=”SYSTEM/M ON DISK.” When periods (.) are included as part of a parameter, as in the case of an IP address, additional quotation marks are required. An example of the COBOL syntax is VALUE OF IOHSTRING IS ”””CREDENTIALS=XXX/YYY IP=172.17.19.188””.” ***Note:*** *The value is terminated with a period (.). The whole string including the period is enclosed in another set of quotation marks with two sets inside.*
Real	The attribute is real-valued.
Translate table	The attribute value is an ALGOL translate table.
Word	The attribute contains 48 bits of information as described in the explanation of the attribute.
Word list	The attribute contains one or more 48-bit words of information as described in the explanation of the attribute. In ALGOL, this is a pointer-valued attribute.

DEFAULT Value	Meaning
A value	Identifies a member of the defined range for the attribute
No default	Used if no default exists for the attribute
Not applicable	Used for read-only attributes and attributes that are event-valued

ABSTRACT VALUE Value	Meaning
DEFAULT	The DEFAULT option can be used by the ALGOL *SETABSTRACTVALUE* statement.
MINIMUM	The MINIMUM option can be used by the ALGOL *SETABSTRACTVALUE* statement.
MAXIMUM	The MAXIMUM option can be used by the ALGOL *SETABSTRACTVALUE* statement.

STORED PERMANENTLY Value	Meaning
Backup file	If the file is a printer file, the value is stored with the backup file.
CD	The value is stored on the CD-ROM.
Disk	The value is stored in the disk file header of the permanent file.
Tape	The value is stored in the tape labels.
No	No value is stored regardless of the kind of file.

PARAMETERS Value	Meaning
Number or numbers	Indicates the number of parameters associated with the attribute and whether they are optional or required
None	Indicates no parameters are associated with the attribute

FTAM FILE ACCESS Value	Meaning
Supported	The attribute is supported by FTAM file access.
Not supported	The attribute is not supported by FTAM file access.
Restricted usage	FTAM file access imposes restrictions on the use of the attribute.
Restricted values	FTAM file access supports only those values listed in the attribute description.

FTAM FILE ACCESS Value	Meaning
Restricted usage/values	FTAM file access imposes restrictions on the use of the attribute and supports only those values listed in the attribute description.

Heritage HOST SERVICES LOGICAL I/O Value	Meaning
Supported	The attribute is supported by Heritage Host Services logical I/O.
Not supported	The attribute is not supported by Heritage Host Services logical I/O.
Restricted usage	Heritage Host Services logical I/O imposes restrictions on the use of the attribute.
Restricted values	Heritage Host Services logical I/O supports only those values listed in the attribute description.
Restricted usage/values	Heritage Host Services logical I/O imposes restrictions on the use of the attribute and supports only those values listed in the attribute description.

PORT SERVICES Value	Meaning
All	This attribute is available for all current and future services.
BNANATIVESERVICE	This attribute is available for the BNANATIVESERVICE.
NETBIOSSESSIONSERVICE	This attribute is available for the NETBIOSSESSIONSERVICE.
OSINATIVESERVICE	This attribute is available for the OSINATIVESERVICE.
OSISESSIONSERVICE	This attribute is available for the OSISESSIONSERVICE.
TCPIPNATIVESERVICE	This attribute is available for the TCPIPNATIVESERVICE.
TCPNATIVESERVICE	This attribute is available for the TCPNATIVESERVICE.
BASICSERVICE	This attribute is available for the BASICSERVICE.
Not applicable	This attributes does not apply to port files.

For a description of each service, refer to the *I/O Subsystem Programming Guide*.

Section 2
General File Attributes A through D

This section presents file attribute descriptions for attributes with names starting with the letter A through names starting with the letter D. Each attribute description begins with a formatted list of attribute characteristics, followed by a narrative discussion of the use of the attribute. Where appropriate, this discussion includes a list of the various combinations of values, mnemonics, and meanings associated with that attribute.

Refer to the tables in Section 1 of this manual for descriptions of the formatted lists.

ACCESSDATE

Kinds	Disk or CD
Interrogate	When assigned
Modify	Never (see following exception)
Type	Integer
Range	70001 through 135365
Default	CREATIONDATE
Abstract Value	Not applicable
Stored Permanently	Disk or CD
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Interrogate the ACCESSDATE attribute to determine when the physical file was last accessed.

The value returned by ACCESSDATE is a decimal integer. When you add the integer 1900000 to the value returned, the result is the date in the form YYYYDDD, where YYYY and DDD represent the year and day, respectively, in Julian form.

The ACCESSDATE value is unaffected by library maintenance and is not updated for files on a disk which is not write-enabled.

ACCESSDATE is updated when ALTERDATE, EXECUTEDATE, or READDATE is updated. Specifically,

- For code files, the ACCESSDATE value is updated upon execution, at the beginning of a job or task.
- For permanent disk files, the ACCESSDATE value is updated when the file is closed if a READ, WRITE, SEEK, or SPACE operation has been performed on the file.

Additionally,

- For control files, guard files, the USERDATAFILE file, and other system files, the ACCESSDATE value is updated when the file is closed.
- For the various disk files referenced by the operating system, the ACCESSDATE value is usually updated when the reference is made. For example, libraries that have been labeled using the SL (Support Library) system command are marked when they are initiated, as are INTRINSIC files, data communication files, and disk firmware files.

The ACCESSDATE attribute is used in conjunction with the ACCESSTIME and ACCESSTZ attributes. See “ACCESSTIME” and “ACCESSTZ” for more information.

The ACCESSDATE attribute can be modified by a valid MCS program if that program previously set the HSFILECOPY file attribute to TRUE.

For CD-ROM files, the ACCESSDATE value never changes.

ACCESSDATEUT

Kinds	Disk
Interrogate	When assigned
Modify	Never
Type	Integer
Range	70001 through 135365
Default	CREATIONDATE
Abstract Value	Not applicable
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Interrogate the ACCESSDATEUT attribute to determine the ACCESSDATE value with respect to UTC time instead of the ACCESSTZ time zone. See “ACCESSDATE” for more information.

When you interrogate the ACCESSDATEUT attribute, you receive an error if the time zone attribute, ACCESSTZ, is 0 (zero). See “ACCESSTZ” for more information.

ACCESSTIME

Kinds	Disk or CD
Interrogate	When assigned
Modify	Never (see following exception)
Type	Integer
Range	0 through 86399999999
Default	CREATIONTIME
Abstract Value	Not applicable
Stored Permanently	Disk or CD
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

Interrogate the ACCESSTIME attribute to determine the time of day, in microseconds since midnight, associated with the ACCESSDATE attribute value. The ACCESSTIME value is updated when the ACCESSDATE value is updated.

The ACCESSTIME attribute can be modified by a valid MCS program if that program previously set the HSFILECOPY file attribute to TRUE.

For CD-ROM files, the ACCESSTIME value never changes.

ACCESSTIMEUT

Kinds	Disk
Interrogate	When assigned
Modify	Never
Type	Integer
Range	0 through 86399999999
Default	CREATIONTIME
Abstract Value	Not applicable
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

Interrogate the ACCESSTIMEUT attribute to determine the ACCESSTIME value with respect to UTC time instead of the ACCESSTZ time zone. See “ACCESSTIME” for more information.

When you interrogate the ACCESSSTIMEUT attribute, you receive an error if the time zone attribute, ACCESSTZ, is 0 (zero). See “ACCESSTZ” in this section for more information.

ACCESSTZ

Kinds	Disk
Interrogate	When assigned
Modify	Never (see following exception)
Type	Integer
Range	0 through 255
Default	CREATIONTZ
Abstract Value	Not applicable
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not supported

Interrogate the ACCESSTZ attribute to determine the number of the time zone that is associated with the ACCESSDATE and ACCESSTIME attribute values. The ACCESSTZ value is updated when the ACCESSDATE value is updated.

To determine the meaning of non-zero values of ACCESSTZ, refer to Appendix G, "Predefined Time Zones." You can use the MCP library procedure TIMEZONENAME to convert the numeric code of a time zone into the name or abbreviation for that time zone. TIMEZONENAME is documented in the *Master Control Program (MCP) System Interfaces Programming Reference Manual.*

If the integer value of the ACCESSTZ attribute is 0 (zero), then there is not a time zone associated with the ACCESSDATE and ACCESSTIME attributes.

The ACCESSTZ value is 0 if the file

- Resides on a disk family that was converted to have FAMILY HEADER VERSION = 7 but the file has not been accessed since the family was converted.
- Resides on a disk family with FAMILY HEADER VERSION = 6.
- Was copied from a tape that was created with version 6 headers and has not been accessed since copied.
- Was copied from a disk family with FAMILY HEADER VERSION = 6 and has not been accessed since that time.
- Was last accessed on a system that did not have a time zone set.

Refer to "Disk File Headers" in Section 1 for more information.

The ACCESSTZ attribute can be modified by a valid MCS program if that program previously set the HSFILECOPY file attribute to TRUE.

ACTUALMAXRECSIZE

Kinds	Port subfile
Interrogate	Anytime
Modify	Never
Type	Integer
Range	1 through 65535 if FRAMESIZE = 8 1 through 10922 if FRAMESIZE = 48
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	All

For all message-oriented services, interrogate the ACTUALMAXRECSIZE attribute to determine the maximum data length in FRAMESIZE units that is handled on READ and WRITE operations.

For TCPIPNATIVESERVICE and TCPNATIVESERVICE, interrogate ACTUALMAXRECSIZE to determine the maximum segment size, in FRAMESIZE units, transmitted by the TCP provider to its peer.

For TCPNATIVESERVICE, the ACTUALMAXRECSIZE value is equal to the REQUESTEDMAXRECSIZE value.

This attribute requires a subfile index as a parameter if the MAXSUBFILES value is greater than 1, except when it is used with a file declaration. If it is used in a file declaration or the subfile index specified is zero, then the value of ACTUALMAXRECSIZE is changed in all allocated subfiles and it becomes the new default for any subfiles that are allocated later, even after increasing MAXSUBFILES.

Message-Oriented Services

User data lengths handled on READ and WRITE operations are affected by the size specification on the READ and WRITE operations, the size of the read and write program buffer, the actual length of the user data, and the value of ACTUALMAXRECSIZE.

Before the dialogue is established, the ACTUALMAXRECSIZE value represents the specified REQUESTEDMAXRECSIZE value. After the dialogue is established, the ACTUALMAXRECSIZE value might be lower than the REQUESTEDMAXRECSIZE value because of the port provider implementation restrictions or because of record size negotiation between the two dialogue endpoints. If you set the PORTSEGMENTIO file attribute value to TRUE, the ACTUALMAXRECSIZE value is set to the value of REQUESTEDMAXRECSIZE when the subport is opened.

The following length restrictions apply to the interprocess communications (IPC) implementations:

Network Implementation	Range
BNA Version 2 local dialogue	1 through 65513
BNA Version 2 remote dialogue	1 through 20000
HLCN	1 through 65435
Local Port Provider (LPP)	1 through 65513
OSI	1 through 64512
TCP remote dialogue	1 through 65535

In a BNA network environment, the system negotiates the ACTUALMAXRECSIZE value for both dialogue endpoints by using the REQUESTEDMAXRECSIZE values of both endpoints. ACTUALMAXRECSIZE is changed to the lower value of the two REQUESTEDMAXRECSIZE values. Note that a host that is not an MCP based system might have a REQUESTEDMAXRECSIZE default value that is different from that of the MCP based host. ACTUALMAXRECSIZE might also be lower than REQUESTEDMAXRECSIZE because of network implementation restrictions. In a BNA network environment, a BNA host—whether or not it is an MCP based host—is always able to support a record size of at least 20000.

The value of this attribute is equivalent to the value returned when MAXRECSIZE is interrogated with the specified subfile index.

Data-Stream-Oriented Services

For TCPIPNATIVESERVICE and TCPNATIVESERVICE, you can use the ACTUALMAXRECSIZE value to optimize the utilization of the underlying transmission mechanisms. If your program delivers data through the WRITE statement to the provider in multiples of ACTUALMAXRECSIZE bytes, the performance of the program might increase.

ADAPTABLE

Kinds	Disk
Interrogate	Anytime
Modify	Anytime
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Abstract Value	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Set the ADAPTABLE attribute of a logical file to TRUE to have the system automatically adjust the values of the ANYSIZEIO and BUFFERSHARING attributes of that logical file if the current values are not supported by the physical file, allowing the file to be successfully opened.

When the value of ADAPTABLE is TRUE:

- If the value of the logical file attribute ANYSIZEIO is TRUE when you attempt to open a physical file that does *not* support an ANYSIZEIO value of TRUE, the value of ANYSIZEIO attribute of the logical file will be changed to FALSE.
- If the value of the logical file attribute BUFFERSHARING is SHARED or EXCLUSIVELYSHARED when you attempt to open a physical file that does *not* support buffer sharing, the BUFFERSHARING attribute of the logical file will be changed to NONE.
- Opening a new or an existing file via Heritage Host Services Logical I/O causes the value of the logical file attribute ANYSIZEIO to be changed to FALSE and the value of BUFFERSHARING to be changed to NONE.

AFTER

Kinds	Disk and printer
Interrogate	Anytime
Modify	Anytime
Type	Character string
Range	Not applicable
Default	Null string
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Use the AFTER attribute to indicate when printing should start. The string value must conform to the syntax for the <starttime spec> variable, which is defined in the *Work Flow Language (WFL) Programming Reference Manual*. The <date> portion of the <starttime spec> variable is optional and, if not stated, is assumed to be the date the print request for the file is received by the Print System.

The file is not printed before the time and date indicated. However, the request is immediately assigned a print request number and added to the list of scheduled print requests. At the specified time and date, the request becomes eligible for printing. For further information about the AFTER file attribute, refer to the *Print System Administration, Operations, and Programming Guide*.

The default value for this attribute is a null string, which indicates no time restriction. If the value of the PRINTDISPOSITION attribute is DIRECTDLP or DIRECTPS, the AFTER attribute does not apply and is ignored.

If you specify the AFTER attribute for a printer file with PRINTDISPOSITION=NOW, the opening of the file is suspended until after the specified time.

The following are examples of valid string values:

```
12:00           (print after 12 noon)

14:00 on +1     (print after 2 pm tomorrow)
```

ALIGNFILE

Kinds	Disk and printer
Interrogate	Anytime
Modify	When closed
Type	Character string
Range	Not applicable
Default	Null string
Abstract Value	DEFAULT
Stored Permanently	Backup file
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

If the value of the ALIGNMENT attribute is TRUE, use the ALIGNFILE attribute to specify the name of a printer backup file that contains an alignment pattern for a particular form. The ALIGNFILE attribute is not valid when the value of the PRINTDISPOSITION attribute is DIRECTDLP or DIRECTPS.

The alignment pattern, which can have more than one page of pattern, is used by the Print System to aid in forms alignment on printers. When the Print System uses an alignment pattern, an operator action is required to initiate the printing of the output file.

The default value for ALIGNFILE is a null string, which indicates that no file is named and the first page of the backup file is used as the alignment pattern.

The syntax for the ALIGNFILE assignment statement is

```
ALIGNFILE = <file name>
```

Refer to the FILENAME attribute for information about a valid MCP disk file name.

ALIGNMENT

Kinds	Disk and printer
Interrogate	Anytime
Modify	When closed
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Abstract Value	DEFAULT
Stored Permanently	Backup file
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Set the ALIGNMENT attribute to TRUE to specify that forms alignment is to be performed on a printer. You can align forms by using a pre-specified alignment pattern, or by using the first 60 lines or first page of the printer file. An alignment pattern is used primarily to align special forms, such as invoices or labels, before printing the actual information on such a form.

If the ALIGNMENT value is TRUE and the ALIGNFILE attribute specifies a file, the alignment pattern in the specified file is printed. If the ALIGNFILE value is a null string or is not specified, the first page of the backup file is used as the alignment pattern. After a pattern is printed, the operator can enter one of the following PS (Print System) system command forms:

- PS OK

  This command starts regular printing.
- PS NOTOK

  This command requests that the alignment pattern be printed again.

For further information about aligning forms on a printer, refer to the *Print System Guide*.

The ALIGNMENT attribute is not valid when the value of the PRINTDISPOSITION attribute is DIRECTDLP or DIRECTPS.

ALLOWSPECIALFILE

Kinds	Disk
Interrogate	Anytime
Modify	Anytime
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Abstract Value	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

Set the ALLOWSPECIALFILE attribute to enable the use of POSIX special files.

The value of ALLOWSPECIALFILE must be TRUE in order to use the POSIX standard capabilities identified by special FILEKIND values of DEVNULL or FIFO. ALLOWSPECIALFILE has no effect on files with other values of FILEKIND.

Note: *Special actions might need to be taken when using POSIX special files. Refer to the FILEKIND attribute description for more information.*

ALTERDATE

Kinds	Disk or CD
Interrogate	When assigned
Modify	Never (see following exception)
Type	Integer
Range	70001 through 135365
Default	CREATIONDATE
Abstract Value	Not applicable
Stored Permanently	Disk or CD
Parameters	None
FTAM File Access	Supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Interrogate the ALTERDATE attribute to determine when the last CLOSE operation was performed on the physical file following some alteration to the data in the file.

The value returned by ALTERDATE is a decimal integer. When you add the integer 1900000 to the value returned, the result is the date in the form YYYYDDD, where YYYY and DDD represent the year and day, respectively, in Julian form.

ALTERDATE is unaffected by library maintenance. Updating a permanent file attribute—for example, FILENAME or SECURITYTYPE—does not cause the ALTERDATE value of the file to update.

The ALTERDATE attribute is used in conjunction with the ALTERTIME and ALTERTZ attributes. See “ALTERTIME” and “ALTERTZ” for more information.

The ALTERDATE attribute can be modified by a valid MCS program if that program previously set the HSFILECOPY file attribute to TRUE.

For CD-ROM files, the ALTERDATE value never changes.

ALTERDATEUT

Kinds	Disk
Interrogate	When assigned
Modify	Never
Type	Integer
Range	70001 through 135365
Default	CREATIONDATE
Abstract Value	Not applicable
Stored Permanently	Disk
Parameters	None
FTAM File Access	Supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Interrogate the ALTERDATEUT attribute to determine the ALTERDATE value with respect to UTC time instead of the ALTERTZ time zone. See “ALTERDATE” for more information.

When you interrogate the ALTERDATEUT attribute, you receive an error if the time zone attribute, ALTERTZ, is 0 (zero). See “ALTERTZ” for more information.

ALTERNATEGROUPS

Kinds	Disk
Interrogate	Anytime
Modify	Anytime
Type	Character string
Range	See below
Default	See following text
Abstract Value	Not applicable
Stored Permanently	Yes
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Use the ALTERNATEGROUPS attribute to associate multiple groupcodes with a file and to specify the application of a specific security mode value for each associated groupcode.

The effect of each groupcode and its corresponding security mode value mnemonic is identical to the effect of assigning that groupcode to the file’s GROUP attribute and assigning the corresponding security mode value mnemonic to the file’s GROUPRWX attribute. The ALTERNATEGROUPS attribute allows more than one such specification for a file.

The ALTERNATEGROUPS specification has no effect on a task running with a USERCODE that is the same as the OWNER attribute of the file.

If the task’s GROUPCODE or any of the groupcodes specified in the task’s SUPPLEMENTARYGRPS attribute match the file’s GROUP or any of the groupcodes specified in the ALTERNATEGROUPS attribute, then the ALTERNATEGROUPS, GROUP, and GROUPRWX attributes function together as a unit to control access to the file. The access provided by all such matches is cumulative. For example, R (read) access is granted if the corresponding security mode value of any matching groupcode includes R; the same is true for W (write) and X (execute) permission.

If the task’s GROUPCODE or any of the groupcodes specified in the task’s SUPPLEMENTARYGRPS attribute do not match the file’s GROUP or any of the groupcodes specified in the ALTERNATEGROUPS attribute, then the file’s OTHERRWX attribute governs file access by the task. See “SECURITYMODE” for further information about file access.

Range

<ALTERNATEGROUPS specification>

```
  ┌<─────────────── , ──────────────┐
──┴─/9\─<group code>── : ─┬─ NO ────┴─┬─────┬────────────────────────────┤
                          ├─ R ────┤  └─ . ─┘
                          ├─ W ────┤
                          ├─ X ────┤
                          ├─ RW ───┤
                          ├─ RX ───┤
                          ├─ WX ───┤
                          └─ RWX ──┘
```

The syntax of the ALTERNATEGROUPS attribute requires a string of EBCDIC characters. It is terminated by the first occurrence of the end of the string, by a null character, or by a period that is not part of a quoted <group code> part. The value returned is terminated by a period.

<group code>

Group codes are identifiers that are acceptable values for the GROUP file attribute. The identifier may be from 1 to 17 alphanumeric characters or it may be 1 to 17 arbitrary characters surrounded by quotes.

A group code that consists of a quoted identifier in parentheses, such as "(USER1)", is treated as the group consisting of just the usercode specified by the identifier string. This group code syntax is limited to usercodes consisting of no more than 15 characters. These single user groups can be used without administrator assistance. Group codes that do not follow this format must be defined using the GROUP and SUPPLEMENTARYGRPS USERDATA attributes by a system administrator.

The string value that follows the colon (:) indicates the access permitted as a result of membership in the corresponding group indicated by the <group code>. The strings correspond to the mnemonics used for the mnemonic subattributes defined in the SECURITYMODE file attribute.

Interrogating the ALTERNATEGROUPS attribute

Interrogating the ALTERNATEGROUPS attribute of a file returns one of the following values:

- If the attribute is interrogated while the logical file is assigned to a physical file, the value from the physical file is returned. A null string is returned if the physical file does not have the ALTERNATEGROUPS attribute set.
- If the attribute is interrogated while the logical file is unassigned, the value returned is the last value assigned to the ALTERNATEGROUPS attribute. A null string is returned if the ALTERNATEGROUPS attribute is not set.

Setting the ALTERNATEGROUPS attribute

Set the ALTERNATEGROUPS attribute to any syntactically valid string. The attribute is cleared by setting it to the null string, to an empty string, or to a string consisting of a period (.). Clearing the attribute returns it to the same state as if it was never set.

When the logical file is assigned to a physical file:

- If the ALTERNATEGROUPS attribute is set to a value that differs from the current value, the physical file's ALTERNATEGROUPS value changes to the specified value if the task has appropriate privileges.
- Only the owner of the file or a privileged user can change this attribute for an existing file.

See "Setting the SECURITYMODE Attribute" in Section 4 for special circumstances that also prevent setting the ALTERNATEGROUPS attribute. These circumstances can also prevent setting the SECURITYMODE (and its subattributes), SECURITYTYPE, SECURITYUSE, SECURITYGUARD, and GROUP attributes.

When the ALTERNATEGROUPS attribute of an existing disk file is modified, the TIMESTAMP attribute is updated.

If the ALTERNATEGROUPS attribute is cleared when the logical file is unassigned, the ALTERNATEGROUPS attribute of the logical file reverts back to its default state (not set). If the ALTERNATEGROUPS attribute is cleared when the logical file is assigned to a physical file, the ALTERNATEGROUPS attribute of the physical file is deleted. When a new disk file is created as part of the file open process, any user-specified ALTERNATEGROUPS value is propagated to the physical file.

When the logical file is used to open an existing disk file, a user-specified ALTERNATEGROUPS value set on the logical file has no effect on the ALTERNATEGROUPS attribute of the physical file. In other words, the ALTERNATEGROUPS attribute is not propagated from the logical file to an existing physical file during the file open process.

Cautions

Specifying a large number of groupcodes in the ALTERNATEGROUPS attribute causes the file open process to take longer, particularly for users that are assigned to a large number of SUPPLEMENTARYGRPS in the USERDATAFILE.

Unlike their affect on the GROUPRWX attribute, setting the SECURITYTYPE or SECURITYUSE attributes has no effect on the ALTERNATEGROUPS attribute. Additionally, unlike the GROUPRWX attribute, the ALTERNATEGROUPS attribute has no effect on the reported values of SECURITYTYPE and SECURITYUSE for a file.

Refer to "Securing Files" in the *Security Features Operations Programming Guide* and "SECURITYMODE" in this book for more information about the use of the ALTERNATEGROUPS attribute.

ALTERTIME

Kinds	Disk or CD
Interrogate	When assigned
Modify	Never (see following exception)
Type	Integer
Range	0 through 86399999999
Default	CREATIONTIME
Abstract Value	Not applicable
Stored Permanently	Disk or CD
Parameters	None
FTAM File Access	Supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Interrogate the ALTERTIME attribute to determine the time of day, in microseconds since midnight, that is associated with the ALTERDATE attribute value. The ALTERTIME value is updated when the ALTERDATE value is updated.

The ALTERTIME attribute can be modified by a valid MCS program if that program previously set the HSFILECOPY file attribute to TRUE.

For CD-ROM files, the ALTERTIME value never changes.

ALTERTIMEUT

Kinds	Disk
Interrogate	When assigned
Modify	Never
Type	Integer
Range	0 through 86399999999
Default	CREATIONTIME
Abstract Value	Not applicable
Stored Permanently	Disk
Parameters	None
FTAM File Access	Supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Interrogate the ALTERTIMEUT attribute to determine the ALTERTIME value with respect to UTC time instead of the ALTERTZ time zone. See “ALTERTIME” for more information.

When you interrogate the ALTERTIMEUT attribute, you receive an error if the time zone attribute, ALTERTZ, is 0 (zero). See “ALTERTZ” for more information.

ALTERTZ

Kinds	Disk
Interrogate	When assigned
Modify	Never (see following exception)
Type	Integer
Range	0 through 255
Default	CREATIONTZ
Abstract Value	Not applicable
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not supported

Interrogate the ALTERTZ attribute to determine the number of the time zone that is associated with the ALTERDATE and ALTERTIME attribute values. The ALTERTZ value is updated when the ALTERDATE value is updated.

To determine the meaning of non-zero values of ALTERTZ, refer to Appendix G, “Predefined Time Zones.” You can use the MCP library procedure TIMEZONENAME to convert the numeric code of a time zone into the name or abbreviation for that time zone. TIMEZONENAME is documented in the *Master Control Program (MCP) System Interfaces Programming Reference Manual.*

If the integer value of the ALTERTZ attribute is 0 (zero), then there is not a time zone associated with ALTERDATE and ALTERTIME.

The ALTERTZ value is 0 if the file

- Resides on a disk family that was converted to have FAMILY HEADER VERSION = 7 but the file has not been altered since the family was converted.
- Resides on a disk family with FAMILY HEADER VERSION = 6.
- Was copied from a tape that was created with version 6 headers and hasn't been altered since that time.
- Was copied from a disk family with FAMILY HEADER VERSION = 6 and hasn't been accessed since that time.
- Was last altered on a system which did not have a time zone set.

Refer to “Disk File Headers” in Section 1 for more information.

The ALTERTZ attribute can be modified by a valid MCS program if that program previously set the HSFILECOPY file attribute to TRUE.

ANYSIZEIO

Kinds	Disk or CD
Interrogate	Anytime
Modify	When closed or closed with retention
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

Set the ANYSIZEIO attribute to TRUE to specify that the transfer of any number of frames in a single I/O operation is not constrained by the MINRECSIZE and MAXRECSIZE attribute values. The only constraints are the size of the data area in the program, the length of the file, and that the I/O length cannot span records with a combined length greater than (((2**20)-1) = 1048575) frames. The file must be defined with the following file attribute values:

- The FILESTRUCTURE value must be STREAM.
- The BLOCKSTRUCTURE value must be FIXED.
- The FILEORGANIZATION value must be NOTRESTRICTED.
- The UPDATEFILE value must be FALSE.

When this attribute is TRUE, a serial I/O operation begins data transfer at the next record boundary following the point at which the previous operation completed data transfer. If the previous operation had transferred an integral number of records, data transfer would begin at the very next frame.

The ANYSIZEIO attribute does not allow reading beyond the end of the file. Data transfer stops at the end of the last record in the file, and the value of the CURRENTRECORDLENGTH attribute includes only those records that were actually accessed.

If you have a program that supplies an arbitrarily large record size in its I/O statements expecting the I/O to be truncated to the MAXRECSIZE value, do not set the ANYSIZEIO value to TRUE. When ANYSIZEIO is TRUE, the maximum transfer length is no longer controlled by the MAXRECSIZE value.

If the ANYSIZEIO value is TRUE, the MAXRECSIZE value is used by the operating system only to locate a record when the file-relative record number of the record is given. When used to find a record, each MAXRECSIZE number of frames is considered one record.

If MAXRECSIZE has a value of 1 and ANYSIZEIO is TRUE, the file is treated as a randomly addressable frame stream. Such a file provides capabilities similar to a file with a BLOCKSTRUCTURE value of EXTERNAL, and additionally allows random access to any frame in the file. A randomly addressable frame stream file also eliminates the possibility of the end of the file occurring in the middle of a record (as could happen with partial-record writes), since all I/O operations transfer integer multiples of MAXRECSIZE.

Open errors occur when ANYSIZEIO is TRUE in the following cases:

- The file KIND is not DISK or CD.
- The FILESTRUCTURE attribute value is not STREAM.
- The BLOCKSTRUCTURE attribute is not FIXED.
- The FILEORGANIZATION attribute value is not NOTRESTRICTED.
- The UPDATEFILE attribute value is TRUE. Note that the UPDATEFILE attribute can have its value changed to FALSE during the OPEN process. Refer to the description of the UPDATEFILE attribute for more information.

APL

Kinds	Disk
Interrogate	When open
Modify	When closed
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Abstract Value	DEFAULT
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

Set the APL attribute to TRUE to specify that only a program whose code file also has its APL attribute set to TRUE can access the file. The APLB language system does not use this attribute. If a program whose code file has its APL attribute set to FALSE creates a file specifying the APL attribute value as TRUE, the file is created with an APL value of FALSE. The APL attribute can be read when the file is open and returns the value for the physical file.

The APL attribute can also be set to TRUE with

- The ALTER statement in WFL; refer to the *Work Flow Language (WFL) Programming Reference Manual* for details
- The MA call in SETSTATUS; refer to the *GETSTATUS/SETSTATUS Programming Reference Manual*

The value of APL cannot be altered by file equation.

APPEND

Kinds	Disk
Interrogate	Anytime
Modify	Anytime
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Set the APPEND attribute to TRUE if you do not want random WRITE operations to be allowed and always want serial WRITE operations to implicitly begin at the current end-of-file position.

If you set APPEND to TRUE and set the BUFFERSHARING file attribute value to a value other than NONE, the operating system implicitly provides locking on WRITE operations. Locking ensures that the WRITE operations are atomic with respect to WRITE operations done to the same physical file by any other task that has the physical file open through a logical file with an APPEND value of TRUE and a BUFFERSHARING value other than NONE.

The use of the APPEND attribute has the following restrictions:

- If the BLOCKSTRUCTURE file attribute value of the file is other than FIXED, the APPEND value must be FALSE.
- Direct I/O files must have an APPEND value of FALSE.
- If the UPDATEFILE file attribute value is TRUE, the APPEND value must be FALSE.
- If the FILEORGANIZATION file attribute value is other than NOTRESTRICTED, the APPEND value must be FALSE.

If you set APPEND to TRUE and set the BUFFERSHARING value to NONE, be aware that WRITE operations from your file that append records to the end of the physical file might conflict with or overlap WRITE operations from another logical file that append records to the same physical file.

APPLICATIONCONTEXT

Kinds	Port subfile
Interrogate	Anytime
Modify	When FILESTATE is equal to CLOSED or OPENRESPONSEPLEASE
Type	Character string
Range	Not applicable
Default	Null string
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	1 optional
FTAM File Access:	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	OSINATIVESERVICE, BNANATIVESERVICE

OSI Usage

Use the APPLICATIONCONTEXT attribute to negotiate the application context between dialogue endpoints. An application context is an explicitly defined set of application service elements, options, and any other information necessary for the interworking of two endpoints on a particular dialogue. The APPLICATIONCONTEXT value contains the object identifier terminated by a period (.). An object identifier must be in number form, and each subidentifier is separated by one or more blanks. The following is an example of an object identifier:

```
"1 3 9999 3."
```

For more information about object identifiers and application contexts, refer to International Standards Organization (ISO) Specifications 8824, 8825, and 8649.

If a null value (".") is specified for APPLICATIONCONTEXT, the mandatory application context field sent with the Open Systems Interconnection (OSI) connection request contains the OSI application context *Nil Application Context* defined by the National Institute of Standards and Technology (NIST) Implementors Agreements. The following is the object identifier for the Nil Application Context:

```
"1 3 14 8 1 1."
```

The application context of the dialogue is negotiated during dialogue establishment. APPLICATIONCONTEXT is used for proposing and counter proposing application context during dialogue establishment. Refer to the *I/O Subsystem Programming Guide* for a discussion about negotiation of application context.

This attribute requires a subfile index as a parameter if the MAXSUBFILES value is greater than 1, except when it is used with a file declaration. If it is used in a file declaration or the subfile index specified is zero, then the value of APPLICATIONCONTEXT is changed in all allocated subfiles and it becomes the new default for any subfiles that are allocated later, even after increasing MAXSUBFILES.

BNA Usage

Use the APPLICATIONCONTEXT attribute to indicate that a port file uses the Host Independent Matching (HIM) feature when opening the subport. Refer to the *I/O Subsystem Programming Guide* for information about using the HIM feature.

APPLICATIONGROUP

Kinds	Port files
Interrogate	Anytime
Modify	When closed
Type	Character string
Range	Not applicable
Default	Null string
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	BNANATIVESERVICE, OSINATIVESERVICE, OSISESSIONSERVICE, BASICSERVICE, TCPIPNATIVESERVICE

Use the APPLICATIONGROUP attribute to identify the user community to which the port belongs. A port communicates only with a correspondent application that has the same APPLICATIONGROUP value.

If the value of APPLICATIONGROUP is not null, the application must be executing under a legal usercode for that application group. Information about valid usercodes for each application group is maintained by the local system. The OPEN or AWAITOPEN operation returns the UNAUTHORIZEDFORAGRSLT (62) open error and an UNAUTHORIZEDFORAPPLICATIONGROUP (11) SUBFILEERROR if the application process is not executing under a legal usercode for the application group. TCPIPSUPPORT returns this error if session establishment is forbidden by the TCP/IP End System Security product. Please refer to the *TCP/IP Implementation and Operations Guide* for more information regarding this feature.

The value of APPLICATIONGROUP must be an identifier of from 1 to 17 characters that begins with an asterisk (*), a digit, or a letter. Application group names beginning with an asterisk are reserved for use by Unisys software or, in some cases, for user programs communicating with Unisys software.

The APPLICATION Authorization Network Operator command can be used to control the use of application groups.

AREAADDRESS

Kinds	Disk or CD
Interrogate	When assigned
Modify	Never
Type	Real
Range	Not applicable
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	Disk or CD
Parameters	1 required
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

Interrogate the AREAADDRESS attribute to determine the physical disk address of an area of a disk file.

The value returned by the AREAADDRESS attribute contains the FAMILYINDEX attribute in the field [41:08] and the sector address in the field [32:33]. A value of 0 (zero) is returned if the area is not allocated. To find out only whether the area is allocated, use the AREAALLOCATED attribute.

The AREAADDRESS attribute requires one parameter—the area number. Area numbers begin at 0.

The following examples show how to interrogate the AREAADDRESS attribute in ALGOL:

```
ADDRESS := DSK(ROWNBR).AREAADDRESS;
```

The following example shows how to interrogate the AREAADDRESS attribute in COBOL74 or COBOL85:

```
IF  ATTRIBUTE AREAADDRESS OF DATAFILE (WS-ROW) = 0
THEN
    GO TO ATTR-ERR.
```

AREAALLOCATED

Kinds	Disk or CD
Interrogate	When assigned
Modify	Never
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Abstract Value	Not applicable
Stored Permanently	Disk or CD
Parameters	1 required
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Interrogate the AREAALLOCATED attribute to determine whether a specific area of a disk file has been allocated. One parameter, the area number, is required; area numbers begin at 0.

The following example shows how to interrogate the AREAALLOCATED attribute in ALGOL :

```
BOOL := F(AREANO).AREAALLOCATED;
```

The following example shows how to interrogate the AREAALLOCATED attribute in COBOL74 or COBOL85:

```
IF  ATTRIBUTE AREAALLOCATED OF DATAFILE (WS-AREA) NOT = VALUE TRUE
THEN
    GO TO ATTR-ERR.
```

AREALENGTH

Kinds	Disk, CD, or printer (if backup)
Interrogate	Anytime
Modify	When unassigned
Type	Integer
Range	0 through 4294967295
Default	See following text
Abstract Value	DEFAULT, MINIMUM, MAXIMUM
Stored Permanently	Disk or CD
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Use the AREALENGTH attribute to determine or specify the number of FRAMESIZE units in a disk file area. When you interrogate AREALENGTH, the value returned includes only those FRAMESIZE units in an area that form part of the file. The value can include control information that is not accessed as part of any record, and it does include any disk space that is assigned to the file purely for disk allocation purposes, such as the space between the end of a block and the next sector boundary.

When you modify the AREALENGTH value to 0 (zero), the AREASIZE attribute value is used instead. However, if the AREASIZE value is also 0, the AREALENGTH default is used. If AREALENGTH has been specified, its value is used regardless of the value of AREASIZE. An attribute error is reported if an attempt is made to modify AREASIZE after AREALENGTH has been specified.

If you do not want to use the default value, assign the desired value to AREALENGTH when you create the file. Assigning a value has no effect once the file has been created because, once the file is assigned, AREALENGTH always reflects the length of an area of the physical disk file.

If you use the file attribute AREALENGTH when creating a printer backup file (KIND = PRINTER, BACKUPKIND = DISK or PACK), the value specified affects the printer backup file created on disk and is interpreted in the context of the attributes of the disk file. When calculating the size of an area, the MAXRECSIZE and BLOCKSIZE of the resulting printer backup file are used. Refer to “Structure of Backup Files” in the *Print System and Remote Print System Administration, Operations, and Programming Guide* for more information about printer backup files.

When a file with a FILESTRUCTURE attribute value of ALIGNED180 is created, the AREALENGTH value is rounded up so that it is divisible by the BLOCKSIZE attribute value. The default value of AREALENGTH for a file with a BLOCKSTRUCTURE attribute value of FIXED is calculated by dividing the number of records in a block into 1000, rounding the quotient up, and multiplying that figure by the BLOCKSIZE value. For other BLOCKSTRUCTURE values, the default value of AREALENGTH is 1000 multiplied by the BLOCKSIZE value.

When a file with a FILESTRUCTURE value of BLOCKED is created, one of the following conditions occurs:

- If the BLOCKSIZE value is such that a single block spans an even number of 180-byte disk sectors, then the AREALENGTH value is rounded up to be divisible by the BLOCKSIZE value.
- If a block spans an odd number of 180-byte disk sectors, then the AREASIZE value is rounded up to a multiple of two times the BLOCKSIZE value.

Note: *For a description of virtual sector size (VSS-2) disks and the rationale for this adjustment, refer to the* Peripherals Information File *on the* Product Documentation Library CD-ROM.

The default value of AREALENGTH for a file with a FILESTRUCTURE value of BLOCKED is also dependent on the number of 180-byte sectors spanned by a block, as follows:

- If a block spans an even number of 180-byte sectors, the default value of AREALENGTH is either
  - The multiple of the BLOCKSIZE value closest to but not exceeding 184320 bytes
  - The BLOCKSIZE value, when it exceeds 184320 bytes
- If a block spans an odd number of 180-byte sectors, the default value of AREALENGTH is either
  - The multiple of two times the value of BLOCKSIZE that is closest to but does not exceed 184320 bytes
  - Two times the value of BLOCKSIZE, when the BLOCKSIZE value exceeds 184320 bytes

If the AREALENGTH value is less than the BLOCKSIZE value, then the AREALENGTH value is changed to either

- The value of BLOCKSIZE, when a block spans an even number of 180-byte sectors
- Two times the value of BLOCKSIZE, when a block spans an odd number of sectors

When a file with a FILESTRUCTURE value of STREAM is created, the AREALENGTH value is not adjusted if the AREASIZE value is greater than or equal to the MAXRECSIZE value and the AREALENGTH value spans an even number of 180-byte disk sectors. If the AREALENGTH value is less than the MAXRECSIZE value, or the AREALENGTH value spans an odd number of 180-byte sectors, the AREALENGTH value is rounded up to the closest multiple of the MAXRECSIZE value that does span an even number of 180-byte sectors.

Notwithstanding the maximum value of the AREALENGTH attribute, the length of an area cannot exceed 381774870 sectors.

Because CD-ROM files always have an AREAS attribute value of 1, when an existing CD-ROM file is opened for input, the AREALENGTH value is the actual length of the file in FRAMESIZE units.

AREAS

Kinds	Disk, CD, or printer (if backup)
Interrogate	Anytime
Modify	When unassigned
Type	Integer
Range	0 through 15000
Default	20 for disk or CD, 15 for printer
Abstract Value	DEFAULT, MINIMUM, MAXIMUM
Stored Permanently	Disk or CD
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Use the AREAS attribute to to specify or determine the number of row address words that have been allocated in the disk file header. The value you specify for the AREAS attribute has no effect when you are opening an existing disk file or reopening a file closed with retention.

When you are creating a new file and have specified a value of 0 (zero) for AREAS, the default value of 20 is used. The initial value of AREAS can be exceeded during file expansion if the FLEXIBLE attribute value is not FALSE. When the file expands beyond the AREAS value and FLEXIBLE is not set to FALSE, the system increases AREAS by ten percent (10%) over the area number of the area the system is about to allocate. However, the number of areas is not increased to over 1000 until an area number of 1000 is actually allocated.

When a file is closed with the CRUNCH option, the value of AREAS can decrease.

If you use the AREAS file attribute when creating a printer backup file (KIND = PRINTER, BACKUPKIND = DISK or PACK), the value specified affects the printer backup file created on disk and is interpreted in the context of the attributes of that disk file. The default value of AREAS for a printer backup file is 15. Additional areas can be added if the value of the FLEXIBLE attribute is not FALSE. Refer to Appendix I, “Structure of Backup Files,” in the *I/O Subsystem Programming Guide* for more information about printer backup files.

The limit on the number of areas in a file is normally 15000. However, under some circumstances a lesser limit is enforced. See “LASTACCESSIBLEAREA” for more information about accessible areas.

Note: *Files with more than 1000 areas cannot be opened on MCPs earlier than SSR 44.2.*

CD-ROM files always have an AREAS value of 1. For information about programming for CD-ROM files, refer to the *I/O Subsystem Programming Guide*.

Refer to the description of the AREASINUSE attribute to determine the actual number of areas allocated for a disk file.

AREASECTORS

Kinds	Disk or CD
Interrogate	When assigned
Modify	Never
Type	Integer
Range	1 through 381774870
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	Disk or CD
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Interrogate the AREASECTORS attribute to determine the number of physical disk sectors necessary to accommodate one area of the file.

If a file is crunched, any unoccupied disk space in the last area is released for reallocation. This last area might then contain fewer sectors than the AREASECTORS value.

The product of the AREASECTORS and SECTORSIZE values indicates the number of bytes of disk space occupied by each area.

Because CD-ROM files always have an AREAS attribute value of 1, when an existing CD-ROM file is opened for input, the AREASECTORS value is the total number of sectors for the file.

AREASINUSE

Kinds	Disk or CD
Interrogate	When assigned
Modify	Never
Type	Integer
Range	0 through 15000
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	Disk or CD
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

Interrogate the AREASINUSE attribute to determine the actual number of areas allocated for a disk file. Refer to the descriptions of the AREAS and AREAALLOCATED attributes. Because disk areas are allocated only when needed, it is possible, using random I/O operations, to create a disk file in which areas in the logical middle of the file are never referenced and hence never allocated.

AREASIZE

Kinds	Disk, CD, or printer (if backup)
Interrogate	Anytime (local host) When unassigned (remote host)
Modify	When unassigned
Type	Integer
Range	0 through 4294967295
Default:	See following text
Abstract Value	DEFAULT, MINIMUM, MAXIMUM
Stored Permanently	Disk or CD
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Restricted usage
Port Services	Not applicable

Use the AREASIZE attribute to determine or specify the number of logical records or blocks in an area of a disk file. Unlike the AREALENGTH attribute, which expresses the size of an area of a file in FRAMESIZE units, AREASIZE expresses the size of an area of a file as follows:

- In logical records if the BLOCKSTRUCTURE attribute value is FIXED
- In blocks if one of the following conditions are true:
  - The BLOCKSTRUCTURE value is EXTERNAL, LINKED, VARIABLE, VARIABLE2, or VARIABLEOFFSET.
  - The FILETYPE attribute value is 1, 2, 3, 4, or 6.

AREASIZE has no meaning for a variable-length file with a FILESTRUCTURE attribute value of STREAM.

Assigning a value to AREASIZE is meaningful only when a new file is to be created because, once the file is assigned, AREASIZE always reflects the size of the area of the physical disk file.

If you use the file attribute AREASIZE when creating a printer backup file (KIND = PRINTER, BACKUPKIND = DISK or PACK), the value specified affects the printer backup file created on disk and is interpreted in the context of the attributes of the disk file. When calculating the size of an area, the MAXRECSIZE and BLOCKSIZE of the resulting printer backup file are used. The AREASIZE for a printer backup file defaults to 150. Refer to “Structure of Backup Files” in the *Print System and Remote Print System Administration, Operations, and Programming Guide* for more information about printer backup files.

When a new disk file is created and the FILESTRUCTURE value is not STREAM, the AREASIZE value is rounded up so that it is also a multiple of the number of records per block. If AREASIZE is unspecified or 0 (zero) and AREALENGTH is unspecified, the default value is assumed. If AREALENGTH is specified, AREASIZE is ignored. If AREASIZE is modified when AREALENGTH has been specified, an attribute error occurs.

Notwithstanding the maximum value of the AREASIZE attribute, the length of an area cannot exceed 381774870 sectors.

Because CD-ROM files always have an AREAS attribute value of 1, when an existing CD-ROM file is opened for input, the AREASIZE attribute value is the actual number of records in the file.

When a file with a FILESTRUCTURE value of ALIGNED180 is created, the value of AREASIZE is rounded up so that it is divisible by the number of records in a block. The default value of AREASIZE is determined by dividing the number of records in a block into 1000, rounding up the quotient, and multiplying that figure by the number of records in a block. Because AREASIZE is expressed in blocks when the BLOCKSTRUCTURE attribute has a value other than FIXED, the number of records per block is considered 1 for such BLOCKSTRUCTURE values, for the purpose of determining the AREASIZE value.

In order to provide optimal utilization of virtual sector size (VSS-2) disks, the AREASIZE value for a file with a FILESTRUCTURE value of BLOCKED or STREAM is always set to a value that yields an area spanning an even number of 180-byte disk sectors.

Note: *For a description of VSS-2 disks and the rationale for this adjustment, refer to the* Peripherals Information File *on the* Product Documentation Library CD-ROM.

When a file with a FILESTRUCTURE value of BLOCKED is created, one of the following results occurs:

- If a block spans an even number of 180-byte sectors, the AREASIZE value is rounded up to a multiple of the number of records in a block.
- If a block spans an odd number of 180-byte sectors, the AREASIZE value is rounded up to the number of records in two blocks.

The default value of AREASIZE for a file with a FILESTRUCTURE value of BLOCKED is the AREALENGTH default value divided by the BLOCKSIZE value and then multiplied by the number of records in a block. If a block spans an odd number of 180-byte sectors, the AREALENGTH default value is divided by the BLOCKSIZE value and then multiplied by the number of records in two blocks. When the BLOCKSTRUCTURE value is other than FIXED, the AREASIZE value is expressed in blocks. A value of 1 is used for the number of records for blocks and for determining the AREASIZE value.

When a file with a FILESTRUCTURE value of STREAM is created, the AREASIZE value is meaningful only if the BLOCKSTRUCTURE value is FIXED, as follows:

- If the AREASIZE value is less than 1 and the MAXRECSIZE value spans an even number of 180-byte sectors, then the AREASIZE value is changed to 1. For all other values of BLOCKSTRUCTURE, the AREASIZE value is not changed.
- If the AREASIZE value spans an odd number of sectors, the value is set to the closest number of MAXRECSIZE records that does span an even number of 180-byte sectors.

The default value of AREASIZE for a file with a FILESTRUCTURE value of STREAM is the default value of AREALENGTH divided by the MAXRECSIZE value, when the result yields an area that spans an even number of 180-byte sectors. If the preceding calculation yields an area that spans an odd number of 180-byte disk sectors, then the AREASIZE value is adjusted by adding the number of MAXRECSIZE records needed to get the area to span an even number of 180-byte sectors.

When the AREASIZE file attribute of a disk file being accessed using Heritage Host Services Logical I/O is interrogated, the value of the AREALENGTH attribute is returned.

ASSIGNTIME

Kinds	Remote
Interrogate	When open
Modify	Never
Type	Integer
Range	0 through 86400
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	No
Parameters	1 required
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

Interrogate the ASSIGNTIME attribute to determine the time, in seconds since midnight, when the station was assigned to the file. The ASSIGNTIME attribute requires one parameter: the relative station number (RSN).

The following ALGOL example shows how to interrogate the ASSIGNTIME attribute:

```
X := F(RSN).ASSIGNTIME;
```

ASSOCIATEDFILENAME

Kinds	Tape
Interrogate	Anytime
Modify	When closed
Type	Character string
Range	See following text
Default	Null string
Abstract Value	DEFAULT
Stored Permanently	Yes
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

Use the ASSOCIATEDFILENAME attribute to specify or determine the name of a disk file that is associated with a tape volume family.

The associated disk file is used to keep track of the status of files on the tape or to store position ID values of records on the tape. A tape with an associated file cannot be purged without the approval of the operator which prevents any accidental purges.

The specified file name can be up to 57 characters. The file name cannot include a usercode, an asterisk (*), or an ON <family name> suffix.

When the file is assigned, the ASSOCIATEDFILENAME attribute can be modified only when it is positioned at the beginning of the tape. The name of the associated disk file is copied from the attribute value when the tape volume labels are written. A value that is not null, results in ANSI87 labels.

By default the associated disk file resides under the usercode of the task that created the tape file on the DL LIBMAINTDIR disk family. The system does not define the content of the disk file.

If the ASSOCIATEDFILENAME attribute is interrogated when the file is assigned, the attribute returns the file name including a usercode or an asterisk (*) prefix in the tape label and the ON <family name> suffix. The family name is that name of the DL LIBMAINTDIR disk family. If the attribute is not interrogated, the returned value is the last specified value or a null string if no value has been specified.

When the ASSOCIATEDFILENAME attribute is interrogated for a library maintenance tape, the returned value is the file name of the LIBMAINTFIR tape directory disk file for that tape.

Note that the ASSOCIATEDFILENAME attribute cannot be specified for a library maintenance tape.

ATTERR

Kinds	All
Interrogate	Anytime
Modify	Never
Type	Boolean
Range	TRUE, FALSE
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	No
FTAM File Access	Supported
Heritage Host Services Logical I/O	Supported
Port Services	All

Interrogate the ATTERR attribute to determine whether the last file attribute action resulted in an error. A value of TRUE indicates that an error occurred. For example, a file might have been opened with incompatible attribute values, or a certain attribute combination might be in the process of deimplementation.

Most attribute errors are nonfatal, but they can be informative to the user program. The ATTERR attribute is set to FALSE after a successful file attribute action. ATTERR used in conjunction with the ATTYPE and ATTVALUE attributes can be helpful in correcting programming errors.

Attribute errors related to event-valued attributes are always fatal.

ATTMODIFYDATE

Kinds	Disk
Interrogate	When assigned
Modify	Never (see following exception)
Type	Integer
Range	70001 through 135365
Default	CREATIONDATE
Abstract Value	Not applicable
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not supported

Interrogate the ATTMODIFYDATE attribute to determine the last time a file attribute of a physical disk file (such as FILENAME or FILEKIND) was explicitly modified, for example by setting an attribute programmatically or via the WFL ALTER statement.

The value returned by ATTMODIFYDATE is a decimal integer. When you add the integer 1900000 to the value returned, the result is the date in the form YYYYDDD, where YYYY and DDD represent the year and date respectively, in Julian form.

When copying a file to disk, library maintenance sets ATTMODIFYDATE to the current date if the user includes a FAMILYINDEX = <number> specification for that destination disk.

The ATTMODIFYDATE attribute is used in conjunction with the ATTMODIFYTIME and ATTMODIFYTZ attributes. See “ATTMODIFYTIME” and “ATTMODIFYTZ” for more information.

The ATTMODIFYDATE attribute can be modified by a valid MCS program if that program previously set the HSFILECOPY file attribute to TRUE.

ATTMODIFYDATEUT

Kinds	Disk
Interrogate	When assigned
Modify	Never
Type	Integer
Range	70001 through 135365
Default	CREATIONDATE
Abstract Value	Not applicable
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not supported

Interrogate the ATTMODIFYDATEUT attribute to determine the value with respect to UTC time instead of the ATTMODIFYTZ time zone. See “ATTMODIFYDATE” for more information.

When you interrogate the ATTMODIFYDATEUT attribute, you receive an error if the time zone attribute, ATTMODIFYTZ, is 0 (zero). See “ATTMODIFYTZ” for more information.

ATTMODIFYTIME

Kinds	Disk
Interrogate	When assigned
Modify	Never (see following exception)
Type	Integer
Range	0 through 86399999999
Default	CREATIONTIME
Abstract Value	Not applicable
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not supported

Interrogate the ATTMODIFYTIME attribute to determine the time of day, in microseconds since midnight, that is associated with the ATTMODIFYDATE attribute value. The ATTMODIFYTIME value is updated when the ATTMODIFYDATE value is updated.

The ATTMODIFYTIME attribute can be modified by a valid MCS program if that program previously set the HSFILECOPY file attribute to TRUE.

ATTMODIFYTIMEUT

Kinds	Disk
Interrogate	When assigned
Modify	Never
Type	Integer
Range	0 through 86399999999
Default	CREATIONTIME
Abstract Value	Not applicable
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not supported

Interrogate the ATTMODIFYTIMEUT attribute to determine the ATTMODIFYTIME value with respect to UTC time instead of the ATTMODIFYTZ time zone. See "ATTMODIFYTIME" for more information.

When you interrogate the ATTMODIFYTIMEUT attribute, you receive an error if the time zone attribute, ATTMODIFYTZ, is 0 (zero). See "ATTMODIFYTZ" for more information.

ATTMODIFYTZ

Kinds	Disk
Interrogate	When assigned
Modify	Never (see following exception)
Type	Integer
Range	0 through 255
Default	CREATIONTZ
Abstract Value	Not applicable
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not supported

Interrogate the ATTMODIFYTZ attribute to determine the number of the time zone that is associated with the ATTMODIFYDATE and ATTMODIFYTIME attribute values. The ATTMODIFYTZ value is updated when the ATTMODIFYDATE value is updated.

To determine the meaning of non-zero values of ATTMODIFYTZ, refer to Appendix G, “Predefined Time Zones.” You can use the MCP library procedure TIMEZONENAME to convert the numeric code of a time zone into the name or abbreviation for that time zone. TIMEZONENAME is documented in the *Master Control Program (MCP) System Interfaces Programming Reference Manual.*

If the integer value of ATTMODIFYTZ is 0 (zero), then there is not a time zone associated with ATTMODIFYDATE and ATTMODIFYTIME.

The ATTMODIFYTZ value is 0 if the file

- Has CREATIONTZ = 0 and ATTMODIFYDATE and ATTMODIFYTIME are equal to CREATIONDATE and CREATIONTIME.
- Resides on a disk family with FAMILY HEADER VERSION = 6. Refer to “Disk File Headers” in Section 1 for more information.
- Has an attribute modified on a system which did not have a time zone set.

The ATTMODIFYTZ attribute can be modified by a valid MCS program if that program previously set the HSFILECOPY file attribute to TRUE.

ATTVALUE

Kinds	All
Interrogate	Anytime
Modify	Never
Type	Real
Range	See following text
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Supported
Heritage Host Services Logical I/O	Supported
Port Services	All

Following an attribute error, interrogate the ATTVALUE attribute to obtain a value that can aid in correcting the error. If the file is open and an attempt is made to set an attribute that requires the file to be closed, ATTVALUE returns a word with all bits on, that is, REAL (NOT FALSE). This word is a small negative number and prints out as 0 (zero).

If the attribute in error is a string attribute, ATTVALUE returns the pointer as an operand—that is, as a string descriptor with a tag equal to 0. Otherwise, ATTVALUE returns the value to which the attribute was to be set. For more information, refer to the ATTERR and ATTYPE attribute descriptions.

ATTYPE

Kinds	All
Interrogate	Anytime
Modify	Never
Type	Mnemonic
Range	See following text
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Supported
Heritage Host Services Logical I/O	Not supported
Port Services	All

Interrogate the ATTYPE attribute to determine the number of the last file attribute that was used incorrectly. You can use ATTYPE along with the ATTERR and ATTVALUE attributes to determine programming errors related to files.

The VALUE function can be used, in the languages that provide it, to ascertain the number of an attribute when the attribute name is given as a parameter. For example, the following ALGOL and COBOL74 statements use the ATTYPE attribute and the VALUE function to determine if BLOCKSIZE was the last attribute to be used incorrectly:

```
IF FILEID.ATTYPE = VALUE(BLOCKSIZE) THEN HANDLEPROBLEM;

IF ATTRIBUTE ATTYPE OF FILEID IS EQUAL TO VALUE BLOCKSIZE
      PERFORM HANDLE-PROBLEM.
```

Refer to the appropriate language manuals for syntax and semantics.

Refer to Appendix D, “Attribute Numbers,” for a numerical and alphabetical listing of the attributes and their numbers.

AUTOUNLOAD

Kinds	Tape
Interrogate	Anytime
Modify	Anytime
Type	Mnemonic
Range	DONTCARE, ON, OFF
Default	DONTCARE
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	None
FTAM File Access	Not Supported
Heritage Host Services Logical I/O	Supported
Port Services	Not Applicable

Use the AUTOUNLOAD attribute to control the behavior of a file close action on tape drives or to determine the behavior of the tape drive when a CLOSE operation is performed on the file. When the AUTOUNLOAD value of a tape file is ON, the associated tape volumes automatically unload when a program has either released the unit or performed a reel switch. A program can use the AUTOUNLOAD attribute to override the effects of the AUTOUNLOAD option of the system command MODE (Unit Mode).

The following are the possible mnemonic and integer values, and their meanings:

Mnemonic Value	Integer Value	Meaning
ON	1	The tape is unloaded if the CLOSE operation released the volume from program control regardless of the automatic-unload mode of the unit.
DONTCARE	0	The tape is unloaded if the CLOSE operation requested that the tape be unloaded or if the unit is in automatic-unload mode and the CLOSE operation released the volume from program control.
OFF	2	The tape is unloaded only if the CLOSE operation requested that the tape be unloaded.

Note: *Refer to the* I/O Subsystem Programming Guide *for a discussion of the various ways a file can be closed.*

If you interrogate this attribute, the value returned does not reflect the setting of the mode of a unit that might be associated with the logical file.

If you have a program that requires that a tape volume remain mounted after the program closes a file, use CLOSE(F,REWIND) if appropriate. If this version of CLOSE is not appropriate, you can set the AUTOUNLOAD file attribute to OFF before you execute the CLOSE(F) action. After the CLOSE(F) code, you can again set AUTOUNLOAD to DONTCARE, as shown in the following ALGOL example:

```
F.AUTOUNLOAD := VALUE(OFF);
CLOSE(F);
F.AUTOUNLOAD := VALUE(DONTCARE);
```

It is possible to traverse a multireel file without operator intervention by sequentially loading the tape cartridges in an automatic cartridge loader in AUTOMATIC mode and by adding the following ALGOL code:

```
F.AUTOUNLOAD := VALUE(ON);
```

AVAILABLE

Kinds	All
Interrogate	Anytime
Modify	Never
Type	Mnemonic
Range	See following text
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Supported
Heritage Host Services Logical I/O	Supported
Port Services	All

Interrogate the AVAILABLE attribute to determine if a file is present and can be opened. If the file cannot be opened, the system reports a reason for the failure without suspending the program or requiring operator intervention. If the file can be opened, the file is opened.

Using an OPEN statement with the AVAILABLE option or setting the AVAILABLEONLY attribute to TRUE for port files is preferable to interrogating the AVAILABLE attribute. For example, in ALGOL the following syntax can be used to open a file:

```
I := OPEN(F,AVAILABLE)
```

If the file is a port file and the AVAILABLE attribute is interrogated, the OPEN operation performed is equivalent to that performed by the OPEN statement with the AVAILABLE option specified and the subfile index equal to 0 (zero). When the AVAILABLE attribute is interrogated for other types of files, the OPEN operation performed on the file and the value returned to the program are the same as when the OPEN statement is used.

Refer to Appendix E, “OPEN, CLOSE, and RESPOND Results,” for information about open results.

AVAILABLEONLY

Kinds	Port subfile
Interrogate	Anytime
Modify	When closed
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	BNANATIVESERVICE, BASICSERVICE, NETBIOSSESSIONSERVICE, OSINATIVESERVICE, OSISESSIONSERVICE

Use the AVAILABLEONLY attribute to specify whether or not the OPEN or AWAITOPEN operation is suspended until a dialogue can be established with the correspondent endpoint.

When the value of AVAILABLEONLY is TRUE, the OPEN or AWAITOPEN operation is terminated if dialogue establishment is not possible at the time the OPEN or AWAITOPEN operation is invoked.

When the value of AVAILABLEONLY is FALSE, the OPEN or AWAITOPEN operation is suspended until a dialogue can be successfully established with the correspondent endpoint.

Regardless of the AVAILABLEONLY value, the point at which process control is returned to the program is dependent on the value of the control option of the OPEN or AWAITOPEN statement.

- If the control option is WAIT, process control is returned to the program only after the operation finishes.
- If the control option is DONTWAIT (NO WAIT in COBOL74), process control is immediately returned to the program, and if AVAILABLEONLY is FALSE, the operation continues in parallel with the program.
- If the control option of the OPEN statement is OFFER, process control is returned to the program as soon as it is determined that the host is reachable or unreachable. If AVAILABLEONLY is FALSE, the operation continues in parallel with the program.

This attribute requires a subfile index as a parameter if the MAXSUBFILES value is greater than 1, except when it is used with a file declaration. If it is used in a file declaration or the subfile index specified is zero, then the value of AVAILABLEONLY is changed in all allocated subfiles and it becomes the new default for any subfiles that are allocated later, even after increasing MAXSUBFILES.

The following table gives network-specific information about the AVAILABLEONLY attribute:

Network	Network-Related Information
BNA	When an OPEN operation is invoked, the service provider sends the AVAILABLEONLY value with the dialogue request. If no matching subfiles are on a correspondent host or the remote host is unreachable, and the AVAILABLEONLY value is TRUE, the OPEN operation fails. If the AVAILABLEONLY value is FALSE, the request is saved on the correspondent host for future matching. If the AVAILABLEONLY value is FALSE when an AWAITOPEN operation is invoked, the subfile is saved for matching with future dialogue requests if no matching request has already been received. Otherwise, if the AVAILABLEONLY value is TRUE, the subfile is matched only to requests that have already been received.
OSI	When your port file is establishing a dialogue with a remote correspondent endpoint, the AVAILABLEONLY attribute has no effect.

BACKUPDATE

Kinds	Disk
Interrogate	When assigned
Modify	Never (see following exception)
Type	Integer
Range	70001 through 135365
Default	CREATIONDATE
Abstract Value	Not applicable
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not supported

Interrogate the BACKUPDATE attribute to determine the last time the physical file was

- Copied from disk by a WFL COPY & BACKUP operation.
- Copied from disk by a WFL ARCHIVE BACKUP operation.
- Marked as having been backed up via a call on the MCP procedure MARK_FILE_BACKED_UP with the file as a parameter. Refer to "Declaring the MARK_FILE_BACKED_UP Procedure" in Section 9 of the *Master Control Program (MCP) System Interfaces Programming Reference Manual* for more information.

Note: *Whenever a file is copied to disk, library maintenance resets its BACKUPDATE, BACKUPTIME, and BACKUPTZ values to match the CREATIONDATE, CREATIONTIME, and CREATIONTZ values of the file respectively.*

The value returned by BACKUPDATE is a decimal integer. When you add the integer 1900000 to the value returned, the result is the date in the form YYYYDDD, where YYYY and DDD represent the year and date, respectively, in Julian form.

The BACKUPDATE value is not updated for files on a disk which is not write-enabled.

The BACKUPDATE attribute is used in conjunction with the BACKUPTIME and BACKUPTZ attributes. See "BACKUPTIME" and "BACKUPTZ" for more information.

The BACKUPDATE attribute can be modified by a valid MCS program if that program previously set the HSFILECOPY file attribute to TRUE.

BACKUPDATEUT

Kinds	Disk
Interrogate	When assigned
Modify	Never
Type	Integer
Range	70001 through 135365
Default	CREATIONDATE
Abstract Value	Not applicable
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not supported

Interrogate the BACKUPDATEUT attribute to determine the BACKUPDATE value with respect to UTC time instead of the BACKUPTZ time zone. See “BACKUPDATE” for more information.

When you interrogate the BACKUPDATEUT attribute, you receive an error if the time zone attribute, BACKUPTZ, is 0 (zero). See “BACKUPTZ” for more information.

BACKUPKIND

Kinds	Printer
Interrogate	Anytime
Modify	When closed
Type	Mnemonic
Range	See following text
Default	DONTCARE
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Use the BACKUPKIND attribute in conjunction with the KIND attribute to specify the printer backup device or interrogate the BACKUPKIND attribute to determine the current backup device.

The allowable mnemonic values for BACKUPKIND are as follows:

Mnemonic Value	Integer Value	Meaning
DONTCARE	0	Use the system default values.
DISK	1	Use a disk pack.
PACK	17	Use a disk pack.
TAPE	45	Use any available tape.

The BACKUPKIND value is used when the file is opened to select an output device for printer backup files. When no output devices are specified with the BACKUPKIND attribute, other system and task options are used to select an online printerdevice. Site management can control the selection of the backup device through the use of the SB (Substitute Backup) system command and the LPBDONLY and CPBDONLY options of the OP (Options) system command. For more information, refer to the *System Commands Reference Manual*.

If you interrogate BACKUPKIND while the file is unassigned, the value that was most recently assigned to the attribute is returned. If the file is assigned and associated with a backup device, the value returned is equal to the KIND value for the backup device. If the file is assigned but is not associated with a backup device because the file is not a printer file, or if the file is directly attached to a printer device, a value of DONTCARE is returned.

The mnemonic value PACK is functionally synonymous with DISK when setting the attribute, although DISK is the preferred value.

Backup files cannot be unlabeled.

BACKUPTIME

Kinds	Disk
Interrogate	When assigned
Modify	Never (see following exception)
Type	Integer
Range	0 through 86399999999
Default	CREATIONTIME
Abstract Value	Not applicable
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not supported

Interrogate the BACKUPTIME attribute to determine the time of day, in microseconds since midnight, that is associated with the BACKUPDATE attribute value. The BACKUPTIME value is updated when the BACKUPDATE value is updated.

The BACKUPTIME attribute can be modified by a valid MCS program if that program previously set the HSFILECOPY file attribute to TRUE.

BACKUPTIMEUT

Kinds	Disk
Interrogate	When assigned
Modify	Never
Type	Integer
Range	0 through 86399999999
Default	CREATIONTIME
Abstract Value	Not applicable
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not supported

Interrogate the BACKUPTIMEUT attribute to determine the BACKUPTIME value with respect to UTC time instead of the BACKUPTZ time zone. See "BACKUPTIME" for more information.

When you interrogate the BACKUPTIMEUT attribute, you receive an error if the time zone attribute, BACKUPTZ, is 0 (zero). See "BACKUPTZ" for more information.

BACKUPTZ

Kinds	Disk
Interrogate	When assigned
Modify	Never (see following exception)
Type	Integer
Range	0 through 255
Default	CREATIONTZ
Abstract Value	Not applicable
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not supported

Interrogate the BACKUPTZ attribute to determine the number of the time zone that is associated with the BACKUPDATE and BACKUPTIME attribute values. The BACKUPTZ value is updated when the BACKUPDATE value is updated.

To determine the meaning of non-zero values of BACKUPTZ, refer to Appendix G, “Predefined Time Zones.” You can use the MCP library procedure TIMEZONENAME to convert the numeric code of a time zone into the name or abbreviation for that time zone. TIMEZONENAME is documented in the *Master Control Program (MCP) System Interfaces Programming Reference Manual.*

If the integer value of BACKUPTZ is 0 (zero), then there is not a time zone associated with BACKUPDATE and BACKUPTIME.

The BACKUPTZ value is 0 if the file

- Has CREATIONTZ = 0 and BACKUPDATE and BACKUPTIME are equal to CREATIONDATE and CREATIONTIME.
- Resides on a disk family with FAMILY HEADER VERSION = 6. Refer to “Disk File Headers” in Section 1 for more information.
- Is backed up and the system does not have a time zone set.

The BACKUPTZ attribute can be modified by a valid MCS program if that program previously set the HSFILECOPY file attribute to TRUE.

BANNER

Kinds	Disk and printer
Interrogate	Anytime
Modify	Anytime
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Abstract Value	DEFAULT
Stored Permanently	Backup file
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

Set the BANNER attribute to TRUE to specify that a banner page is to be printed before the file is printed. A banner helps identify the file in the printed output. If the value of the PRINTDISPOSITION attribute is DIRECTDLP or DIRECTPS, the BANNER attribute is ignored; the LABEL attribute determines whether standard banner pages are printed.

The content of the banner page depends on the contents of the NOTE attribute or the banner contents that the site administrator has identified through the PS BANNER system command. If the banner content has been specified by the system operator, the NOTE value is ignored. If the NOTE value is not ignored and is a null string, the file title—not the INTNAME value—is printed in block format. Otherwise, the string of characters specified in the NOTE attribute is printed in block format.

BLANK

Kinds	All except printer files
Interrogate	Anytime
Modify	Anytime
Type	Mnemonic
Range	NULL, ZERO
Default	NULL
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	All

Use the BLANK attribute to dynamically control the interpretation of blanks in numeric input fields. This attribute has functional meaning only for programs written in FORTRAN77. If NULL is specified, all blank characters in numerically formatted input fields are ignored, except that a field of all blanks has a value of ZERO. If ZERO is specified, all blanks except leading blanks are treated as zeros.

The mnemonic values and integer values for this attribute follow:

Mnemonic Value	Integer Value
NULL	0
ZERO	1

The following example shows how to specify the BLANK attribute in FORTRAN77:

```
OPEN(5,   BLANK='NULL');

OPEN(5,   BLANK='ZERO');
```

BLOCK

Kinds	All except CD, port, and remote
Interrogate	Anytime
Modify	Never
Type	Integer
Range	Not applicable
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

For a disk file with a FILESTRUCTURE value other than STREAM and all other device files, interrogate the BLOCK attribute to determine the number of the block referenced in the last I/O statement. Do not interrogate the BLOCK attribute if the file is a direct I/O disk file.

A value of -1 is returned under the following conditions:

- When the file is closed
- When the file is open, but no I/O operation has taken place on the file
- After an I/O operation is attempted beyond the end of the file

The first block number in a file is 0 (zero).

If you interrogate the BLOCK attribute for a file with the FILESTRUCTURE value of STREAM, an attribute error occurs and a value of 0 is returned.

BLOCKEDTIMEOUT

Kinds	Port subfile
Interrogate	Anytime
Modify	When CLOSED, or when OPEN has completed
Type	Integer
Range	0 through 1440 minutes
Default	0
Abstract Value	DEFAULT, MINIMUM, MAXIMUM
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	BNANATIVESERVICE, OSINATIVESERVICE, OSISESSIONSERVICE, TCPIPNATIVESERVICE, TCPNATIVESERVICE, BASICSERVICE

If you are using BNANATIVESERVICE, OSINATIVESERVICE, OSISESSIONSERVICE, or BASICSERVICE, use the BLOCKEDTIMEOUT attribute to specify the amount of time, in minutes, the system waits for data to be successfully transmitted to the corresponding application before allowing automatic invocation of the deactivation procedures. When data cannot be successfully transmitted, the value of FILESTATE is BLOCKED. A value of 0 (zero) indicates that the deactivation procedures are invoked by the system that depends on the value of HOSTUNREACHABLETIMEOUT. For more information on setting HOSTUNREACHABLETIMEOUT, refer to the *BNA Version 2 Operations Reference Manual*.

If you are using TCPIPNATIVESERVICE or TCPNATIVESERVICE, use the BLOCKEDTIMEOUT attribute to specify the amount of time, in minutes, that “keep-alive” packets continue to be transmitted. To activate the use of keep-alive packets that keep a dialogue open when a remote host does not respond, use the DIALOGCHECKINTERVAL file attribute.

If the MAXSUBFILES value is greater than 1, the BLOCKEDTIMEOUT attribute requires a subfile index as a parameter, except when it is used with a file declaration or an ALGOL multiple-file assignment statement.

BLOCKSIZE

Kinds	All except CD and port
Interrogate	Anytime
Modify	When closed
Type	Integer
Range	0 through 65535
Default	MAXRECSIZE
Abstract Value	DEFAULT, MINIMUM, MAXIMUM
Stored Permanently	Disk or tape
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Restricted values
Port Services	Not applicable

Use the BLOCKSIZE attribute to determine or specify the length of a block. BLOCKSIZE is expressed in FRAMESIZE units.

For files with a FILESTRUCTURE value of STREAM:

- BLOCKSIZE is ignored if specified.
- A value of 0 (zero) is returned along with an attribute error if BLOCKSIZE is interrogated.
- Buffering can be controlled by using the BUFFERSIZE attribute.

Heritage Host Services logical I/O requires that BLOCKSIZE be less than 65,486 characters.

When the file is opened, the BLOCKSIZE value is changed under the following conditions:

- If the FILEORGANIZATION attribute value is not RELATIVE and the BLOCKSIZE value is less than the MAXRECSIZE attribute value, the BLOCKSIZE value is changed to the MAXRECSIZE value.
- If the FILEORGANIZATION value is RELATIVE and the BLOCKSIZE value is less than or equal to the MAXRECSIZE value, the BLOCKSIZE value is changed to the MAXRECSIZE value plus 1.

For relative files, if a COBOL74 file description contains a *BLOCK CONTAINS n RECORDS* clause, where *n* is a constant, the value used for BLOCKSIZE is calculated as follows:

```
((n + FRAMESIZE - 1) DIV FRAMESIZE) + (n * MAXRECSIZE)
```

Files with BLOCKSTRUCTURE values other than LINKED are blocked only when the BLOCKSIZE value is greater than the MAXRECSIZE value.

If the BLOCKSTRUCTURE value is FIXED, the BLOCKSIZE value must be a nonnegative-integer multiple of the MAXRECSIZE value.

The default value of BLOCKSIZE depends on both the kind of physical unit assigned to the file and the MAXRECSIZE value. Refer to the description of the MAXRECSIZE attribute.

Opening an existing disk file with a logical BLOCKSIZE value unequal to the physical BLOCKSIZE value is allowed only when both values are multiples of the disk sector size.

A tape file is determined to have an inconsistency if a logical BLOCKSIZE value is less than the physical BLOCKSIZE value.

The BLOCKSIZE value is ignored when accessing direct I/O files, however, the BLOCKSIZE value is stored with the physical file for disk and tape files.

For additional information, refer to the discussions of disk files and tape files in the *I/O Subsystem Programming Guide*.

The BLOCKSIZE and MAXRECSIZE attributes have length restrictions for card files and remote files. These restrictions are enforced when the file is opened. If a restriction is violated, an attribute error is issued and the BLOCKSIZE value is reduced appropriately. If a direct I/O remote file attempts to write a record that is too long, the I/O operation is canceled. Refer to the descriptions of the IOCANCEL, IOERRORTYPE, and IORESULT buffer attributes in Section 7, “Direct I/O Buffer Attributes.”

Card files have the following limits:

- A character-oriented file has a limit of 80 characters.
- A word-oriented EBCDIC file has a 14-word limit.
- A binary file has a 20-word limit.

The limit for remote files is approximately 500 words.

For magnetic tapes, refer to the *I/O Subsystem Programming Guide* for information about the minimum and maximum BLOCKSIZE values.

The operating system issues a run-time warning to programs that interrogate the BLOCKSIZE attribute of disk files that have a FILEKIND attribute value of BACKUPPRINTER.

Note: *Programs must interrogate the MAXRECSIZE attribute instead of the BLOCKSIZE attribute to determine the length of a backup block. Refer to the* Print System Guide *for a description of a backup block.*

BLOCKSTRUCTURE

Kinds	All
Interrogate	Anytime
Modify	When closed
Type	Mnemonic
Range	See following text
Default	See following text
Abstract Value	Not applicable
Stored Permanently	Disk or tape
Parameters	None
FTAM File Access	Restricted values
Heritage Host Services Logical I/O	Restricted values
Port Services	BNANATIVESERVICE, NETBIOSSESSIONSERVICE, OSINATIVESERVICE, OSISESSIONSERVICE, BASICSERVICE

Use the BLOCKSTRUCTURE attribute to specify or determine the structure of the file and the format of the records of the file. If you specify a BLOCKSTRUCTURE value, the FILETYPE value is assigned the value corresponding to the BLOCKSTRUCTURE value. Once the FILETYPE value is assigned in this manner, an attribute error is reported if an attempt is made to change the FILETYPE value.

If you do not specify a BLOCKSTRUCTURE value, this attribute is assigned the value corresponding to the FILETYPE attribute value. If you do not specify a FILETYPE attribute, the FIXED value is used.

When BLOCKSTRUCTURE = LINKED, VARIABLE2, or VARIABLEOFFSET, double-byte (16-bit) and mixed multibyte (mixed 8-bit and 16-bit with FRAMESIZE = 8) character sets are not supported.

The following table lists each of the possible mnemonic and integer values of BLOCKSTRUCTURE together with the corresponding value and meaning of the FILETYPE attribute.

Mnemonic Value	Integer Value	Corresponding FILETYPE Value	Meaning
EXTERNAL	1	3	Variable-length records. Neither the record itself nor the structure of the file contains information about the length of the record; length information must be specified externally in the I/O statement. For unblocked tape files, operator display terminal (ODT) files, remote files, and port files, the I/O subsystem determines the actual length to be read and returns the length information in the CURRENTRECORDLENGTH and STATE attributes. For blocked files, if a READ operation specifies a record length that is larger than the remainder of the block, the record is truncated at the end of the block. For blocked files, if a WRITE operation specifies a record length that is longer than the remainder of the block, a new block is started. If the file is a tape file, the previous block is written as a short block.
FIXED	0	0	Blocked or unblocked, fixed-length records.
LINKED	3	6	Variable-length records containing link words which allow the records to be located. The link words are maintained by the I/O subsystem and are not part of the data. The INTMODE value of the file is assumed to be SINGLE, and software translation is never attempted. Such a file is often used for FORTRAN unformatted I/O.

Mnemonic Value	Integer Value	Corresponding FILETYPE Value	Meaning
VARIABLE	2	1	Variable-length records. The record size is contained, in decimal form, in the first four INTMODE characters of the record. If the INTMODE value is OCTETSTRING, the first four characters are EBCDIC characters. If the INTMODE value is SINGLE, the record size is stored in the first word of the record and is represented in binary format. This size includes the length of the size field—for example, for character files the size is four units greater than the length of the data. The size is expressed in FRAMESIZE units. If the value of the SIZEVISIBLE attribute is TRUE, the record length field is included within the record. Double-byte (16-bit) and mixed multibyte (mixed 8-bit and 16-bit with FRAMESIZE = 8) character sets are supported only when the character sets are 8-bit EBCDIC-based, ASCII-based, UCS2, or UCS2NT characters and the file is character oriented.
VARIABLE2	4	2	Variable-length records. The record size is contained in the first two INTMODE characters of the record and is represented in binary format. If the INTMODE value is SINGLE, the record size is stored in the first word of the record. The size is expressed in FRAMESIZE units. This size includes the length of the size field—for example, for character files the size is two units greater than the length of the data.

Mnemonic Value	Integer Value	Corresponding FILETYPE Value	Meaning
VARIABLEOFFSET	5	4	Variable-length records. The record size is contained in a fixed location in the record. The attributes SIZEMODE, SIZEOFFSET, and SIZE2 determine the mode of the information, the position of the record-size information, and the length of the field containing the information, respectively. The size is expressed in FRAMESIZE units and includes the length of the size field.

The following table identifies the situations when only certain mnemonic values are supported:

Situation	Supported Values
Host Service logical I/O	When the file is not blocked, FIXED and EXTERNAL are supported.
File Transfer, Access, and Management (FTAM)	For files with a DOCUMENTTYPE attribute value of FTAM1, FTAM2, or FTAM3, the following values are supported: FIXED, EXTERNAL, and VARIABLE. For files with a DOCUMENTTYPE value of INTAP1, the following values are supported: FIXED and VARIABLE.
File with a KIND attribute value of CD	Only FIXED is supported.
File with a BUFFERSHARING attribute value other than NONE	Only FIXED is supported.
File with an APPEND attribute value of TRUE	Only FIXED is supported.
FILEORGANIZATION value RELATIVE, FILESTRUCTURE value ALIGNED180	Only FIXED is supported.
FILEORGANIZATION value RELATIVE, FILESTRUCTURE value STREAM	FIXED and VARIABLE are supported.
Any FILEORGNIZATION value, FILESTRUCTURE value STREAM or BLOCKED	All values are supported except LINKED.

If the BLOCKSTRUCTURE value has defaulted to FIXED and the MINRECSIZE value is greater than 0 (zero) but less than the MAXRECSIZE value, the BLOCKSTRUCTURE value is changed to EXTERNAL when the file is opened.

Variable-length, blocked records are processed differently than fixed-length, blocked records. If the record to be written plus the offset into the block is greater than the BLOCKSIZE value, the current block is written to the physical file. The record to be written becomes the first record of the next block. If the logical file is assigned to a peripheral unit that allows variable-length blocks, such as a tape file, the part of each block containing valid data is written. However, if the logical file is assigned to a peripheral unit that requires fixed-length blocks, such as a disk file, a field of null characters equal in length to the MINRECSIZE attribute is written into the remainder of the block as an end-of-block marker, and the entire block is written to the physical file.

To facilitate random access to FILEORGANIZATION=RELATIVE, FILESTRUCTURE=STREAM, BLOCKSTRUCTURE=VARIABLE files, the system uses record slots of MAXRECSIZE plus one byte for all records—the extra byte indicates whether the record in a given slot is valid or deleted. Because the system uses a fixed size slot for these records, the file requires more space than might be expected when records are stored in the file that are less than MAXRECSIZE bytes long.

The value of the MINRECSIZE attribute is affected by the BLOCKSTRUCTURE value in the following ways:

BLOCKSTRUCTURE Value	**MINRECSIZE Value**
FIXED	The value is changed to the MAXRECSIZE value when the file is opened, even if a MINRECSIZE value has been explicitly specified.
EXTERNAL	The default value is 0. Logical I/O issues a data error if the record being processed is smaller than the specified MINRECSIZE value.
LINKED, VARIABLE, VARIABLE2, VARIABLEOFFSET	The value must be equal to or greater than the length of the data area, and the linked field or the size field of the record.

Files with a BLOCKSTRUCTURE value of LINKED are always blocked and cannot have a FILESTRUCTURE attribute value of STREAM. Files with BLOCKSTRUCTURE values other than LINKED are blocked only when the value of BLOCKSIZE is greater than the MAXRECSIZE value.

If the BLOCKSTRUCTURE value is EXTERNAL, the system displays a fatal “ILLEGAL SEEK” message when the program attempts to space to any record or attempts to use the SEEK statement to locate a record other than record 0.

If the BLOCKSTRUCTURE value is VARIABLE, VARIABLE2, or VARIABLEOFFSET, the system displays a fatal “ILLEGAL SEEK” message when the program attempts to space backwards to a record or to seek backward to a record other than record 0.

If the BLOCKSTRUCTURE value is VARIABLE, the SIZEVISIBLE attribute can be used. When SIZEVISIBLE is FALSE at file-creation time, the values of MINRECSIZE and MAXRECSIZE do not include the length of the system-maintained record size field, and the I/O subsystem adjusts these attribute values upward before the values are stored in the physical file header. If this adjustment results in the MAXRECSIZE value being greater than the BLOCKSIZE value, the BLOCKSIZE value is also adjusted upward. When you interrogate the MINRECSIZE and MAXRECSIZE attributes, the logical values originally specified before the adjustment are returned. When you interrogate the BLOCKSIZE, the actual size of the block being used is returned.

Port Files

BLOCKSTRUCTURE is meaningful only for READ operations, and only the FIXED and EXTERNAL values can be used.

If the BLOCKSTRUCTURE value is FIXED, the application buffer is blank-filled on the right up to the minimum of: the length of the I/O buffer, READ operation request length, and the ACTUALMAXRECSIZE file attribute value. When the BLOCKSTRUCTURE value is FIXED, the CURRENTRECORDLENGTH attribute value always equals the ACTUALMAXRECSIZE attribute value.

If the BLOCKSTRUCTURE value is EXTERNAL, only the data received by the service provider is placed into the application buffer. The CURRENTRECORDLENGTH value always equals the amount of data received by the service provider.

For either mnemonic value, you can determine if truncation occurred by interrogating the STATE attribute. You can also determine the CURRENTRECORDLENGTH file attribute value if you interrogate the STATE attribute.

The READ operation performance is somewhat slower when the BLOCKSTRUCTURE value is FIXED, because of the blank-filling process.

Attempting to open a port file with an unsupported BLOCKSTRUCTURE value results in an open error.

BUFFERS

Kinds	All
Interrogate	Anytime
Modify	When closed
Type	Integer
Range	0 through 63
Default	See following text
Abstract Value	DEFAULT, MINIMUM, MAXIMUM
Stored Permanently	No
Parameters	1 optional (port files)
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	BNANATIVESERVICE, NETBIOSSESSIONSERVICE

Use the BUFFERS attribute to determine or specify the number of buffers that are to be used when processing the file. If BUFFERS has a value of 0 (zero), the default value is used. The default value of BUFFERS is 3 for disk files with an UPDATEFILE attribute value of TRUE. Remote files have a BUFFERS default value of 1, and all other types of files have a BUFFERS default value of 2. Using more than the default number of buffers adds to the efficiency of the I/O operations of a file only in exceptional cases, such as when printers are used directly.

If you interrogate BUFFERS while the file is open, the number of buffers currently being used is returned.

The BUFFERS attribute is not used by indexed files. All I/O operations for files that are defined as INDEXED or INDEXEDNOTRESTRICTED by the FILEORGANIZATION attribute are done through the SYSTEM/KEYEDIO or SYSTEM/KEYEDIOII support libraries. The BUFFERS attribute is set to 0 (zero) when the file is opened.

If you interrogate the BUFFERS attribute while the file is assigned to an indexed disk file, a value of 0 is returned.

Port Files

The BUFFERS attribute can be used to specify the number of outputs to a remote host that can be outstanding between the host that the port file program is running on and the network processor that the subfile is using. The BUFFERS attribute is not interrogated for subfiles with local dialogues.

This attribute requires a subfile index as a parameter if the MAXSUBFILES value is greater than 1, except when it is used with a file declaration. If it is used in a file declaration or the subfile index specified is zero, then the value of BUFFERS is changed in all allocated subfiles and it becomes the new default for any subfiles that are allocated later, even after increasing MAXSUBFILES.

If the MAXSUBFILES value is greater than 1, the subfile index parameter is required, except when it is used with a file declaration or an ALGOL multiple-file assignment statement.

BUFFERSHARING

Kinds	Disk
Interrogate	Anytime
Modify	When closed
Type	Mnemonic
Range	See following text
Default	NONE
Abstract Value	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

Use the BUFFERSHARING attribute to determine if buffer sharing is being used or specify that buffer sharing is to be initiated. When buffer sharing is initiated, regions of the file can be locked or unlocked. Refer to the *I/O Subsystem Programming Guide* for information about using the locking and unlocking feature.

The allowable mnemonic values for BUFFERSHARING are as follows:

Mnemonic Value	Integer Value	Meaning
NONE	1	Indicates that neither buffer sharing nor record level locking is to be initiated.
SHARED	2	Indicates that when the logical file being defined is opened, the file shares a single set of buffers with other logical files that are accessing the same physical file and have a BUFFERSHARING value of SHARED or EXCLUSIVELYSHARED. If you use this mnemonic, other logical files with a BUFFERSHARING value of NONE can also open the same physical file, but those logical files have unique buffers for their I/O operations.
EXCLUSIVELYSHARED	3	Indicates that when the logical file being defined is opened, the file shares a single set of buffers with other logical files that are accessing the same physical file and have a BUFFERSHARING value of SHARED or EXCLUSIVELYSHARED. If you use this mnemonic, other logical files with a BUFFERSHARING value of NONE cannot open the same physical file until all the logical files that are sharing buffers are closed.

If you set the BUFFERSHARING value to SHARED or EXCLUSIVELYSHARED, you should be aware of the following information:

- Variable record sized files are not supported, so do not set the BLOCKSTRUCTURE file attribute value to any value other than FIXED.
- If you set the FILESTRUCTURE file attribute value to ALIGNED180 for the logical file, do not set the BLOCKSIZE file attribute value to a value that is different from the value stored in the disk file header.
- If the FILESTRUCTURE value is STREAM or BLOCKED and your file is the first logical file to open the physical file, the buffer size is calculated as your file opens. Once the buffer size is calculated, all other logical files that share that buffer pool use that buffer size.
- The MAXRECSIZE value must be equal to the stored MAXRECSIZE value in the disk file header.
- The UPDATEFILE file attribute must be set to FALSE.
- The FILEORGANIZATION file attribute value must be NOTRESTRICTED.
- The LASTRECORD file attribute value is updated after every WRITE operation that extends the end of the file and is coordinated among all logical files that have the same physical file open and have a BUFFERSHARING value of SHARED or EXCLUSIVELYSHARED.

  The end of file stored on disk in the disk file header is extended to the value of LASTRECORD (when the end of file is currently before the LASTRECORD value) under the following circumstances:

  - At the end of a WRITE operation that specifies the SYNCHRONIZE option.
  - At the end of a WRITE operation when the SYNCHRONIZE file attribute value is OUT.
  - After the completion of a physical WRITE operation is handled that caused a new area to be added at or after the end of the file.
  - At the start of an I/O operation that is other than a serial WRITE operation, but that follows a sequence of serial WRITE operations that extended the file.
  - When the file is closed.
- No checkpoints can be taken by a program that has a file open with buffer sharing.
- Unwritten file space added by writing beyond the end of file from the subject logical file is unconditionally scrubbed with zeros as long as the PROTECTION file attribute value is other than PROTECTED. As a result, the CLEARAREAS file attribute needs to be set to TRUE only if it is required that unwritten file space added by (or in unwritten areas touched by) logical files with a BUFFERSHARING value of NONE be scrubbed with zeros.

  Note: *When the PROTECTION value is PROTECTED, some unwritten file space might get scrubbed with the binary pattern used by the PROTECTED mechanism to detect the end of file.*

- Unallocated file areas are not allocated until the point where a physical WRITE operation is to be issued to the area. Reading from an unallocated area returns zeros and does not allocate the area.
- Writing beyond the end of file to a CRUNCHED file is not allowed.
- Your logical file immediately “sees” any extensions to the end of file made by any other logical file that has the file open and has a BUFFERSHARING value of SHARED or EXCLUSIVELYSHARED. However, extensions to the end of file made by logical files that have a BUFFERSHARING value of NONE are not “seen” by the current shared buffer pool in use by logical files that have a BUFFERSHARING value of SHARED or EXCLUSIVELYSHARED.

If a physical file is opened by a logical file with a BUFFERSHARING value of NONE and that physical file is already being used by a file that has a BUFFERSHARING value of EXCLUSIVELYSHARED, the OPEN operation waits until the file with the BUFFERSHARING value of EXCLUSIVELYSHARED is closed. If the OPEN operation was initiated by an OPEN statement with the AVAILABLE option, an OPENWITHBUFFERSHARINGRSLT (250) OPEN error is returned.

If a physical file is opened by a logical file with a BUFFERSHARING value of EXCLUSIVELYSHARED and the physical file is already being used by a file that has a BUFFERSHARING value of NONE, the OPEN operation waits until the file with the BUFFERSHARING value of NONE is closed. If the OPEN operation was initiated by an OPEN statement with the AVAILABLE option, an OPENWITHOUTBUFFERSHARINGRSLT (252) OPEN error is returned.

Note: *When a COPY operation is being performed on a physical file that is currently being used by a logical file with a BUFFERSHARING value of EXCLUSIVELYSHARED, Library Maintenance reads the physical file through the shared buffer pool.*

When a permanent disk file is open and data is added beyond the current end of file, and the system-wide DISKSCRUB option is set or the disk file attribute CLEARAREAS is TRUE, considerable performance improvements are possible when the BUFFERSHARING value is EXCLUSIVELYSHARED.

Direct I/O files must have the BUFFERSHARING value set to NONE. However, direct I/O files can use the record locking feature. Refer to the *I/O Subsystem Programming Guide* for information about using the locking feature.

When the BUFFERSHARING value is NONE and the FILESTRUCTURE value is ALIGNED180 or BLOCKED, writing beyond the end of file to a CRUNCHED file is allowed within the last block.

BUFFERSIZE

Kinds	All except port and remote
Interrogate	Anytime
Modify	When closed
Type	Integer
Range	0 through 65535 (Limited further by other factors.)
Default	See following text
Abstract Value	DEFAULT, MINIMUM, MAXIMUM
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Interrogate the BUFFERSIZE attribute to determine the number of words in memory that each buffer area occupies. The value is the maximum possible size of an I/O transfer into and out of memory from and to the physical I/O device. These buffer areas are not allocated when no physical I/O device is involved—for example, remote and port files. The total memory area used by the buffers of a program is determined by multiplying the BUFFERSIZE value by the BUFFERS value.

The BUFFERSIZE value is used to allocate buffers when a file is opened only if the file is a disk file and the FILESTRUCTURE attribute value is STREAM or BLOCKED. The allocated size is not necessarily the same number of words as the BUFFERSIZE value because of the need to accommodate an integral number of sectors or blocks, as appropriate to the FILESTRUCTURE value, in each buffer. For all other files, buffers can accommodate only one block.

In general, you do not need to specify a value for this attribute. The default value that is assigned by the system or by the site administrator is adequate. Refer to the information about the BUFFERGOAL factor of the SF (Set Factor) system command in the *System Commands Reference Manual*.

However, if your system is experiencing performance and memory utilization problems, the following possible solutions might solve the problem. The potential of these possible solutions is dependent on the characteristics of the disk device, of the program, and of the other programs that are running at the same time. In particular, a program that fills memory with an excess of buffer space might degrade total system performance, including its own performance.

If the file you are creating or accessing . . .	**Then specify . . .**
Is shorter than the default BUFFERSIZE value	An appropriate value for the AREALENGTH or AREASIZE file attribute. This action causes the buffer allocation to be reduced and causes the allocated BUFFERSIZE value to become smaller. If you control the AREASIZE of a short file, you save disk and memory space.
Is a large randomly accessed file	A BUFFERSIZE value that is as small as one record plus two sectors when a file has a FILESTRUCTURE file attribute value of STREAM, or specify a BUFFERSIZE value the size of one block when a file has a FILESTRUCTURE value of BLOCKED.
Is a large serially accessed file with little processing	A BUFFERSIZE value as large as possible, up to the size of an area.

If you interrogate the attribute while the file is open, the actual buffer size is returned. If you interrogate the attribute at any other time, the program-specified value or a 0 (zero) if the attribute has not been explicitly set is returned. For a backup file, the BUFFERSIZE value is the size of the buffer used for the physical tape or disk file. For remote and port files, the value returned is 0 (zero). For direct I/O files, the value returned is 0, because the buffer size is provided by the user program, not the I/O subsystem.

CARRIAGECONTROL

Kinds	Printer
Interrogate	Anytime
Modify	When closed
Type	Mnemonic
Range	See following text
Default	STANDARD
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Use the CARRIAGECONTROL attribute to specify how the first character in the record of a printer file is to be interpreted when the record is printed.

Nonstandard carriage control is allowed only for character-oriented or EBCDIC printer files. A character-oriented printer file has a FRAMESIZE value other than 48 or a UNITS value of CHARACTERS. An EBCDIC printer file has an INTMODE value of EBCDIC.

The mnemonic values and meanings associated with CARRIAGECONTROL are as follows:

Mnemonic Value	Integer Value	Meaning	
CTLASA	1	The line is printed after the carriage motion has been completed, according to the first character of the record, as follows:	
		Character	**Carriage Motion**
		A space	Single spacing
		0 (zero)	Double spacing
		--	Triple spacing
		+	No carriage motion
		1	Skip to channel 1
		2	Skip to channel 2
		3	Skip to channel 3
		4	Skip to channel 4
		5	Skip to channel 5
		6	Skip to channel 6

Mnemonic Value	Integer Value	Meaning	
		7	Skip to channel 7
		8	Skip to channel 8
		9	Skip to channel 9
		It is possible to skip to channel 10, 11, or 12. There are, however, no graphic EBCDIC characters that can be used as the first character for those cases. To skip to channel 10, the first 8 bits should be 48"FA", which is a logical extension of the EBCDIC number set. For channel 11, use 48"FB" and for channel 12, use 48"FC".	
CTL360	2	The first character controls carriage motion using the various fields in the character as follows:	
		Field	**Action**
		[0:1]	If this field is 1, printing occurs before carriage control is performed.
		[1:1]	If this field is 1, no printing occurs, only carriage control is performed.
		[2:1]	Ignored
		[6:4]	The value of this field is either the channel number (if skipping to a channel) or the line count (if spacing to a line) according to field [7:1].
		[7:1]	If this field is 1, a skip to a channel is the carriage control; otherwise, the printer is spaced the number of spaces specified by field [6:4].
STANDARD	0	The first character of the record has no meaning and is treated as valid data. Carriage control is controlled by using the [<I/O option or carriage control>] syntax of ALGOL and the ALGOL $ WRITEAFTER compiler control option, and the BEFORE/AFTER ADVANCING syntax of COBOL74 or COBOL85.	

CCSVERSION

Kinds	Disk
Interrogate	Anytime
Modify	When unassigned
Type	Mnemonic
Range	See following text
Default	NOTSET
Abstract Value	DEFAULT
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

When you create a file, you can use the CCSVERSION attribute to specify the rules to be used to process the character data in the file. For example, the rules by which characters of the file should be collated or which lowercase letters should be converted to uppercase letters. The value you specify is stored in the header of the file and can be used by other programs that open the file.

For a complete discussion of the concept of CCSVERSION and the rules associated with a particular CCSVERSION, refer to the *MultiLingual System (MLS) Administration, Operations, and Programming Guide*.

The CCSVERSION attribute has the following mnemonic and integer values:

Mnemonic Value	Integer Value	Meaning
ARABIC20	20	Use the rules of the Arabic ccsversion (previously known as “Version 20”).
ASERIESNATIVE	0	Use the rules of the default ccsversion of the MCP environment.
BRAZILIAN	45	Use the rules of the Brazilian ccsversion.
CANADAEBCDIC	74	Use the rules of the Canada EBCDIC ccsversion.
CANADAGP	75	Use the rules of the Canada general purpose ccsversion.
CZECHOSLOVAKIA	53	Use the rules of the Czechoslovakia ccsversion.
FRANCE	35	Use the rules of the French ccsversion.
HUNGARIAN	85	Use the rules of the Hungarian ccsversion.
LATINGREEK	49	Use the rules of the Latin-Greek ccsversion.
NORWAY	71	Use the rules of the Norwegian ccsversion.

Mnemonic Value	Integer Value	Meaning
POLISH	86	Use the rules of the Polish ccsversion.
ROMANIAN	96	Use the rules of the Romanian ccsversion.
RUSSIAN	88	Use the rules of the Russian ccsversion.
SPANISH	98	Use the rules of the Spanish ccsversion.
SWEDISH1	99	Use the rules of the Swedish ccsversion.
SWISS	64	Use the rules of the Swiss ccsversion.
TURKISH	80	Use the rules of the Turkish ccsversion.

The following table identifies additional values that can be returned by logical I/O, but cannot be assigned:

Mnemonic Value	Integer Value	Meaning
NOTSET	-1	The CCSVERSION has not been specified.
INVALIDCCSVERSION	-6	The CCSVERSION attribute has been set, but a condition exists that could cause the program to behave incorrectly if it attempts to process data read from the file according to the CCSVERSION value of the physical file. Specifically, the value is returned whenever logical I/O translation of the data in the file is occurring or whenever the EXTMODE value of the file is inconsistent with the CCSVERSION value of the file.

The following actions are taken when a new file is created:

- If a CCSVERSION value is specified and is found in the SYSTEM/CCSFILE, the INTMODE attribute value you specified or the compiler specified is overridden by the value of the coded character set number associated with the CCSVERSION value found in the SYSTEM/CCSFILE. If a CCSVERSION value is specified and is not found in the SYSTEM/CCSFILE, the INTMODE value is not changed.
- If the EXTMODE attribute value is not specified, the EXTMODE value defaults to the INTMODE value.
- If logical I/O translation will occur and a value is specified for CCSVERSION, a CCSVERSION attribute error is issued and the CCSVERSION value is not written to the physical file. If the CCSVERSION attribute is interrogated subsequently while the file is assigned, the INVALIDCCSVERSION value (-6) is returned. If translation will not occur and a CCSVERSION value is specified, that value is written to the disk file header even if that value is not found in the SYSTEM/CCSFILE.

The following actions are taken when an existing file is opened:

- If the physical file CCSVERSION value is found in the SYSTEM/CCSFILE and the EXTMODE value does not match the coded character set number associated with the CCSVERSION, the I/O subsystem treats CCSVERSION as invalid and the INVALIDCCSVERSION value (-6) is returned if the CCSVERSION attribute is interrogated while the file is assigned. The value in the file header is not changed. If the physical file CCSVERSION value is not found in the SYSTEM/CCSFILE, no actions are taken.
- If logical I/O translation will occur, the logical file CCSVERSION value is treated as invalid and the INVALIDCCSVERSION value (-6) is returned if the CCSVERSION attribute is interrogated while the file is assigned. The value in the file header is not changed.

If you interrogate the CCSVERSION attribute when a file is assigned, the physical file value is returned except in those cases previously described when INVALIDCCSVERSION is returned.

Considerations When Using the WFL *ALTER* Statement with CCSVERSION

You can modify the CCSVERSION value of a permanent disk file with the WFL *ALTER* statement, subject to the following conditions:

- When the CCSVERSION attribute is specified in the <alter attribute list> and the EXTMODE attribute is not, the CCSVERSION value is modified in the file and the EXTMODE value is modified to the value compatible with the CCSVERSION value specified.
- When the EXTMODE attribute is specified in the <alter attribute list> and the CCSVERSION attribute is not, the EXTMODE value is modified in the file. The CCSVERSION value is unaffected if it is compatible with the EXTMODE value specified; otherwise, the CCSVERSION attribute reflects a value of NOTSET (-1) in the file.
- When both the CCSVERSION and EXTMODE attributes are specified in the <alter attribute list> and are compatible with one another, then both values are modified in the file. Incompatible values of CCSVERSION and EXTMODE cause the ALTER statement to fail and an error message results.
- For character-oriented files, if modifying the value of CCSVERSION or EXTMODE causes the character size to change, the ALTER statement fails and an error message results.

Refer to the *Work Flow Language (WFL) Programming Reference Manual* for details on the ALTER statement.

CDCOPIES

Kinds	CD
Interrogate	Anytime
Modify	When Closed
Type	Integer
Range	1 to 255
Default	1
Abstract Value	None
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not Supported
Port Services	Not Supported

Use the CDCOPIES attribute to specify how many copies of a CD-R disc are to be burned with the attribute CDCOPIES. The default value of CDCOPIES is 1.

When data is copied to a CD-R disc, it is first copied from the source media to a temporary disk file. The data in the temporary disk file is formatted exactly as it is to appear on the CD-ROM. The temporary disk file is then copied to a CD-R drive. CDCOPIES causes the disk file to be copied to the CD-R drive as many times as you specify, without having to re-copy the data from the source media.

Only library maintenance uses the CDCOPIES attribute.

For further information about the CDCOPIES statement, refer to the *Work Flow Language (WFL) Programming Reference Manual*.

CENSUS

Kinds	Remote, port, or port subfile
Interrogate	Anytime
Modify	Never
Type	Integer
Range	Not applicable
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	BNANATIVESERVICE, NETBIOSSESSIONSERVICE, OSINATIVESERVICE, OSISESSIONSERVICE, BASICSERVICE

Interrogate the CENSUS attribute to determine the number of messages currently in the input queue of a port file, port subfile, or remote file. If the file is a remote file, the value returned is the total number of messages queued for that file.

If the file is a port file and no subfile index is given, the value returned is the sum of the messages queued for all subfiles in that file. If a subfile index is given, the value returned is the number of messages queued for the specified subfile only.

If the file is an OSINATIVESERVICE or OSISESSIONSERVICE port file and the PORTSEGMENTIO file attribute is TRUE, the value returned is the number of segments in the queue. If no subfile index is given, the value returned is the sum of the segments queued for all subfiles in that file.

The range value of the CENSUS attribute should be in the range of the MAXCENSUS attribute. However, it is possible for the CENSUS attribute to have a value greater than the maximum value of the MAXCENSUS attribute, if the Stop Sending message is received after sending data that caused the range to exceed the maximum MAXCENSUS value.

For services that are not message-oriented, use the FRAMESIZECENSUS attribute.

CHANGEDSUBFILE

Kinds	Port
Interrogate	Anytime
Modify	Never
Type	Integer
Range	0 through the MAXSUBFILES attribute value
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	All

Interrogate the CHANGEDSUBFILE attribute to obtain the index of a subfile that has a CHANGEEVENT attribute that is currently in a happened state. If the CHANGEEVENT attribute of more than one subfile is in the happened state, the index of the subfile that had the CHANGEEVENT value changed to the happened state first is returned. If no subfile CHANGEEVENT is in the happened state, a value of 0 (zero) is returned.

The CHANGEDSUBFILE attribute continues to return the same index until the CHANGEEVENT value for that subfile is reset. The CHANGEEVENT value is reset when the FILESTATE attribute of that subfile is interrogated.

CHANGEDSUBFILE is implemented as a time ordered list of subfiles that have a value of Happened for the ChangeEvents parameter. Then the filestate value is interrogated, the event value is reset and the subfile is removed from the list.

When the state of the file changes and the subport is not in the list, the subport is inserted at the end of the list. Any subsequent changes to the state of the file before the Filestate parameter is interrogated will not change the position of the subport in the list. When Filestate is interrogated, the current state value is returned, the ChangeEvent value for the subport is reset, and the subport is removed from the list. This cycle starts again whenever there is a change is the state value.

CHANGEEVENT

Kinds	Port or port subfile
Interrogate	Anytime
Modify	Never
Type	Event
Range	Not applicable
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	All

Use the CHANGEEVENT attribute to determine when the FILESTATE attribute value for a port file or port subfile has changed. The CHANGEEVENT value of a subfile is set whenever the subfile FILESTATE value changes. The CHANGEEVENT value of a subfile is reset when the FILESTATE attribute of the subfile is interrogated. The value of FILESTATE may change several times between the setting of CHANGEEVENT and the interrogation of FILESTATE. No history of these changes to FILESTATE is kept. The CHANGEEVENT value of the file remains in a happened state as long as the CHANGEEVENT value of any of its subfiles is also in a happened state. You can use the CHANGEDSUBFILE attribute to determine which subfile the FILESTATE attribute has changed. The CHANGEEVENT value for the file is reset after all subfile CHANGEEVENT values are reset.

Because CHANGEEVENT is a read-only attribute, it should not be passed as a parameter to any of the following ALGOL statements: CAUSEANDRESET, FIX, FREE, LIBERATE, PROCURE, RESET, SET, or WAITANDRESET. If the CHANGEEVENT attribute is used as a parameter to these statements, an error message is not issued, and the program performs incorrectly.

The ATTACH and DETACH statements can be used with the CHANGEEVENT attribute; however, up-level attach errors can occur as described in the discussion of interprocess communication in the *Task Management Programming Guide* or the *ALGOL Programming Reference Manual, Volume 1: Basic Implementation*.

CHECKPOINT

Kinds	Disk and printer
Interrogate	Anytime
Modify	When closed
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Set the CHECKPOINT attribute to TRUE to specify that checkpoints are to be taken while the file is printing. Checkpoints are restart points used in the event that printing is interrupted and must be restarted.

The CHECKPOINT attribute is ignored if the printer file is a direct I/O file or if the value of the PRINTDISPOSITION attribute is DIRECTDLP, DIRECTPS, or NOW. If the printer to which the print request is directed is not configured to take checkpoints and the backup file has a CHECKPOINT value of TRUE, checkpoints are taken at 10,000 line intervals.

If the CHECKPOINT value is FALSE, printing restarts at the beginning of the print file unless the device has been configured to take checkpoints unconditionally. For information about setting and using checkpoints, refer to the *Print System Guide*.

CLEARAREAS

Kinds	Disk
Interrogate	Anytime
Modify	When closed
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Abstract Value	DEFAULT
Stored Permanently	Yes
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Set the CLEARAREAS attribute to TRUE to specify that zeros or NULLS (all bits off) be written to each area as the area is allocated. Such an action is known as disk scrubbing. As a result of disk scrubbing, when a program reads a record that has never been written and the file was created with the CLEARAREAS value of TRUE, the returned record contains zeros or NULLS.

If CLEARAREAS is FALSE, disk scrubbing occurs on a file only if it is a RELATIVE file, or, on all files, if the system security option DISKSCRUB is SET. If disk scrubbing is not in effect for a file when an area is allocated and a program reads a record that has never been written, the content of the returned record is unpredictable.

COMPRESSING

Kinds	Port subfile or tape
Interrogate	Anytime
Modify	Never
Type	Boolean
Range	TRUE, FALSE
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	Tape
Parameters	1 optional
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	BNANATIVESERVICE, BASICSERVICE, NETBIOSSESSIONSERVICE, OSINATIVESERVICE, OSISESSIONSERVICE

Interrogate the COMPRESSING attribute to determine the following information:

File KIND Value	Information
PORT	If the data being sent by this port subfile is compressed during transmission. Compression could improve efficiency by stripping contiguous characters such as blanks, NULs, and ESCs.
TAPE	If the assigned tape was created with compression.

Port Files

Compression is supported only by BNANATIVESERVICE and BASICSERVICE. For services that do not support compression or when a provider determines that the benefit of compressing data is ineffective or detrimental, the COMPRESSING attribute always returns the value FALSE.

Use the COMPRESSIONCONTROL attribute to specify whether you or the I/O subsystem controls compression.

Tape Files

If the tape drive is unassigned, FALSE is always returned. If the tape drive is assigned, FALSE is returned if the file is mounted on a tape drive that is not compression capable or the file is on a volume that indicates the data on the tape is not compressed.

Refer to the *I/O Subsystem Programming Guide* for more information about compression.

COMPRESSION

Kinds	Port subfile
Interrogate	Anytime
Modify	When closed, or when an open operation is completed
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	BNANATIVESERVICE, BASICSERVICE

Changing the COMPRESSION value implicitly changes the COMPRESSIONREQUESTED attribute value to the same value.

If you interrogate the COMPRESSION attribute, the current value of the COMPRESSING attribute is returned if the FILESTATE value of the subfile is OPENED, BLOCKED, or SHUTTINGDOWN; otherwise, the current COMPRESSIONREQUESTED value is returned.

If the MAXSUBFILES value is greater than 1, the COMPRESSION attribute requires a subfile index as a parameter, except when it is used with a file declaration or an ALGOL multiple-file assignment statement.

Use of the COMPRESSING, COMPRESSIONCONTROL, and COMPRESSIONREQUESTED attributes is preferred.

Refer to the *I/O Subsystem Programming Guide* for more information about compression.

COMPRESSIONCONTROL

Kinds	Port subfile or tape
Interrogate	Anytime
Modify	When closed, or when an open operation is completed (port subfile). When unassigned (tape).
Type	Mnemonic
Range	USER, SYSTEM
Default	USER (port) See following text (tape)
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	BNANATIVESERVICE, BASICSERVICE

Use the COMPRESSIONCONTROL attribute in one of the following ways:

File KIND Value	Use
PORT	To specify whether your program or the I/O subsystem controls the stripping of contiguous characters such as blanks, NULs, and ESCs for a subfile. Compressing data that is passed between dialogue endpoints can improve efficiency.
TAPE	When you are creating a file, to specify if the operating system determines if compression will be used based on the COMPRESSIONREQUESTED file attribute value or on the value stored in the tape label of the tape volume.

The mnemonic values and integer values for this attribute follow:

Mnemonic Value	Integer Value
SYSTEM	1
USER	0

Port Files

During dialogue establishment, both hosts negotiate for support of compression. If the COMPRESSIONCONTROL value is SYSTEM and compression can be supported by both endpoints, the system selectively sets the COMPRESSING attribute based on the evaluated benefit of compressing data in the current network state. If the COMPRESSIONCONTROL value is USER and compression can be supported by both endpoints, compression is controlled by the value of the COMPRESSIONREQUESTED attribute. In this case, the COMPRESSING attribute reflects the COMPRESSIONREQUESTED value.

This attribute requires a subfile index as a parameter if the MAXSUBFILES value is greater than 1, except when it is used with a file declaration. If it is used in a file declaration or the subfile index specified is zero, then the value of DIALOGPRIORITY is changed in all allocated subfiles and it becomes the new default for any subfiles that are allocated later, even after increasing MAXSUBFILES.

Tape Files

The following table describes the action that is initiated when you create a file and specify the identified mnemonic:

Mnemonic Value	Action
SYSTEM	The operating system performs compression based on the compression value of the volume tape label. The compression value of the tape volume can be controlled by using the SN (Serial Number) or PG (Purge) system commands or can be controlled programmatically when compression is requested by the first file on the volume. Refer to the *System Commands Reference Manual* for information about using these two commands.
USER	The operating system performs compression based on the value of the COMPRESSIONREQUESTED file attribute.

If you do not specify a value for COMPRESSIONCONTROL, the value defaults to one of the following values:

Current State of the File	Default Value
Unassigned to a unit	USER
Assigned to a unit	SYSTEM

If you are creating a multifile tape volume, all files are compressed if the compression value of the tape volume indicates compression is to be used.

If your program creates a multivolume file, all volumes will have the same compression specification as the first volume of the set.

Refer to the *I/O Subsystem Programming Guide* for more information about compression.

COMPRESSIONREQUESTED

Kinds	Port subfile or tape
Interrogate	Anytime
Modify	When closed, or when an open operation is completed (port subfile). When unassigned (tape).
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	BNANATIVESERVICE, BASICSERVICE

Set the COMPRESSIONREQUESTED attribute to TRUE if the COMPRESSIONCONTROL file attribute value is set to USER for the subfile or file.

The following actions are taken when the COMPRESSIONREQUESTED value is TRUE:

File KIND Value	Action
PORT	Contiguous characters such as blanks, NULs, and ESCs are stripped from data during transmission.
TAPE	The data is compressed.

Port Files

If the COMPRESSIONCONTROL value is USER, and compression is supported by both hosts, the value of the COMPRESSING attribute changes to the value of the COMPRESSIONREQUESTED attribute. Otherwise, the COMPRESSIONREQUESTED attribute does not affect the COMPRESSING value.

If you interrogate the COMPRESSION attribute, the COMPRESSING value is returned if a subfile can be written to. Otherwise, the COMPRESSIONREQUESTED value is returned.

Tape Files

If the COMPRESSIONCONTROL value is USER and the COMPRESSIONREQUESTED value is TRUE, the operating system compresses data when the file being created is assigned to a compression capable unit.

If you set the value to FALSE or allow the value to default to FALSE, and the COMPRESSIONCONTROL value is USER, the output data is not compressed.

Refer to the *I/O Subsystem Programming Guide* for more information about compression.

COPYDESTDATE

Kinds	Disk
Interrogate	When assigned
Modify	Never (see following exception)
Type	Integer
Range	70001 through 135365
Default	CREATIONDATE
Abstract Value	Not applicable
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not supported

Interrogate the COPYDESTDATE attribute to determine when the physical file was copied to disk.

When library maintenance copies a file to disk, it sets the COPYDESTDATE, COPYDESTTIME, and COPYDESTTZ values to the current date. At that time, library maintenance resets BACKUPDATE, BACKUPTIME, and BACKUPTZ and COPYSOURCEDATE, COPYSOURCETIME, and COPYSOURCETZ respectively to the CREATIONDATE, CREATIONTIME, and CREATIONTZ values of the file.

The value returned by COPYDESTDATE is a decimal integer. When you add the integer 1900000 to the value returned, the result is the date in the form YYYYDDD, where YYYY and DDD represent the year and date, respectively, in Julian form.

The COPYDESTDATE attribute is used in conjunction with the COPYDESTTIME and COPYDESTTZ attributes. See “COPYDESTTIME” and “COPYDESTTZ” for more information.

The COPYDESTDATE attribute can be modified by a valid MCS program if that program previously set the HSFILECOPY file attribute to TRUE.

COPYDESTDATEUT

Kinds	Disk
Interrogate	When assigned
Modify	Never
Type	Integer
Range	70001 through 135365
Default	CREATIONDATE
Abstract Value	Not applicable
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not supported

Interrogate the COPYDESTDATEUT attribute to determine the COPYDESTDATE value with respect to UTC time instead of the COPYDESTTZ time zone. See "COPYDESTDATE" file attribute for more information.

When you interrogate the COPYDESTDATEUT attribute, you receive an error if the time zone attribute, COPYDESTTZ, is 0 (zero). See "COPYDESTTZ" for more information.

COPYDESTTIME

Kinds	Disk
Interrogate	When assigned
Modify	Never (see following exception)
Type	Integer
Range	0 through 86399999999
Default	CREATIONTIME
Abstract Value	Not applicable
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not supported

Interrogate the COPYDESTTIME attribute to determine the time of day, in microseconds since midnight, that is associated with the COPYDESTDATE attribute value. The COPYDESTTIME value is updated when the COPYDESTDATE value is updated.

The COPYDESTTIME attribute can be modified by a valid MCS program if that program previously set the HSFILECOPY file attribute to TRUE.

COPYDESTTIMEUT

Kinds	Disk
Interrogate	When assigned
Modify	Never
Type	Integer
Range	0 through 86399999999
Default	CREATIONTIME
Abstract Value	Not applicable
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not supported

Interrogate the COPYDESTTIMEUT to determine the COPYDESTTIME value with respect to UTC time instead of the COPYDESTTZ time zone. See “COPYDESTIME” for more information.

When you interrogate the COPYDESTTIMEUT attribute, you receive an error if the time zone attribute, COPYDESTTZ, is 0 (zero). See “COPYDESTTZ” for more information.

COPYDESTTZ

Kinds	Disk
Interrogate	When assigned
Modify	Never (see following exception)
Type	Integer
Range	0 through 255
Default	CREATIONTZ
Abstract Value	Not applicable
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not supported

Interrogate the COPYDESTTZ attribute to determine the number of the time zone that is associated with the COPYDESTDATE and COPYDESTTIME attribute values. The COPYDESTTZ value is updated when the COPYDESTDATE value is updated.

To determine the meaning of non-zero values of COPYDESTTZ, refer to Appendix G, “Predefined Time Zones.” You can use the MCP library procedure TIMEZONENAME to convert the numeric code of a time zone into the name or abbreviation for that time zone. TIMEZONENAME is documented in the *Master Control Program (MCP) System Interfaces Programming Reference Manual.*

If the integer value of COPYDESTTZ is 0 (zero), then there is not a time zone associated with COPYDESTDATE and COPYDESTTIME.

The COPYDESTTZ value is 0 if the file

- Has CREATIONTZ = 0 and COPYDESTDATE and COPYDESTTIME are equal to CREATIONDATE and CREATIONTIME.
- Resides on a disk family with FAMILY HEADER VERSION = 6. Refer to “Disk File Headers” in Section 1 for more information.
- Was copied to disk and the system did not have a time zone set at the time of the copy.

The COPYDESTTZ attribute can be modified by a valid MCS program if that program previously set the HSFILECOPY file attribute to TRUE.

COPYSOURCEDATE

Kinds	Disk
Interrogate	When assigned
Modify	Never (see following exception)
Type	Integer
Range	70001 through 135365
Default	CREATIONDATE
Abstract Value	Not applicable
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not supported

Interrogate the COPYSOURCEDATE attribute to determine the last date the physical file was copied from disk with a WFL ADD or COPY statement that did not specify the & BACKUP option.

When library maintenance copies the file from disk, it sets COPYSOURCEDATE, COPYSOURCETIME, and COPYSOURCETZ to the current date.

Note: *Whenever the file is copied back to disk, library maintenance resets COPYSOURCEDATE, COPYSOURCETIME, and COPYSOURCETZ to the CREATIONDATE, CREATIONTIME, and CREATIONTZ values of the file respectively.*

The value returned by COPYSOURCEDATE is a decimal integer. When you add the integer 1900000 to the value returned, the result is the date in the form YYYYDDD, where YYYY and DDD represent the year and date, respectively, in Julian form.

The COPYSOURCEDATE value is not updated for files on a disk which is not write-enabled.

The COPYSOURCEDATE attribute is used in conjunction with the COPYSOURCETIME and COPYSOURCETZ attributes. See “COPYSOURCETIME” and “COPYSOURCETZ” for more information.

The COPYSOURCEDATE attribute can be modified by a valid MCS program if that program previously set the HSFILECOPY file attribute to TRUE.

COPYSOURCEDATEUT

Kinds	Disk
Interrogate	When assigned
Modify	Never
Type	Integer
Range	70001 through 135365
Default	CREATIONDATE
Abstract Value	Not applicable
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not supported

Interrogate the COPYSOURCEDATEUT attribute to determine the COPYSOURCEDATE value with respect to UTC time instead of the COPYSOURCETZ time zone. See “COPYSOURCEDATE” for more information.

When you interrogate the COPYSOURCEDATEUT attribute, you receive an error if the time zone attribute, COPYSOURCETZ, is 0 (zero). See “COPYSOURCETZ” for more information.

COPYSOURCETIME

Kinds	Disk
Interrogate	When assigned
Modify	Never (see following exception)
Type	Integer
Range	0 through 86399999999
Default	CREATIONTIME
Abstract Value	Not applicable
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not supported

Interrogate the COPYSOURCETIME attribute to determine the time of day, in microseconds since midnight, associated with the COPYSOURCEDATE attribute value. The COPYSOURCETIME value is updated when the COPYSOURCEDATE value is updated.

The COPYSOURCETIME attribute can be modified by a valid MCS program if that program previously set the HSFILECOPY file attribute to TRUE.

COPYSOURCETIMEUT

Kinds	Disk
Interrogate	When assigned
Modify	Never
Type	Integer
Range	0 through 86399999999
Default	CREATIONTIME
Abstract Value	Not applicable
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not supported

Interrogate the COPYSOURCETIMEUT attribute to determine the COPYSOURCETIME value with respect to UTC time instead of the COPYSOURCETZ time zone. See “COPYSOURCETIME” for more information.

When you interrogate the COPYSOURCETIMEUT attribute, you receive an error if the time zone attribute, COPYSOURCETZ, is 0 (zero). See “COPYSOURCETZ” for more information.

COPYSOURCETZ

Kinds	Disk
Interrogate	When assigned
Modify	Never (see following exception)
Type	Integer
Range	0 through 255
Default	CREATIONTZ
Abstract Value	Not applicable
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not supported

Interrogate the COPYSOURCETZ attribute to determine the number of the time zone that is associated with the COPYSOURCEDATE and COPYSOURCETIME attribute values. The COPYSOURCETZ value is updated when the COPYSOURCEDATE value is updated.

To determine the meaning of non-zero values of COPYSOURCETZ, refer to Appendix G, "Predefined Time Zones." You can use the MCP library procedure TIMEZONENAME to convert the numeric code of a time zone into the name or abbreviation for that time zone. TIMEZONENAME is documented in the *Master Control Program (MCP) System Interfaces Programming Reference Manual.*

If the integer value of COPYSOURCETZ is 0 (zero), then there is not a time zone associated with COPYSOURCEDATE and COPYSOURCETIME.

The COPYSOURCETZ value is 0 if the file

- Has CREATIONTZ = 0 and COPYSOURCEDATE and COPYSOURCETIME are equal to CREATIONDATE and CREATIONTIME.
- Resides on a disk family with FAMILY HEADER VERSION = 6. Refer to "Disk File Headers" in Section 1 for more information.
- Was last copied from disk where the system did not have a time zone set.

The COPYSOURCETZ attribute can be modified by a valid MCS program if that program previously set the HSFILECOPY file attribute to TRUE.

CREATEPASSWORD

Kinds	Disk
Interrogate	Anytime
Modify	When closed
Type	Character string
Range	Not applicable
Default	Null string
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	None
FTAM File Access	Supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

When you are creating a File Transfer, Access, and Management (FTAM) file on a remote host, use the CREATEPASSWORD attribute to specify a password string. The syntax of the CREATEPASSWORD attribute is the same as that for the USERCODE attribute with the following exception: only a single piece (or node) of information can be specified. For further details, refer to the USERCODE file attribute.

The value that you supply is used as the CREATEPASSWORD parameter of the FTAM CREATE protocol dialogue unit.

CREATIONDATE

Kinds	Disk, CD, or tape
Interrogate	When assigned
Modify	Never (see following exception)
Type	Integer
Range	70001 through 135365
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	Disk, CD, or tape
Parameters	None
FTAM File Access	Supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Interrogate the CREATIONDATE attribute to determine the date when the physical file was first opened for creation.

The value returned by CREATIONDATE is a decimal integer. When you add the integer 1900000 to the value returned, the result is the date in the form YYYYDDD, where YYYY and DDD represent the year and day, respectively, in Julian form.

When a file is created, the current date of the system is used as the CREATIONDATE value for the file. The values returned by the CREATIONDATE attribute are unaffected by library maintenance.

The CREATIONDATE attribute is used in conjunction with the CREATIONTIME and CREATIONTZ attributes. See “CREATIONTIME” and “CREATIONTZ” for more information.

The CREATIONDATE attribute can be modified by a valid MCS program if that program previously set the HSFILECOPY file attribute to TRUE.

CREATIONDATEUT

Kinds	Disk
Interrogate	When assigned
Modify	Never
Type	Integer
Range	70001 through 135365
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	Disk
Parameters	None
FTAM File Access	Supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Interrogate the CREATIONDATEUT attribute to determine the CREATIONDATE value with respect to UTC time instead of the CREATIONTZ time zone. See "CREATIONDATE" for more information.

When you interrogate the CREATIONDATEUT attribute, you receive an error message if the time zone attribute, CREATIONTZ, is 0 (zero). See "CREATIONTZ" for more information.

CREATIONTIME

Kinds	Disk or CD
Interrogate	When assigned
Modify	Never (see following exception)
Type	Integer
Range	0 through 86399999999
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	Disk or CD
Parameters	None
FTAM File Access	Supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Interrogate the CREATIONTIME attribute to determine the time of day, in microseconds since midnight, associated with the CREATIONDATE attribute value.

The CREATIONTIME attribute is used in conjunction with the CREATIONDATE attribute. For more information, refer to the description of the CREATIONDATE attribute.

The CREATIONTIME attribute can be modified by a valid MCS program if that program previously set the HSFILECOPY file attribute to TRUE.

CREATIONTIMEUT

Kinds	Disk
Interrogate	When assigned
Modify	Never
Type	Integer
Range	0 through 86399999999
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	Disk
Parameters	None
FTAM File Access	Supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Interrogate the CREATIONTIMEUT attribute to determine the CREATIONTIME value with respect to UTC time instead of the CREATIONTZ time zone. See “CREATIONTIME” for more information.

When you interrogate the CREATIONTIMEUT attribute, you receive an error if the time zone attribute, CREATIONTZ, is 0 (zero). See “CREATIONTZ” for more information.

CREATIONTZ

Kinds	Disk
Interrogate	When assigned
Modify	Never (see following exception)
Type	Integer
Range	0 through 255
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	Yes
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Interrogate the CREATIONTZ attribute to determine the number of the time zone that is associated with the CREATIONDATE and CREATIONTIME attribute values. The CREATIONTZ value is updated when the CREATIONDATE value is updated.

To determine the meaning of non-zero values of CREATIONTZ, refer to Appendix G, "Predefined Time Zones." You can use the MCP library procedure TIMEZONENAME to convert the numeric code of a time zone into the name or abbreviation for that time zone. TIMEZONENAME is documented in the *Master Control Program (MCP) System Interfaces Programming Reference Manual.*

If the integer value of CREATIONTZ is 0 (zero), then there is not a time zone associated with CREATIONDATE and CREATIONTIME.

The CREATIONTZ value is 0 if the file

- Was created on a disk family with FAMILY HEADER VERSION = 6.
- Was copied from a disk family with FAMILY HEADER VERSION = 6.
- Was copied from a tape that was created with version 6 headers.
- Was created at a time when the system did not have a time zone set.
- Resides on a disk family with FAMILY HEADER VERSION = 6.

Refer to "Disk File Headers" in Section 1 for more information.

The CREATIONTZ attribute can be modified by a valid MCS program if that program previously set the HSFILECOPY file attribute to TRUE.

CRUNCHED

Kinds	Disk or CD
Interrogate	When assigned
Modify	Never
Type	Boolean
Range	TRUE, FALSE
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Interrogate the CRUNCHED attribute to determine if a permanent disk file was closed with the CRUNCH option. When a disk file is *crunched,* all disk space beyond the end of file—that is, the remainder of the area in which the end of the file lies and any subsequent areas that have been allocated—is returned to the system. If the FILESTRUCTURE value of the file is STREAM, the disk space beyond the sector containing the end of the file is returned to the system and the file cannot be extended. If the FILESTRUCTURE value of the file is not STREAM, the disk space beyond the last sector of the last block is returned to the system and the file can be extended with new records until that block is filled.

A crunched file can be rewritten or updated only when the value of the BLOCKSIZE attribute of the logical file is equal to the physical BLOCKSIZE value; otherwise, the file can only be read.

A file can be crunched only when the CRUNCH system option is set. A file is crunched when it is closed with a CLOSE statement with the CRUNCH option set or if it is a newly created code file or backup file.

A file is not actually crunched until the file is no longer in use, or, if a halt/load occurs, until the file is used again.

CURRENTBLOCK

CURRENTBLOCK is a nonpreferred synonym for CURRENTBLOCKLENGTH. Refer to the description of the CURRENTBLOCKLENGTH attribute.

CURRENTBLOCKLENGTH

Kinds	All except CD and port
Interrogate	Anytime
Modify	Never
Type	Integer
Range	0 through the BLOCKSIZE attribute value
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Interrogate the CURRENTBLOCKLENGTH attribute to determine the size, in FRAMESIZE units, of the block that is currently in use. Normally, this value is the same as the BLOCKSIZE attribute value. The CURRENTBLOCKLENGTH value becomes of interest for a tape file when the system encounters a short block. Your program receives a short block indicator in the STATE attribute word.

When the CURRENTBLOCKLENGTH attribute is interrogated for a file with a FILESTRUCTURE value of STREAM, an attribute error results and a value of 0 (zero) is returned.

When the CURRENTBLOCKLENGTH attribute of a direct file is interrogated, a value of 0 is returned.

CURRENTRECORD

CURRENTRECORD is a nonpreferred synonym for CURRENTRECORDLENGTH. Refer to the description of the CURRENTRECORDLENGTH attribute.

CURRENTRECORDLENGTH

Kinds	All
Interrogate	Anytime
Modify	Never
Type	Integer
Range	0 through the MAXRECSIZE attribute value
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Supported
Heritage Host Services Logical I/O	Supported
Port Services	All

Interrogate the CURRENTRECORDLENGTH attribute to determine the number of FRAMESIZE units in the last record read or written. This number is not necessarily the number of FRAMESIZE units requested by the I/O statement that caused the record to be read or written.

For disk files that have a FILESTRUCTURE attribute value of STREAM and an ANYSIZEIO attribute value of TRUE, a READ or WRITE operation can cross record boundaries, and can end at a point other than a record boundary. In such a case, the CURRENTRECORDLENGTH attribute returns the total number of FRAMESIZE units in all records that were accessed by the last I/O operation.

A disk file with a BLOCKSTRUCTURE attribute value of FIXED and any FILESTRUCTURE attribute value can end in the middle of a record. This situation can happen when a file has been expanded by a program that used a logical MAXRECSIZE value that is different from the MAXRECSIZE value of the permanent file or when your program uses the MAXRECSIZE attribute in this manner. If you want to determine that the end-of-file (EOF) position does not occur on a MAXRECSIZE boundary, divide the FILELENGTH value by the MAXRECSIZE value. Any remainder (MOD) that is a nonzero value indicates the number of frames in the last logical record that occurs before the EOF position. It is recommended that this check be made in any program that accesses a file with a MAXRECSIZE value that is different from the MAXRECSIZE value of the permanent file.

For files other than disk files, if the value of BLOCKSTRUCTURE is FIXED, the CURRENTRECORDLENGTH value equals the MAXRECSIZE value.

The CURRENTRECORDLENGTH value for a disk file with a BLOCKSTRUCTURE value of EXTERNAL and a FILESTRUCTURE value of STREAM is the actual number of FRAMESIZE units transferred by the last I/O operation. This number normally equals the requested number of units, except when the request exceeds the MAXRECSIZE attribute value or would cause the I/O operation to go beyond the end of the file.

For other files with a BLOCKSTRUCTURE value of EXTERNAL, the CURRENTRECORDLENGTH value depends on whether the file is blocked or unblocked.

For an unblocked file, the size of the record is the size of the block; therefore, the CURRENTBLOCKLENGTH value is returned. If the file is a tape file, an operator display terminal (ODT) file, or a remote file, the I/O subsystem determines the actual length to be read. When a short block is encountered, the value is less than the MAXRECSIZE value. For a blocked file, there is no record-size information available other than the number of FRAMESIZE units requested by the last READ or WRITE operation performed by the program, reduced to MAXRECSIZE or the remainder of the block, as necessary. The record size is not reduced because the user record area is not large enough. Therefore, that number is the value returned.

The CURRENTRECORDLENGTH attribute returns a value of 0 (zero) under any of the following conditions:

- The file is open but no I/O action has occurred.
- The file is closed.
- The DUMMYFILE attribute value is TRUE.
- The last operation caused an I/O error.

If the previous action was a SEEK operation, the value of CURRENTRECORDLENGTH is undefined.

Port Files

When a READ operation is invoked, the quantity of data copied into the buffer of your program is the smallest of the following:

- The length specified in the READ statement
- The size of the buffer
- The size of the actual message
- The value of the ACTUALMAXRECSIZE attribute

If the BLOCKSTRUCTURE value is FIXED, the buffer is blank-filled—when necessary—and the CURRENTRECORDLENGTH attribute always returns the ACTUALMAXRECSIZE value. If the BLOCKSTRUCTURE value is EXTERNAL, the buffer is not blank-filled and the CURRENTRECORDLENGTH attribute returns the quantity of data received by the service provider. If truncation occurred, the CURRENTRECORDLENGTH value will be greater than the actual quantity of data copied into the buffer. For either BLOCKSTRUCTURE mnemonic value, interrogate the STATE attribute to determine if truncation occurred.

The CURRENTRECORDLENGTH value can be obtained from the STATE attribute.

When a WRITE operation is invoked, the smallest of the following is returned:

- The length specified in the WRITE statement
- The length of the array passed to the WRITE statement
- The value of the ACTUALMAXRECSIZE attribute

Remote Files

The value of CURRENTRECORDLENGTH is the smaller of the following:

- The size of the actual message
- The value of the MAXRECSIZE attribute

The value is not bounded by the length specified in the READ statement nor by the size of the buffer, although the amount of data actually transferred is bounded by the size of the buffer.

CYCLE

Kinds	Disk or tape
Interrogate	Anytime
Modify	When not in directory (disk) When closed (tape)
Type	Integer
Range	0 through 9999
Default	1
Abstract Value	DEFAULT, MINIMUM, MAXIMUM
Stored Permanently	Disk or tape
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Use the CYCLE attribute in conjunction with the VERSION attribute to specify the different generations of a permanent file. The initial, and default, CYCLE value is 1. The value of the CYCLE attribute can be modified only when the file is closed, except in the case of the creation of a disk file, where CYCLE can be changed any time before the file is entered into the directory. Refer to the description of the PROTECTION attribute for information about forcing an entry into the directory.

Changing the CYCLE attribute to 0 (zero) has two results:

- Both the CYCLE and VERSION attributes change to their default values of 1 and 0, respectively.
- No specific genealogy checking is done when the logical file is first assigned to the permanent file. Thus, the permanent file with the best genealogy is assigned to the logical file. The best genealogy is defined to be the highest CYCLE value and the highest VERSION value of that cycle.

If you modify the CYCLE or VERSION attribute to a value in the acceptable value range, a permanent file is assigned only if a permanent file is found that matches the genealogy as well as the other prerequisites you might have established through other attributes. If the proper file cannot be found and does not match the genealogy, an “UNMATCHED GENEALOGY” message is displayed. The operator can respond by making the file available or by using the FA (File Attribute), NF (No File), or DS (Discontinue) system command. Refer to the *System Operations Guide* for information about using these commands.

If the CYCLE attribute is interrogated when the file is assigned, the value of the physical file is returned.

DATE

DATE is a nonpreferred synonym for *CREATIONDATE*. Refer to the description of the CREATIONDATE attribute.

DENSITY

Kinds	Tape
Interrogate	Anytime
Modify	When closed
Type	Mnemonic
Range	See following text
Default	See following text
Abstract Value	DEFAULT
Stored Permanently	Tape
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Use the DENSITY attribute to specify or determine the media type of a magnetic tape file.

The mnemonic and integer values associated with the DENSITY attribute are as follows:

Mnemonic Value	Integer Value	Media Type	Tape Subsystems
BPI800	0	9-track NRZ reel-to-reel tape	2145, 4125
BPI1600	3	9-track PE reel-to-reel tape	2145, 4125
BPI6250	4	9-track GCR reel-to-reel tape	2145, 4125
BPI38000	5	18-track half-inch cartridge tape and 4mm cartridge tape	RM5073, HS4400 Notes: *The HS4400 4mm tape emulates a half-inch cartridge tape, thereby inheriting the density of the RM5073 tape.* *18-track tapes are read- only on subsystems supporting 36-track HIC media.*
BPI1250	6	Quarter-inch cartridge tape	QIC, QIC1000
BPI11000	7	8mm cartridge tape	HS8500 (Exabyte)
FMT36TRK	8	36-track half-inch cartridge tape	CTS5136, OST5136, CTS5236, and CLU9710-36T

Mnemonic Value	Integer Value	Media Type	Tape Subsystems
FMTQIC1000	10	Quarter-inch cartridge tape	QIC1000
FMTDLT3	13	Digital linear tape—DLTIII	CLU9710-DLT4 and DLT7 ***Note:*** *Tapes are read-only on these subsystems.*
FMTDLT6	14	Digital linear tape—DLTIII	CLU9710-DLT4 and DLT7 ***Note:*** *Tapes are read-only on these subsystems.*
FMTDLT10	15	Digital linear tapes—DLTIII or DLTIIIxt	CLU9710-DLT4, DLT7, and DLT8
FMTDLT20	16	Digital linear tape—DLTIV	CLU9710-DLT4, DLT7, and DLT8
FMTDLT35	17	Digital linear tape—DLTIV	CLU9710-DLT7 and DLT8
FMTST9840	18	High-capacity cartridge tape	CTS9840
FMTDDS	19	DDS-1 cartridge tape	HS4400 and ALP430
FMTDDS2	9	DDS-2 cartridge tape	HS4400 and ALP430
FMTDDS3	11	DDS-3 cartridge tape	ALP430
FMTAIT	21	AIT-1 cartridge tape	ALP920
FMTAIT2	22	AIT-2 cartridge tape	ALP920
FMTDLT40	23	Digital linear tape—DLTIV	CLU9710-DLT8

The following considerations apply when creating an output tape file:

- If you do not specify a DENSITY value, the system does not take a tape-volume density setting into consideration when assigning the file. The current density of the tape volume is used when writing the new file.
- If you specify a DENSITY value, the system searches for a tape unit capable of writing at the specified density. If the selected tape unit supports software selection of density, the system sets the specified density.

The DENSITY value for input files is the density at which the tape was written.

You should use the DENSITY attribute to select the type of tape unit, because the KIND attribute values of TAPE7, TAPE9, and PETAPE are no longer valid. If a valid DENSITY value is used, but no tape unit that supports the specified density is available, the user task is placed in the waiting state, and a request is issued for a tape unit that supports that density.

The following conditions cause the specified DENSITY value to be overwritten:

- When a multifile tape is created, the DENSITY value of the first file is used for all subsequent files
- When a multivolume file is created, the DENSITY value remains the same from volume to volume

If an OU (Output Unit) system command is entered by an operator to indicate a continuation reel on an incompatible media type, an error message is displayed, and the original waiting message is reissued.

The quarter-inch cartridge tape drives (BPI1250 and FMTQIC1000) can be used only with the library maintenance statements such as ADD, COPY, ARCHIVE FULL, and so forth, with the FILEDATA *TDIR* command, by the Enterprise Database Server DMUTILITY and COPYAUDIT programs, by MAPPER applications, and by DUMPANALYZER.

The following table shows the nonpreferred synonyms for two of the mnemonics in the preceding table:

Mnemonic	**Synonym**
BPI800	HIGH
BPI1600	SUPER

DEPENDENTINTMODE

Kinds	All except port files
Interrogate	Anytime
Modify	When closed
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	None
FTAM File Access	Supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Set the DEPENDENTINTMODE attribute to TRUE if you want the INTMODE attribute value to assume the value of the EXTMODE attribute from the permanent file when the file is opened.

An open error is issued if the new INTMODE value is inconsistent with the default or user-specified FRAMESIZE attribute value. This error cannot occur if the DEPENDENTSPECS attribute value is TRUE or the FILETYPE attribute value is 7 or 8.

DEPENDENTSPECS

Kinds	All except port
Interrogate	Anytime
Modify	When closed
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	None
FTAM File Access	Supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Use the DEPENDENTSPECS attribute to determine whether the attributes of the physical file or the logical file take precedence. If the DEPENDENTSPECS attribute is TRUE when the file is opened, the logical file assumes the structure of the associated physical file. In other words, the following attributes assume values equal to those used to create the physical file:

- BLOCKSIZE
- BLOCKSTRUCTURE
- FILESTRUCTURE
- FILETYPE
- MAXRECSIZE
- MINRECSIZE
- SIZEMODE
- SIZEOFFSET
- SIZE2
- UNITS

The FRAMESIZE attribute is set to a value that is compatible with the INTMODE attribute value. If a new file is being created or the associated permanent file is unlabeled, DEPENDENTSPECS is ignored.

If the value of DEPENDENTSPECS is FALSE, the structure of the logical file is determined by the default or user-specified values of the attributes listed previously.

If DEPENDENTSPECS has been explicitly assigned the value FALSE and the FILETYPE value is 7 or 8, FILETYPE is assigned a value of 0 (zero) when any attempt is made to open the file. An attribute error is reported if a value of DEPENDENTSPECS has been specified and an attempt is made to modify the FILETYPE value to 7 or 8.

DESTINATION

Kinds	Printer
Interrogate	Anytime
Modify	Anytime
Type	Character string
Range	Not applicable
Default	Null string
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

Use the DESTINATION attribute to specify a list of destinations to which a printer backup file should be routed for printing. When this attribute is specified, the string value must contain the device name and, if desired, a colon (:) followed by the number of copies requested. If the number of copies is not specified, the value designated with the PRINTCOPIES attribute is used.

Any combination of site devices (line printers or image printers), remote devices, virtual devices, or device groups can be specified in this string value. The following are examples of valid string values:

```
LP34:2

CP4:1, MYAP9215, LP34:2 AT WHQ, BNAGROUP, MICROFICHE
```

For further information, refer to the discussion of the DESTINATION file attribute in the *Print System Guide*.

If the DESTINATION attribute is not specified, the number of copies specified by the PRINTCOPIES attribute is printed at a printer in the default destination group. For information about establishing the default destination group, refer to the *Print System Guide*.

If the DESTINATION attribute is explicitly specified, such as at file creation time or when changed by the PRINTDEFAULTS task attribute, the value of the DESTINATION attribute overrides the DESTNAME task attribute value. Refer to the *Task Attributes Programming Reference Manual* for information about the DESTNAME task attribute.

The DESTINATION attribute does not apply when the PRINTDISPOSITION attribute is DIRECTDLP or when the BDNAME task attribute is set.

DIALOGCHECKINTERVAL

Kinds	Port subfile
Interrogate	Anytime
Modify	When CLOSED, or when OPEN is completed
Type	Integer
Range	0 through 1440 minutes
Default	0
Abstract Value	DEFAULT, MINIMUM, MAXIMUM
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	BNANATIVESERVICE, TCPIPNATIVESERVICE, TCPNATIVESERVICE

If you are using BNANATIVESERVICE, use the DIALOGCHECKINTERVAL attribute to specify the minimum amount of time, in minutes, that the system waits on a silent dialogue before it automatically invokes inactivity handshaking procedures. If the inactivity handshake that attempts to verify that the dialogue is still intact fails, the subfile is deactivated. The DIALOGCHECKINTERVAL default value is 0 (zero) and indicates that the inactivity handshaking procedures are never invoked. A port provider might choose to always return the default of 0 based on the state of the provider or the location of the dialogue, or both.

If you are using TCPIPNATIVESERVICE or TCPNATIVESERVICE use the DIALOGCHECKINTERVAL attribute to request that “keep-alive” packets be sent to a remote host that is currently not responding to messages. Use the BLOCKEDTIMEOUT attribute to control the length of time keep-alive packets are to be sent.

This attribute requires a subfile index as a parameter if the MAXSUBFILES value is greater than 1, except when it is used with a file declaration. If it is used in a file declaration or the subfile index specified is zero, then the value of DIALOGPRIORITY is changed in all allocated subfiles and it becomes the new default for any subfiles that are allocated later, even after increasing MAXSUBFILES.

DIALOGPRIORITY

Kinds	Port subfile
Interrogate	Anytime
Modify	When CLOSED, or when OPEN is completed
Type	Integer
Range	0 through 7
Default	0
Abstract Value	DEFAULT, MINIMUM, MAXIMUM
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	BNANATIVESERVICE, TCPIPNATIVESERVICE

Use the DIALOGPRIORITY attribute to specify the priority of transmissions from the current port subfile relative to other port subfiles. User programs and distributed system services (DSSs) should normally use the values 0 (zero) through 2.

For BNANATIVESERVICE the values 0, 1, and 2 are supported.

For TCPIPNATIVESERVICE the values 0, 1, and 2 are supported.

The Precedence sub-field of the IP header Type-of-Service field is set to the value of DIALOGPRIORITY. Some router implementations allow priority queuing based on the value of the Type-of-Service field within the IP header. If this field is set to a value other than zero and the router is configured accordingly, data transmission is prioritized by the router.

If both DIALOGPRIORITY and IPTYPEOFSERVICE have values other than zero, the IPTYPEOFSERVICE value overrides the DIALOGPRIORITY value.

Any attempt to set the DIALOGPRIORITY attribute to a value larger than 2 results in a value of 2 after the subport is opened.

A port provider can choose to support only the default value of 0 (zero) based on the state of the provider or the location of the dialog, or both.

This attribute requires a subfile index as a parameter if the MAXSUBFILES value is greater than 1, except when it is used with a file declaration. If it is used in a file declaration or the subfile index specified is zero, then the value of DIALOGPRIORITY is changed in all allocated subfiles and it becomes the new default for any subfiles that are allocated later, even after increasing MAXSUBFILES.

DIOFILESTRUCTURE

Kinds	Disk or CD
Interrogate	Anytime
Modify	When closed
Type	Mnemonic
Range	See following text
Default	ALIGNED180
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

Use the DIOFILESTRUCTURE attribute with direct I/O files only to indicate to the operating system the FILESTRUCTURE values that the program understands and to indicate the way in which the program intends to access the file. Additionally, the DIOFILESTRUCTURE value determines the semantics of the READ and WRITE operations on the file. Because DIOFILESTRUCTURE is a statement of the intention of the programmer, changing the DIOFILESTRUCTURE value by file equation or by the use of the FA (File Attribute) system command is not allowed.

The mnemonic and integer values associated with the DIOFILESTRUCTURE attribute are as follows:

Mnemonic Value	Integer Value	Meaning
ALIGNED180 BLOCKED STREAM	0 5 1	These values indicate that the program is capable of handling only files with a matching FILESTRUCTURE value. An open error occurs if an attempt is made to open an existing file with a FILESTRUCTURE value that does not match the DIOFILESTRUCTURE value. The value of the DEPENDENTSPECS attribute has no effect on this check.
DEPENDENT	3	Indicates that the program is capable of handling files with any FILESTRUCTURE value and is allowed to open an existing file with any FILESTRUCTURE value, subject to the normal rules for opening files. Such a program often specifies the DEPENDENTSPECS value as TRUE, but that specification is not required. In any case, the FILESTRUCTURE value is changed to the value of the FILESTRUCTURE of the permanent file. The FILESTRUCTURE value governs the semantics of READ and WRITE operations.

Mnemonic Value	Integer Value	Meaning
SECTORSTREAM	2	Indicates that the program requests sector-oriented access to the file, regardless of the FILESTRUCTURE value of the physical file. The program can open an existing file with any FILESTRUCTURE value. The DEPENDENTSPECS attribute must have a value of TRUE or the FILETYPE attribute must have a value of 7 or 8. Regardless of the FILESTRUCTURE value of the physical file, the semantics of the READ and WRITE operations are the same as those of a direct I/O file with a FILESTRUCTURE value of STREAM, except that the entire last sector of an area is visible to a program when the data is being read.

When a new file is created, the DIOFILESTRUCTURE value is used as the FILESTRUCTURE value. As a result, if the DIOFILESTRUCTURE value is not ALIGNED180, BLOCKED, or STREAM, the FILESTRUCTURE value must be specified.

For more information on the semantics of direct I/O and stream files, refer to the *I/O Subsystem Programming Guide*.

DIRECTION

Kinds	Tape
Interrogate	Anytime
Modify	Anytime (direct I/O) When closed (not direct I/O)
Type	Mnemonic
Range	FORWARD, REVERSE
Default	FORWARD
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Use the DIRECTION attribute to specify the direction in which the records of a tape file are to be read.

The mnemonic and integer values associated with the DIRECTION attribute are as follows:

Mnemonic Value	Integer Value	Meaning
FORWARD	0	Reads proceed toward the end of the tape.
REVERSE	1	Reads proceed toward the beginning of the tape.

You can change the value of the DIRECTION attribute in a direct I/O program at any time. When the DIRECTION attribute is changed while a direct I/O file is open, the direction bit in the I/O control word (IOCW) is also changed. If a change in direction from forward to reverse is concurrent with a change from writing to reading, system label records are written on the tape before the READ operation is initiated.

For a file that is accessed by a program other than a direct I/O program, the DIRECTION attribute can be modified only when the file is closed.

A file cannot be read in a reverse direction unless the value of BLOCKSTRUCTURE is FIXED or the file is unblocked.

For a description of how the buffer is filled when the system is reading in reverse mode, refer to the discussion of tape files in the *I/O Subsystem Programming Guide*.

DISPOSITION

Kinds	Remote
Interrogate	When open
Modify	Never
Type	Integer
Range	0 through 6
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	No
Parameters	1 required
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Interrogate the DISPOSITION attribute to determine if the remote file can communicate with the station. The DISPOSITION attribute requires one parameter: the relative station number (RSN).

The mnemonic and integer values associated with the DISPOSITION attribute are as follows:

Mnemonic Value	Integer Value	Meaning
UNKNOWNDISPOSITION	0	Disposition unknown
ASSIGNEDDISPOSITION	1	Assignment made
DENIEDDISPOSITION	2	Assignment denied
POSTPONEDDISPOSITION	4	Assignment postponed
ILLEGALUSEDISPOSITION	6	Assignment denied because an illegal use was attempted

For information about how stations are assigned to remote files, refer to the *I/O Subsystem Programming Guide*.

The following ALGOL example shows how to interrogate the DISPOSITION attribute:

```
I := DCFILE (RSN).DISPOSITION;
```

DOCUMENTTYPE

Kinds	Disk or CD
Interrogate	Anytime
Modify	See following text
Type	Mnemonic
Range	See following text
Default	DONTCARE
Abstract Value	DEFAULT
Stored Permanently	Disk
Parameters	None
FTAM File Access	Supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

For File Transfer, Access, and Management (FTAM) file access, use the DOCUMENTTYPE attribute to specify the abstract data type of the contents of the file and the structuring information of the file. The value of the DOCUMENTTYPE attribute is set at the time the file is created and cannot be altered. If the file is a CD-ROM file, the DOCUMENTTYPE attribute can only be interrogated. When a logical file is assigned to an existing physical file, the physical file DOCUMENTTYPE value is used.

The mnemonic and integer values associated with DOCUMENTTYPE for disk files are as follows:

Mnemonic Value	Integer Value	Meaning
DONTCARE	0	Unspecified DOCUMENTTYPE value
FTAM1	1	Unstructured text file
FTAM2	2	Sequential text file
FTAM3	3	Unstructured binary file
INTAP1	4	Record-oriented binary file

If the DOCUMENTTYPE value is not specified—for instance, if the MCP file was created before FTAM was available—or is DONTCARE, an FTAM document type is determined based on the BLOCKSTRUCTURE, FILEKIND, and EXTMODE attribute values.

The following table describes what DOCUMENTTYPE value is selected by the logical I/O subsystem when the BLOCKSTRUCTURE, FILEKIND, and EXTMODE attribute values have certain values:

EXTMODE	FILEKIND	BLOCKSTRUCTURE	Resulting Document Type
EBCDIC, ASCII, IA5STRING, ISOGENERALSTRING, ISOVISIBLESTRING, ISOGRAPHICSTRING	Symbolic values	FIXED, EXTERNAL	FTAM-1
EBCDIC, ASCII, IA5STRING, ISOGENERALSTRING, ISOVISIBLESTRING, ISOGRAPHICSTRING	Symbolic values	VARIABLE	FTAM-2
EBCDIC, ASCII, IA5STRING, ISOGENERALSTRING, ISOVISIBLESTRING, ISOGRAPHICSTRING	Non-symbolic values	FIXED, EXTERNAL, VARIABLE	FTAM-3
SINGLE, OCTETSTRING	Any value	FIXED, EXTERNAL, VARIABLE	FTAM-3

Note: *To use the INTAP-1 document type, the DOCUMENTTYPE value of the file must be INTAP1.*

If a remote host does not support FTAM-2 documents, an FTAM-2 file is handled as though it were an FTAM-1 document. Some file characteristics are not retained during this process of simplification.

Note: *All files with a FILEKIND of SEQDATA, CSEQDATA, TEXTDATA, or xSYMBOL (where xSYMBOL represents a kind of compiler symbol file, such as ALGOLSYMBOL or NDLSYMBOL) are considered symbolic files. Symbolic files normally contain only displayable characters.*

Refer to the *I/O Subsystem Programming Guide* for information about programming with FTAM.

DONOTSEARCHNETWORK

Kinds	Port subfile
Interrogate	Anytime
Modify	When closed
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	See following text
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	BNANATIVESERVICE, TCPIPNATIVESERVICE, TCPNATIVESERVICE

The value of DONOTSEARCHNETWORK is significant only if the YOURHOST attribute value is a null string.

Set the DONOTSEARCHNETWORK attribute to TRUE to indicate to BNA that the network is not to be searched for a subport matching the subport that is being opened. If the system cannot open the subport without searching the network, the system either suspends the OPEN operation when the AVAILABLEONLY value is FALSE or abandons it when the AVAILABLEONLY value is TRUE. Note that for two port subfiles on different hosts to match, at least one of the port files must have a DONOTSEARCHNETWORK value of FALSE or have a YOURHOST value other than a null string.

When a TCPIP connection is opened, the TCPIPSUPPORT library attempts to map a hostname and domain name to the remote IP address by default. If the IP address is not in the TCPIPSUPPORT internal table, it calls the Resolver to find the hostname and domain name associated with the IP address by sending requests to the configured Domain Name Server(s). If a hostname is found before the open completes, YOURHOSTNAME and YOURDOMAINNAME subfile attributes are updated. If the application does not have these attributes updated, they may be suppressed by setting DoNotSearchNetwork to TRUE.

This attribute requires a subfile index as a parameter if the MAXSUBFILES value is greater than 1, except when it is used with a file declaration. If it is used in a file declaration or the subfile index specified is zero, then the value of DIALOGPRIORITY is changed in all allocated subfiles and it becomes the new default for any subfiles that are allocated later, even after increasing MAXSUBFILES.

DONTCLEARAREASBYDEFAULT

Kinds	Disk
Interrogate	Anytime
Modify	Closed
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Abstract Value	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

The DONTCLEARAREASBYDEFAULT attribute has effect only when the following occur:

- The BUFFERSHARING attribute of the logical file has the value SHARED or EXCLUSIVELYSHARED.
- The CLEARAREAS attribute of the physical file has the value FALSE.
- The DISKSCRUB option of the SECOPT (Security Options) system command is not set.
- The subject logical file is the first logical file to open the physical file with a BUFFERSHARING attribute value of SHARED or EXCLUSIVELYSHARED.

If all these conditions are met, a DONTCLEARAREASBYDEFAULT value of TRUE will suppress the scrubbing with zeros of unwritten file space added by writing beyond end-of-file.

Thus, setting DONTCLEARAREASBYDEFAULT can improve the performance of programs that leave large unwritten gaps in a file, and then write to those gaps.

However, the following situation can occur when DONTCLEARAREASBYDEFAULT is TRUE:

1. A program reads a record X from an unallocated area within an unwritten gap. The program receives zero data, and the area remains unallocated.
2. The area becomes allocated due to a write to some other record within the area. Because DONTCLEARAREASBYDEFAULT is set to TRUE, the remainder of the area is not zeroed (that is, not cleared).
3. No write to record X occurs.
4. The program reads record X again, but does not receive zero data. This is a direct result of not zeroing the area. Programs that set DONTCLEARAREASBYDEFAULT to TRUE must be prepared to handle this situation.

DUMMYFILE

Kinds	All except port and remote
Interrogate	Anytime
Modify	When closed and unassigned
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Set the DUMMYFILE attribute to TRUE to indicate the following:

- The logical file is not to be assigned to a permanent file.
- No buffers are to be allocated.
- Logical I/O operations on the file do not result in physical I/O operations.

When an output operation is invoked, a successful result is returned and the program continues with no data transfer or physical I/O operation taking place. An input operation is processed in the same way, except that an end-of-file (EOF) indicator is returned.

The DUMMYFILE attribute is useful for debug files, or when the output of the program is not needed. The attribute is not useful for a program that reads records that it has previously written. For example, the DUMMYFILE attribute should not be set to TRUE for a code file of a compiler because compilers generally reread some of the records written to code files.

When the value of DUMMYFILE is TRUE and a file attribute is interrogated, the value returned is that assigned by the program or is the default value of the attribute, where applicable. If the attribute has not been assigned a value, does not have a default value, and requires a physical file, an attribute error results and a value consistent with an unassigned file is returned. For example, when the LASTRECORD attribute is interrogated, a value of 0 (zero) is returned and an attribute error message is issued.

Section 3
General File Attributes E through M

This section presents file attribute descriptions for attributes with names starting with the letter E through names starting with the letter M. Each attribute description begins with a formatted list of attribute characteristics, followed by a narrative discussion of the use of the attribute. Where appropriate, this discussion includes a list of the various combinations of values, mnemonics, and meanings associated with that attribute.

Refer to the tables in Section 1 for descriptions of the formatted lists.

ENABLEINPUT

Kinds	Remote
Interrogate	When open
Modify	Never
Type	Boolean
Range	TRUE, FALSE
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

Interrogate the ENABLEINPUT attribute to determine if a remote file is enabled for input. If no parameters are specified, the value TRUE is returned if any station in the file is enabled for input. If the relative station number (RSN) parameter is specified, the value TRUE is returned if the specified relative station is enabled for input.

EOF

Kinds	Tape
Interrogate	When open
Modify	Never
Type	Boolean
Range	TRUE, FALSE
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

Interrogate the EOF attribute to determine if an end-of-file condition has been reached on a tape file. When the value of TRUE is returned, the end of the file has been reached. When the value of FALSE is returned, the end of the file has not been reached.

Depending on the situation, it is possible to clear an end-of-file condition by accessing a valid record. To access a valid record, perform a backward read or backspace.

The EOF attribute is not valid for direct I/O files. For the corresponding information concerning direct I/O files, refer to the description of the IOEOF buffer attribute in Section 7, "Direct I/O Buffer Attributes."

ESTIMATEDRECORDS

Kinds	Disk or Printer
Interrogate	Anytime
Modify	When closed
Type	Integer
Range	1–549755813887
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Use the ESTIMATEDRECORDS attribute to specify how many records are to be written to a file with a PRINTDISPOSITION attribute value of DIRECTPS, FILEOPEN, or NOW.

The Print System uses the ESTIMATEDRECORDS value as the volume for the resulting print request. If ESTIMATEDRECORDS is not specified, the Print System uses the system default, which is defined by the PS DEFAULT ESTIMATEDRECORDS system command.

ESTIMATEDRECORDS is ignored for disk files unless the PRINTDISPOSITION value is set to FILEOPEN.

EXCLUSIVE

Kinds	Disk
Interrogate	Anytime
Modify	When unassigned
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	None
FTAM File Access	Supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

For non-FTAM files, set the EXCLUSIVE attribute to TRUE to ensure that the file is the only logical file assigned to a permanent file. For FTAM files, set the EXCLUSIVE attribute to indicate that you want control over the Concurrency-Control parameter information. Refer to the *I/O Subsystem Programming Guide* for further information about controlling the Concurrency-Control parameter information.

If the EXCLUSIVE attribute is set to TRUE, one of the following situations occurs when a program opens a permanent disk file implicitly or explicitly through the OPEN or PRESENT attribute:

- If the file exists and is in use by another program, the opening program waits with a “WAITING ON” message until the first program closes the file, unless the I/O operation included an AVAILABLE option, in which case the NOTAVAILABLERSLT (0) OPEN error is returned. Note that the logical file of the opening program is now bound to the physical file. Thus, when the first program closes the file, the opening program opens the file that was bound, even if the first program renamed or purged the file.
- If the file exists and is not in use by any other program, the file is opened and all future programs attempting to open the file must wait for this program to close the file.
- When a program attempts to access a file through FTAM, and the remote system denies access to the file because of exclusive conflicts with another program, the open process fails and produces an OPEN result value of 20.

If the file does not exist and an explicit or implicit OPEN operation is initiated, the program waits for a “NO FILE” response. If the PRESENT attribute is interrogated, FALSE is returned.

Note that a file opened with the EXCLUSIVE attribute set to TRUE can be removed from the system directory.

The RESIDENT attribute is unaffected when the EXCLUSIVE attribute value is TRUE. RESIDENT returns TRUE if the permanent file exists, whether or not the file can be assigned at that moment.

For more information about the OPEN, PRESENT, and RESIDENT attributes, refer to their descriptions.

EXECUTEDATE

Kinds	Disk
Interrogate	When assigned
Modify	Never (see following exception)
Type	Integer
Range	70001 through 135365
Default	CREATIONDATE
Abstract Value	Not applicable
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not supported

Interrogate the EXECUTEDATE attribute to determine the last time the physical file was run as a program or library.

The value returned by EXECUTEDATE is a decimal integer. When you add the integer 1900000 to the value returned, the result is the date in the form YYYYDDD, where YYYY and DDD represent the year and date respectively, in Julian form.

The EXECUTEDATE attribute is updated at the beginning of a job or task. The attribute value is unaffected by library maintenance and is not updated for files on a disk which is not write-enabled.

The EXECUTEDATE attribute is used in conjunction with the EXECUTETIME and EXECUTETZ attributes. See “EXECUTETIME” and “EXECUTETZ” for more information.

The EXECUTEDATE attribute can be modified by a valid MCS program if that program previously set the HSFILECOPY file attribute to TRUE.

EXECUTEDATEUT

Kinds	Disk
Interrogate	When assigned
Modify	Never
Type	Integer
Range	70001 through 135365
Default	CREATIONDATE
Abstract Value	Not applicable
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not supported

Interrogate the EXECUTEDATEUT attribute to determine the EXECUTEDATE value with respect to UTC time instead of the EXECUTETZ time zone. See “EXECUTEDATE” for further information.

When you interrogate the EXECUTEDATEUT attribute, you receive an error if the time zone attribute, EXECUTETZ, is 0 (zero). See “EXECUTETZ” for more information.

EXECUTETIME

Kinds	Disk
Interrogate	When assigned
Modify	Never (see following exception)
Type	Integer
Range	0 through 86399999999
Default	CREATIONTIME
Abstract Value	Not applicable
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not supported

Interrogate the EXECUTETIME attribute to determine the time of day, in microseconds since midnight, that is associated with the EXECUTEDATE attribute value. The EXECUTETIME value is updated when the EXECUTEDATE value is updated.

The EXECUTETIME attribute can be modified by a valid MCS program if that program previously set the HSFILECOPY file attribute to TRUE.

EXECUTETIMEUT

Kinds	Disk
Interrogate	When assigned
Modify	Never
Type	Integer
Range	0 through 86399999999
Default	CREATIONTIME
Abstract Value	Not applicable
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not supported

Interrogate the EXECUTETIMEUT attribute to determine the EXECUTETIME value with respect to UTC time instead of the EXECUTETZ time zone. See "EXECUTETIME" for more information.

When you interrogate the EXECUTETIMEUT attribute, you receive an error if the time zone attribute, EXECUTETZ, is 0 (zero). See "EXECUTETZ" for more information.

EXECUTETZ

Kinds	Disk
Interrogate	When assigned
Modify	Never (see following exception)
Type	Integer
Range	0 through 255
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	Yes
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Interrogate the EXECUTETZ attribute to determine the number of the time zone that is associated with the EXECUTEDATE and EXECUTETIME attribute values. The EXECUTETZ value is updated when the EXECUTEDATE value is updated.

To determine the meaning of non-zero values of EXECUTETZ, refer to Appendix A, “Predefined Time Zones.” You can use the MCP library procedure TIMEZONENAME to convert the numeric code of a time zone into the name or abbreviation for that time zone. TIMEZONENAME is documented in the *Master Control Program (MCP) System Interfaces Programming Reference Manual.*

If the integer value of EXECUTETZ is 0 (zero), then there is not a time zone associated with EXECUTEDATE and EXECUTETIME.

The EXECUTETZ value is 0 if the file

- Has CREATIONTZ = 0 and EXECUTEDATE and EXECUTETIME are equal to CREATIONDATE and CREATIONTIME.
- Resides on a disk family with FAMILY HEADER VERSION = 6. Refer to “Disk File Headers” in Section 1 for more information.
- Was executed on a system that did not have a time zone set.

The EXECUTETZ attribute can be modified by a valid MCS program if that program previously set the HSFILECOPY file attribute to TRUE.

EXTDELIMITER

Kinds	All
Interrogate	Anytime
Modify	See following explanation
Type	Mnemonic
Range	UNSPECIFIED, CR, CRCC, CRLF, and NL
Default	See following explanation
Abstract Value	None
Stored Permanently	See following explanation
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	All

Use the EXTDELIMITER attribute to specify delimiter characters, if any, for identifying the separation of records in a physical file. EXTDELIMITER is meaningful primarily for FILECLASS=CHARACTERSTREAM files.

Mnemonic Value	Integer Value	Meaning
Unspecified	0	The actual delimiter characters, if any, are unspecified.
CR	1	Carriage return.
CRCC	4	Carriage return, optionally followed by another carriage control character that is a line feed or a form feed.
CRLF	3	Carriage return followed by a line feed.
NL	2	New line (line feed).

When a file is created with a KIND attribute value of PRINTER, a BACKUPKIND attribute value of DISK, and a FILESTRUCTURE attribute value of STREAM, the EXTDELIMITER value of the resulting printer backup file is set to CRCC. In addition, the actual delimiter characters are placed in the file by the MCP to correspond to the carriage control applied to the printer file.

For all other files, the default value of EXTDELIMITER is UNSPECIFIED. Other values of EXDELIMITER are valid only for files with a FILECLASS attribute value of CHARACTERSTREAM. Specifying EXTDELIMITER serves only as notification regarding which delimiter should be placed in the file by programs writing to that file, so that programs reading from that file can interpret the delimiter characters appropriately.

For files other than disk files, the value of the EXTDELIMITER attribute is not stored permanently and is not transmitted to the correspondent of a port file.

The EXTDELIMITER value of a logical file can be modified while the file is unassigned. If the logical file is assigned, modifying EXTDELIMITER is allowed only when the file is assigned to a permanent disk file and is not open. In this case, the value of EXTDELIMITER stored in the permanent file is also changed.

EXTMODE

Kinds	All
Interrogate	Anytime
Modify	When closed
Type	Mnemonic
Range	See following text
Default	For new files, the INTMODE value For existing files, the EXTMODE value of the physical file
Abstract Value	DEFAULT
Stored Permanently	Disk or tape
Parameters	1 optional
FTAM File Access	Restricted values
Heritage Host Services Logical I/O	Restricted values
Port Services	BNANATIVESERVICE

Use the EXTMODE attribute to specify the external or physical character encoding of the records in a file. The mnemonic and integer values and meanings associated with this attribute are listed in the following table. The CENTRALSUPPORT library might contain additional valid values, up to a 65,535 maximum.

Mnemonic Value	Integer Value	Meaning
ASCII	5	8-bit data encoded with the ASCII coded character set
BCL	3	6-bit data encoded with the BCL coded character set ***Note:*** *Valid for printer under certain conditions. Refer to the* I/O Subsystem Programming Guide *for information about translation.*
EBCDIC or ASERIESEBCDIC	4	8-bit data encoded with the ClearPath MCP servers EBCDIC character set
HEX	2	4-bit, packed decimal
SINGLE	0	Word mode, binary, or 48-bit data

The following table displays EXTMODE values that are valid only for files of KIND=DISK and KIND=PRINTER:

Mnemonic Value	Integer Value	Meaning
ARABICV201B	52	Arabic ASCII-based 8-bit character set used for printing
ARABICV202D	53	Arabic ASCII-based 8-bit character set used for printing
ARABIC20EBCDIC	34	Arabic EBCDIC 8-bit character set (previously known as version 20 EBCDIC)
ARABIC20ISO	35	Arabic ISO 8-bit character set (previously known as version 20 ASCII)
ASTASCIIGB2312	112	Mixed multibyte character set for Chinese ASCII-based terminal use
ASTASCIIKSC5601	106	Mixed multibyte character set for Korean ASCII-based terminal use
ASTASCIIUTL	109	Mixed multibyte character set for Taiwanese ASCII-based terminal use
ASTUTL	83	16-bit character set for Taiwanese ASCII-based terminal use
ASUTL	82	16-bit character set for Taiwanese ClearPath MCP servers use
CANSUPPLEBCDIC	16	Canadian EBCDIC 8-bit character set with general purpose supplementary graphics
CANSUPPLISO	17	Canadian ISO 8-bit character set with general purpose supplementary graphics
CODEPAGE437	36	MS-DOS 8-bit code page for United States use
CODEPAGE850	18	MS-DOS 8-bit code page for multilingual use
CODEPAGE851	21	MS-DOS 8-bit code page for Latin/Greek use
CODEPAGE852	28	MS-DOS 8-bit code page for Eastern European use
CODEPAGE857	44	MS-DOS 8-bit code page for Turkish use
CODEPAGE864	54	MS-DOS 8-bit code page for Arabic use
CODEPAGE866	31	MS-DOS 8-bit code page for Russian use
CODEPAGE932	102	MS-DOS and Windows mixed multibyte code page for Japanese PC use
CODEPAGE936	113	MS-DOS and Windows mixed multibyte code page for Chinese PC use
CODEPAGE949	107	MS-DOS and Windows mixed multibyte code page for Korean PC use

Mnemonic Value	Integer Value	Meaning
CODEPAGE950	110	MS-DOS and Windows mixed multibyte code page for Taiwanese PC use
CODEPAGE1250	33	Windows 8-bit code page for Eastern European use
CODEPAGE1251	28	Windows 8-bit code page for Russian use
CODEPAGE1252	37	Windows 8-bit code page for multilingual use
CODEPAGE1253	45	Windows 8-bit code page for Greek use
CODEPAGE1254	43	Windows 8-bit code page for Turkish use
CODEPAGE1256	55	Windows 8-bit code page for Arabic use
EBCDICGB2312	111	Mixed multibyte character set for Chinese ClearPath MCP servers use
EBCDICKSC5601	105	Mixed multibyte character set for Korean ClearPath MCP servers use
EBCDICUTL	108	Mixed multibyte character set for Taiwanese ClearPath MCP servers use
EUCJP	103	Mixed multibyte character set for Japanese UNIX systems use
FRENCHARABICE	22	French/Arabic EBCDIC 8-bit character set
FRENCHARABICISO	23	French/Arabic ISO 8-bit character set
HUNGARIANBTOS	46	Hungarian BTOS 8-bit character set
IA5STRING	10	A string of 8-bit frames containing data encoded in the ISO646 G0 and C0 character sets
IBMSWEDENEBCDIC	51	8-bit EBCDIC-based Swedish IBM character set
IBM297	39	8-bit EBCDIC-based IBM character set
ISOGENERALSTRING	7	A string of 8-bit frames containing data encoded in the ISO646 C0 and G0 character sets and the ISO8859-1 G0 and G1 character sets
ISOGRAPHICSTRING	8	A string of 8-bit frames containing data encoded in the ISO646 G0 character set and ISO8859-1 G0 and G1 character sets
ISOVISIBLESTRING	9	A string of 8-bit frames containing data encoded in the ISO646 G0 character set
JAPANEBCDIC	41	Japanese EBCDIC 8-bit character set
JAPANEBCDICJBIS8	100	Mixed multibyte character set for Japanese ClearPath MCP servers use
JAPANV24	49	Japanese Version 24 8-bit character set
JAPANV24JBIS8	114	Japanese Version 24 mixed, multibyte character set

Mnemonic Value	Integer Value	Meaning
JBIS7	81	16-bit character set for Japanese ASCII-based terminal use
JBIS8	80	16-bit character set for Japanese ClearPath MCP servers use
JISASCII	42	Japanese ASCII 8-bit character set
JISASCIIJBIS7	101	Mixed multibyte character set for Japanese ASCII-based terminal use
LATINCYRILLICEBC	29	Cyrillic EBCDIC 8-bit character set
LATINCYRILLICISO	30	Cyrillic ISO 8-bit character set
LATINGREEKBTOS	25	Latin/Greek BTOS 8-bit character set
LATINGREEKEBCDIC	19	Latin/Greek EBCDIC 8-bit character set
LATINGREEKISO	20	Latin/Greek ISO 8-bit character set
LATIN1EBCDIC	12	Latin1 EBCDIC 8-bit character set
LATIN1ISO	13	Latin1 ISO 8-bit character set
LATIN2EBCDIC	26	Latin2 EBCDIC 8-bit character set
LATIN2ISO	27	Latin2 ISO 8-bit character set
LATIN5EBCDIC	14	Latin5 EBCDIC 8-bit character set
LATIN5ISO	15	Latin5 ISO 8-bit character set
LATIN9EBCDIC	47	Latin9 EBCDIC 8-bit character set
LATIN9ISO	48	Latin9 ISO 8-bit character set
LETSJ	104	Mixed multibyte character set for Japanese ClearPath IX servers use
LOCALEBCDIC	50	8-bit EBCDIC-based locally defined character set
MACROMAN	38	Macintosh 8-bit code page for multilingual use
NORWAYBTOS	24	Norwegian BTOS 8-bit character set
OCTETSTRING	11	A string of 8-bit frames each containing any binary value from hexadecimal 00 to hexadecimal FF
UCS2	85	Universal 16-bit character set based on Multilingual Plane of the ISO 10646-1 standard
UCS2NT	84	Universal 16-bit character set based on the Basic Multilingual Plane of the ISO 10646-1 standard, encoded for Intel processors, Windows NT and Windows 95 systems use

Note: *Mixed multibyte character sets contain both 16-bit and 8-bit characters. For such character sets, data is transferred in units of 8 bits (FRAMESIZE = 8).*

For a description of the ASCII, BCL, and EBCDIC coded character sets, refer to the *ALGOL Reference Manual, Volume 1*. For a description of the ISO646 and ISO8859-1 coded character sets, refer to the *I/O Subsystem Programming Guide*. For a description of the remaining coded character sets, refer to the *MultiLingual System (MLS) Administration, Operations, and Programming Guide*.

The following table indicates the IPC providers and the distributed systems services (DSSs) that support the various EXTMODE values:

	Distributed Systems Services (DSSs)		**IPC Providers**	
Mnemonic Value	**Heritage Host Services Logical I/O**	**FTAM File Access**	**BNA Version 2 Port Files**	**Local Port Provider (LPP)**
ARABIC20EBCDIC	Yes	No	No	No
ARABIC20ISO	Yes	No	No	No
ASCII	Yes	No	Yes	Yes
ASTASCIIGB2312	Yes	No	No	No
ASTASCIIKSC5601	Yes	No	No	No
ASTASCIIUTL	Yes	No	No	No
ASTUTL	Yes	No	No	No
ASUTL	Yes	No	No	No
BINARY	Yes	No	No	No
CANSUPPLEBCDIC	Yes	No	No	No
CANSUPPLISO	Yes	No	No	No
CODEPAGE437	Yes	No	No	No
CODEPAGE850	Yes	No	No	No
CODEPAGE851	Yes	No	No	No
CODEPAGE852	Yes	No	No	No
CODEPAGE857	Yes	No	No	No
CODEPAGE866	Yes	No	No	No
CODEPAGE932	Yes	No	No	No
CODEPAGE936	Yes	No	No	No
CODEPAGE949	Yes	No	No	No
CODEPAGE950	Yes	No	No	No
CODEPAGE1253	Yes	No	No	No
CODEPAGE1254	Yes	No	No	No

	Distributed Systems Services (DSSs)		IPC Providers	
Mnemonic Value	**Heritage Host Services Logical I/O**	**FTAM File Access**	**BNA Version 2 Port Files**	**Local Port Provider (LPP)**
EBCDIC	Yes	No	Yes	Yes
EBCDICGB2312	Yes	No	No	No
EBCDICKSC5601	Yes	No	No	No
EBCDICUTL	Yes	No	No	No
EUCJP	Yes	No	No	No
FRENCHARABICE	Yes	No	No	No
FRENCHARABICISO	Yes	No	No	No
HUNGARIANBTOS	Yes	No	No	No
IA5STRING	Yes	Yes	No	No
ISOGENERALSTRING	Yes	Yes	No	No
ISOGRAPHICSTRING	Yes	Yes	No	No
ISOVISIBLESTRING	Yes	Yes	No	No
JAPANEBCDIC	Yes	No	No	No
JAPANEBCDICJBIS8	Yes	No	No	No
JAPANV24	Yes	No	No	No
JAPANV24EBCDIC	Yes	No	No	No
JBIS7	Yes	No	No	No
JBIS8	Yes	No	No	No
JISASCII	Yes	No	No	No
JISASCIIJBIS7	Yes	No	No	No
LATINCYRILLICEBC	Yes	No	No	No
LATINCYRILLICISO	Yes	No	No	No
LATINGREEKBTOS	Yes	No	No	No
LATINGREEKEBCDIC	Yes	No	No	No
LATINGREEKISO	Yes	No	No	No
LATIN1EBCDIC	Yes	No	No	No
LATIN1ISO	Yes	No	No	No
LATIN2EBCDIC	Yes	No	No	No
LATIN2ISO	Yes	No	No	No

	Distributed Systems Services (DSSs)		IPC Providers	
Mnemonic Value	Heritage Host Services Logical I/O	FTAM File Access	BNA Version 2 Port Files	Local Port Provider (LPP)
LATIN5EBCDIC	Yes	No	No	No
LATIN5ISO	Yes	No	No	No
LATIN9EBCDIC	Yes	No	No	No
LATIN9ISO	Yes	No	No	No
LETSJ	Yes	No	No	No
NORWAYBTOS	Yes	No	No	No
OCTETSTRING	Yes	Yes	No	No
SINGLE	Yes	No	Yes	Yes
STDEBCDIC	Yes	No	No	No
UCS2	Yes	No	No	No
UCS2NT	Yes	No	No	No

The EXTMODE value cannot be changed for an existing labeled, permanent file. If a logical file is assigned to a permanent physical file, the physical file EXTMODE value is used as the logical file EXTMODE value while the assignment exists.

For CD–ROM files that were not created by a ClearPath system, the value is always OCTETSTRING.

The default value for the EXTMODE attribute is the INTMODE attribute value, unless the INTMODE value is in conflict with the requirements of the associated physical unit. The EXTMODE value of remote files or ODT files is always EBCDIC. If the value of PRINTDISPOSITION is either DIRECTPS or DIRECTDLP, the EXTMODE value must be EBCDIC.

Whenever the EXTMODE and INTMODE attributes differ, there is a possibility that the data is being translated. Refer to the *I/O Subsystem Programming Guide* for information about translation.

Port Files

The EXTMODE value is negotiated during dialogue establishment and is based on the INTMODE and TRANSLATE values. Refer to the *I/O Subsystem Programming Guide* for more information.

The SINGLE mnemonic is always treated as EBCDIC. When EXTMODE is interrogated, it requires a subfile index as a parameter if the MAXSUBFILES value is greater than 1, except when it is used in a file declaration. Modifying EXTMODE has no effect on the port subfile.

Printer Backup Files

You can create printer backup files with EXTMODE set to HEX, 8-bit, 16-bit, or mixed multibyte characters; however, the Print System prints only files with 8-bit characters.

For delimited backup files, the EXTMODE must correspond to a character set that is EBCDIC-based, ASCII-based, UCS2, or UCS2NT.

Considerations When Using the WFL *ALTER* Statement with EXTMODE

You can modify the EXTMODE value of a permanent disk file with the WFL *ALTER* statement, subject to the following conditions:

- When the CCSVERSION attribute is specified in the <alter attribute list> and the EXTMODE attribute is not, the CCSVERSION value is modified in the file and the EXTMODE value is modified to the value compatible with the CCSVERSION value specified.
- When the EXTMODE attribute is specified in the <alter attribute list> and the CCSVERSION attribute is not, the EXTMODE value is modified in the file. The CCSVERSION value is unaffected if it is compatible with the EXTMODE value specified; otherwise, the CCSVERSION attribute reflects a value of NOTSET (-1) in the file.
- When both the CCSVERSION and EXTMODE attributes are specified in the <alter attribute list> and are compatible with one another, then both values are modified in the file. Incompatible values of CCSVERSION and EXTMODE cause the ALTER statement to fail and an error message results.
- For character-oriented files, if modifying the value of CCSVERSION or EXTMODE causes the character size to change, the ALTER statement fails and an error message results.

Refer to the *Work Flow Language (WFL) Programming Reference Manual* for details on the ALTER statement.

FAMILY

FAMILY is a nonpreferred synonym for STATIONLIST. Refer to the description of the STATIONLIST attribute.

FAMILYINDEX

Kinds	Disk or CD
Interrogate	Anytime
Modify	Anytime
Type	Integer
Range	0 through 255
Default	0
Abstract Value	DEFAULT, MINIMUM, MAXIMUM
Stored Permanently	Disk or CD
Parameters	1 optional
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

If a file is to be written to a multidisk family, you can use the FAMILYINDEX attribute to override the normal allocation of disk file areas, if necessary, for a specific application. The value specified for the attribute identifies a particular family member on which the file areas are to be allocated. Each disk of a multidisk family has a family-relative index number that is used to identify the physical volume within the multidisk family. Use the appropriate relative index number when modifying the FAMILYINDEX attribute. If the value of FAMILYINDEX is 0 (zero), the disk areas are allocated in the normal rotational order used by the system.

Modifying the value of the FAMILYINDEX attribute when the logical file is not assigned to a physical file is effective only on subsequent OPEN operations that cause a new physical file to be created. When the logical file is assigned to a physical file, the FAMILYINDEX attribute requires the area number as a parameter.

Unless overridden by a change in the FAMILYINDEX value for a specific area, the FAMILYINDEX value when the file is opened is used to allocate each area of the file. A FAMILYINDEX value can be specified for a specific area after the file is opened, but only before the disk space is allocated for that area.

If space is unavailable on the family member specified by the FAMILYINDEX value, the system displays the following message:

```
PKnn (FAMILYINDEX nnn) sss SECTORS REQUIRED
<file name> ON <family name>; OK TO IGNORE FAMILY INDEX
```

An operator can perform one of the following tasks after receiving the preceding message:

- Allow the file to overflow to another unit in the family by entering the OK (Reactivate) system command.
- Discontinue the program by entering the DS (Discontinue) system command.

- Take some action to free up space on that family member. Possible actions include the following:
  - If the required pack is online and write-enabled, issue the system command SQUASH. Refer to the *System Commands Manual* for details.
  - If the required pack is online and write-enabled, remove some files.
  - If the required pack is online and write-enabled, issue the WFL statement ARCHIVE ROLLOUT or ARCHIVE RELEASE. Refer to the *System Operations Guide* for details.

If the OK command is used, the following message is displayed to indicate where the extra space was found:

```
PKnn SECTORS ALLOCATED ON FAMILYINDEX nnn ON <family name>
```

When library maintenance is used to copy a file, the family indexes of the areas of the disk file assigned through the FAMILYINDEX attribute are preserved, but you can override them by specifying the FAMILYINDEX attribute in the COPY statement.

If interrogated when the file is assigned, FAMILYINDEX returns the value for the specified area of the physical file. If you do not specify an area number when you interrogate the value when the file is assigned, the value for area 0 is returned.

Note: *After the system has allocated sectors to an area, any interrogation of the FAMILYINDEX value of that area returns the actual family index on which the area was allocated, even if you did not specify the FAMILYINDEX attribute.*

The following examples show how to modify the value of FAMILYINDEX in ALGOL:

```
DISKFILE.FAMILYINDEX := 1;

DISKFILE(ROWNUMBER).FAMILYINDEX := 3;

DUPDISKFILE(ROWNUMBER,COPYNUMBER).FAMILYINDEX := 23;
```

The following examples show how to modify the value of FAMILYINDEX for a file in COBOL74 or COBOL85. Note that COBOL74 accepts one parameter only.

```
CHANGE ATTRIBUTE FAMILYINDEX OF DISKFILE TO 1.

CHANGE ATTRIBUTE FAMILYINDEX OF DISKFILE (ROW-NBR) TO 3.
```

FAMILYNAME

Kinds	Disk, CD, printer
Interrogate	Anytime
Modify	When closed
Type	Character string
Range	Any valid family name
Default	DISK
Abstract Value	DEFAULT
Stored Permanently	Disk, CD or backup files
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Use the FAMILYNAME attribute to determine or specify the name or label of the disk family on which the physical file is located.

A valid family name is an identifier that is 1 to 17 characters long and contains any combination of the letters A through Z and the numbers 0 through 9. The value TAPE is an invalid disk family name. In addition, the family name for a CD-ROM file can have underscores (_) in the name.

If family substitution is in effect for the specified or default family name of the file, the primary and alternate FAMILYNAME values are used to search for the file. When a file is created, the primary FAMILYNAME value is used.

If FAMILYNAME is interrogated when the file is assigned, the value returned is the name of the family on which the physical file resides.

For backup files, refer to the USERBACKUPNAME attribute for information about how the FAMILYNAME attribute value is affected when USERBACKUPNAME is set to TRUE.

FAMILYOWNER

Kinds	Tape
Interrogate	Anytime
Modify	When unassigned
Type	Character string
Range	A valid usercode
Default	See following text
Abstract Value	DEFAULT
Stored Permanently	Tape
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

The system uses the FAMILYOWNER attribute only when the *SECOPT TAPECHECK* form of the SECOPT (Security Options) system command is set to AUTOMATIC. Refer to the *Security Administration Guide* for an explanation of the SECOPT command.

Use the FAMILYOWNER attribute to specify the usercode associated with the tape volume family.

A usercode has two valid formats:

- A name consisting of from 1 to 17 characters that begins with a letter or a number. The remaining 16 characters can be any combination of letters, numbers, hyphens (-), or underscores (_).
- An identifier consisting of from 1 to 17 characters that is enclosed in quotation marks ('' ''). The EBCDIC character can be any character that has a hexadecimal code greater than or equal to a hexadecimal 4"40" and cannot be an EBCDIC quotation mark.

If interrogated when a file is assigned, the FAMILYOWNER attribute returns the owner of the tape volume family for the physical file. Otherwise, the FAMILYOWNER attribute returns the last specified value or a null string if no value was specified.

When an OPEN operation creates a new tape volume family, the FAMILYOWNER value is determined by one of the following methods:

- If a FAMILYOWNER value is specified, that value becomes the owner of the newly created tape volume family.
- If a FAMILYOWNER value is not specified or the specified value is a null string, the usercode portion of the FILENAME attribute is used. If the FILENAME values no usercode, the usercode of the process that owns the logical file is used.
- If the FAMILYOWNER value is an asterisk (*) or the FILENAME value contains no usercode and the process that owns the logical file is running with no usercode, then the asterisk becomes the owner of the new tape volume family.

Only a privileged user can create a tape volume family with a usercode other than his or her own.

When an OPEN operation requests an existing tape file, one of the following actions is taken:

- If a FAMILYOWNER value is specified, only tapes that are mounted and owned by the FAMILYOWNER are searched.
- If the FAMILYOWNER value is an asterisk, only tapes that have no usercode are searched.
- If a FAMILYOWNER value is not specified or a null string is specified, only tapes belonging to the user indicated by the usercode part of the FILENAME attribute are searched.

  If the FILENAME contains no usercode, any mounted tapes meeting the matching and security criteria are searched.

Attempting to open a file when the FAMILYOWNER value and the usercode portion of the FILENAME attribute do not match results in an attribute error and the open result error BADATTRIBUTERSLT (66) is returned.

If the SECURITYLABELS tape volume security option is TRUE, the value of the FAMILYOWNER attribute is stored in the file label and ANSI87 tape labels are created. Refer to the *WFL Reference Manual* for information about the SECURITYLABELS tape volume security option.

FAMILYSIZE

FAMILYSIZE is a nonpreferred synonym for STATIONCOUNT. Refer to the description of the STATIONCOUNT attribute.

FILECLASS

Kinds	All
Interrogate	Anytime
Modify	Never
Type	Mnemonic
Range	RECORDORIENTED, CHARACTERSTREAM, and WORDSTREAM
Default	See following explanation
Abstract Value	DEFAULT
Stored Permanently	See following explanation
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	All

Interrogate the FILECLASS attribute to determine whether a physical file is a traditional record-oriented file, a file that is a continuous stream of characters, or a file that is a continuous stream of words.

For an existing file, the FILECLASS attribute is derived as follows.

Mnemonic Value	Integer Value	Meaning
CHARACTERSTREAM	2	One of the following: • Any disk file, printer backup file, or CD-ROM file with a MAXRECSIZE attribute value of 1, a FILESTRUCTURE attribute value of STREAM, and an EXTMODE attribute value other than SINGLE • Any port file with a SERVICE attribute value of TCPNATIVESERVICE or TCPIPNATIVESERVICE and an EXTMODE attribute value other than SINGLE • FIFO special disk files and pipes
WORDSTREAM	3	One of the following: • Any disk file or CD-ROM file with a MAXRECSIZE value of 1, a FILESTRUCTURE value of STREAM, and an EXTMODE value of SINGLE • Any port file with a SERVICE attribute value of TCPNATIVESERVICE or TCPIPNATIVESERVICE and an EXTMODE value of SINGLE
RECORDORIENTED	1	All other physical files

Interrogating the FILECLASS attribute while a logical file is assigned to a physical file returns the file class of the physical file regardless of how it is being accessed.

FILEEQUATED

Kinds	All
Interrogate	Anytime
Modify	Never
Type	Boolean
Range	TRUE, FALSE
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	No
Parameters	2 optional
FTAM File Access	Supported
Heritage Host Services Logical I/O	Supported
Port Services	All

Interrogate the FILEEQUATED attribute to determine if file equation was specified for the file when the program was compiled or at the time that it was initiated. If TRUE is returned, one of these conditions exists.

You can interrogate a particular file attribute by identifying the attribute in your language statement. In addition, if you are using a port file that has a MAXSUBFILES value that is greater than 1, you can use the subfile index to identify the particular subfile for the file attribute you want to interrogate.

If you do not specify a particular file attribute, TRUE is returned if any of the attributes for the file were file-equated.

The following ALGOL statement determines if the TITLE attribute has been file-equated in subfile 1:

```
TITLE_EQUATED := F(TITLE).FILEEQUATED;
```

The following COBOL74 or COBOL85 example determines whether or not the TITLE attribute has been file-equated:

```
IF ATTRIBUTE FILEEQUATED OF F(VALUE(TITLE)) = VALUE(TRUE) THEN
        MOVE 1 TO TITLE-EQUATED
   ELSE
        MOVE 0 TO TITLE-EQUATED.
```

File equation is reevaluated whenever the INTNAME attribute is modified. In the following example, FILE_EQUATED is TRUE if any of the attributes of the file with an INTNAME value of NEWINTNAME were file-equated, and TITLE_EQUATED is TRUE if the TITLE attribute was file-equated:

```
REPLACE F.INTNAME BY "NEWINTNAME.";
FILE_EQUATED := F.FILEEQUATED
TITLE_EQUATED := F(TITLE).FILEEQUATED;
```

When a file has been assigned to a global file through a WFL job using the WFL global file assignment *FILE F:= GLOBALFILE*, the FILEEQUATED value, regardless of the presence of a parameter, is always FALSE.

The FILEEQUATED attribute is not affected by the FA (File Attribute) system command. Refer to the *System Operations Guide* or the *CANDE Operations Reference Manual* for more information about the FA command.

FILEKIND

Kinds	Disk, CD, tape
Interrogate	Anytime
Modify	See following text
Type	Mnemonic
Range	See following text
Default	DATA
Abstract Value	DEFAULT
Stored Permanently	Disk or tape
Parameters	None
FTAM File Access	Restricted values
Heritage Host Services Logical I/O	Restricted usage/values
Port Services	Not applicable

Use the FILEKIND attribute to determine or specify the internal structure and purpose for a disk file. When using Heritage Host Services logical I/O, the LABELKIND attribute must be used for tape files.

Returned Information

When a program interrogates the FILEKIND attribute, the following information is returned:

- If the file is a tape file, the LABELKIND file attribute value is returned.
- If the file is assigned to a disk file, the value for the physical file is returned.
- If the file is unassigned, the last value assigned to the FILEKIND attribute is returned.

Changing the FILEKIND attribute

A program can modify the FILEKIND attribute when the file is unassigned or when the file is assigned if the file is a disk file. If a program specifies a value before a file is assigned, that value is used to give the newly created file a FILEKIND value or is ignored if the program is accessing an existing file.

A program can change the FILEKIND value of an existing disk file when the following conditions are met:

- The file must be assigned.
- The program must have write access to the file.
- The specified change must be consistent with the privileges of the program, as detailed in the restrictions listed later. For example, a compiler program can change a data file to a code file, but other types of programs can only change the value to a non-codefile value.

FILEKIND Integer and Numeric Values

The integer and mnemonic values associated with FILEKIND for disk files are as follows:

0	NULLFILE
1	DIRECTORY or VERSIONDIRECTORY
2	SYSTEMDIRECTORY
3	CATALOG
4	BACKUPPRINTER or BACKUPDISK
5	RECONSTRUCTIONFILE
6	SYSTEMDIRFILE
7	JOBDESCFILE
8	ARCHIVELOG
9	BOOTCODE
10	BACKUPVSID
11	DASDLDATA
12	CPAFILE
13	FTAUDIT
14	RESTRICTEDFILE
15	XDISKFILE
16	KEYSFILE
18	PRINTFILE
19	REMOTEAUDIT
20	COMPILERCODEFILE
21	CHECKPOINTFILE
22	CPJOBFILE
23	DCPCODE
24	NDLCODE
25	NDLIICODE
26	RECOVERYFILE
27	SCHEDULEFILE
28	INFOFILE
29	LIBRARYCODE
30	INTRINSICFILE
31	MCPCODEFILE
32	ALGOLCODE
33	COBOLCODE
34	FORTRANCODE
35	XALGOLCODE
36	PLICODE
37	JOVIALCODE
39	ESPOLCODE
40	DCALGOLCODE
41	BASICCODE
42	XFORTRANCODE
43	JOBCODE
44	DMALGOLCODE
45	NEWPCODE
46	SANSCODE
47	PASCALCODE
48	RPGCODE
50	FORTRAN77CODE
53	SORTCODE
54	COBOL74CODE
55	CCCODE
57	COBOL85CODE
59	PASCAL83CODE
60	JAVACODE
62	BOUNDCODE
63	CODEFILE
64	ALGOLSYMBOL
65	COBOLSYMBOL
66	FORTRANSYMBOL
67	XALGOLSYMBOL
68	PLISYMBOL
69	JOVIALSYMBOL
71	ESPOLSYMBOL
72	DCALGOLSYMBOL
73	BASICSYMBOL
74	XFORTRANSYMBOL
75	JOBSYMBOL
77	VFORTRANSYMBOL
78	SFORTRANSYMBOL
79	NEWPSYMBOL
80	SANSSYMBOL
81	PASCALSYMBOL
82	RPGSYMBOL

83	NDLIISYMBOL
84	FORTRAN77SYMBOL
85	SORTSYMBOL
86	COBOL74SYMBOL
87	CCSYMBOL
89	COBOL85SYMBOL
91	PASCAL83SYMBOL
92	JAVASYMBOL
94	BINDERSYMBOL
95	DASDLSYMBOL
96	DMALGOLSYMBOL
97	DCPSYMBOL
98	NDLSYMBOL
99	LOG
104	MDLCODE
105	MDLSYMBOL
106	VFORTRANCODE
107	VLINKEDCODE
108	VMCPCODE
109	LCOBOLSYMBOL
110	LCOBOLSL3CODE
111	LCOBOLSL5CODE
112	SFORTRANCODE
113	SLINKEDCODE
114	SMCPCODE
115	FIRMWARE
116	OHNESYMBOL
165	MDTTEST
166	MDTDUMP
167	PROMBURNERDATA
168	CONFIGURATIONDATA
169	CONFIDENCECODE
176	MP1FIRMWARE
177	MP1ALGOLCODE
181	FONT
182	FORM
183	GRAPHICS
184	LIBMAINTDIRECTORY
185	WRAPPEDDATA
186	CONTAINERDATA
190	PERMDIR
191	BDDATA
192	DATA
193	SEQDATA
194	GUARDFILE
195	APLFILE or APLDATA
196	APLWORKSPACE or APLWORKSOURCE
197	CDATA
198	CSEQDATA
199	DBRESTARTSET
200	DBDATA
201	TEXTDATA
203	FTPDATA
204	PDUMPFILE
205	FOLDER
206	MSGDRAFT
207	MAILMSGSTORE
208	JAVACLASS
238	DEVNULL
240	FIF0

Note: *The FILEKIND values 241 through 255 are available for customer use. All other values are reserved.*

FILEKIND Classes for Disk Files

The following table groups FILEKIND values into classes. These classes are used to identify restrictions on FILEKIND value changes in subsequent text.

Class	FILEKIND Value	
System file	0 (NULLFILE) through 20 (COMPILERCODEFILE), and 99 (LOG)	
Compiler	20 (COMPILERCODEFILE)	
Code file	20 (COMPILERCODEFILE) through 25 (NDLIICODE), and 28 (INFOFILE) through 63 (CODEFILE)	
Data file	64 (ALGOLSYMBOL) through 189, and 191 (BDDATA) through 207 (MAILMSGSTORE)	
Symbolic file	64 (ALGOLSYMBOL) through 98 (NDLSYMBOL)	
Special files	Permanent Directories	190 (PERMDIR)
	POSIX Files	232 through 240 (FIFO)
Customer	241 through 255	

Disk Files Restrictions

Based on the classes identified in the previous table, the following restrictions exist for disk files:

- A user program can change the FILEKIND value of a file that is in the data file class to another FILEKIND value in the data file class.
- A user program cannot change the FILEKIND value of a file in the system file class or an in-use file in the code file class to a FILEKIND value in the data file class.
- A user program can change the FILEKIND value of a file in the code file class that is not in-use to a FILEKIND value in the data class.
- A compiler program can change a data file to a code file through the use of file declarations, run-time attribute change statements, or both. However, a compiler cannot create any code files on another host when Heritage Host Services Logical I/O is being used.
- Only the operating system can change or assign a FILEKIND value to a file in the system file class or assign a FILEKIND value to a file in the compiler file class.
- Once a file is created with a FILEKIND value in the Special File class, the value of the FILEKIND attribute cannot be changed.
- The FILEKIND values in the Customer class will not be used by Unisys software products.
- Only programs with a message control system (MCS) status can create files with a FILEKIND value of RECOVERYFILE or SCHEDULEFILE.
- If the FILEKIND attribute value of a file is currently equal to or greater than 20—for example, COMPILERCODEFILE and not JOBCODE---an MCS can, after setting the

HSFILECOPY attribute value to TRUE, set the FILEKIND value of the file to most values shown in the previous table. The value for FILEKIND cannot be set to the RESTRICTEDFILE, KEYSFILE, or LOG values.

- Modifying an existing disk file with a FILEKIND of BACKUPPRINTER can be accomplished if the program declaring and using the file has privileged status. The program must modify the file in such a way that it can still be correctly interpreted by the Print System.

User programs have the following restrictions:

- They cannot write to files classed as code files, compilers, or system files. Refer to the previous table for class information.
- They cannot open or access disk files with a FILEKIND value of FTAUDIT.
- If the system security option MODIFYGUARDFILE value is NOTOK, they cannot write to existing files with a FILEKIND value of GUARDFILE. (A user program can create a file with a FILEKIND value of GUARDFILE.)

Other Restrictions

When creating or accessing a File Transfer, Access, and Management (FTAM) file, use the DATA or CDATA values only. If other values are used, an error message is issued and the value is changed to DATA.

Special Files Restrictions

When a disk file with one of these FILEKIND values is opened and certain other attributes of the permanent disk file and the declared file have the proper values, the opened file has special semantics. To obtain these special semantics, the attribute settings (other than FILEKIND) required for the permanent disk file are:

- BLOCKSTRUCTURE is FIXED
- EXTMODE is EBCDIC or some other 8 bit character type
- FILEORGANIZATION is NOTRESTRICTED
- FILESTRUCTURE is STREAM
- FRAMESIZE is 8 or UNITS is not WORDS
- LASTRECORD is –1 (the file is empty)
- MAXRECSIZE is 1

Files with FILEKIND equal to PERMDIR are used for permanent directories. The attribute settings required for the declared file are

- ANYSIZEIO is TRUE
- BLOCKSTRUCTURE is FIXED
- FILEORGANIZATION is NOTRESTRICTED
- FILESTRUCTURE is STREAM
- FRAMESIZE is 8 or UNITS is not WORDS
- INTMODE is EBCDIC or some other 8 bit character
- MAXRECSIZE is 1 (one)
- The logical file cannot be a DIRECT file
- UPDATEFILE is FALSE

The special semantics that are applied when all of the above conditions are met cause the file to be treated as a permanent directory. Data can not be read from or written to a permanent directory. For more information on the use of permanent directories, refer to the *System Operations Guide*.

Files with FILEKIND values in the POSIX files class are used by the POSIX implementation. The attribute settings required for the declared file are

- ALLOWSPECIALFILE is TRUE
- ANYSIZEIO is TRUE
- BLOCKSTRUCTURE is FIXED
- FILEORGANIZATION is NOTRESTRICTED
- FILESTRUCTURE is STREAM
- FRAMESIZE is 8 or UNITS is not WORDS
- INTMODE is EBCDIC or some other 8 bit character
- MAXRECSIZE is 1 (one)
- The logical file cannot be a DIRECT file
- UPDATEFILE is FALSE

The special semantics that are applied when all of the conditions above are met cause the data read from or written to one of these files to be redirected to some other entity, depending on the following values of FILEKIND:

DEVNULL This value identifies a file that discards any data written to the file and returns an end-of-file result if an attempt is made to read from the file. This is similar to the behavior of a DISK file with the DUMMYFILE attribute set to TRUE.

FIFO This value identifies a file that redirects data written to the file to a first-in, first-out queue that can be opened and read by other processes running on the same host. Although FIFO files have headers on disk, their data storage areas exist only in memory.

For more information about the use of POSIX special files, refer to the *I/O Subsystem Programming Guide.*

FILEKIND Values and Formats

Files that have certain FILEKIND values are generally assumed to have particular record formats. The following table lists the FILEKIND values and their assumed formats:

FILEKIND Value	Format Model	Text Field	Sequence Field	ID Field	Minimum Record Length in Characters
ALGOLSYMBOL	ALGOL	1-72	73-80	81-90	80
BASICSYMBOL	BASIC	5-72	1-4	73-80	72
BINDERSYMBOL	FORTRAN	1-72	73-80	N/A	80
CCSYMBOL	ALGOL	1-72	73-80	81-90	80
COBOLSYMBOL	COBOL	7-72	1-6	73-80	72
COBOL74SYMBOL	COBOL	7-72	1-6	73-80	72
COBOL85SYMBOL	COBOL	7-72	1-6	73-80	72
CSEQDATA	CSEQDATA	7-80	1-5	N/A	N/A
DASDLSYMBOL	ALGOL	1-72	73-80	81-90	80
DCALGOLSYMBOL	ALGOL	1-72	73-80	81-90	80
DCPSYMBOL	ALGOL	1-72	73-80	81-90	80
DMALGOLSYMBOL	ALGOL	1-72	73-80	81-90	80
ESPOLSYMBOL	ESPOL	1-72	73-80	81-88	80
FORTRANSYMBOL	FORTRAN	1-72	73-80	N/A	80
FORTRAN77SYMBOL	ALGOL	1-72	73-80	81-90	80
JAVASYMBOL	ALGOL	1-72	73-80	81-90	80
JOBSYMBOL	JOB	1-80	83-90	N/A	90
LCOBOLSYMBOL	COBOL	7-72	1-6	73-80	72

FILEKIND Value	Format Model	Text Field	Sequence Field	ID Field	Minimum Record Length in Characters
NDLSYMBOL	ALGOL	1-72	73-80	81-90	80
NDLIISYMBOL	ALGOL	1-72	73-80	81-90	80
NEWPSYMBOL	ALGOL	1-72	73-80	81-90	80
PASCALSYMBOL	ALGOL	1-72	73-80	81-90	80
PASCAL83SYMBOL	ALGOL	1-72	73-80	81-90	80
PLISYMBOL	ALGOL	1-72	73-80	81-90	80
RPGSYMBOL	RPG	6-80	1-5	81-90	80
SANSSYMBOL	ALGOL	1-72	73-80	81-90	80
SEQDATA	ALGOL	1-72	73-80	81-90	80
SORTSYMBOL	ALGOL	1-72	73-80	81-90	80
TEXT	ALGOL	1-72	73-80	81-90	80
XFORTRANSYMBOL	FORTRAN	1-72	73-80	N/A	80

Note: *CSEQDATA files have an arbitrary length; the text field extends through the end of the record.*

The FILEKIND attribute returns the value RESTRICTEDFILE if an existing system file, compiler, or code file is opened and one of the following conditions exists:

- The unit or volume on which the file resides was restricted by the RESTRICT (Set Restrictions) system command.
- The file itself was restricted by the RESTRICT system command.
- The file is an existing file on a remote host, and the HOSTSRESTRICTED option of the SECOPT system command is set on the local host.
- The file was created using Heritage Host Services logical I/O at a host with the HOSTSRESTRICTED option of the SECOPT system command set.
- The file was copied from a restricted unit or volume.
- The file was copied to a host that had the HOSTSRESTRICTED option of the SECOPT command set.

For information on restrictions of files, units, volumes, and hosts, refer to the *Security Administration Guide*.

The mnemonic value FTPDATA indicates that the file was copied with the File Transfer Protocol (FTP) mechanism. The file must be converted using the FTP utility before it can be viewed, printed, or processed. For information on FTP, refer to the *Distributed Systems Service (DSS) Operations Guide*.

FILELENGTH

Kinds	Disk or CD
Interrogate	When open
Modify	Never
Type	Real
Range	0 through 5.49755813887E + 11
Default	0
Abstract Value	Not applicable
Stored Permanently	Disk or CD
Parameters	None
FTAM File Access	Supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

Interrogate the FILELENGTH attribute to determine the length of a file in FRAMESIZE units. The length is counted from the first record position in the file to the end of the file, regardless of which records have been written or have had space allocated for them on the physical device. The total length does not include any unused bytes beyond the end of a block when that block is not an integral number of sectors.

The FILELENGTH value is evaluated by the I/O subsystem when the file is opened. When the file is extended, the FILELENGTH value is not updated. The LASTRECORD attribute is updated to include the added records.

The maximum size of a file in FRAMESIZE units is 549755813887. In the future, if larger files are supported, the value returned by the FILELENGTH attribute will be a floating-point number with 11 decimal digits of precision. For smaller files, the FILELENGTH attribute returns an integer value with complete precision.

FILENAME

Kinds	All
Interrogate	Anytime
Modify	Anytime (disk) When all subfiles are closed (port) When unassigned (all others)
Type	Character string
Range	See following text
Default	Value of INTNAME, if TITLE is not set
Abstract Value	DEFAULT
Stored Permanently	Disk, CD, tape
Parameters	None
FTAM File Access	Supported
Heritage Host Services Logical I/O	Supported
Port Services	BNANATIVESERVICE, OSINATIVESERVICE, OSISESSIONSERVICE, BASICSERVICE

Use the FILENAME attribute to specify or determine the external file name of a file. FILENAME is used primarily to identify the physical file. The default value for FILENAME is the INTNAME attribute value but only if the TITLE attribute is not set. The default of FILENAME is taken from the TITLE attribute, less any ON clause, if the TITLE attribute is set.

Note: *The FILENAME, LFILENAME, TITLE, LTITLE and PATHNAME file attributes are all used for naming a file. Each of these attributes has its own unique characteristics that makes it the right interface to use in certain circumstances. However, setting any one of these attributes causes a change in the value of all of the others. For disk files, the interpretation of the FILENAME attribute is also affected by the SEARCHRULE file attribute. For further information, see the "PATHNAME" and "SEARCHRULE" file attributes in this manual.*

Disk Files

The FILENAME and TITLE attribute values can differ. When the file is open, the TITLE value includes both the FILENAME and the FAMILYNAME values separated by the word ON, whereas the FILENAME value identifies only the name of the file.

You can change the FILENAME value at any time. Any changes to the FILENAME value while the file is assigned change the FILENAME value in the disk file header of the physical file. You can change the FILENAME value only if you own the file or if the task that is accessing the file is privileged.

If your program is creating a disk file, you can request that certain parts of the file name contain system-supplied character strings, such as the mix number or the system serial number of the system.

The MCP no longer supports the creation of duplicated disk files; however, copies of existing duplicated files can be accessed as non-duplicated files by specifying the full name of the copy with FILENAME.

The valid keywords that identify the system-supplied character strings follow. These keywords must be preceded by a displayable character that is specified by using the UNIQUETOKEN file attribute.

Keyword	Meaning
MIXNO	Current mix number
JOBNO	Current job number
HOSTNAME	Host name
JDATE	Julian date in YYYYDDD format
DATE	Current date in YYYYMMDD format
TIME	Current time in HHMMSS format
TIMEMIC	Ten digit current time in multiples of 2.4 microseconds, comparable to ALGOL TIME(6)
SYSTEMSERIAL	System serial number

The system-supplied character strings are provided at the time the FILENAME attribute is modified.

Non-Disk Files

FILENAME and TITLE return the same value.

The FILENAME value can be changed only when the file is unassigned.

Port Files

If the service supports this attribute, the value is used in the matching process and the naming of the dialogue endpoints of the subfile.

Remote Files

If the STATION task attribute value is 0 (zero), you can use this attribute to identify the name of the DATACOMINFO file that is to be assigned to the remote file. If a file with the FILENAME value is not found in the DATACOMINFO file, the FILENAME value becomes the first station in the station list.

The STATIONNAME and STATIONLIST file attributes are used to determine the name of a station in the station name list, and change the contents of the station name list, respectively.

Valid File Names in the MCP Environment

In SSR 44.2 and subsequent releases, the MCP supports two forms of file names: “traditional” file names and “long” file names. Long file names should only be used under restricted circumstances and are primarily intended for interoperability with workstation programs that create files in the MCP environment through Client Access Services. Long file names are specified using the LFILENAME file attribute. The SYSOPS option LONGFILENAMES determines if long file names are allowed. Refer to “Long File Name Features” in the *System Administration Guide* for more information on long file names.

The MCPSUPPORT library procedure TITLELIMITS can be used by applications to determine whether the SYSOPS LONGFILENAMES option is set or reset and to obtain information about title limits in display, standard, and path forms. Refer to “TITLELIMITS” in the *MCP System Interfaces Programming Manual* for more information on how to call TITLELIMITS.

Both forms of file names in the MCP environment contain an optional usercode in parentheses or an optional asterisk (*) followed by one or more nodes. Adjacent nodes are separated by slashes (/). A usercode and a node consist of an identifier with the following characteristics:

- Any combination of EBCDIC uppercase letters A through Z or EBCDIC digits 0 through 9. Additionally, a hyphen (-) or an underscore (_) can be included in the identifier, but neither of those characters can be the first character.
- An identifier enclosed in quotation marks ("). The EBCDIC characters can be any character that has a hexadecimal code greater than or equal to a hexadecimal 4"40" and cannot be an EBCDIC character quotation mark.
- A keyword that identifies a system-supplied character string preceded by a displayable character that is specified with the UNIQUETOKEN file attribute value. Refer to “Disk Files” in this subsection for a list of valid keywords.
- Quotation marks do not count toward the limits.
- Usercodes of 1 to 17 characters. Usercodes longer than 17 characters are truncated to 17 characters.

Traditional file names in the MCP environment have the following limits:

- A maximum of 12 nodes.
- A maximum of 17 characters per node.
- Any node longer than 17 characters is invalid.
- Any file name with more than 12 nodes is invalid.

If the SYSOPS LONGFILENAMES option is reset, long file names have the same limits as traditional file names.

If the SYSOPS LONGFILENAMES option is set, long file names have the following limits:

- A maximum of 20 nodes.
- A maximum of 215 characters in all nodes, including the slash characters.
- Any long file name with more than 215 characters in the nodes is invalid.
- Any long file name with more than 20 nodes is invalid.

The primary enhancement provided by long file names is the support of nodes that are longer than 17 characters; the total supported long file name length is not much larger than that of traditional file names.

When read, if the file is associated with a physical file, the FILENAME attribute returns the file name, traditional or long, of the physical file. If not associated with a physical file, the FILENAME attribute returns the last file name, traditional or long, that was assigned. Because a long file name can be returned, the longest possible file name that can be returned is increased from 260 to 276 characters. Note that this does not include the period (.) many languages append when reading this attribute.

A long file name that does not conform to the traditional file name limits can be assigned to the LFILENAME attribute, but not to the FILENAME attribute. When read, the FILENAME and LFILENAME attributes always return the same value, even when the value is a long file name that cannot be assigned to the FILENAME attribute by a user program. For more information on file name limits, refer to TITLE_LIMITS in the *Master Control Program (MCP) System Interfaces Programming Reference Manual*.

Examples

The following are examples of valid traditional file names in the MCP environment:

- ACCOUNTING/RECEIPTS/021689
- PERSONNEL/''EMPLOY.LIST''
- (KELLY)DOCUMENTS/CLIENT/6
- *SYSTEM/LOGANALYZER
- ACCOUNTING/''@JOBNO''/''@MIXNO''/RECEIVABLES (For disk files only)

The following are examples of valid long file names in the MCP environment where the SYSOPS LONGFILENAMES option is set:

- (MEMYSELFANDI)1/2/3/4/5/6/7/8/8/10/12/13/14/15/16/17/18/19/20
- THISISAVERYVERYVERYLONGFILENODE/ANDTHISISANOTHERONE
- (''17 character name'') ''This is a more readable''/''very,very,very,long''/''filename''
- ACCOUNTING/''@JOBNO''/''@MIXNO''/RECEIVABLES (For disk files only).

Valid Tape File Names in the MCP Environment

A tape file name in the MCP environment has the same characteristics as a file name in the MCP environment, except that the file name written to the tape label only contains the first node and the last node of the name. These file name nodes are a maximum of 17 characters in length. If more than one file resides on a tape, the first node must be the same for all files and the second node must be different for each file.

The following are examples of valid tape file names in the MCP environment:

- ACCOUNTING
- ACCOUNTS/MASTER
- STUDENT/GRADES
- STUDENT/CLASSES

Valid Port File Names

A port file name has a 1 to 17 character node. That node can contain an identifier with the following characteristics:

- Any combination of EBCDIC uppercase letters A through Z or EBCDIC digits 0 through 9.
- A hyphen (-) or an underscore (_) can be included in the identifier, but neither of those characters can be the first character.

The following are examples of valid port file names:

- RESERVATIONS
- ''RESERVATION_REQ''

Valid Names for Disk Files on Systems Outside the MCP Environment

When you are accessing a system that is not an MCP based system through a network and the file name is not valid in the MCP environment, enclose the name in apostrophes ('). Before the I/O subsystem sends the file name information to the remote host, the apostrophes are removed and all other characters are sent to the remote host exactly as they are. The name can contain a maximum of 250 EBCDIC characters, and the EBCDIC characters can be any character that has a hexadecimal code greater than or equal to a hexadecimal 4"40" and cannot be an EBCDIC character quotation mark. A pair of apostrophes can be used to specify a single apostrophe in the file name.

The following are examples of valid foreign host file names:

- 'MY\FILE\031789'
- 'A:ACCOUNTS.PAY'
- '[Sys]<Sys>InstallSpl.Run'
- 'ACCOUNTING*PAYABLES.'

FILEORGANIZATION

Kinds	Disk or CD
Interrogate	Anytime
Modify	When closed
Type	Mnemonic
Range	See following text
Default	NOTRESTRICTED
Abstract Value	DEFAULT
Stored Permanently	Disk
Parameters	None
FTAM File Access	Restricted values
Heritage Host Services Logical I/O	Restricted values
Port Services	Not applicable

Use the FILEORGANIZATION attribute to specify the organization of the file. The following are the mnemonic and integer values and meanings associated with the FILEORGANIZATION attribute:

Mnemonic Value	Integer Value	Meaning
INDEXED	2	The file is created with an indexed file organization and it is implemented through the KEYEDIO library. For more information about KEYEDIO, refer to the *System Software Utilities Operations Reference Manual*.
INDEXEDNOTRESTRICTED	3	The file is created with relative keys and can have indexed keys. It can be accessed when the logical file is declared as NOTRESTRICTED, INDEXED, or INDEXEDNOTRESTRICTED. This value is implemented through the KEYEDIO library. For more information about KEYEDIO, refer to the *System Software Utilities Manual*.
KEYEDIOII	5	The physical file is created as a KEYEDIOII data file, and it is implemented through the KEYEDIOII library. For information about creating and using KEYEDIOII files, refer to the *KEYEDIOII Programming Reference Manual*.

Mnemonic Value	Integer Value	Meaning
KEYEDIOIISET	6	The physical file is created as a KEYEDIOII index file, and it is implemented through the KEYEDIOII library. For information about KEYEDIOII, refer to the *KEYEDIOII Reference Manual*.
NOTRESTRICTED	0	No restrictions are applied.
RELATIVE	1	The file is created with a relative file organization. This organization is derived from the COBOL language.

Double-byte (16-bit) and mixed multibyte (mixed 8-bit and 16-bit with FRAMESIZE = 8) character sets are supported only when the value of FILEORGANIZATION is NOTRESTRICTED.

The NOTRESTRICTED and the RELATIVE mnemonics are the only values that can be used if a file has a FILESTRUCTURE value of STREAM. The NOTRESTRICTED mnemonic is the only value that can be used if a file:

- Is to be accessed or created using Heritage Host Services Logical I/O
- Is to be accessed or created using File Transfer, Access, and Management (FTAM)
- Has a FILESTRUCTURE value of BLOCKED

Direct I/O files do not use the KEYEDIO or KEYEDIOII library.

The FILEORGANIZATION values for the physical file and the logical file must be compatible when existing files are accessed. An open error occurs if these values are incompatible. The compatible values are as follows:

Physical File Value	Compatible Logical File Value
INDEXEDNOTRESTRICTED, KEYEDIOII, and KEYEDIOIISET	Any value other than RELATIVE (However, the physical file value is used.)
INDEXED	INDEXED, INDEXEDNOTRESTRICTED, or direct file
RELATIVE	RELATIVE or direct file
PLIISAM	PLIISAM
NOTRESTRICTED	NOTRESTRICTED

The value of FILEORGANIZATION does not affect the value of the RESIDENT attribute.

Specifying the DEPENDENTSPECS attribute as TRUE or the FILETYPE attribute as 7 or 8 does not alter the value of FILEORGANIZATION when the file is opened.

The FILEORGANIZATION attribute returns the value for the physical file if interrogated when the file is assigned. When the run-time KEYEDIOII system option is set, creating a new file with a logical file FILEORGANIZATION value of INDEXED or INDEXNOTRESTRICTED results in a permanent file with a FILEORGANIZATION value of KEYEDIOII.

FILESECTION

Kinds	Tape
Interrogate	Anytime
Modify	When closed
Type	Integer
Range	1 through 9999
Default	1
Abstract Value	DEFAULT, MINIMUM, MAXIMUM
Stored Permanently	Tape
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

For a multireel tape volume set, use the FILESECTION attribute to determine or specify the file-relative volume number of the volume currently in use or to be used. The file-relative volume number of the volume on which a file begins is always 1. The number is incremented each time the file switches to a new volume. The semantics of this attribute are established by the American National Standards Institute (ANSI) tape label format standard.

For tape files, the file section number indicates the part or section of a tape file which is in use or is located on a given tape volume. The following illustrates how this term may be used:

- If a tape file is totally contained on one tape volume, the file section number of the entire file is 1 (FILESECTION = 1).
- If a tape file is contained on more than one tape volume, the file section number of the part or section on each volume indicates which part or section of the file comes first, which section comes second, etc. The file section number of that part of the file on the second volume is 2 (FILESECTION = 2).
- If a multi-file, multi-reel volume set is created in which the first file occupies all of the first volume, and continues onto the second, and the second file occupies the remainder of the second volume and continues on to the third, then FILESECTION=1 for the second file is physically located on the second volume of the set and FILESECTION=2 for the second file is physically located on the third volume of the set.

When FILESECTION is interrogated, a value of 1 is returned if the file is closed or if an operator has entered the FR (Final Reel) system command in response to a waiting message that requested the next volume. Refer to the *System Operations Guide* for information about using the FR command.

OPEN uses the FILESECTION value in tape file assignments along with the KIND and FILENAME attribute values and, when appropriate, the CYCLE and VERSION attribute values.

FILESTATE

Kinds	All
Interrogate	Anytime
Modify	Never
Type	Mnemonic
Range	See following text
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	No
Parameters	1 required (remote files) 1 optional (port files)
FTAM File Access	Supported
Heritage Host Services Logical I/O	Supported
Port Services	All

Interrogate the FILESTATE attribute to determine the logical state of the file.

For a remote file, the relative station number (RSN) parameter must be used. If the number is invalid, an attribute error occurs.

For a port file with more than one subfile, the subfile index parameter must be used. When you interrogate the FILESTATE attribute for a port subfile, the CHANGEEVENT happened state is reset. When you interrogate the CHANGEDSUBFILE attribute, a 0 (zero) is returned if the CHANGEEVENT happened state of all the port subfiles are reset. For information on FILESTATE transitions, refer to information about opening and closing port files in the *I/O Subsystem Programming Guide*.

The following tables list the FILESTATE values and their meanings:

Mnemonic Value	Integer Value	Meaning
CLOSED	0	Indicates that the file is closed. For remote files, this file state indicates that the specified relative station is closed. For port files, this file state indicates that no dialogue is in progress. Dialogue establishment is not being initiated nor is dialogue establishment being awaited. A READ and WRITE operation returns an EOF indicator in this file state.

Mnemonic Value	Integer Value	Meaning
OPENED	3	Indicates for files other than remote files and port files that the file is opened. For remote files, this file state indicates that the specified relative station is opened. For port files, this file state indicates that the service provider has established a dialogue for the subfile with a correspondent application. Both READ and WRITE operations are legal during this file state. Depending on the service provider, the subfile remains in this state until one of the following events occurs: Communication with the correspondent application is interrupted for some reason. Urgent data is queued for the subfile. The local system notifies the subfile that shutdown is in progress. The local application issues a CLOSE ORDERLY or ABORT statement. The correspondent application issues a CLOSE ORDERLY or ABORT statement. The service provider on either end of the dialogue issues an ABORT statement.

Mnemonic Value	Integer Value	Meaning
DENIED	9	Indicates that the file assignment of the remote file was rejected.
DENIEDILLEGALUSE	11	Indicates that an illegal OPEN operation for a remote file was executed.
OFFERED	2	Indicates that the user program has just added the specified RSN to the remote file via an open request or the STATIONLIST attribute. The station (RSN) has been offered for addition to the remote file and to its controlling MCS. The MCS has not yet responded with the disposition of the additional request.
POSTPONED	10	Indicates that the file assignment for the specified remote file has been postponed.

Mnemonic Value	Integer Value	Meaning
AWAITINGHOST	1	Indicates that the host specified by the YOURHOST subfile attribute cannot be reached at this time, or it indicates that the provider of the service specified by the SERVICE attribute is not available at this time. Once communication with the specified host or service provider is possible, the FILESTATE value is changed to OFFERED or AWAITINGOFFER, and the dialogue establishment continues for the subfile. If the host becomes unavailable again, the FILESTATE value is changed to AWAITINGHOST and the dialogue establishment is suspended. When the FILESTATE attribute has an AWAITINGHOST value, a READ or WRITE operation returns an EOF indicator.
AWAITINGOFFER	13	Indicates that the program invoked an AWAITOPEN operation and the subfile is awaiting an offer for dialogue establishment. When a matching OPEN indication is received, the FILESTATE value of the subfile is changed to OPENED or OPENRESPONSEPLEASE if the PARTICIPATE option is specified with an AWAITOPEN statement. When the FILESTATE attribute has an AWAITINGOFFER value, a READ or WRITE operation returns an EOF indicator.
BLOCKED	5	Indicates that data cannot be successfully transmitted to the correspondent application at this time. A subfile can enter this state for a variety of service provider-dependent reasons. When the FILESTATE attribute has a BLOCKED value, READ operations continue to be legal until queued messages do not remain for the subfile. When the FILESTATE attribute has a BLOCKED value, WRITE operations also continue to be legal; however, resources can be exhausted quickly, which results in a WRITE operation returning a no-buffer indication to the application.

Mnemonic Value	Integer Value	Meaning
CLOSECOLLRESPONSEPLEASE	14	Indicates that a collision of ORDERLY CLOSE requests occurred. The FILESTATE value of the subfile was CLOSEREQUESTED, but instead of receiving a response, an ORDERLY CLOSE request was received from the correspondent application. If the endpoint that received the ORDERLY CLOSE request is the initiator of the dialogue, the program must respond to the ORDERLY CLOSE request of the correspondent endpoint first by invoking a RESPOND ACCEPTCLOSE statement. The FILESTATE value of the subfile returns to CLOSEREQUESTED, and the subfile waits for the other endpoint to respond to its CLOSE request. When the response is received, the FILESTATE value of the subfile changes to CLOSED or CLOSEDINPUTPENDING. If the endpoint that received the ORDERLY CLOSE request is the responder—not the initiator—of the dialogue, the FILESTATE value of the subfile does not change to CLOSECOLLRESPONSEPLEASE until the initiator has responded to the CLOSE request of this endpoint. The FILESTATE value of the subfile changes from CLOSECOLLRESPONSEPLEASE to CLOSED or CLOSEDINPUTPENDING when the subfile invokes a RESPOND ACCEPTCLOSE statement. If FILESTATE has a value of CLOSECOLLRESPONSEPLEASE, a READ operation retrieves any queued read data and a WRITE operation returns an EOF indicator.
CLOSEDINPUTPENDING	16	Indicates that the underlying dialogue that supports the subfile is now CLOSED or CLOSEPENDING, but data to be received is still in the read queue. The FILESTATE value of the subfile changes to CLOSED (perhaps after spending a brief period of time in the CLOSEPENDING file state) as soon as the read queue is empty, or if the user program issues a CLOSEABORT request. When FILESTATE has a CLOSEDINPUTPENDING value, a WRITE operation returns an EOF indicator.

Mnemonic Value	Integer Value	Meaning
CLOSEPENDING	6	Indicates a CLOSE operation was invoked by the user program. The dialogue supported by the subfile is considered closed, but the CLOSE operation is still executing. The FILESTATE value of the subfile changes to CLOSED when the CLOSE operation completes. When the FILESTATE attribute has a CLOSEPENDING value, a READ or WRITE operation returns an EOF indicator.
CLOSEREQUESTED	17	Indicates that the local user program has issued an ORDERLY CLOSE request for this subfile and the user program is waiting for the response to its request. If a positive response is received, the FILESTATE value of the subfile changes to CLOSEDINPUTPENDING or CLOSED. If a negative response is received, the FILESTATE value returns to OPENED. When the FILESTATE attribute has a CLOSEREQUESTED value, a READ operation can be initiated and a WRITE operation returns an EOF indicator. Rejection of ORDERLY CLOSE requests is not yet supported.
CLOSEREQUESTRECEIVED	19	Indicates that the correspondent application has finished sending data and wants to terminate the dialogue. The user program should finish processing and issue a corresponding CLOSEREQUEST request. Both READ and WRITE operations are valid in the file state.

Mnemonic Value	Integer Value	Meaning
CLOSERESPONSEPLEASE	15	Indicates that the correspondent application has sent an ORDERLY CLOSE request to the dialogue and the user program is being prompted for a response. When the program responds by invoking the RESPOND ACCEPTCLOSE request for OSI services, or when the program responds by invoking the CLOSE ORDERLY request for TCPIPNATIVESERVICE, the FILESTATE value changes to CLOSEDINPUTPENDING or CLOSED. When the FILESTATE attribute has a CLOSERESPONSEPLEASE value, both the READ and WRITE operations are legal. The RESPOND REJECTCLOSE request is not yet supported.
DEACTIVATED	8	Indicates that the dialogue was established for the subfile and was terminated by a correspondent-initiated ABORT request or provider-initiated ABORT request, and that there are no messages queued for the subfile to read. The subfile remains in this state until the user program issues a CLOSE ABORT statement. When the FILESTATE attribute has a DEACTIVATED value, READ and WRITE operations return an EOF indicator.
DEACTIVATIONPENDING	7	Indicates that the dialogue was established on the subfile and was terminated by a correspondent-initiated ABORT request or a provider-initiated ABORT request, and there are messages queued for the subfile to read. The subfile remains in this state until all the information has been read, at which point the FILESTATE value changes to DEACTIVATED, or until the user program issues a CLOSE ABORT statement. When the FILESTATE attribute has a DEACTIVATIONPENDING value, READ operations are legal and WRITE operations return an EOF indicator.

Mnemonic Value	Integer Value	Meaning
OFFERED	2	Indicates that the user program has invoked an OPEN operation for the subfile and a dialogue request has been sent for the offered subfile. If the OPEN operation succeeds, the FILESTATE value of the subfile changes to OPENED. If the OPEN operation fails or if a provider- or correspondent-initiated ABORT request is received, the FILESTATE value of the subfile changes to CLOSED or CLOSEDINPUTPENDING. If a change in the status of the host to which the OPEN request was directed is detected, the FILESTATE value of the subfile changes to AWAITINGHOST. When the FILESTATE attribute has an OFFERED value, a READ or WRITE operation returns an EOF indicator.
OPENRESPONSEPLEASE	18	Indicates that the subfile had a FILESTATE value of AWAITINGOFFER because an AWAITOPEN PARTICIPATE statement was invoked and now the subfile has a matched OPEN indication. The user program is being prompted for a response. The subfile can interrogate and modify attributes such as APPLICATIONCONTEXT that can be negotiated or exchanged when the subfile is responding to a dialogue request. The subfile can then respond with ACCEPTOPEN request, which causes the FILESTATE value to change to OPENED, or REJECTOPEN, which then changes the FILESTATE value to CLOSED or CLOSEDINPUTPENDING. If an ABORT request is received, the FILESTATE value of the subfile is changed to CLOSED or CLOSEDINPUTPENDING. When FILESTATE has an OPENRESPONSEPLEASE value, a READ operation retrieves OPENDATA from the read queue if user data arrived with the dialogue request. Otherwise, the READ operation returns an EOF indicator. A WRITE operation returns an EOF indicator.

Mnemonic Value	Integer Value	Meaning
SHUTTINGDOWN	4	Indicates that the local system operator has requested a shutdown of networking services. When all subfiles on the system are closed, an orderly shutdown of the network commences. When FILESTATE has a SHUTTINGDOWN value, both READ and WRITE operations are legal. For a Transmission Control Protocol/Internet Protocol (TCP/IP) network, the URGENTDATAWAITING value of FILESTATE takes precedence over the SHUTTINGDOWN value.
URGENTDATAWAITING	12	Indicates that urgent data is queued for this subfile. The subfile remains in this file state until all urgent data for this subfile has been read by the local user program. When the FILESTATE attribute has an URGENTDATAWAITING value, the READ and WRITE operations are legal.

FILESTRUCTURE

Kinds	CD, Disk, or Printer
Interrogate	Anytime
Modify	When closed
Type	Mnemonic
Range	ALIGNED180, BLOCKED, and STREAM
Default	ALIGNED180 (disk)
Abstract Value	DEFAULT
Stored Permanently	Disk
Parameters	None
FTAM File Access	Restricted values
Heritage Host Services Logical I/O	Restricted values
Port Services	Not applicable

Use the FILESTRUCTURE attribute to determine or specify the structure of a disk file. You can provide a value for this attribute for a file that is not a disk file, in anticipation of a possible file equation to a disk file.

The following table lists the possible mnemonic and integer values of FILESTRUCTURE and their meanings:

Mnemonic Value	Integer Value	Meaning
ALIGNED180	0	The file is divided into fixed-length blocks. Each block begins at a modulo 180-byte boundary. The file can be accessed with a BLOCKSIZE attribute set to a value different from that with which it was created. This attribute value is intended to ensure compatibility with programs that depend, explicitly or implicitly, on the division of the file into 180-byte sectors. ALIGNED180 is the default value.
BLOCKED	5	The file is divided into fixed-length blocks. Each block begins at a physical disk sector boundary. The file must be accessed with a logical BLOCKSIZE attribute value equal to the physical BLOCKSIZE value.
STREAM	1	The file is a continuous stream of records and is not divided into blocks.

For files with a KIND attribute value of CD, the value of FILESTRUCTURE must be STREAM.

For backup files with a KIND attribute value of PRINTER, setting FILESTRUCTURE to STREAM results in the creation of delimited character-stream disk files, which can be read at a future time by other disk files and can be transferred to other operating systems with a mechanism such as FTP. Delimited backup files are created with the following attributes and values:

- FILECLASS = CHARACTERSTREAM
- EXTDELIMITER = CRCC
- FILEKIND = PRINTFILE

Although printer backup files are character-stream files, there is no change to the structure of the logical files. The files continue to be record-oriented and each block represents one print record. The MCP inserts the appropriate delimiters at the end of each block based upon the carriage control specified when writing the files:

- Double spacing is represented in printer backup files by an additional carriage return-line feed pair after the delimiters at the end of the first print line.
- A channel skip is represented by a carriage return-form feed pair, regardless of the destination channel.

Unlike files with a FILEKIND attribute value of BACKUPPRINTER, there is no embedded binary control information. Software translation is allowed and the delimiters are inserted in the EXTMODE character set. The EXTMODE attribute value must correspond to a character set that is EBCDIC-based, ASCII-based, UCS2, or UCSNT.

If BLOCKED or STREAM is selected, the following features are not supported:

- Update I/O with synchronization
- Binary I/O
- A BLOCKSTRUCTURE value of LINKED
- FILEORGANIZATION values other than NOTRESTRICTED when FILESTRUCTURE has a value of BLOCKED
- FILEORGANIZATION values other than NOTRESTRICTED or RELATIVE when FILESTRUCTURE has a value of STREAM
- Checkpoint or restart capabilities

When an existing file is opened, the FILESTRUCTURE value is used by the operating system to determine the assumptions made by the programmer regarding the capabilities required of the file being accessed. Careful consideration should precede the decision to file-equate the FILESTRUCTURE attribute. A permanent file with the requested FILESTRUCTURE value can be assigned to the logical file, or a permanent file with a different FILESTRUCTURE value can be assigned to the logical file as long as its FILESTRUCTURE value provides the capabilities that are required by the program. The value returned by the FILESTRUCTURE attribute, however, does not differ from the specification of the program unless DEPENDENTSPECS has a value of TRUE.

The values of other attributes of the logical file can prevent file assignment. In particular, the allowable values of the MAXRECSIZE and BLOCKSIZE attributes depend on the combination of FILESTRUCTURE values. The following table summarizes when a file can be assigned:

	Permanent File Value		
Logical File Value	**ALIGNED180**	**BLOCKED**	**STREAM**
ALIGNED180	Allowed	Allowed, if the logical file meets BLOCKED conditions.	Not allowed
BLOCKED	Allowed; BLOCKED conditions are checked.	Allowed	Not allowed
STREAM	Allowed, unless attributes valid only for files with a FILESTRUCTURE value of STREAM are used or the BLOCKSTRUCTURE value is EXTERNAL.	Allowed, unless attributes valid only for files with a FILESTRUCTURE value of STREAM are used or the BLOCKSTRUCTURE value is EXTERNAL.	Allowed

An explanation of each combination follows:

- A permanent file that has a FILESTRUCTURE value of STREAM can be assigned only if the logical file has specified that value. This condition is true because the physical file has no blocks and specifying a FILESTRUCTURE value of ALIGNED180 or BLOCKED indicates that the program expects blocks.
- If the logical file has a FILESTRUCTURE value of BLOCKED, the operating system expects the program to adhere to the restrictions of that FILESTRUCTURE value. A permanent file with a FILESTRUCTURE value of ALIGNED180 can be assigned if the attributes that the program has specified are also valid for a file with a BLOCKED value. The method of access for a file with a FILESTRUCTURE value of ALIGNED180 or BLOCKED are very similar.

- If the logical file has a FILESTRUCTURE value of ALIGNED180, a permanent file with a FILESTRUCTURE value of BLOCKED can be assigned only if the specified logical BLOCKSIZE and MAXRECSIZE values are acceptable for a file with a FILESTRUCTURE of BLOCKED.
- If the logical file has a FILESTRUCTURE value of STREAM, a permanent file with a FILESTRUCTURE value of ALIGNED180 or BLOCKED can be assigned, provided that no other attributes unique to a STREAM file, such as ANYSIZEIO, have been specified and provided that the BLOCKSTRUCTURE attribute does not have a value of EXTERNAL.

When creating a direct I/O file, use the DIOFILESTRUCTURE attribute value to specify the FILESTRUCTURE value of the physical file. If you assign DIOFILESTRUCTURE the DEPENDENT or SECTORSTREAM value, you must also assign the FILESTRUCTURE attribute a value.

When an MCP host is the initiator, File Transfer, Access, and Management (FTAM) file access requires a FILESTRUCTURE value of STREAM. When an MCP host is the responder, FTAM file access supports all FILESTRUCTURE values.

Heritage Host Services logical I/O does not support a FILESTRUCTURE value of BLOCKED.

The following table shows what you must do or what the operating system does for each FILESTRUCTURE value when you set the BUFFERSHARING file attribute value to a value other than NONE.

FILESTRUCTURE Value	Action
ALIGNED180	The BLOCKSIZE and MAXRECSIZE file attribute values must be equal to the BLOCKSIZE and MAXRECSIZE values stored in the disk file header of the permanent file.
BLOCKED	The buffer size is calculated when a shared buffer pool is first set up for a given disk file and the calculated size applies to all users of that buffer pool.
STREAM	The MAXRECSIZE file attribute values must be equal to the MAXRECSIZE values stored in the disk file header of the permanent file. The buffer size is calculated when a shared buffer pool is first set up for a given disk file and the calculated size applies to all users of that buffer pool.

A program that specifies the DEPENDENTSPECS value as TRUE or the FILETYPE value as 7 or 8 is able to access files of any FILESTRUCTURE value. A warning is issued if the program interrogates attributes that are invalid for some FILESTRUCTURE values, unless the program has interrogated the FILESTRUCTURE attribute while the file was open since the file was last assigned.

FILETYPE

Kinds	All
Interrogate	When unassigned (Heritage Host Services logical I/O) Anytime (local host)
Modify	When closed
Type	Integer
Range	0 through 4; 6 through 8
Default	0
Abstract Value	DEFAULT
Stored Permanently	Disk or tape
Parameters	None
FTAM File Access	Restricted values
Heritage Host Services Logical I/O	Restricted values/usage
Port Services	All

You can use the FILETYPE attribute to specify the format of the records and the structure of the file. However, use of the BLOCKSTRUCTURE attribute is preferred for files with a FILETYPE value of 0, 1, 2, 3, 4, or 6.

The preferred attributes BLOCKSTRUCTURE and DEPENDENTSPECS can override and change the value of FILETYPE. If neither BLOCKSTRUCTURE nor DEPENDENTSPECS has been specified, FILETYPE determines the structure of the logical file. An attribute error is reported if a value of DEPENDENTSPECS has been specified and an attempt is made to modify the FILETYPE value to 7 or 8.

The values and meanings associated with the FILETYPE attribute values 7 and 8 are listed in the following table. Refer to the discussion of BLOCKSTRUCTURE for the descriptions of the 0, 1, 2, 3, 4, or 6 values.

Value	Meaning
7	The format of the records and the structure of the logical file are to be determined by the structure of the associated permanent file—that is, the attributes BLOCKSIZE, BLOCKSTRUCTURE, FILETYPE, FILESTRUCTURE (unless a direct I/O file), INTMODE, MAXRECSIZE, MINRECSIZE, SIZEMODE, SIZEOFFSET, SIZE2, and UNITS are changed to the values used in the permanent file. The FRAMESIZE attribute is not changed, but it can be changed by setting the DEPENDENTSPECS attribute to TRUE. The INTMODE attribute is changed to the mode of the permanent file, which was, in fact, the EXTMODE value of the logical file that created the permanent file. If no permanent file is associated with the logical file—that is, if a new file is being created—FILETYPE is set to 0 (zero) or 3, depending upon the initial value of MINRECSIZE, and all the attributes mentioned previously are set to their default values if the attributes were not explicitly set before the file was opened. Setting the DEPENDENTSPECS and DEPENDENTINTMODE attributes to TRUE is preferable to using the value 7 for FILETYPE.
8	The action is the same as for a FILETYPE value of 7, with the following exceptions: the INTMODE value of the logical file is not changed when a permanent file is associated with the logical file, and, if a new file is being created, the attributes BLOCKSIZE, BLOCKSTRUCTURE, FILETYPE, FILESTRUCTURE (unless a direct I/O file), MAXRECSIZE, MINRECSIZE, SIZEMODE, SIZEOFFSET, SIZE2, and UNITS are changed to their default values. Setting the DEPENDENTSPECS attribute to TRUE is preferable to using the value 8 for FILETYPE.

Heritage Host Services logical I/O does not support FILETYPE values 7 and 8, while File Transfer, Access, and Management (FTAM) supports both those values.

Port Files

Attempting to open a port file with an unsupported FILETYPE value results in an open error.

FILEUSE

Kinds	All except port
Interrogate	Anytime
Modify	When closed
Type	Mnemonic
Range	See following text
Default	IO
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	None
FTAM File Access	Supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Use the FILEUSE attribute to specify the way in which the file is to be used. FILEUSE is used for remote files and for disk file security checking. The mnemonic and integer values, and their meanings for the FILEUSE attribute are as follows:

Mnemonic Value	Integer Value	Meaning
IN	1	Input only
IO	3	Both input and output
OUT	2	Output only

If the value of FILEUSE is IN and a WRITE operation is performed, a fatal run-time error occurs, the program is terminated, and the message “WRITE ON INPUT FILE” is displayed. Similarly, if the value of FILEUSE is OUT and a READ operation is performed, a fatal run-time error occurs and the message “READ ON OUTPUT FILE” is displayed. In some languages, such as COBOL85, the errors are not fatal, but an exception result of WRITEONINPUT (118) or READONOUTPUT (119) for the I/O operation is returned in the I/O result descriptor.

Unlike the MYUSE attribute, the values of the FILEUSE attribute have no effect on the file search algorithms—that is, whether to search for an existing file or create a new one.

For information about using the FILEUSE attribute when creating or accessing File Transfer, Access, and Management (FTAM) files, refer to the *I/O Subsystem Programming Guide*.

FLEXIBLE

Kinds	Disk or printer (if backup)
Interrogate	Anytime
Modify	Anytime
Type	Boolean
Range	TRUE, FALSE
Default	TRUE
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Set the FLEXIBLE attribute to FALSE to indicate that the number of areas originally specified by the AREAS attribute cannot be increased. Refer to the description of the AREAS attribute for more information.

When the file expands beyond the AREAS value and if FLEXIBLE is not set to FALSE, the system increases AREAS by ten percent (10%) over the area number of the area the system is about to allocate. The value of the FLEXIBLE attribute is not retained by the physical file, and the action of the I/O subsystem is controlled by the setting of the attribute in the logical file that currently has the file open. The setting of FLEXIBLE is ignored if the file has been crunched.

FORMID

Kinds	Disk and printer
Interrogate	Anytime
Modify	When closed
Type	Character string
Range	Not applicable
Default	Null string
Abstract Value	DEFAULT
Stored Permanently	Backup file
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Use the FORMID attribute to specify an identifier that contains up to 100 characters. This identifier is used to indicate the kind of paper that must be used to print a file. If you specify the FORMID file attribute for a backup file, the file can be printed only on a device configured with the same FORMID or the special value *DONTCARE*. For information about configuring a device for special forms, refer to the *Print System Guide*.

If the FORMID attribute is specified for the following types of files and a device with a matching form identification is not available, then the string assigned to the FORMID attribute is displayed as part of the message requesting operator action. These files include

- Direct I/O files and other printer files with a PRINTDISPOSITION value of DIRECT when the value of the DIRECTPRINTER option of the SYSOPS system command is DIRECTDLP
- Files with a PRINTDISPOSITION value of DIRECTDLP

The program is suspended until a device that has a matching string becomes available or until the operator responds with a Print Subsystem PS CONFIGURE system command or an FM (Form Message) system command. Refer to the *System Commands Reference Manual* for descriptions of these commands.

If the file is placed on a backup medium and no device with a matching string is available when a print request is made, the print request for the backup file is marked with the word WAITING in the print request list. Refer to the PS SHOWREQUESTS command in the *Print System Guide* for more information about displaying a print request list.

If the FORMID attribute is specified for an unlabeled backup file, header and trailer pages are suppressed when the file is printed. For example, if you create a backup file with the *WRITE <filename>:FORMID <formid>* CANDE command, the header and trailer pages are suppressed. To print header and trailer pages, perform one of the following before the print request is created:

- Set the HEADER and TRAILER print modifiers for the task or session to UNCONDITIONAL by using the *PDEF* CANDE command in the CANDE session or by using the PRDEF selection of the PS menu in MARC.
- Have the system administrator set the HEADER and TRAILER print modifiers in the PRINTDEFAULTS section of your USERDATAFILE entry. Refer to the *Security Administration Guide* for information about specifying print modifiers in the USERDATAFILE file.

FORMMESSAGE

FORMMESSAGE is a nonpreferred synonym for FORMID. Refer to the description of the FORMID attribute.

FRAMESIZE

Kinds	All
Interrogate	Anytime
Modify	When closed
Type	Integer
Range	4, 8, 16, 48
Default	Value corresponding to UNITS
Abstract Value	DEFAULT
Stored Permanently	Disk or tape
Parameters	None
FTAM File Access	Restricted values
Heritage Host Services Logical I/O	Restricted values
Port Services	All

Use the FRAMESIZE attribute to specify or determine the number of bits transferred as a unit of data during a programmatic I/O operation on a file. The following attributes are expressed in FRAMESIZE units:

- ACTUALMAXRECSIZE
- AREALENGTH
- BLOCKSIZE
- CURRENTBLOCKLENGTH
- CURRENTRECORDLENGTH
- FILELENGTH
- FRAMESIZECENSUS
- MAXFRAMESIZECENSUS
- MAXRECSIZE
- MINRECSIZE
- REQUESTEDMAXRECSIZE

The values and meanings for FRAMESIZE are as follows:

Value	Meaning
4	Data is transferred in units of 4 bits or as hexadecimal characters. The INTMODE attribute value must be HEX. This value is not supported by Heritage Host Services logical I/O or File Transfer, Access, and Management (FTAM), and is not valid for port files.
8	Data is transferred in units of 8 bits. INTMODE must not be BCL, BINARY, HEX, SINGLE, or any of the double-byte (16-bit) character sets. Eight (8) is the recommended value for TCPNATIVESERVICE. The value must be 8 for TCPIPNATIVESERVICE.
16	Data is transferred in units of 16 bits. The INTMODE value can only assume the values that correspond to double-byte (16-bit) character sets, such as UCS2NT(84) or UCS2(85). This value is not supported by Heritage Host Services Logical I/O or File Transfer, Access, and Management (FTAM) and is only valid for disk and printer backup files.
48	Data is transferred in units of 48 bits or as full words. This value is compatible with all values of INTMODE.

If the values of FRAMESIZE and INTMODE are not compatible when the file is opened, an open error is reported.

If FRAMESIZE is unspecified when the file is opened, a value is assigned as follows:

- If the value of UNITS is WORDS, FRAMESIZE is set to 48.
- If the file is a CD-ROM file, FRAMESIZE is always set to 8.
- In all other cases, FRAMESIZE is set to the number of bits in a character as defined by the INTMODE attribute.

If FRAMESIZE has a value, it is used regardless of the value of UNITS, and the UNITS value is changed to WORDS if the FRAMESIZE value is 48 or to CHARACTERS if the FRAMESIZE value is any other value. An attribute error is reported if FRAMESIZE has a value and an attempt is made to specify any value for UNITS.

If FRAMESIZE is interrogated when the file is open, the value is computed from the physical file and returned even if the logical file had no value specified.

FRAMESIZECENSUS

Kinds	Port or port subfile
Interrogate	Anytime
Modify	Never
Type	Integer
Range	Not applicable
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	TCPIPNATIVESERVICE, TCPNATIVESERVICE

Interrogate the FRAMESIZECENSUS attribute to determine the number of FRAMESIZE units currently in the input queue of a port file or port subfile.

FRAMESIZECENSUS is valid only for port files with a service that is not message oriented. For message-oriented services, use the CENSUS attribute. Because the services for which the FRAMESIZECENSUS attribute is valid require a FRAMESIZE value of 8, the attribute value is expressed in characters.

If the port file is open and no subfile index is given, the value returned is the sum of the characters queued for all subfiles in that file. If a subfile index is given, the value returned is the number of characters queued for the specified subfile only.

The range value of the FRAMESIZECENSUS attribute should be in the range of the MAXFRAMSIZECENSUS attribute. However, it is possible for the FRAMESIZECENSUS attribute to have a value greater than the maximum value for the MAXFRAMSIZECENSUS attribute, if the Stop Sending message is received after sending data that caused the range to exceed the maximum MAXFRAMESIZECENSUS value.

If the file is closed, the value 0 (zero) is returned.

GENERATION

Kinds	Disk or tape
Interrogate	Anytime
Modify	When closed
Type	Integer
Range	0 through any valid catalog level
Default	0
Abstract Value	DEFAULT, MINIMUM
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

If cataloging is enabled, use the GENERATION attribute in conjunction with the CYCLE and VERSION attributes to select a copy of a file other than the one with the latest time and date. The larger the value of GENERATION, up to the limit of the references specified by the catalog level of the system, the less recent the update or creation of the file.

Use the WM (What MCP) system command to determine the catalog level of a system.

The GENERATION attribute is ignored when the value of USECATALOG is FALSE.

GROUP

Kinds	Disk or tape
Interrogate	Anytime
Modify	Anytime (disk) When unassigned (tape)
Type	Character string
Range	Not applicable
Default	See following text
Stored Permanently	Disk or tape
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

The GROUP attribute accepts or returns a groupcode value or a null string. A null string indicates that the group attribute is not set. The related attribute, ALTERNATEGROUPS, provides the capability of associating additional groupcodes with a file.

Interrogating the GROUP attribute

Interrogating the GROUP attribute of a file returns one of the following values:

- If the attribute is interrogated while the logical file is assigned to a physical file, the value from the physical file is returned. A null string is returned if the physical file does not have the GROUP attribute set.
- If the attribute is interrogated while the logical file is unassigned, the value to which the GROUP attribute was last set is returned. A null string is returned if the GROUP attribute is not set.

Setting the GROUP attribute

The GROUP attribute can be changed to any syntactically valid groupcode or to the null string. Only the owner of the file or a privileged user or task can change the GROUP attribute. A groupcode is an identifier consisting of from 1 to 17 alphanumeric characters or of 1 to 17 arbitrary characters surrounded by quotes.

A groupcode that consists of a quoted identifier in parentheses, such as "(USER1)", is treated as the group consisting of just the usercode specified by the identifier string. This syntax is limited to usercodes consisting of no more than 15 characters. These single user groups can be used without administrator assistance. Groupcodes that do not follow this format must be defined using the GROUP and SUPPLEMENTARYGRPS USERDATA attributes by a system administrator.

When the logical file is assigned to a physical file, if the GROUP attribute is set to a permitted value different from the current value, the GROUP attribute of the physical file changes.

Whenever the GROUP attribute of an existing disk file is modified, the TIMESTAMP attribute is updated.

If the GROUP attribute is set to a null string when the logical file is unassigned, the GROUP attribute of the logical file reverts back to its default state (that is, not set). If the GROUP attribute is set to a null string when the logical file is assigned to a physical file, the GROUP attribute of the physical file is deleted.

When a new disk file is created as part of the file open process, any user-specified GROUP value is propagated to the physical file after the value has been validated. If the validation fails, an open error is generated.

When the logical file is used to open an *existing* disk file, a user-specified GROUP value set on the logical file has no effect on the GROUP attribute of a physical file. In other words, the GROUP attribute is not propagated from the logical file to an existing physical file during the file open process.

Tape Files

The GROUP attribute can only be used for tape files if the system SECOPT TAPECHECK security option has the value AUTOMATIC. The GROUP value is assigned when a tape volume family is created and cannot subsequently be changed programmatically.

When a new tape volume family is created as part of the file open process, any user-specified GROUP value is propagated to the new volume family after it has been appropriately validated. If validation fails, an open error is generated.

Restrictions

Prior to the 44.2 release, any attempt to set the GROUP attribute while the ENABLEPOSIX system option is RESET results in a READONLY attribute value. It is no longer necessary to set or reset the ENABLEPOSIX option to control any functionality. The ENABLEPOSIX option will be deimplemented in a future release.

For More Information

Refer to “Securing Files” in the *Security Features Operations Programming Guide* for more information about the use of the GROUP attribute.

HOSTNAME

Kinds	All
Interrogate	Anytime
Modify	When closed
Type	Character string
Range	Not applicable
Default	Name of local host
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Supported
Heritage Host Services Logical I/O	Supported
Port Services	All

Use the HOSTNAME attribute to determine or specify the name of the host on which the physical file or correspondent port file exists or is to be created. The name can be 1 to 17 characters long and can contain uppercase alphanumeric characters. For a port subfile, *HOSTNAME* is a nonpreferred synonym for *YOURHOST*.

If the file is a port file and more than one subfile exists, HOSTNAME requires a subport index as a parameter.

If HOSTNAME is interrogated while the file is closed and unassigned and HOSTNAME has never been set, the name of the local host is returned. If HOSTNAME is interrogated while the file is assigned, the name of the host on which the file resides is returned.

Heritage Host Services logical I/O does not permit an operator to modify the HOSTNAME attribute by using the FA (File Attribute) system command. File Transfer, Access, and Management (FTAM) permits an operator to modify the HOSTNAME attribute by using the system command FA (File Attribute). For a description of the FA command, refer to the *System Commands Reference Manual*.

HSFILECOPY

Kinds	Disk
Interrogate	Anytime by an MCS or DSS
Modify	Only by an MCS or DSS
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

If your program is a message control system (MCS) or a distributed systems service (DSS), set the HSFILECOPY attribute to TRUE to specify that the program can modify certain permanent file attributes that normally cannot be modified.

Setting the HSFILECOPY value to TRUE allows a user-written file transfer utility to maintain the original values for such attributes as CREATIONDATE and CREATIONTIME for a file.

The following file attribute values can either be modified or the range of values can be increased when the HSFILECOPY value is TRUE:

- ALTERDATE
- ALTERTIME
- CREATIONDATE
- CREATIONTIME
- FILEKIND
- TIMESTAMP
- USEDATE
- USETIME

Refer to the descriptions for the individual attribute for specific information.

INPUTEVENT

Kinds	Port, port subfile, or remote
Interrogate	Anytime (port or port subfile) When open (remote)
Modify	Never
Type	Event
Range	Not applicable
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	No
Parameters	1 optional (port or port subfile)
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	All

The INPUTEVENT attribute is a system-defined event that you can use to determine whether input data are waiting for a remote file, port file, or port subfile. Using this attribute in a WAIT statement is simpler and more efficient than periodically interrogating the CENSUS or FRAMESIZECENSUS attribute.

For port files, when no subfile index is given, the INPUTEVENT happened state is TRUE if the CENSUS or FRAMESIZECENSUS attribute for the file is greater than 0.

Because INPUTEVENT is a read-only attribute, it should not be passed as a parameter to the following statements: CAUSEANDRESET, FIX, FREE, LIBERATE, PROCURE, RESET, SET, and WAITANDRESET. If the INPUTEVENT attribute is used as a parameter to these statements, the program performs incorrectly, but an error indication is not issued.

The ATTACH and DETACH statements can be used with INPUTEVENT; however, up-level attach errors can occur as described in the discussion of interprocess communication in the *Task Management Guide* or the *ALGOL Reference Manual, Volume 1*.

Attribute errors on event-valued attributes result in the termination of the program.

The following ALGOL examples shows how to use the INPUTEVENT attribute in a multiple WAIT statement:

```
I := WAIT((5),F.INPUTEVENT);

WAIT(F.INPUTEVENT);

WHAT_HAPPENED := WAIT(MYSELF.EXCEPTIONEVENT,REM.INPUTEVENT,EV2);
```

INPUTTABLE

Kinds	All except port
Interrogate	Never
Modify	Anytime
Type	Translate table
Range	Not applicable
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Use the INPUTTABLE attribute to supply a translate table to be used for software translation during a READ operation.

If the assigned translate table is declared in a program, that table must be either the first table in a list of translate tables or the only table in that declaration. Each time the logical file is closed, the value of the INPUTTABLE attribute is discarded.

An example of the ALGOL syntax for assigning a translate table to INPUTTABLE is as follows:

```
FILEID.INPUTTABLE := ASCIITOBCL;
```

For further information regarding translation, refer to the *I/O Subsystem Programming Guide*.

INTERACTIVEFILE

Kinds	All
Interrogate	Anytime
Modify	Never
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Abstract Value	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	All

Interrogate the INTERACTIVEFILE attribute to determine if your file is assigned to an interactive file. An interactive file is a port file, a remote file, or a POSIX FIFO file.

The attribute returns TRUE if the file is assigned to an interactive file or FALSE if the file is not interactive.

INTMODE

Kinds	All
Interrogate	Anytime
Modify	When closed
Type	Mnemonic
Range	See following text
Default	See following text
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	None
FTAM File Access	Restricted values
Heritage Host Services Logical I/O	Restricted values
Port Services	BNANATIVESERVICE

Use the INTMODE attribute to determine or specify the internal or logical character encoding of the file. The mnemonic and integer values and meanings associated with the INTMODE attribute are listed in the following table. The CENTRALSUPPORT library might contain additional valid values, up to a 65,535 maximum.

Mnemonic Value	Integer Value	Meaning
ASCII	5	8-bit data encoded with the ASCII coded character set
BCL	3	Receives an OPEN error
EBCDIC or ASERIESEBCDIC	4	8-bit data encoded with the ClearPath MCP servers EBCDIC character set
HEX	2	4-bit, packed decimal
SINGLE	0	Word mode, binary, or 48-bit data

The following table displays INTMODE values that are valid only for files of KIND=DISK and KIND=PRINTER:

Mnemonic Value	Integer Value	Meaning
ARABIC20EBCDIC	34	Arabic EBCDIC 8-bit character set (previously known as version 20 EBCDIC)
ARABIC20ISO	35	Arabic ISO 8-bit character set (previously known as version 20 ASCII)
ARABICV201B	52	Arabic ASCII-based 8-bit character set used for printing

Mnemonic Value	Integer Value	Meaning
ARABICV202D	53	Arabic ASCII-based 8-bit character set used for printing
ASTASCIIGB2312	112	Mixed multibyte character set for Chinese ASCII-based terminal use
ASTASCIIKSC5601	106	Mixed multibyte character set for Korean ASCII-based terminal use
ASTASCIIUTL	109	Mixed multibyte character set for Taiwanese ASCII-based terminal use
ASTUTL	83	16-bit character set for Taiwanese ASCII-based terminal use
ASUTL	82	16-bit character set for Taiwanese ClearPath MCP servers
CANSUPPLEBCDIC	16	Canadian EBCDIC 8-bit character set with general-purpose supplementary graphics
CANSUPPLISO	17	Canadian ISO 8-bit character set with general-purpose supplementary graphics
CODEPAGE437	36	MS-DOS 8-bit code page for United States use
CODEPAGE850	18	MS-DOS 8-bit code page for multilingual use
CODEPAGE851	21	MS-DOS 8-bit code page for Latin/Greek use
CODEPAGE852	28	MS-DOS 8-bit code page for Eastern European use
CODEPAGE857	44	MS-DOS 8-bit code page for Turkish use
CODEPAGE864	54	MS-DOS 8-bit code page for Arabic use
CODEPAGE866	31	MS-DOS 8-bit code page for Russian use
CODEPAGE932	102	MS-DOS and Windows mixed multibyte code page for Japanese PC use
CODEPAGE936	113	MS-DOS and Windows mixed multibyte code page for Chinese PC use
CODEPAGE949	107	MS-DOS and Windows mixed multibyte code page for Korean PC use
CODEPAGE950	110	MS-DOS and Windows mixed multibyte code page for Taiwanese PC use
CODEPAGE1250	33	Windows 8-bit code page for Eastern European use
CODEPAGE1251	28	Windows 8-bit code page for Russian use
CODEPAGE1252	37	Windows 8-bit code page for multilingual use
CODEPAGE1253	45	Windows 8-bit code page for Greek use

Mnemonic Value	Integer Value	Meaning
CODEPAGE1254	43	Windows 8-bit code page for Turkish use
CODEPAGE1256	55	Windows 8-bit code page for Arabic use
EBCDICGB2312	111	Mixed multibyte character set for Chinese ClearPath MCP servers use
EBCDICKSC5601	105	Mixed multibyte character set for Korean ClearPath MCP servers use
EBCDICUTL	108	Mixed multibyte character set for Taiwanese ClearPath MCP servers use
EUCJP	103	Mixed multibyte character set for Japanese UNIX systems use
FRENCHARABICE	22	French/Arabic EBCDIC 8-bit character set
FRENCHARABICISO	23	French/Arabic ISO 8-bit character set
HUNGARIANBTOS	46	Hungarian BTOS 8-bit character set
IA5STRING	10	A string of 8-bit frames containing data encoded in the ISO646 G0 and C0 character sets
IBMSWEDENEBCDIC	51	8-bit EBCDIC-based Swedish IBM character set
IBM297	39	8-bit EBCDIC-based IBM character set
ISOGENERALSTRING	7	A string of 8-bit frames containing data encoded in the ISO646 C0 and G0 character sets and the ISO8859-1 G0 and G1 character sets
ISOGRAPHICSTRING	8	A string of 8-bit frames containing data encoded in the ISO646 G0 character set and ISO8859-1 G0 and G1 character sets
ISOVISIBLESTRING	9	A string of 8-bit frames containing data encoded in the ISO646 G0 character set
JAPANEBCDIC	41	Japanese EBCDIC 8-bit character set
JAPANEBCDICJBIS8	100	Mixed multibyte character set for Japanese ClearPath MCP servers use
JAPANV24	49	Japanese Version 24 8-bit character set
JAPANV24JBIS8	114	Japanese Version 24 mixed, multibyte character set
JBIS7	81	16-bit character set for Japanese ASCII-based terminal use
JBIS8	80	16-bit character set for Japanese ClearPath MCP servers use
JISASCII	42	Japanese ASCII 8-bit character set

Mnemonic Value	Integer Value	Meaning
JISASCIIJBIS7	101	Mixed multibyte character set for Japanese ASCII-based terminal use
LATINCYRILLICEBC	29	Cyrillic EBCDIC 8-bit character set
LATINCYRILLICISO	30	Cyrillic ISO 8-bit character set
LATINGREEKBTOS	25	Latin/Greek BTOS 8-character set
LATINGREEKEBCDIC	19	Latin/Greek EBCDIC 8-bit character set
LATINGREEKISO	20	Latin/Greek ISO 8-bit character set
LATIN1EBCDIC	12	Latin1 EBCDIC 8-bit character set
LATIN1ISO	13	Latin1 ISO 8-bit character set
LATIN2EBCDIC	26	Latin2 EBCDIC 8-bit character set
LATIN2ISO	27	Latin2 ISO 8-bit character set
LATIN5EBCDIC	14	Latin5 EBCDIC 8-bit character set
LATIN5ISO	15	Latin5 ISO 8-bit character set
LATIN9EBCDIC	47	Latin9 EBCDIC 8-bit character set
LATIN9ISO	48	Latin9 ISO 8-bit character set
LETSJ	104	Mixed multibyte character set for Japanese ClearPath IX servers use
LOCALEBCDIC	50	8-bit EBCDIC-based locally defined character set
MACROMAN	38	Macintosh® 8-bit code page for multilingual use
NORWAYBTOS	24	Norwegian BTOS 8-bit character set
OCTETSTRING	11	A string of 8-bit frames each containing any binary value from hexadecimal 00 to hexadecimal FF
UCS2	85	Universal 16-bit character set based on the Multilingual Plane of the ISO 10646-1 standard
UCS2NT	84	Universal 16-bit character set based on the Basic Multilingual Plane of the ISO 10646-1 standard, encoded for Intel processors, Windows NT and Windows 95 systems use

Note: *Mixed multibyte character sets contain both 16-bit and 8-bit characters. For such character sets, data is transferred in units of 8 bits (FRAMESIZE = 8).*

For a description of the ASCII, BCL, and EBCDIC coded character sets, refer to the *ALGOL Reference Manual, Volume 1*. For a description of the ISO646 and ISO8859-1 coded character sets, refer to the *I/O Subsystem Programming Guide*. For a description of the remaining coded character sets, refer to the *MLS Guide*.

The following table indicates the IPC providers and the distributed systems services (DSSs) that support the various INTMODE values:

	Distributed Systems Services (DSSs)		**IPC Providers**	
Mnemonic Value	**Heritage Host Services Logical I/O**	**FTAM File Access**	**BNA Version 2 Port Files**	**Local Port Provider (LPP)**
ARABIC20EBCDIC	Yes	No	No	No
ARABIC20ISO	Yes	No	No	No
ASCII	Yes	Yes	Yes	Yes
ASTASCIIGB2312	Yes	No	No	No
ASTASCIIKSC5601	Yes	No	No	No
ASTASCIIUTL	Yes	No	No	No
ASTUTL	Yes	No	No	No
ASUTL	Yes	No	No	No
CANSUPPLEBCDIC	Yes	No	No	No
CANSUPPLISO	Yes	No	No	No
CODEPAGE437	Yes	No	No	No
CODEPAGE850	Yes	No	No	No
CODEPAGE851	Yes	No	No	No
CODEPAGE852	Yes	No	No	No
CODEPAGE857	Yes	No	No	No
CODEPAGE866	Yes	No	No	No
CODEPAGE932	Yes	No	No	No
CODEPAGE936	Yes	No	No	No
CODEPAGE949	Yes	No	No	No
CODEPAGE950	Yes	No	No	No
CODEPAGE1250	Yes	No	No	No
CODEPAGE1251	Yes	No	No	No
CODEPAGE1252	Yes	No	No	No
CODEPAGE1253	Yes	No	No	No
CODEPAGE1254	Yes	No	No	No

	Distributed Systems Services (DSSs)		IPC Providers	
Mnemonic Value	Heritage Host Services Logical I/O	FTAM File Access	BNA Version 2 Port Files	Local Port Provider (LPP)
EBCDIC	Yes	Yes	Yes	Yes
EBCDICGB2312	Yes	No	No	No
EBCDICKSC5601	Yes	No	No	No
EBCDICUTL	Yes	No	No	No
EUCJP	Yes	No	No	No
FRENCHARABICE	Yes	No	No	No
FRENCHARABICISO	Yes	No	No	No
HUNGARIANBTOS	Yes	No	No	No
IA5STRING	Yes	Yes	No	No
ISOGENERALSTRING	Yes	Yes	No	No
ISOGRAPHICSTRING	Yes	Yes	No	No
ISOVISIBLESTRING	Yes	Yes	No	No
JAPANEBCDIC	Yes	No	No	No
JAPANEBCDICJBIS8	Yes	No	No	No
JAPANV24	Yes	No	No	No
JAPANV24JBIS8	Yes	No	No	No
JBIS7	Yes	No	No	No
JBIS8	Yes	No	No	No
JISASCII	Yes	No	No	No
JISASCIIJBIS7	Yes	No	No	No
LATINCYRILLICEBC	Yes	No	No	No
LATINCYRILLICISO	Yes	No	No	No
LATINGREEKBTOS	Yes	No	No	No
LATINGREEKEBCDIC	Yes	No	No	No
LATINGREEKISO	Yes	No	No	No
LATIN1EBCDIC	Yes	No	No	No
LATIN1ISO	Yes	No	No	No
LATIN2EBCDIC	Yes	No	No	No
LATIN2ISO	Yes	No	No	No

	Distributed Systems Services (DSSs)		IPC Providers	
Mnemonic Value	**Heritage Host Services Logical I/O**	**FTAM File Access**	**BNA Version 2 Port Files**	**Local Port Provider (LPP)**
LATIN5EBCDIC	Yes	No	No	No
LATIN5ISO	Yes	No	No	No
LATIN9EBCDIC	Yes	No	No	No
LATIN9ISO	Yes	No	No	No
LETSJ	Yes	No	No	No
NORWAYBTOS	Yes	No	No	No
OCTETSTRING	Yes	Yes	No	No
SINGLE	Yes	No	Yes	Yes
STDEBCDIC	Yes	No	No	No
UCS2	Yes	No	No	No
UCS2NT	Yes	No	No	No

Most compilers assign a default value for each file when creating the code file; otherwise, the default value of INTMODE is SINGLE. The following table summarizes the defaults.

Language	Default Value	
ALGOL	EBCDIC	
COBOL	EBCDIC if the first 01-level entry of the file is USAGE DISPLAY. HEX if the first 01-level entry of the file is USAGE COMP-2	
COBOL74	EBCDIC if the first 01-level entry of the file is USAGE DISPLAY. HEX if the first 01-level entry of the file is USAGE COMP	
COBOL85	EBCDIC	
COBOL	HEX if the first 01-level entry of the file is USAGE COMP and the B2500 compiler option is set	
Pascal	Depends on the component type:	
	Component Type	**INTMODE Default Value**
	Packed array with 4-bit elements	HEX
	Packed array of characters	EBCDIC or ASCII depending on the setting of the STRINGS compiler control option
	All other component types	SINGLE

Whenever the INTMODE and EXTMODE attributes differ, there is a possibility that the data is being translated. For information about translation, refer to the *I/O Subsystem Programming Guide*.

INTMODE, in combination with the FRAMESIZE attribute or the UNITS attribute, determines whether the data transfer is a word-oriented or a character-oriented transfer. When INTMODE and FRAMESIZE are not compatible, the open error "FRAMESIZE AND INTMODE INCOMPATIBLE" is reported and the open result error FRAMESIZEINCOMPATIBLERSLT (106) is returned.

An open error is issued if an attempt is made to open a file with an INTMODE value of BCL.

When the INTMODE value is SINGLE, the FRAMESIZE value must be 48. As a result, when a logical file with a DEPENDENTSPECS attribute value of TRUE or a FILETYPE attribute value of 8, and a DEPENDENTINTMODE attribute value of FALSE is used to open a file that has a UNITS attribute value and an EXTMODE attribute value that require a FRAMESIZE value other than 48, a "LOGICAL FILE INTMODE INCOMPATIBLE WITH PERMANENT FILE FRAMESIZE" open error is reported and the open result error INTMODEFRAMESIZECONFLICTRSLT (144) is returned.

If an I/O operation is attempted in which the character size of the array, pointer, or string specified in the I/O statement is different from the INTMODE value of the file, an INVALID OP or INVALID STACK ARGUMENT fault occurs in the user program.

Port Files

For BNANATIVESERVICE, the INTMODE attribute is used during matching. If the INTMODE attributes of the two subfiles do not match, at least one of the subfiles must have a TRANSLATE attribute value of FULLTRANS or USERTRANS or the subfiles are not matched. Refer to the *I/O Subsystem Programming Guide* for more information.

Printer Files

If you specify a value of SINGLE, the INTMODE value is changed to EBCDIC. If the value of PRINTDISPOSITION is either DIRECTPS or DIRECTDLP, the INTMODE value must be EBCDIC.

Printer Backup Files

When a logical file is assigned to an existing printer backup disk file, the value of INTMODE takes on the EXTMODE value. Translation occurs only if the TRANSLATE attribute has a value of FORCESOFT. When the logical file is no longer assigned to the physical file, INTMODE is restored to its original, pre-assignment value.

INTNAME

Kinds	All
Interrogate	Anytime
Modify	When unassigned
Type	Character string
Range	Not applicable
Default	As declared in program
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	None
FTAM File Access	Supported
Heritage Host Services Logical I/O	Supported
Port Services	All

Use the INTNAME attribute to change the internal file name before the logical file is opened and assigned to a physical file. The new identifier can be 1 to 17 characters long and can contain any combination of EBCDIC uppercase letters A through Z or EBCDIC digits 0 through 9. Additionally, a hyphen (-) or an underscore (_) can be included in the identifier, but neither of those characters can be the first character. By default, the first 17 characters of the file declaration identifier in a program are used as the internal file name. Thus, file identifiers in any given program must be unique in the first 17 characters.

File equation is accomplished by matching the internal file name to the file name specified in the task initiation statement. The INTNAME attribute cannot be modified through the Work Flow Language (WFL). When the internal name of a file is changed, file equation action is initiated by using the new internal name.

The following example shows how the internal name of a file can be changed before the file is opened:

```
?BEGIN JOB TEST/INTNAME;
 COMPILE X ALGOL GO;
 FILE NEWINT (KIND=DISK);
 ALGOL DATA
 BEGIN
 FILE F (KIND=TAPE);
 REPLACE F.INTNAME BY "NEWINT.";
 OPEN(F);
 END.
 ?END JOB
```

IOCLOCKS

Kinds	All
Interrogate	Anytime
Modify	Never
Type	Integer
Range	0 through 549755813887
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	All

Interrogate the IOCLOCKS attribute to determine the accumulated I/O time for the file in units of 2.4 microseconds.

The value of the IOCLOCKS attribute is always 0 (zero) for remote and port files.

IOHFUNCTIONNAME

Kinds	Virtual
Interrogate	Anytime
Modify	When assigned
Type	Simple name
Range	Not applicable
Default	See explanation
Abstract Value	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

Use the IOHFUNCTIONNAME attribute to specify the library function name to be used when linking to the IOHANDLER library to open a virtual file. A virtual file is a file with KIND=VIRTUAL specified.

You use the IOHFUNCTIONNAME attribute to select the IOHANDLER library when the IOHLIBACCESS attribute has been set to BYFUNCTION. If you do not specify the IOHFUNCTIONNAME attribute when the IOHLIBACCESS attribute is set to BYFUNCTION, then a library function name of IOHANDLER is used.

IOHINTERFACENAME

Kinds	Virtual
Interrogate	Anytime
Modify	When unassigned
Type	Character string
Range	Not applicable
Default	Null string
Abstract Value	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

Use the IOHINTERFACENAME attribute to specify a particular connection library when linking to the IOHANDLER library to open a virtual file. IOHINTERFACENAME specifies the library interface name that corresponds to the library INTERFACENAME attribute.

A virtual file is a file with KIND=VIRTUAL specified.

IOHLIBACCESS

Kinds	Virtual
Interrogate	Anytime
Modify	When unassigned
Type	Mnemonic
Range	BYTITLE, BYFUNCTION, and BYINITIATOR
Default	BYTITLE
Abstract Value	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

Use the IOHLIBACCESS attribute to specify how the IOHANDLER library object code file is accessed when a virtual file is opened. A virtual file is a file with KIND=VIRTUAL specified.

IOHLIBACCESS can be set to one of the following mnemonic values.

Mnemonic Value	Meaning
BYTITLE	IOHLIBACCESS uses the value of the IOHTITLE attribute of the virtual file to access the object code file for the IOHANDLER library.
BYFUNCTION	IOHLIBACCESS uses the value of the IOHFUNCTIONNAME attribute of the virtual file to access the operating system library function table, and then uses the object code file associated with the FUNCTIONNAME attribute to link to the IOHANDLER library.
BYINITIATOR	IOHLIBACCESS uses the object code file that declared the virtual file to link to the IOHANDLER library.

IOHLIBPARAMETER

Kinds	Virtual
Interrogate	Anytime
Modify	When unassigned
Type	String
Range	Not applicable
Default	Unassigned
Abstract Value	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

Use the IOHLIBPARAMETER attribute to specify the library parameter to be passed as the LIBPARAMETER value when linking to the IOHANDLER library to open a virtual file. A virtual file is a file with KIND=VIRTUAL specified.

A virtual file provides a mechanism through which a program in an MCP environment can access features provided outside the MCP by using native language constructs such as OPEN, CLOSE, READ, and WRITE.

IOHPREFIX

Kinds	Virtual
Interrogate	Anytime
Modify	When unassigned
Type	Simple name
Range	Not applicable
Default	Unassigned
Abstract Value	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

Use the IOHPREFIX attribute to dynamically select the name of the IOHANDLER entry points when linking to the IOHANDLER library to open a virtual file. A virtual file is a file with KIND=VIRTUAL specified.

The prefix string is combined with the entry point names OPEN, CLOSE, READ, and WRITE to produce the actual name used during the library linkage, such as <IOHPREFIX>OPEN, <IOHPREFIX>CLOSE, <IOHPREFIX>READ, and <IOHPREFIX>WRITE.

Example

The following statement sets the IOHPREFIX attribute to the string “MY_”:

```
REPLACE F.IOHPREFIX BY "MY_.";
```

The entry point names used during library linkage are

```
MY_OPEN, MY_CLOSE, MY_READ and MY_WRITE.
```

IOHSTRING

Kinds	Virtual
Interrogate	Anytime
Modify	Anytime
Type	String
Range	Not applicable
Default	Unassigned
Abstract Value	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

Use the IOHSTRING attribute to pass information between an IOHANDLER library and the application using a virtual file. A virtual file is a file with KIND=VIRTUAL specified.

When the IOHSTRING attribute is interrogated for an unassigned logical file, the value to which the attribute was last set on the unassigned file is returned. If the value is not set, then "." is returned. When the IOHSTRING attribute is interrogated on an assigned virtual file, then any value that is returned is generated by the appropriate IOHANDLER library, if the library has provided the appropriate IOH_GETATTRIBUTE interface and has provided the appropriate support for that attribute.

When the IOHSTRING attribute is assigned on an unassigned logical file, the value is stored in the logical file. When the attribute is assigned on an assigned virtual file, then the value is passed to the appropriate IOHANDLER library, if the library has provided the appropriate IOH_SETATTRIBUTE interface. The IOHANDLER library must process the assignment.

IOHTITLE

Kinds	Virtual
Interrogate	Anytime
Modify	When unassigned
Type	Name
Range	Not applicable
Default	See explanation
Abstract Value	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

Use the IOHTITLE attribute to specify the library code file title to be used when linking to the IOHANDLER library to open a virtual file. A virtual file is a file with KIND=VIRTUAL specified.

You specify the IOHTITLE attribute to select the IOHANDLER library when the IOHLIBACCESS attribute has been set to BYTITLE or is unspecified. If you do not specify the IOHTITLE attribute and the IOHLIBACCESS attribute is set to BYTITLE or is unspecified, then a library title of IOHANDLER is used.

The IOHTITLE attribute enables one library to implement multiple types of virtual files.

IOINERROR

Kinds	Disk, CD, or tape
Interrogate	When open
Modify	Never
Type	Boolean
Range	TRUE, FALSE
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

Interrogate the IOINERROR attribute to determine if a physical I/O error has occurred on the buffer currently in use for the file. When IOINERROR is TRUE, an I/O error has occurred.

For files with more than one buffer, the usual case, the physical and logical I/O operations occur asynchronously. Therefore, error results are returned to the program when the buffer is reused. Also, the value returned as the result of an I/O operation contains additional information about the error.

When IOINERROR is TRUE, the RECORDINERROR, RESULTLIST, and STATE attributes can be consulted for more information.

IPTYPEOFSERVICE

Kind	Port subfile
Interrogate	Anytime
Modify	When closed
Type	Integer
Range	0 through 255
Default	0
Abstract Value	DEFAULT, MIMIMUM, MAXIMUM
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	TCPIPNativeService

Use the IPTYPEOFSERVICE attribute to set the Type-of-Service field of the IP header on outgoing frames.

Some router implementations allow priority queuing based on the value of the Type-of-Service field within the IP header. If this field is set to a non-zero value and the router is configured accordingly, data transmission is prioritized by the router.

The file attribute DIALOGPRIORITY sets the Precedence sub-field in the IP Type-of-Service field. If the IPTYPEOFSERVICE attribute is zero or unspecified, the DIALOGPRIORITY semantics are used. If the IPTYPEOFSERVICE attribute is a value other than zero, that value is used in the IP Type-of-Service field.

This attribute requires a subfile index as a parameter if the MAXSUBFILES value is greater than 1, except when it is used in a file declaration. If it used in a file declaration or the subfile index specified is zero, then the value of IPTYPEOFSERVICE is changed in all closed, allocated subfiles and it becomes the new default for any subfiles that are allocated later, even after increasing MAXSUBFILES.

KERBEROSACCESS

Kind	Disk
Interrogate	Anytime
Modify	When closed
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Abstract Value	DEFAULT
Stored Permanently	Yes
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

The KERBEROSACCESS attribute is used by Kerberos software to restrict access to certain disk files. Only programs that have been marked with the MP + KERBEROS system command can set the KERBEROSACCESS file attribute to TRUE. Likewise, only programs that have been marked with the MP + KERBEROS system command can access a disk file with KERBEROSACCESS set to TRUE.

KIND

Kinds	All
Interrogate	Anytime
Modify	When closed
Type	Mnemonic
Range	See following text
Default	DONTCARE
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	None
FTAM File Access	Restricted values
Heritage Host Services Logical I/O	Restricted values
Port Services	All

Use the KIND attribute to determine or specify the peripheral unit associated with the logical file. This attribute must be used when Heritage Host Services logical I/O is being used.

The mnemonic and integer values associated with the KIND attribute are as follows. The first mnemonic for each value is the preferred mnemonic.

Mnemonic Value	Integer Value	Supported by Heritage Host Services Logical I/O	Supported by FTAM
CD	4	No	No
DISK or SERIAL	1	Yes	Yes
DONTCARE	0	No	No
HC	20	No	No
HY	8	No	No
LBP	36	No	No
ODT or SPO	2	No	No
PACK or DISKPACK	17	Yes	Yes
PORT	19	No	No
PRINTER	7	Yes	No
READER	9	Yes	No
REMOTE or DC	3	Yes	No
TAPE	45	Yes	No
TSP	5	No	No
VC	40	No	No
VIRTUAL	51	No	No

The KIND value is set to PACK automatically when the TITLE attribute is set to a file title that includes an ON clause.

If the KIND value is DONTCARE and a new file is to be created when the file is opened, the value is converted to TAPE. If the KIND value is DONTCARE and an existing file is to be opened, the logical I/O subsystem searches for the file with each of the KIND values TAPE, READER, PACK, and ODT.

The value CD is used for CD-ROM files.

The FA (File Attribute) system command cannot be used to change the KIND value to PORT.

If the KIND attribute is interrogated when the file is assigned, the value for the physical file is returned.

The mnemonic value PACK is functionally synonymous with DISK when setting the attribute, although DISK is the preferred value.

The use of voice channel (VC) and telephony services platform (TSP) files is limited to the Network Application Platform (NAP) software. NAP declares various direct files with a KIND attribute value of VC or TSP to interface with VC and TSP units. VC and TSP files can be used only with direct I/O. The use of VC and TSP files outside of the NAP software is not supported at this time.

A laser beam printer (LBP) file is used for KPSUPPORT and SYSTEM/LBPBACKUP to control printing output to a 16-bit laser beam printer (LBP). The LBP mnemonic is limited to direct files and is intended for the special software mentioned above only.

For POSIX special files (FILEKIND values of DEVNULL and FIFO), KIND must be set to DISK. Although special files do not reside on disk, they do have directory entries.

Double-byte (16-bit) and mixed multibyte (mixed 8-bit and 16-bit with FRAMESIZE set to 8) character sets are supported only for KIND = DISK and KIND = PRINTER (with BACKUPKIND = DISK or PACK). Additionally, for PRINTER files, the BLOCKSTRUCTURE value must be FIXED, and the INTMODE and EXTMODE values must be equal. While you can create printer backup files with double-byte or mixed multibyte characters, the Print System cannot print these files.

The virtual file feature provides a mechanism through which a program in an MCP environment can access features provided outside the MCP by using native language constructs such as OPEN, CLOSE, READ, and WRITE.

For detailed information about virtual files, refer to the *I/O Subsystem Programming Guide.*

LABEL

Kinds	All except port
Interrogate	Anytime
Modify	Never (remote) When unassigned (others)
Type	Mnemonic
Range	See following text
Default	OMITTED (remote) STANDARD (others)
Abstract Value	DEFAULT (remote)
Stored Permanently	Backup file
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Interrogate the LABEL attribute to determine if the file has label records. For files other than remote files, use the LABEL attribute to specify whether file labels are to be created.

The mnemonic and integer values associated with LABEL are as follows:

Mnemonic Value	Integer Value
OMITTED	1
OMITTEDEOF	3
STANDARD	0

Tape and printer files have specific characteristics.

Tape Files

If the value of LABEL is STANDARD when a tape file is created, the I/O subsystem writes labels compatible with the ANSI X3.27-1969 or ANSI X3.27-1987 standard at the beginning and ending of the file. The beginning labels of files are succeeded by tape marks and the ending labels are preceded by tape marks. Refer to the *Task Attribute Reference Manual* for information about the LABELFORMAT task attribute that enables you to control the type of labels to be written. Refer to the *I/O Subsystem Programming Guide* for standard tape formats.

Label records are not included if the value of LABEL is OMITTED or OMITTEDEOF.

When a labeled tape that has a LABEL value of OMITTED or OMITTEDEOF is read, the DENSITY and PARITY attributes values are taken from the logical file description.

When an unlabeled tape is read and a tape mark is encountered, the action taken is dependent on the value of LABEL. If the value of LABEL is OMITTEDEOF, an end-of-file (EOF) action occurs; if the value of LABEL is OMITTED, a volume switch is attempted. If the value of LABEL is OMITTED, the operator can use the FR (Final Reel) system command to indicate that the end of the file has been reached. Refer to the *System Operations Guide* for information about designating a final reel.

Printer Files

If a file has a PRINTDISPOSITION attribute value of DIRECTDLP and a LABEL value of STANDARD, standard banner pages are printed at the beginning and end of the file. If the LABEL value is OMITTED or OMITTEDEOF, banner pages are not printed.

If a file has a PRINTDISPOSITION value of DIRECTPS and a LABEL value of STANDARD, header and trailer pages are printed. A LABEL value of OMITTED or OMITTEDEOF prevents header and trailer pages from printing.

For files with PRINTDISPOSITION values other than DIRECTPS, header and trailer pages are not printed unless the following conditions are met:

- The LABEL attribute value is OMITTED or OMITTEDEOF.
- The FORMID attribute is specified for the file.
- The task or session HEADER and TRAILER print modifiers of the user are explicitly set to UNCONDITIONAL before the print request is created.

The following table summarizes when a top-of-page (skip-to-channel 1) action is performed based on the LABEL attribute:

LABEL Attribute Value	Action Taken
STANDARD	A top-of-page action is performed when the file is closed, unless it is a printer backup disk file with a FILESTRUCTURE value of STREAM or BLOCKED.
OMITTED	A top-of-page action is performed when the file is closed, unless it is a printer backup disk file with a FILESTRUCTURE value of STREAM or BLOCKED. No header or trailer is printed.
OMITTEDEOF	No top-of-page action is performed when the file is closed. No header or trailer is printed.

However, a top-of-page action is always performed when a print request is completed. For more information about using the LABEL attribute for a printer file, refer to the *Print System Guide*.

LABELKIND

Kinds	Tape
Interrogate	When assigned
Modify	Never
Type	Integer
Range	0 through 14
Default	See following text
Abstract Value	Not applicable
Stored Permanently	Tape
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Interrogate the LABELKIND attribute to determine whether a tape is labeled and, if labeled, to determine the kind of labels.

The following are the default values:

LABEL File Attribute Value	Mnemonic Value
STANDARD	ASERIESDATA
OMITTED	UNLABELLED
OMITTEDEOF	UNLABELLED

The values and meanings associated with LABELKIND are as follows:

Mnemonic Value	Integer Value	Meaning
ASERIESDATA	0	ClearPath MCP servers data
UNLABELLED	1	Unlabeled
B5500STANDARD	2	B 5500 standard
SYSTEMSTANDARD	3	System standard
B3500USASI	4	B 3500 USASI
B3500STANDARD	5	B 3500 standard
ASERIESPRINTERBACKUP	6	ClearPath MCP servers printer backup tape
B6500LIBRARYTAPE	7	ClearPath MCP servers library maintenance tape
B6500LOADCONTROLTAPE	8	ClearPath MCP servers load control tape

Mnemonic Value	Integer Value	Meaning
ASERIESLIBRARYTAPE	9	ClearPath MCP servers library tape
B5500BACKUPTAPE	10	B 5500 backup tape
B5500LIBRARYTAPE	11	B 5500 library tape
ANSISTANDARDDATA	12	ANSI standard data for systems other than ClearPath MCP servers
USERDEFINEDLABEL	13	User-defined label
FIXEDBLOCKLIBRARYTAPE	14	Fixed-block library tape

Note that the mnemonics B6500USASI, B6500BACKUPTAPE, and NONBURROUGHSUSASI are the nonpreferred mnemonic synonyms for ASERIESDATA, ASERIESPRINTERBACKUP, and ANSISTANDARDDATA, respectively.

The valid values for library maintenance tapes are 7 (B6500LIBRARYTAPE), 9 (ASERIESLIBRARYTAPE), and 14 (FIXEDBLOCKLIBRARYTAPE). Refer to the *I/O Subsystem Programming Guide* for information about library maintenance tape formats.

For information on constructing value arrays of nonstandard tape labels through the use of the USERDEFINEDLABEL mnemonic, refer to the RLTABLEGEN utility in the *System Software Utilities Operations Reference Manual*.

LABELTYPE

LABELTYPE is a nonpreferred synonym for LABEL. Refer to the description of the LABEL attribute.

LASTACCESSIBLEAREA

Kinds	Disk
Interrogate	When open
Modify	Never
Type	Integer
Range	0 through 14999
Default	Not applicable
Abstract Value	None
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Use the LASTACCESSIBLEAREA attribute to determine the highest area number accessible in an open disk file.

Normally the LASTACCESSIBLEAREA attribute value is either one less than the maximum value of the AREAS attribute or one less than the existing AREAS value.

There is a relationship between LASTACCESSIBLEAREA and the FLEXIBLE attribute as follows:

- If the value of the FLEXIBLE attribute is FALSE, then the LASTACCESSIBLEAREA value does not exceed one less than the value of the AREAS attribute.
- If the value of the FLEXIBLE attribute is TRUE, a higher value than described above is possible for LASTACCESSIBLEAREA because LASTACCESSIBLEAREA reports the highest potentially accessible area, whether or not that area number has ever been used.

For files with CRUNCHED = TRUE, the value of LASTACCESSIBLEAREA does not exceed one less than the number of existing areas.

The following two situations reduce the number of accessible areas if you request a large number of AREAS, or a large AREASIZE or AREALENGTH:

- The combination of AREAS and AREASIZE or AREALENGTH can cause the potential number of sectors to be too large. The limit on the number of AREAS in a file is calculated based on the maximum number of sectors in a file (549,755,551,743 sectors). If the number of AREAS is too large, it is reduced to a number that prevents exceeding the maximum number of sectors in a file when it is combined with the designated AREASIZE or AREALENGTH.
- Each record of both the logical and physical file must be addressable with an integer address or the system may not be able to reach all of the records. If the number of AREAS is too large, it is reduced to a size that ensures all records are addressable. For a new file, the record size and addressing unit of the logical file are considered.

For an existing file, the record size of the physical file and addressing unit of the logical file are considered.

Because all records in all areas of logical I/O are addressed with an integer value, the record size used to calculate the limit on the number of AREAS is determined as follows:

- For fixed-length records, MAXRECSIZE is used.
- For variable-length records, MINRECSIZE is used. If the MINRECSIZE value is zero, it is replaced for character-oriented BLOCKSTRUCTURE = VARIABLE files with 4 and with 1 for other files.
- For files with FILESTRUCTURE = STREAM and FILEORGANIZATION = RELATIVE, MAXRECSIZE + 1 is used whether they are fixed or variable-length records.

The addressing unit is normally the same as the record size for the logical file. In the case of a DIRECT file it can be different depending on the values of DIOFILESTRUCTURE, FILESTRUCTURE, and BLOCKSTRUCTURE as summarized in the following table:

DIOFILESTRUCTURE	**FILESTRUCTURE**	**BLOCKSTRUCTURE**	**Addressing Unit**
SECTORSTREAM	Any	Any	Sector
STREAM or DEPENDENT	STREAM	Any	Sector
BLOCKED or DEPENDENT	BLOCKED	Any	Block
ALIGNED180 or DEPENDENT	ALIGNED180	FIXED	Record
ALIGNED180 or DEPENDENT	ALIGNED180	Not FIXED	Block

An open error is produced if an existing file has an allocated row that exceeds the LASTACCESSIBLEAREA. This occurs if the addressing unit of the logical file is smaller than the record size of the physical file.

LASTRECORD

Kinds	Disk or CD
Interrogate	When assigned
Modify	When assigned (see following text)
Type	Integer
Range	–1 through 549755813887
Default	–1
Abstract Value	DEFAULT, MINIMUM, MAXIMUM
Stored Permanently	Disk or CD
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Use the LASTRECORD attribute to determine or specify the record number of the last record in the file.

When you interrogate the LASTRECORD value, the value is calculated in terms of the blocking of the logical file. LASTRECORD might not be correct during periods when the file is being expanded because the end-of-file calculations are made only when necessary, such as during the transition from writing to reading. Since the first record of a file is record number 0, LASTRECORD returns 0 when there is only one record in the file. LASTRECORD returns –1 for an empty file.

The LASTRECORD value of a permanent file is stored only when the logical file is closed or when a new space is allocated. Therefore, if a system failure occurs while the file is open and records have been added to the file, the new records might be lost because the LASTRECORD value was not updated before the halt/load. Specifying a PROTECTION attribute value of PROTECTED can prevent this loss of records.

The number of the block rather than the number of the last record is returned when the LASTRECORD attribute is interrogated and either of the following conditions exist:

- FILESTRUCTURE has a value of either ALIGNED180 or BLOCKED and the BLOCKSTRUCTURE attribute value is not FIXED.
- FILESTRUCTURE has a value of either ALIGNED180 or BLOCKED and the FILETYPE attribute value is not 0 (zero).

If you interrogate the LASTRECORD attribute for a file with a logical or physical file FILESTRUCTURE value of STREAM and the logical or permanent file BLOCKSTRUCTURE value is not FIXED, a value of 0 is returned and an attribute error is issued.

You can modify the LASTRECORD value to change the end-of-file position. Doing so, however, does not allocate or deallocate space on the disk. You can modify the LASTRECORD attribute only when all the following conditions exist:

- The program has write access to the file.
- The PROTECTION attribute value is not PROTECTED.
- The CRUNCHED attribute is FALSE.
- The FILEKIND attribute value is DATA, DBDATA, or DBRESTARTSET.
- The logical file is closed with retention.
- No other logical file is currently assigned to the physical file.
- The FILESTRUCTURE value is not STREAM or the BLOCKSTRUCTURE value of both the logical and permanent files is FIXED.

The value of LASTRECORD cannot be altered by file equation.

LASTSTATION

LASTSTATION is a nonpreferred synonym for LASTSUBFILE. Refer to the description of the LASTSUBFILE attribute.

LASTSUBFILE

Kinds	Port or remote
Interrogate	Anytime
Modify	Never (port) When open (remote)
Type	Integer
Range	0 through port MAXSUBFILES attribute value Valid RSNs (remote)
Default	0
Abstract Value	DEFAULT (remote)
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	All

The LASTSUBFILE attribute has several uses depending on the type of file.

Port Files

Interrogate the LASTSUBFILE attribute to determine the index of the subfile that last had a successful I/O operation.

The LASTSUBFILE value is updated only upon completion of a successful I/O operation. After a broadcast WRITE operation, the LASTSUBFILE attribute returns the subfile index of the highest numbered open subfile. If the operation was a nonselective READ operation, the LASTSUBFILE value is the subfile index of the subfile that provided the message. If all the subfiles are closed, the LASTSUBFILE value is 0 (zero).

Multiple Stacks Operating on the Same Port

When multiple stacks are operating on the same port file, the following information is true:

- There is no way to determine which stack initiated the I/O operation that the LASTSUBFILE value is reporting.
- It is possible for the LASTSUBFILE value to be updated by the system between the time a user program performs a non-selective READ operation, and subsequently inquires on the LASTSUBFILE attribute.

  This problem might be encountered when the following code is used:

  ```
  READ (PORTF[SUBFILE 0], 72, IOBUF);
  INX :=PORTF.LASTSUBFILE;
  ```

It is possible for another stack that is using the same port file to perform an I/O operation immediately after the READ operation of the first stack, and as a result change the LASTSUBFILE value to reflect the new subfile. In that case, the INX variable might not contain the index of the subfile that performed the non-selective READ operation in the prior statement.

In order to avoid the timing hole described here, it is recommended that the port-file user program be coded as follows:

```
READ (PORTF[SUBFILE INX:0], 72, IOBUF);
```

In this example, the INX variable contains the correct index of the subfile that performed the non-selective READ operation.

Updating the LASTSUBFILE Attribute

When multiple stacks are operating on the same port file, it is possible for the LASTSUBFILE attribute to be updated by the system between the time a user program performs a nonselective READ operation, and subsequently inquires on the LASTSUBFILE attribute.

This problem can be encountered in a program coded such as the following:

```
READ (PORTF[SUBFILE 0], 72, IOBUF);
INX := PORTF.LASTSUBFILE;
```

It is possible for another stack using the same port file to perform an I/O operation immediately after the READ by the first stack, and thus, change the LASTSUBFILE attribute to reflect the new subfile. In that case, INX might not return the index of the subfile that performed the nonselective READ in the prior statement.

In order to avoid the timing hold described here, it is recommended that the port file user program be coded as follows:

```
READ (PORTF[SUBFILE INX:0], 72, IOBUF);
```

In this case, INX will correctly return the index of the subfile that performed the nonselective READ.

Nondirect I/O Remote Files

Interrogate the LASTSUBFILE attribute when the file is open to determine the relative station number (RSN) of the station from which the last message was received or the RSN of a newly added station. When a station is added to the station list of a remote file, the LASTSUBFILE value contains the RSN of the newly added station if no attribute error occurred during the addition. If the program has modified the LASTSUBFILE value, the modified value is returned.

Modify the LASTSUBFILE attribute to a valid RSN to direct the output to a specific station. Input operations are always nonselective. If a program sets the LASTSUBFILE value to an invalid RSN or if the station is denied, a WRITE operation to the file returns an end-of-file (EOF) indicator. Valid RSNs are greater than 0 (zero) and correspond to a valid station in the STATIONLIST attribute. When the LASTSUBFILE value is 0, a WRITE statement is broadcast to every station assigned to the file. If the value of the POPULATION attribute is 1, a broadcast WRITE operation is the same as a WRITE operation directed to a single station. The LASTSUBFILE value also can be modified by using the *[STATION <arithmetic expression>]* form of the *[<record number or carriage control>]* part of the ALGOL *WRITE* statement.

For more information regarding RSNs, refer to the discussion of remote files in the *I/O Subsystem Programming Guide*. The description of the STATIONLIST attribute also provides helpful information.

Direct I/O Remote Files

Do not interrogate the LASTSUBFILE attribute to determine the originating RSN. Use the IORECORDNUM buffer attribute because IORECORDNUM is associated with the particular buffer rather than the file.

The following are two methods to direct output to a specific RSN station:

- If the LASTSUBFILE value is 0 and you do not wish to modify that value, change the value of the IORECORDNUM buffer attribute to the desired RSN.
- Change the value of the LASTSUBFILE attribute to the desired RSN.

You can send information to a particular RSN or broadcast the information to all stations. The following ALGOL examples show how to code for either of the preceding situations:

```
WRITE(<file id>[STATION 0],80,A);   %To broadcast

WRITE (<file id>[STATION 3],...); %To station 3 only
```

LFILENAME

Kinds	All
Interrogate	Anytime
Modify	Anytime (disk) When all subfiles are closed (port) When unassigned (all others)
Type	Character string
Range	See following text
Default	Value of INTNAME, if TITLE is not set
Abstract Value	DEFAULT
Stored Permanently	Disk, CD, or tape
Parameters	None
FTAM File Access	Supported
Heritage Host Services Logical I/O	Supported
Port Services	BNANATIVESERVICE, OSINATIVESERVICE, OSISESSIONSERVICE3, BASICSERVICE

Use the LFILENAME attribute to specify a long file name that does not conform to the traditional file naming limits. When read, FILENAME and LFILENAME return the same value. See “FILENAME” for more information.

LIBMAINTAPPEND

Kinds	Tape
Interrogate	Anytime
Modify	Never (See following text)
Type	Enumerated
Range	TOEND, NO
Default	NO
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

Only library maintenance uses the LIBMAINTAPPEND attribute.

Set LIBMAINTAPPEND to TOEND for a destination tape in a WFL COPY or ADD statement or a WFL ARCHIVE statement to cause library maintenance to append the files it copies to the end of the existing library maintenance tape. There must be a LIBMAINTDIR tape directory disk file for the destination tape resident on the DL LIBMAINTDIR disk family or library maintenance will not copy any files to the destination tape.

For further information about the COPY, ADD, or ARCHIVE statements, refer to the *Work Flow Language (WFL) Programming Reference Manual.*

The system ignores any value you specify for LIBMAINTAPPEND if you are running other programs.

LIBMAINTDIR

Kinds	Tape
Interrogate	Anytime
Modify	When closed
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Abstract Value	DEFAULT
Stored Permanently	Yes
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

Only library maintenance uses the LIBMAINTDIR attribute.

For a destination tape, set LIBMAINTDIR to TRUE in a WFL COPY or ADD statement or in a WFL ARCHIVE statement to cause library maintenance to create a LIBMAINTDIR tape directory disk file for that tape. The file contains information about the files copied and the serial numbers of all the tapes used.

For a source tape, set LIBMAINTDIR to TRUE in a WFL COPY or ADD statement, or in an ARCHIVE, RESTORE, or RESTOREADD statement, to cause library maintenance to use the information in the LIBMAINTDIR tape directory disk file for that tape. This speeds up searching for the input files that are to be copied.

For further information about the COPY, ADD, or ARCHIVE statements, refer to the *Work Flow Language (WFL) Programming Reference Manual.*

The system ignores any value you specify for LIBMAINTDIR if you are running other programs.

LICENSEKEY

Kinds	Disk
Interrogate	When open
Modify	When opened (with restrictions)
Type	Character string
Range	Not applicable
Default	Null string
Abstract Value	DEFAULT
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Restricted usage/values
Port Services	Not applicable

Use the LICENSEKEY attribute to determine or specify a key. This key can be used to control library maintenance COPY operations from a *conditioned* tape. A user can only copy a file that has a LICENSEKEY value assigned from a conditioned tape to disk if a matching key and password are present in the SYSTEM/KEYSFILE file. A conditioned tape is created when the COPY statement that copies the keyed file to a tape contains the SW8=TRUE syntax.

Normal backup mechanisms can be used for any disk file with a LICENSEKEY value; that is, they can be copied and retrieved from non conditioned tapes without a key.

The LICENSEKEY attribute returns a null string when no key has been assigned, or the display form of the key, without its password, when a key has been assigned.

A program can assign a value to the LICENSEKEY attribute only if the process performing the assignment is privileged or is being run under a privileged usercode.

Note: *When the LICENSEKEY attribute is assigned, it is permanently stored with the file and cannot be changed. Although two files can have the same LICENSEKEY value, you cannot assign the assigned LICENSEKEY value of one file to another file.*

The format of the value is

```
<key>:<password>
```

The key can consist of one or more alphanumeric fields that contain no embedded blanks, and each field can be separated by a dash (—). The password can consist of 10 to 17 characters with no embedded blanks. Note that the separator between the key and the password is a colon (:). The total length of the key, the colon, and the password cannot exceed 150 characters.

Files with a LICENSEKEY value that is not null can be transferred using Heritage Host Services logical I/O if both systems are using an operating system with a Mark 3.7-or-higher release.

The LICENSEKEY attribute cannot be modified or interrogated in a file that resides on another host.

For information about how to create a conditioned tape for controlled distribution of application programs and how to add a <key>:<password> variable to the SYSTEM/KEYSFILE, refer to the *I/O Subsystem Programming Guide*.

LINENUM

Kinds	Printer
Interrogate	Anytime
Modify	Anytime
Type	Integer
Range	0 through 255
Default	Not applicable
Abstract Value	MINIMUM, MAXIMUM
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

If the PAGESIZE attribute is greater than 0 (zero), interrogate the LINENUM attribute to determine the current line number on the logical page as defined by the PAGESIZE value or modify the LINENUM attribute to change when an end-of-page result is returned.

If the PAGESIZE value is not 0, the LINENUM attribute is initialized to 1 when the printer file is opened; in addition, the LINENUM value is incremented by 1 every time a serial WRITE statement is invoked. When the LINENUM value equals the PAGESIZE value, an end-of-page result occurs, which can be detected by interrogating the STATE attribute. Any special action, such as the printing of subtotals after the end-of-page result is detected, must be done by the program. After the end-of-page result is returned, the LINENUM attribute is reinitialized to 1 and the PAGE attribute value is incremented by 1.

Refer to the *I/O Subsystem Programming Guide* for information about controlling printing.

The error field [16:17] of the result returned for the end-of-page and end-of-file (EOF) conditions has the same format. The I/O result enumeration field uniquely identifies these errors. For more information, refer to the description of ENDOFFILE (46) and ENDOFPAGE (49) in Appendix F.

The LINENUM value cannot be altered by a file equation.

LOCATECAPABLE

Kinds	Tape
Interrogate	Anytime
Modify	When unassigned
Type	Enumerated
Range	ON, OFF, DONTCARE
Default	DONTCARE
Abstract Value	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Not Supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Set the LOCATECAPABLE attribute to ON to indicate that the file requires a tape drive that is capable of processing the READ POSITION and LOCATE BLOCK ID tape commands for fast tape access.

Set the LOCATECAPABLE attribute to OFF to indicate that the file requires a tape drive that does not have Read Position and Locate Block capabilities.

Set the LOCATECAPABLE attribute to DONTCARE to indicate that it does not matter whether or not the file is assigned to a tape drive containing Read Position and Locate Block capabilities. The tape's Locate capability is irrelevant; a tape drive with or without the Read Position and Locate Block capabilities can be used as long as other specified criteria are met.

Interrogate the attribute when the file is assigned to know whether or not the tape drive is capable of processing the READ POSITION and LOCATE BLOCK ID tape commands.

Interrogate the attribute when the file is unassigned will return the value assigned to the attribute.

LOCKEDFILE

Kinds	Disk, tape, printer
Interrogate	Anytime
Modify	Disk anytime Tape when unassigned
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Abstract Value	DEFAULT
Stored Permanently	Yes
Parameters	None
FTAM File Access	Not Supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Use the LOCKEDFILE attribute to determine or specify that a permanent disk file not be removed, replaced, or have its name changed. Also use this attribute to determine or specify that a tape file cannot be accidentally purged. The attribute is effective on tape only when used to create the first file on the tape and applies to all subsequent reels. The tape label contains a value that indicates that the tape was created with LOCKEDFILE set to TRUE.

The following conditions occur when LOCKEDFILE has a value of TRUE:

- When a program requests a write-enabled tape for output that has the same serial number as a tape created with a LOCKEDFILE value of TRUE, the operator must first manually purge the tape before it can be assigned to the program.
- A program cannot close a tape or disk file with the PURGE option until the value of LOCKEDFILE has been changed to FALSE by a privileged user or the owner of the file.
- The WFL *REMOVE* and *CHANGE* commands have no effect on disk files that have LOCKEDFILE set to TRUE.
- Programs cannot replace a disk file that has LOCKEDFILE set to TRUE.

When a logical file is assigned to a physical file and the LOCKEDFILE attribute is interrogated, whether the file is on disk or tape, the attribute value of the physical file is displayed.

Disk Files

An attempt to replace a disk file that has a LOCKEDFILE value of TRUE causes the following message to be displayed:

```
DUP FILE <file name> (LOCKEDFILE)
```

If the file must be removed, a privileged user or the owner of the file must use the WFL *ALTER* statement to change the value of the LOCKEDFILE attribute to FALSE, and then enter the *<mix>OK* command in response to the message. If the *<mix> OF* command is entered, the job continues without replacing the locked file and a close error is returned to the program.

When a CLOSE operation that has a disposition of purge is invoked and the file has a LOCKEDFILE value of TRUE, the CLOSE operation completes successfully without removing the file. The message "FILE NOT REMOVED ON DISK" is issued.

When the LOCKEDFILE value is changed for a logical file that is assigned to a disk file, the value for the permanent disk file is also changed.

A disk file can be removed after it is processed by the Print System as part of a print request. The decision by the system to remove a disk file is based on the following factors:

- The SAVEPRINTFILE attribute
- The LOCKEDFILE attribute
- The REMOVELOCKEDFILES option of the Print System, which is assigned by the PS DEFAULT system command
- Whether the print request for the file was generated automatically or by the WFL *PRINT* statement

If the SAVEPRINTFILE value is TRUE, the file is not removed.

If the SAVEPRINTFILE value is FALSE and the LOCKEDFILE value is also FALSE, the file is removed.

If the SAVEPRINTFILE value is FALSE and the LOCKEDFILE value is TRUE, the removal of the file is determined by the REMOVELOCKEDFILES option of the Print System as follows:

- When the value of REMOVELOCKEDFILES is NEVER, the file is not removed.
- When the value of REMOVELOCKEDFILES is AUTOMATIC, the file is removed only if the print request was generated automatically and not by the WFL *PRINT* statement.

Tape Files

When a tape file has been created with LOCKEDFILE set to TRUE, the following RSVP message is displayed when the PG (Purge) or the SN (Serial Number) system command is used to purge the tape:

```
MTnn CONFIRM PURGE <tape name> <serialno>
```

Note: *If the TAPEMANAGER library authorizes the purge operation, the preceding message does not appear and the tape is purged.*

Respond with an OK (Reactivate) system command to allow the purge to occur. Respond with the DS (Discontinue) system command to stop the purge from occurring.

When a CLOSE operation that has a disposition of purge is invoked and the file has a LOCKEDFILE value of TRUE, the CLOSE operation receives the warning result WARNPURGELOCKEDFILETAPERSLT (25). The tape is not purged.

Printer Files

A printer backup file can be removed after it is processed by the Print System as part of a print request. The decision by the system to remove a printer backup file is based on the following factors:

- The values of the SAVEPRINTFILE and SAVEBACKUPFILE attributes. Refer to the SAVEPRINTFILE attribute for information on how these attributes are reconciled.
- The value of the LOCKEDFILE attribute
- The value of the REMOVELOCKEDFILES option of the Print System
- Whether the print request for the printer backup file was generated automatically or by the WFL *PRINT* statement

If the values of the SAVEPRINTFILE and SAVEBACKUPFILE attributes are TRUE, the file is not removed.

If the values of the SAVEPRINTFILE and SAVEBACKUPFILE attributes are FALSE and the LOCKEDFILE value is also FALSE, the file is removed.

If the values of the SAVEPRINTFILE and SAVEBACKUPFILE attributes are FALSE, and the LOCKEDFILE value is TRUE, then the removal of the file is determined by the REMOVELOCKEDFILES option of the Print System (assigned with the PS DEFAULT system command) and the source of the print request as follows:

- When the value of REMOVELOCKEDFILES is NEVER, the file is not removed.
- When the value of REMOVELOCKEDFILES is AUTOMATIC, the file is removed only if the print request for the file was generated automatically and not by the WFL *PRINT* statement.

LTITLE

Kinds	All
Interrogate	Anytime
Modify	Anytime (disk) When closed (other devices)
Type	Character string
Range	Not applicable
Default	FILENAME
Abstract Value	DEFAULT
Stored Permanently	Disk, CD, or tape
Parameters	1 optional
FTAM File Access	Restricted Use
Heritage Host Services Logical I/O	Supported
Port Services	All

Use the LTITLE attribute to assign a file title that does not conform to the traditional file naming limits. When read, TITLE and LTITLE return the same value. See "TITLE" for more information.

MAXCENSUS

Kinds	Port file or subfile
Interrogate	Anytime
Modify	When CLOSED, or when OPEN is completed
Type	Integer
Range	–1 through 63
Default	63
Abstract Value	DEFAULT, MINIMUM, MAXIMUM
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	BNANATIVESERVICE, NETBIOSSESSIONSERVICE, OSINATIVESERVICE, OSISESSIONSERVICE, BASICSERVICE

Use the MAXCENSUS attribute to accomplish either of the following tasks:

- Specify the maximum number of input messages that can be queued for a port file or a subfile before the service provider requests that the correspondent dialogue endpoint stop sending messages. If the correspondent endpoint is a subfile in the MCP environment, it is suspended when a WRITE WAIT operation is attempted.
- Specify the maximum number of segments that can be queued before the service provider requests that the correspondent dialogue endpoint stop sending messages when you are using the OSINATIVESERVICE or OSISESSIONSERVICE port service and have set the PORTSEGMENTIO file attribute to TRUE.

If the MAXCENSUS attribute is unspecified or 0 (zero), the default value of 63 is assigned when the subfile is opened.

If MAXCENSUS is –1, the operating system requests that the correspondent endpoint stop sending messages immediately, regardless of the number of input messages queued.

The CENSUS attribute returns either the current number of input messages queued if the PORTSEGMENTIO attribute is FALSE or the current number of queued segments if the PORTSEGMENTIO file attribute is TRUE. When the CENSUS value for the subfile reaches the MAXCENSUS value, a request to stop the correspondent endpoint from sending messages is made. A request allowing the correspondent endpoint to resume sending messages is made when enough messages have been read by the subfile to bring its CENSUS value several messages below the MAXCENSUS value. If the MAXCENSUS value is –1, a request allowing the correspondent endpoint to resume sending messages is made when the MAXCENSUS value is changed by the application program to a value from 0 through 63 and enough messages have been read by the subfile to bring its CENSUS value several messages below the MAXCENSUS value.

Because of the time lag in communication between the two dialogue endpoints, some messages sent by the correspondent endpoint might already be in transit when the STOP request is sent. It is important to note that changing the MAXCENSUS value to –1 does not guarantee immediate suspension of the transmission of the correspondent endpoint and that using this value can result in heavier memory utilization for the integrated communications processor (ICP) subsystem.

For port services that are not message-oriented, use the MAXFRAMESIZECENSUS attribute.

This attribute requires a subfile index as a parameter if the MAXSUBFILES value is greater than 1, except when it is used with a file declaration. If it is used in a file declaration or the subfile index specified is zero, then the value of DIALOGPRIORITY is changed in all allocated subfiles and it becomes the new default for any subfiles that are allocated later, even after increasing MAXSUBFILES.

MAXFRAMESIZECENSUS

Kinds	Port file or subfile
Interrogate	Anytime
Modify	Anytime for TCPNATIVESERVICE When closed for TCPIPNATIVESERVICE When CLOSED, or when OPEN has completed
Type	Integer
Range	–1 through 65535 for TCPIPNATIVESERVICE –1 through 16777212 for TCPNATIVESERVICE when FRAMESIZE is equal to 8 –1 through 2796202 for TCPNATIVESERVICE when FRAMESIZE is equal to 48
Default	32768 for TCPIPNATIVESERVICE 120960 for TCPNATIVESERVICE when FRAMESIZE is equal to 8 20160 for TCPNATIVESERVICE when FRAMESIZE is equal to 48
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	TCPIPNATIVESERVICE, TCPNATIVESERVICE

Use the MAXFRAMESIZECENSUS attribute to specify the number of inbound FRAMESIZE units that can be queued for a subfile or a file before the service provider requests that the correspondent dialogue endpoint stop sending messages.

The MAXFRAMESIZECENSUS and FRAMESIZECENSUS attributes have functions that are similar to those of the MAXCENSUS and CENSUS attributes. The MAXFRAMESIZECENSUS and FRAMESIZECENSUS attributes are used for port services that are not message-oriented. Refer to the MAXCENSUS and CENSUS attribute descriptions for information about the functionality of these attributes.

This attribute requires a subfile index as a parameter if the MAXSUBFILES value is greater than 1, except when it is used with a file declaration. If it is used in a file declaration or the subfile index specified is zero, then the value of DIALOGPRIORITY is changed in all allocated subfiles and it becomes the new default for any subfiles that are allocated later, even after increasing MAXSUBFILES.

If the MAXFRAMESIZECENSUS value is –1, the operating system requests the correspondent endpoint to stop sending messages immediately, regardless of the number of input bytes queued.

The FRAMESIZECENSUS attribute returns the current number of input bytes queued. When the FRAMESIZECENSUS value for the subfile reaches the MAXFRAMESIZECENSUS value, a request is made to stop the correspondent endpoint from sending bytes. A request enabling the correspondent endpoint to resume sending bytes is made when enough data has been read by the subfile to bring its FRAMESIZECENSUS value several bytes below the MAXFRAMESIZECENSUS value.

If the MAXFRAMESIZECENSUS value is –1, a request is made enabling the correspondent endpoint to resume sending bytes when the MAXFRAMESIZECENSUS value is changed by the application program to a value from zero (0) through the maximum MAXFRAMESIZECENSUS value, and enough bytes have been read by the subfile to bring its FRAMESIZECENSUS value several bytes below the MAXFRAMESIZECENSUS value.

Because of the time lag in communication between the two dialogue endpoints, some bytes sent by the correspondent endpoint might already be in transit when the STOP request is sent. Changing the MAXFRAMESIZECENSUS value to –1 does not guarantee immediate suspension of the transmission of the correspondent endpoint, and using this value can result in heavier memory utilization for the integrated communications processor (ICP) subsystem.

If you interrogate the value of this attribute for a subport that never has been opened, the value –1 is returned.

MAXRECORDNUMBER

Kinds	Disk
Interrogate	Anytime
Modify	Anytime
Type	Integer
Range	0 through 549755813887
Default	549755813887
Abstract Value	DEFAULT, MINIMUM, MAXIMUM
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

Use the MAXRECORDNUMBER attribute to indicate the largest valid record number allowed in a sequential read or write operation on a file with FILORGANIZATION= RELATIVE.

When performing sequential read or write operations on a relative file, the logical I/O subsystem will return an error if the relative record number is larger than the value of this attribute.

The attribute applies only to sequential I/O and relative files.

MAXRECSIZE

Kinds	All
Interrogate	Anytime
Modify	When closed
Type	Integer
Range	See following text
Default	See following text
Abstract Value	DEFAULT
Stored Permanently	Disk or tape
Parameters	1 optional
FTAM File Access	Restricted values
Heritage Host Services Logical I/O	Restricted values
Port Services	BNANATIVESERVICE, NETBIOSSESSIONSERVICE, OSINATIVESERVICE, OSISESSIONSERVICE, TCPNATIVESERVICE, BASICSERVICE

Use the MAXRECSIZE attribute to determine or specify the maximum size of records in the logical file. The MAXRECSIZE value is expressed in FRAMESIZE units. If the MAXRECSIZE value is unspecified or 0 (zero), a default value is assigned when the file is opened. The default value depends upon the KIND attribute value and the BLOCKSIZE attribute value. Almost all CD-ROM files have a MAXRECSIZE value of 1.

For port files, use the ACTUALMAXRECSIZE attribute for port subfiles and the REQUESTEDMAXRECSIZE attribute for port files, instead of using the MAXRECSIZE attribute. See the descriptions of the ACTUALMAXRECSIZE and REQUESTEDMAXRECSIZE attributes for more details.

The legal MAXRECSIZE range for Heritage Host Services logical I/O files is 1 through 65487. The legal MAXRECSIZE range for all other files except port files is 1 through 65535 FRAMESIZE units. The legal MAXRECSIZE range for FTAM files is shown in the following table.

When an existing disk file is being opened, the acceptable MAXRECSIZE values depend on the value of the logical FILESTRUCTURE attribute, the value of the permanent FILESTRUCTURE attribute, and the BLOCKSTRUCTURE attribute values. The rules for the acceptable MAXRECSIZE values follow:

- If the permanent FILESTRUCTURE value is ALIGNED180 or BLOCKED, the logical MAXRECSIZE value must be consistent with the permanent BLOCKSIZE value.
- If the permanent FILESTRUCTURE value is STREAM and the permanent BLOCKSTRUCTURE value is FIXED or EXTERNAL, the logical MAXRECSIZE value can be any number within the legal range for disk files provided that the number is less than or equal to the AREALENGTH attribute value. If the permanent

BLOCKSTRUCTURE value is not FIXED or EXTERNAL, the logical MAXRECSIZE value must be equal to the permanent MAXRECSIZE value.

- Unless both the permanent and logical FILESTRUCTURE values are ALIGNED180, a logical file with a BLOCKSTRUCTURE value other than FIXED or EXTERNAL can be opened only if the logical MAXRECSIZE value is equal to the permanent MAXRECSIZE value.

When you create or access File Transfer, Access, and Management (FTAM) files, the maximum length of a record depends on the values of the DOCUMENTTYPE, BLOCKSTRUCTURE, and EXTMODE attributes. Refer to the following table for detailed information:

DOCUMENTTYPE	**BLOCKSTRUCTURE**	**EXTMODE**	**Maximum Record Length**
FTAM1	FIXED EXTERNAL	IA5STRING ISOVISIBLESTRING ISOGRAPHICSTRING ISOGENERALSTRING	64000 9995
	VARIABLE	IA5STRING ISOVISIBLESTRING ISOGRAPHICSTRING ISOGENERALSTRING	9995
FTAM2	VARIABLE	IA5STRING ISOVISIBLESTRING ISOGRAPHICSTRING ISOGENERALSTRING	9995
FTAM3	FIXED, EXTERNAL	OCTETSTRING	64000
	VARIABLE	OCTETSTRING	9995
INTAP1	FIXED	OCTETSTRING	64000
	VARIABLE	OCTETSTRING	9995

For printer backup files, the maximum length of a logical record is 255 words or 1530 characters. User records exceeding this size are truncated when they are stored in the backup file.

The BLOCKSTRUCTURE, KIND, MAXRECSIZE, and MINRECSIZE attributes are closely related. The MAXRECSIZE value must be less than or equal to the BLOCKSIZE value. If the BLOCKSTRUCTURE value is FIXED, the BLOCKSIZE value must be a multiple, possibly 0 or 1 times, of the MAXRECSIZE value. If the MINRECSIZE value is greater than the MAXRECSIZE value, the MINRECSIZE value is changed to the MAXRECSIZE value.

If the MINRECSIZE value is greater than 0 and less than the MAXRECSIZE value, and the BLOCKSTRUCTURE attribute is unspecified, the BLOCKSTRUCTURE value is set to EXTERNAL. If the MAXRECSIZE value is 0 and the BLOCKSIZE value is greater than 0, the MAXRECSIZE value is changed to the BLOCKSIZE value.

If both MAXRECSIZE and BLOCKSIZE values are 0 when the file is opened, the default values assigned to both attributes depend upon the kind of peripheral unit assigned to the file. The default values are as follows:

Peripheral Unit	**Default MAXRECSIZE and BLOCKSIZE Values**
DISK or PACK	30 words
ODT	10 words
REMOTE	12 words
PRINTER	22 words
TAPE	10 words
PORT	320 words

If the FRAMESIZE attribute value is other than 48 or if the UNITS value is CHARACTERS, these values are multiplied by the number of characters per word, as specified by the INTMODE attribute.

If a file has a BLOCKSTRUCTURE value of VARIABLE or a FILETYPE value of 1, the SIZEVISIBLE attribute value affects the values of the MINRECSIZE, MAXRECSIZE, and BLOCKSIZE attributes. If the SIZEVISIBLE value is FALSE when a file is created, the I/O subsystem assumes that the MINRECSIZE and MAXRECSIZE values do not include the length of the system-maintained record size field. Thus, the MINRECSIZE and MAXRECSIZE values are adjusted upward accordingly before being stored in the physical file header. If this adjustment results in the MAXRECSIZE value being greater than the BLOCKSIZE value, the BLOCKSIZE value is also adjusted upward. The values returned when you interrogate the MINRECSIZE and MAXRECSIZE attributes reflect the logical values originally specified before they were adjusted to include the system overhead fields. The BLOCKSIZE attribute always returns the actual size of the block being used.

When an existing disk or tape file is being opened and the DEPENDENTSPECS attribute is FALSE and the FILETYPE attribute is not equal to 7 or 8, the logical file MAXRECSIZE value is checked against the MAXRECSIZE value of the physical file for consistency. If an inconsistency is detected, a run-time warning is issued.

A file is considered to be inconsistent if one of the following conditions is true:

- The file is being opened with a BLOCKSTRUCTURE value that is not EXTERNAL or with a FILETYPE value of 0, 1, 2, 4, or 6, and the logical file MAXRECSIZE value is unequal to the physical file MAXRECSIZE value.
- The file is being opened with a BLOCKSTRUCTURE value that is not FIXED or with a FILETYPE value of 1, 2, 3, 4, or 6, and the file has a logical MAXRECSIZE value that causes a blocked file to be accessed as an unblocked file or an unblocked file to be accessed as a blocked file.

  A blocked file is accessed as unblocked when the physical MAXRECSIZE value is unequal to the physical BLOCKSIZE and the logical MAXRECSIZE value is equal to the logical BLOCKSIZE value. An unblocked file is accessed as blocked when the physical MAXRECSIZE value is equal to the physical BLOCKSIZE value and the logical MAXRECSIZE value is unequal to the logical BLOCKSIZE value.

For additional information about inconsistent blocking, refer to the discussion of disk files and magnetic tape files in the *I/O Subsystem Programming Guide*.

MAXSUBFILES

Kinds	Port
Interrogate	Anytime
Modify	See following text
Type	Integer
Range	1 through 65535
Default	1
Abstract Value	DEFAULT, MINIMUM, MAXIMUM
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	All

Use the MAXSUBFILES attribute to specify the maximum number of subfiles that can be opened for the file. The subfiles are assigned indexes from 1 through the value of the MAXSUBFILES attribute, inclusive. Refer to the *I/O Subsystem Programming Guide* for information about port-subfile indexing.

The value of the MAXSUBFILES attribute can be increased anytime. However, the attribute value can be decreased only if all subfiles in the port file are closed. If the value of the MAXSUBFILES attribute is decreased, the amount of memory used by the task is decreased.

If the value of the MAXSUBFILE attribute is increased, the amount of memory used by the task is increased as the additional subfiles are touched. In addition, the values of the attributes of the newly created subfiles have default values unless those attributes were specified on a global basis, either by being set in a file declaration or by being set at run time with a subfile index of zero specified.

MINRECSIZE

Kinds	All except port
Interrogate	Anytime
Modify	When closed
Type	Integer
Range	0 through MAXRECSIZE
Default	Depends on BLOCKSTRUCTURE value
Abstract Value	DEFAULT, MINIMUM, MAXIMUM
Stored Permanently	Disk or tape
Parameters	None
FTAM File Access	Supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Use the MINRECSIZE attribute to determine or specify the minimum size of records in the logical file. MINRECSIZE is expressed in FRAMESIZE units.

If the MINRECSIZE attribute is unspecified or 0 (zero), a default value that depends upon the value of the MAXRECSIZE attribute is assigned when the file is opened. The default value of MINRECSIZE is 0 when the value of BLOCKSTRUCTURE is EXTERNAL. If the BLOCKSTRUCTURE attribute has been specified as FIXED, the default value of MINRECSIZE is the MAXRECSIZE value. However, if the FILETYPE value is 0, the default value of MINRECSIZE is 0.

If MINRECSIZE is greater than MAXRECSIZE, the MINRECSIZE value is changed to the MAXRECSIZE value. If FILETYPE is 0 and MINRECSIZE is greater than 0 but less than MAXRECSIZE, the FILETYPE attribute value is changed to 3 when the file is opened.

The minimum record size used by the I/O subsystem for deblocking a file is the largest of the following values:

- The value of MINRECSIZE
- The minimum record size when the file was created
- The minimum allowable record size, which is dependent upon the values of the BLOCKSTRUCTURE and FILETYPE attributes

Files with a BLOCKSTRUCTURE value that is not equal to FIXED or files with a FILETYPE value that is not 0 require that the minimum record size be large enough to contain the link word or record length information.

If the file has a BLOCKSTRUCTURE value of VARIABLE or a FILETYPE attribute value of 1, the SIZEVISIBLE attribute value affects the values of the MINRECSIZE, MAXRECSIZE, and BLOCKSIZE attributes. If SIZEVISIBLE is FALSE when the file is created, the I/O subsystem assumes that the MINRECSIZE and MAXRECSIZE values do not include the length of the system-maintained record size field. Thus, the MINRECSIZE and MAXRECSIZE values are adjusted upward appropriately before being stored in the physical file header. If this adjustment results in the MAXRECSIZE value being greater than the BLOCKSIZE value, the BLOCKSIZE value is also adjusted upward. The values returned when you interrogate the attributes MINRECSIZE and MAXRECSIZE reflect the logical values originally specified before they were adjusted to include the system overhead fields. BLOCKSIZE always returns the actual size of the block being used.

When an existing disk or tape file is being opened and DEPENDENTSPECS is FALSE or FILETYPE is not equal to 7 or 8, the logical file MINRECSIZE value is checked for consistency against the MINRECSIZE value of the physical file. If the file is being opened with a BLOCKSTRUCTURE value that is not FIXED or EXTERNAL, or with a FILETYPE value of 1, 2, 4, or 6, and if the file has a logical MINRECSIZE value unequal to the physical MINRECSIZE value, a run-time warning is issued.

For additional information about inconsistent blocking, refer to the discussion of disk files and magnetic tape files in the *I/O Subsystem Programming Guide*.

MULTIVOLUME

Kinds	CD
Interrogate	Anytime
Modify	When closed
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Abstract Value	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not supported

Only library maintenance uses the MULTIVOLUME attribute.

If the data being copied overflows a single CD-ROM and MULTIVOLUME is TRUE, a multivolume set of CD-ROMs is written. If MULTIVOLUME is FALSE when data overflow occurs, the copy is terminated with an error.

When a multivolume set is written, data for a given CD-ROM is written to a temporary disk file. That CD-ROM is burned and the temporary disk file is removed before data for the next CD-ROM is written. Free disk space for one CD-ROM is needed.

On copies to multivolume sets, files might be split between CD-ROMs. When MULTIVOLUME is TRUE, it is possible to copy files to CD-ROM that are much larger than the capacity of a single CD-ROM.

Each CD-ROM in a multivolume set has a complete library maintenance directory listing all files in the entire set.

For further information about the MULTIVOLUME attribute, refer to the *Work Flow Language (WFL) Programming Reference Manual*.

MYDOMAINNAME

Kinds	Port
Interrogate	Anytime
Modify	Never
Type	Character string
Range	MYHOST value or value of TCPIPHN
Default	MYHOST value or value of TCPIPHN
Abstract Value	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	TCPIPNATIVESERVICE, TCPNATIVESERVICE

Interrogate the MYDOMAINNAME attribute to determine the complete domain name of the local host after the port file has been opened.

The value of TCPIPHN is set by using the NW TCPIP TCPIPHOSTNAME <name> system command.

MYHOST

Kinds	Port
Interrogate	Anytime
Modify	Never
Type	Character string
Range	Not applicable
Default	Local hostname
Abstract Value	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	BNANATIVESERVICE, OSINATIVESERVICE, OSISESSIONSERVICE, TCPIPNATIVESERVICE, TCPNATIVESERVICE, BASICSERVICE

Interrogate the MYHOST attribute to determine the name of the local host where the logical file is declared. MYHOST is a pointer-valued attribute that contains the name of the local host. This name is used during the subfile matching process.

The value of MYHOST must match the host being addressed by the correspondent dialogue endpoint.

Refer to the *I/O Subsystem Programming Guide* for more information about the use of the MYHOST attribute for matching port subfiles.

MYHOSTGROUP

Kinds	Port
Interrogate	Anytime
Modify	Never
Type	Character string
Range	Not applicable
Default	Host group of local host
Abstract Value	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	BNANATIVESERVICE

Interrogate the MYHOSTGROUP attribute to determine the group of hosts to which the local host belongs. In port matching, the MYHOSTGROUP of a port must match the YOURHOSTGROUP of the correspondent port.

MYHOSTNAME

MYHOSTNAME is the nonpreferred synonym for *MYHOST*. Refer to the description of the MYHOST attribute.

MYIPADDRESS

Kinds	Port subfile
Interrogate	Anytime
Modify	When closed
Type	Character string
Range	Valid IP address
Default	Null
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	TCPIPNATIVESERVICE, TCPNATIVESERVICE

Use the MYIPADDRESS attribute to determine or specify the IP address of the subfile. This address is used in the matching process. The value of MYIPADDRESS must match the IP address specified by the correspondent endpoint of the dialogue.

It is possible for a host to have multiple IP addresses, known as a multihomed host, when the host is connected to multiple IP networks through one or more integrated communication processors (ICPs).

If you do not specify a value for MYIPADDRESS and interrogate the attribute after the subfile is opened, the returned value is one of the valid IP addresses of the local host.

This attribute requires a subfile index as a parameter if the MAXSUBFILES value is greater than 1, except when it is used with a file declaration. If it is used in a file declaration or the subfile index specified is zero, then the value of MYIPADDRESS is changed in all allocated subfiles and it becomes the new default for any subfiles that are allocated later, even after increasing MAXSUBFILES.

MYNAME

Kinds	Port
Interrogate	Anytime
Modify	When closed
Type	Character string
Range	Not applicable
Default	Null string
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	All

Use the MYNAME attribute to specify the name used to identify the port file. The MYNAME value is part of the address information for the subfile and is used at the time the subfile is matched to a correspondent dialogue endpoint. The MYNAME value must match the endpoint being addressed by the correspondent dialogue endpoint before the subfile can be opened. Refer to the description of the YOURNAME attribute and to the discussion of port files in the *I/O Subsystem Programming Guide*.

The MYNAME attribute is a string of 0 to 100 characters terminated by a period (.).

If the MYNAME value is a null string, the following information is true for the identified service:

Service	Information
BNANATIVESERVICE	The correspondent endpoint must have a null YOURNAME attribute value to match.
NETBIOSSESSIONSERVICE	The system generates a unique value, referred to as the *permanent node name*, and uses it for subfile matching instead of the MYNAME value (which remains null). There is no simple way to predict the permanent node name that will be generated by the system. Therefore, the correspondent endpoint typically uses a null YOURNAME attribute value in order to match.
All other services	An OPEN or AWAITOPEN error occurs.

NETBIOSSESSIONSERVICE, and TCPNATIVESERVICE impose limits on the range of values accepted for the MYNAME attribute. Refer to the descriptions of these port services in the *I/O Subsystem Programming Guide*.

MYUSE

Kinds	All except port
Interrogate	When unassigned (Heritage Host Services logical I/O) Anytime (local host)
Modify	When unassigned (Heritage Host Services logical I/O) When closed (local host)
Type	Mnemonic
Range	See following text
Default	See following text
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Restricted usage
Port Services	Not applicable

You can use the MYUSE attribute to determine or specify how the file is to be opened. However, if you are creating a file, it is recommended that you set the NEWFILE attribute to TRUE and use the FILEUSE attribute to specify the way that the file is to be used.

The following are the mnemonic and integer values associated with the MYUSE attribute:

Mnemonic Value	Integer Value
CLOSED	0
IN	1
IO	3
OUT	2

If the file is not explicitly opened and the MYUSE value is not IO, the opening READ or WRITE operation determines the MYUSE value that is assigned by the I/O subsystem: IN or OUT, respectively. If the opening I/O operation is a SEEK or a SPACE operation or if MYUSE has not been set and the file is opened explicitly, MYUSE is set to IN. However, if the peripheral unit is an output-only device, MYUSE is set to OUT.

When you are using Heritage Host Services logical I/O, if you do not specify the NEWFILE attribute, the MYUSE value of OUT sets the value of NEWFILE to TRUE. All other values of MYUSE set the value of NEWFILE to FALSE. If NEWFILE is specified, the value of MYUSE is ignored.

The MYUSE attribute does not imply a request for the update method of access when the logical file is assigned to a disk file. Only the value of the UPDATEFILE attribute determines the method of access. Refer to the description of the UPDATEFILE attribute for more information about the update method of access.

When the file is opened, the KIND, NEWFILE, and MYUSE attributes determine whether a new file is created or a permanent file is assigned. The NEWFILE attribute overrides any effect that MYUSE has on this process. Refer to the explanation of the NEWFILE attribute for more information.

For remote files, a MYUSE value of OUT means output only. MYUSE specified as IN or IO allows both input and output. For more information about remote files, refer to the *I/O Subsystem Programming Guide*.

Heritage Host Services logical I/O does not support modification or interrogation of the MYUSE attribute when the file is assigned.

Section 4
General File Attributes N through S

This section presents file attribute descriptions for attributes with names starting with the letter N through names starting with the letter S. Each attribute description begins with a formatted list of attribute characteristics, followed by a narrative discussion of the use of the attribute. Where appropriate, this discussion includes a list of the various combinations of values, mnemonics, and meanings associated with that attribute.

Refer to the tables in Section 1 for descriptions of the formatted lists.

NETACCESSPOINT

Kinds	Port subfile
Interrogate	Anytime
Modify	When closed
Type	Character string
Range	Not applicable
Default	Null string
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	NETBIOSSESSIONSERVICE

Use the NETACCESSPOINT attribute to specify the network access point through which to access the matching port application. A network access point is a name used to access a local area network (LAN) by way of a particular network processor. Network access points are defined by the network administrator with the following system command:

```
NW HLCN ADD NETACCESSPOINT <identifier> (NP = <np#>)
```

For a discussion of this command, refer to the *Host LAN Connection (HLCN) Installation and Configuration Guide.*

The following are the possible values of NETACCESSPOINT, and their effects:

Mnemonic Value	**Meaning**
Null	This is the default value. The null value can be used only for subfiles that are opened by an AWAITOPEN operation. The null value causes the system to monitor all active network access points for dialogue-establishment requests from the matching application. If other network access points become active while the AWAITOPEN operation is still pending, the program also receives any requests for dialogue establishment from the newly active network access points. When a successful connection is established, the system updates the NETACCESSPOINT value to the specific network access point over which the connection is made.
*NONE	The subfile can match only with applications on the active host.
<identifier>	The system searches for the matching application only on the network access point specified by the identifier. The identifier can be from 1 to 17 characters in length. The characters must be uppercase letters, digits, hyphens (–), or underscores (_).

This attribute requires a subfile index as a parameter if the MAXSUBFILES value is greater than 1, except when it is used with a file declaration. If it is used in a file declaration or the subfile index specified is zero, then the value of NETACCESSPOINT is changed in all allocated subfiles and it becomes the new default for any subfiles that are allocated later, even after increasing MAXSUBFILES.

NEWFILE

Kinds	All except port
Interrogate	Anytime
Modify	When closed
Type	Boolean
Range	Unspecified, TRUE, FALSE
Default	Unspecified
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	None
FTAM File Access	Supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Use the NEWFILE attribute to specify that a new file is to be created. The NEWFILE attribute has three states: unspecified, TRUE, and FALSE.

The following table shows how the I/O subsystem uses the KIND, NEWFILE, and MYUSE values to determine if a new file is created or if a permanent file is assigned when a file is opened:

Kind of Device	NEWFILE Value	MYUSE Value	Existing File Found	Action Taken
Output-only devices such as KIND=PRINTER	Not considered	Not considered	Not applicable	New file created
Input-only devices such as KIND=READER	Not considered	Not considered	Yes	Existing file opened
			No	NO FILE message issued
Input/output devices such as KIND=DISK or KIND=TAPE	TRUE	Not considered	Not applicable	New file created
	FALSE	Not considered	Yes	Existing file opened
			No	NO FILE message issued

Kind of Device	NEWFILE Value	MYUSE Value	Existing File Found	Action Taken
	Unspecified	IN or IO	Yes	Existing file opened
			No	NO FILE message issued
		OUT	Not applicable	New file created

If NEWFILE is TRUE and the file is opened with a READ statement, the end-of-file (EOF) process takes place. This situation is analogous to a READ operation on an empty file.

If you do not specify a NEWFILE value and then you interrogate the NEWFILE attribute, FALSE is returned even if a new file was created according to the default criteria.

For tape files, header labels are written when the NEWFILE attribute is TRUE and are not written when the NEWFILE attribute is FALSE. Header labels can overwrite an existing file on a tape because the NEWFILE value does not change when a file is closed with retention. Thus, if the NEWFILE attribute is TRUE and you close a file by using a *CLOSE (F,REWIND)* statement in ALGOL or a *CLOSE F* statement in COBOL, header labels overwrite an existing file on the tape if you reopen the file and do not change the NEWFILE value to FALSE.

NEXTRECORD

Kinds	All except port and remote
Interrogate	Anytime
Modify	Never
Type	Integer
Range	Not applicable
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Interrogate the NEXTRECORD attribute to determine the current position of the file.

The returned value specifies the record in the file at which a serial READ operation is to be performed if a READ operation is the next I/O operation performed on the file. This record is also where a serial WRITE operation is to be performed, unless the UPDATEFILE attribute value is TRUE. For an explanation of update I/O action, refer to the description of the UPDATEFILE attribute.

If you interrogate the NEXTRECORD attribute before an I/O operation has performed on the file, a 0 (zero) is returned. If a CLOSE operation is performed on a file, the NEXTRECORD value becomes 0.

For direct I/O files with a FILESTRUCTURE value of STREAM or a DIOFILESTRUCTURE attribute value of SECTORSTREAM, the sector number that is to be accessed next by a serial READ operation is returned.

For direct I/O files with a FILESTRUCTURE value of BLOCKED, the block number that is to be accessed next by a serial READ operation is returned.

NONBLOCK

Kinds	Disk
Interrogate	Anytime
Modify	Anytime
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Abstract Value	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

Set NONBLOCK to TRUE so that READ and WRITE operations on a file with a FILEKIND of FIFO do not wait for corresponding data transfers: OPEN operations for read-only do not wait for corresponding WRITE operations, and OPEN operations for write-only do not wait for corresponding READ operations.

The NONBLOCK attribute has no effect unless the FILEKIND value is FIFO.

NORECOVERY

Kinds	Disk
Interrogate	Anytime
Modify	When unassigned
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

Set the NORECOVERY attribute to TRUE to specify that KEYEDIOII data file updates should not be audited.

If the NORECOVERY value is TRUE and a system failure occurs while a KEYEDIOII file is open, the file must be reloaded from a backup copy and the program must be rerun.

The NORECOVERY attribute should be set to TRUE only for batch-type applications that can be rerun easily. When the NORECOVERY value is TRUE, system performance improves significantly.

NORESOURCEWAIT

Kinds	Disk
Interrogate	Anytime
Modify	Anytime
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

Set the NORESOURCEWAIT attribute to TRUE to indicate that a program using the file should not wait for disk space to become available when allocating a new area or activating an old one.

If NORESOURCEWAIT is TRUE and disk space is not available or the continuation pack on which the area of the file being accessed is allocated is not present, the program receives an immediate error result for that I/O operation. If the error occurs, the following events happen:

- The I/O result INSUFFICIENTSPACE (71) is in the result enumeration field ([26:10]).
- The exception ([0:1]) and end-of-tape ([8:1]) bits are set.

Additionally, if the file is a direct I/O file, the following events occur:

- The IORESULT buffer attribute has the exception ([0:1]) and end-of-tape ([8:1]) bits set.
- The IOERRORTYPE buffer attribute value is 5.
- The IOEOF buffer attribute is FALSE.

Regardless of the value of NORESOURCEWAIT, the program waits if one or more of the following conditions occur:

- The file is protected.
- A temporary file has a CLOSE WITH LOCK statement invoked that causes sectors to be required for the directory.
- The FLEXIBLE attribute value of the file is TRUE and directory sectors are required to increase the header size.

NOTE

Kinds	Printer or disk
Interrogate	Anytime
Modify	See following text
Type	Character string
Range	Not applicable
Default	Null string
Abstract Value	DEFAULT
Stored Permanently	Disk and backup file
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

For a printer file, use the NOTE attribute to store a message of up to 250 characters that is to be printed on the banner page preceding the file.

If BANNER is FALSE (the default value), no banner page is printed for the file.

If you assign the BANNER file attribute the value TRUE, the NOTE string is printed in block characters on the banner page. If the NOTE string is null (the default value), then the title of the file being printed is shown in block characters on the banner page. Because the print system sometimes prints a copy of a file rather than the original, the title of the file being printed may not be the same as the title of the file originally requested. For example, when BNA ReprintS is used the file being printed is a renamed copy of the original. For the file name format used by BNA ReprintS, refer to the discussion of remote printing in the *Print System Guide*.

This default content (NOTE string or file title) and format (block characters) of the banner page can be overridden by the PS BANNER system command. For further information about formatting the banner page, refer to the *Print System Guide*.

If you want a period (.) to appear in your NOTE string and the language in which you are writing requires a period at the end of character string, enclose the NOTE string in quotation marks ('' '') and place a period after the last quotation mark. The following are examples of syntax that would generate the strings ABC and ABC.DEF:

ALGOL

```
REPLACE LINE.NOTE BY "ABC."

REPLACE LINE.NOTE BY ""ABC.DEF"""."
```

COBOL74 and COBOL85

```
CHANGE ATTRIBUTE NOTE OF LYNE TO "ABC."

CHANGE ATTRIBUTE NOTE OF LYNE TO """ABC.DEF"""."
```

If you specify a null string, the default, a banner page is not printed.

If the value of the PRINTDISPOSITION attribute is DIRECTDLP or DIRECTPS, the NOTE attribute is ignored.

For a printer file the NOTE attribute can be modified at any time. The contents of the NOTE attribute are stored in the backup file when the file is opened and are stored in the print request whenever the NOTE value is modified.

For disk files, use the NOTE attribute to store a permanent message if the program has write access to the file. That message is printed as a banner when the contents of the file is printed if the following conditions are true:

- The BANNER attribute value is TRUE.
- The print job was initiated by a WFL *PRINT* statement.

You can interrogate the information on the disk at any time and can modify the contents if the program has write access to the file. If the program does not have write access to the file, the program is terminated and a security error is issued when the NOTE attribute is modified.

ODDBLOCKSIZE

Kinds	Tape
Interrogate	Anytime
Modify	When unassigned
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Abstract Value	None
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

Set the ODDBLOCKSIZE attribute of a file with KIND = TAPE specified to ensure that the tape unit assigned to the file can write blocks containing an odd number of bytes, and, for non-direct I/O files, to prevent the system from padding odd-length blocks with an additional byte to make them into even-length blocks.

Odd-length blocks are supported only on the following SCSI tape units:

- CLU9710-DLT
- HS8500
- USR5073, USR5118
- HS4400
- 2145-03
- OST5136
- CTS5236, CLU9710-36T

Note: *Odd-length blocks are not supported on Host Data Unit (HDU) systems and are not supported on DLP connections through MLI channels on Resource Management Module (RMM) systems.*

If you interrogate ODDBLOCKSIZE while the file is assigned, the odd-block-size capability of the unit (given its connectivity to the system) is returned.

OFFSITE

Kinds	Tape
Interrogate	Anytime
Modify	Anytime
Type	Boolean
Range	FALSE/TRUE
Default	FALSE
Abstract Value	Default
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

Use the OFFSITE attribute to instruct the system to update the volume library and volume directory entries for a tape volume when the tape file is closed or when a reel switch occurs. OFFSITE has the following values and results:

Value	Result
FALSE	The CLOSE procedure closes the tape, releases the unit, and updates the volume library and volume directory entries for the tape volume to indicate a status of ONSITE.
TRUE	The CLOSE procedure closes the tape, releases the unit, and updates the volume library and volume directory entries for the tape volume to indicate a status of OFFSITE.
Not explicitly set	The CLOSE procedure does not update the OFFSITE or ONSITE status of the volume library and volume directory entries for the tape volume. Likewise, if the unit is not released during a CLOSE call, the CLOSE procedure does not update the OFFSITE status of the entries for the tape, the volume library, or the volume directory.

The CLOSE procedure updates the volume library entry for the tape only if the site is using the cataloging feature and the tape volume is listed in the volume library. Refer to the OP + Cataloging system command in the *System Commands Operations Reference Manual* and the VOLUME ADD statement in the *Work Flow Language (WFL) Programming Reference Manual*.

The CLOSE procedure updates the volume directory entry for the tape only if the site is using the security tape check feature and the tape volume is listed in the volume directory. Refer to the SECOPT TAPECHECK = AUTOMATIC system command in the *System Commands Operations Reference Manual* and the VOLUME ADD statement in the *Work Flow Language (WFL) Programming Reference Manual*.

The volume library and volume directory entries for the tape are automatically updated to indicate a status of ONSITE under the following conditions:

- During the OPEN procedure, whenever the first file on the tape is overwritten
- During a PURGE procedure, when the tape is purged with the PG MT <unit number> system command or with the CLOSE file (PURGE) system command

OFNOTIFICATION

Kinds	All except port and printer files
Interrogate	Anytime
Modify	When closed or unassigned
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	None
FTAM File Access	Supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

If the OPTIONAL attribute is also TRUE, you can set the OFNOTIFICATION attribute to TRUE to enable your program to distinguish between the following conditions:

- A successful file OPEN operation in which the physical file was assigned to the logical file.
- A successful file OPEN operation in which the operator used the OF (Optional File) system command to cause the program to proceed without a physical file being assigned to the logical file. Refer to the *System Operations Guide* for information about using the OF command.

When the OFNOTIFICATION value is TRUE, the OPEN I/O result enumerated values returned for the conditions are

- OFRSLT (3) if the OF command was entered
- OKRSLT (1) if a physical file was assigned

When the OFNOTIFICATION value is FALSE, the OPEN I/O result enumerated value returned for both conditions is always OKRSLT (1). Refer to Appendix E of this book for more information.

OFNOTIFICATION has no effect if the OPTIONAL attribute is FALSE.

OPEN

Kinds	All
Interrogate	Anytime
Modify	Anytime
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	None
FTAM File Access	Supported
Heritage Host Services Logical I/O	Supported
Port Services	All

Interrogate the OPEN attribute to determine whether a file is open. You can set the OPEN attribute to TRUE to open a closed file, or set the OPEN attribute to FALSE to close an opened file. However, using an OPEN statement is preferable to setting the OPEN attribute to TRUE, and using a CLOSE statement is preferable to setting the OPEN attribute to FALSE, because

- If the requested operation fails, a specific error result is returned if the OPEN or CLOSE statement was used, but only a generic attribute error is issued if the OPEN attribute value was set.
- The OPEN and CLOSE statements can be issued with various options, while setting the OPEN attribute allows only the default option.

When a closed file is opened by using the OPEN attribute, the equivalent ALGOL statements are as follows:

- For port files: OPEN (F[SUBFILE 0],WAIT)
- For all other file kinds: OPEN (F)

When an opened file is closed by using the OPEN attribute, the equivalent ALGOL statements are as follows:

- For port files: CLOSE (F[SUBFILE 0],WAIT)
- For all other file kinds: CLOSE (F,REWIND)

If the file cannot be opened when an attempt is made to set the OPEN attribute to TRUE, or the file cannot be closed when an attempt is made to set the OPEN attribute to FALSE, an attribute error is issued.

If the OPEN attribute is interrogated for a port file or a port subfile, the attribute CHANGEEVENT is reset.

The OPEN attribute cannot be specified in a file declaration or used in a file equation.

OPTIONAL

Kinds	All except port and printer files
Interrogate	Anytime
Modify	When closed
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	None
FTAM File Access	Supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Set the OPTIONAL attribute to TRUE to indicate that the permanent file is an optional file, or interrogate the attribute to determine if a permanent file is allowed to be an optional file.

When the OPTIONAL attribute has been set to TRUE and an attempt is made to open an existing permanent file that cannot be found on the system, an operator can use the OF (Optional File) system command to indicate that the program can proceed without a physical file being assigned to the logical file.

If the OF command is entered, the file is a read-only file. All READ statements result in an end-of-file (EOF) action. Writing to such a file is invalid and causes program termination. A program can determine whether or not an OF command was entered by using the OFNOTIFICATION attribute.

When a permanent file is to be opened only if it is present and available, use an OPEN statement with the AVAILABLE option, or the AVAILABLE attribute, instead of the OPTIONAL attribute. The OPTIONAL attribute is ignored for such operations.

Regardless of the value of the OPTIONAL attribute, if the appropriate file is not present and an available type OPEN was not done, the operator can respond with an FA (File Attribute), NF (No File), or DS (Discontinue) system command.

Refer to the *System Operations Guide* for information about using the OF, FA, NF, and DS commands.

OUTPUTEVENT

Kinds	Port subfile
Interrogate	Anytime
Modify	Never
Type	Event
Range	Not applicable
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	All

The OUTPUTEVENT attribute is a system-defined event that you can use to determine whether or not resources are available to perform a WRITE operation on the subfile.

The OUTPUTEVENT happened state is TRUE when output buffers are available and it is possible to send data to the correspondent program. Otherwise, the OUTPUTEVENT happened state is FALSE. A WRITE WAIT operation suspends processing indefinitely, and a WRITE DONTWAIT operation returns a NOBUFFERFORWRITE result in bit [8:1] of the STATE attribute.

Because the OUTPUTEVENT attribute can be reset between the time it is tested and the time that a WRITE operation is performed, determining if the OUTPUTEVENT happened state is TRUE does not ensure that the next WRITE operation will succeed.

Because OUTPUTEVENT is a read-only attribute, it should not be passed as a parameter to the following statements: CAUSEANDRESET, FIX, FREE, LIBERATE, PROCURE, RESET, SET, and WAITANDRESET. If OUTPUTEVENT is used as a parameter to these statements, the program performs incorrectly, but an error indication is not issued.

The ATTACH and DETACH statements can be used with OUTPUTEVENT; up-level attach errors can occur as described in the discussion of the ATTACH statement in the *ALGOL Reference Manual, Volume 1*.

Attribute errors on event-valued attributes terminate the program.

Refer to the *I/O Subsystem Programming Guide* for a more detailed explanation of the impact of flow control on the WRITE operation when port files are being used.

OUTPUTTABLE

Kinds	All except port
Interrogate	Never
Modify	Anytime
Type	Translate table
Range	Valid translate table identifier
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Use the OUTPUTTABLE attribute to supply a translate table to be used for software translation during a WRITE operation.

If the assigned translate table is declared in a program, that table must be either the first table in a list of tables or the only table in that declaration. Each time the logical file is closed, the value of the OUTPUTTABLE attribute is discarded.

An example of the ALGOL syntax for assigning a translate table to OUTPUTTABLE is as follows:

```
FILEID.OUTPUTTABLE := BCLTOASCII;
```

For further information regarding software translation, refer to the description of the TRANSLATE attribute.

OVERRIDEEXTMODE

Kinds	Disk, CD, or Tape
Interrogate	Anytime
Modify	When closed
Type	Mnemonic
Range	See following text
Default	NEVER
Abstract Value	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Normally, when an existing physical file is opened, the EXTMODE value of the logical file unconditionally assumes the EXTMODE value of the physical file.

Use the OVERRIDEEXTMODE attribute to cause the logical file EXTMODE value to override the EXTMODE of the physical file, thus causing the MCP to treat the physical file as though it had the EXTMODE you specified.

Whether or not the logical file EXTMODE overrides the actual EXTMODE of the physical file, the EXTMODE attribute of the physical file retains the value it had before the file was opened.

The following table lists the mnemonic values of the OVERRIDEEXTMODE attribute and explains their behavior:

Mnemonic Value	Integer Value	Meaning
NEVER	0	The EXTMODE value of the logical file does not override the EXTMODE value of the physical file. In other words, during a successful open of the file, the EXTMODE value of the logical file is unconditionally set to the EXTMODE value of the physical file.
ALWAYS	1	The EXTMODE value of the logical file overrides the EXTMODE value of the physical file if the EXTMODE values are compatible; otherwise, an open error is returned. Specifically, during a successful open of the file, the MCP uses the EXTMODE value of the logical file rather than that of the physical file when the EXTMODE value of the logical file represents 8-bit or 16-bit characters. Additional conditions are described below. If the appropriate conditions are not met, then the file open operation fails and the open error INCOMPATIBLEEXTMODERSLT is returned. **Eight-Bit Characters** If the EXTMODE value of the logical file represents 8-bit characters, then one of the following conditions must be met: The EXTMODE value of the physical file represents 8-bit characters. or The EXTMODE value of the physical file is HEX and The BLOCKSTRUCTURE value of the physical file is FIXED. The FRAMESIZE value of the physical file is 48 or 4. When the FRAMESIZE value of the physical file is 4, then either the MAXRECSIZE value of the physical file must be even, or, if the FILECLASS value of the physical file is CHARACTERSTREAM, the FILELENGTH value of the physical file must be even. The FILEORGANIZATION value of the physical file is NOTRESTRICTED. The DEPENDENTSPECS value of the physical file is TRUE.

Mnemonic Value	Integer Value	Meaning
		Sixteen-Bit Characters If the EXTMODE value of the logical file represents 16-bit characters, then one of the following conditions must be met: The EXTMODE value of the physical file represents 16-bit characters. or The EXTMODE value of the physical file represents 8-bit characters and The BLOCKSTRUCTURE value of the physical file is FIXED. The FRAMESIZE value of the physical file is 48 or 8. If the FRAMESIZE value of the physical file is 8, then the MAXRECSIZE value of the physical file must be even, or, if the FILECLASS value of the physical file is CHARACTERSTREAM, the FILELENGTH value of the physical file must be even. The FILEORGANIZATION value of the physical file is NOTRESTRICTED. The DEPENDENTSPECS value of the physical file is TRUE.
IFOCTETSTRING	2	During a successful open of the file, the MCP uses the EXTMODE value of the logical file rather than that of the physical file when both of the following conditions are met: The EXTMODE value of the logical file represents eight-bit characters (that is, the value is not SINGLE, BCL, BINARY, or HEX) The EXTMODE value of the physical file is OCTETSTRING Otherwise, the EXTMODE value is handled as described for OVERRIDEEXTMODE=NEVER.

Mnemonic Value	Integer Value	Meaning
IFNECESSARY	3	The EXTMODE value of the logical file overrides the EXTMODE value of the physical file if the combination of the INTMODE value and the physical file EXTMODE value requires that translation occur, but the open operation would fail because the required translation tables are not present. Specifically, during a successful open of the file, the MCP uses the EXTMODE value of the logical file rather than that of the physical file when all of the following conditions are true: The EXTMODE value of the logical file represents eight-bit characters (that is, the value is SINGLE, BCL, BINARY, or HEX) If after applying all attributes and conditions affecting the translation of the physical file to the logical file, the opening of the physical file would fail because of lack of translation. Note that translation is never needed when DEPENDENTINTMODE is TRUE The EXTMODE of the physical file represents eight-bit characters (that is, the value is not BCL or HEX) The TRANSLATE attribute has the value DEFAULTTRANS, FORCESOFT, FULLTRANS, or SOFTONLY (that is, system-supplied translation is allowed) Translation is available between the INTMODE value and the EXTMODE value that is stored in the logical file at the time it is opened, or no translation is required because the values are the same. Note that translation can occur in one or both directions as needed according to the IN/OUT/IO use of the file Otherwise, EXTMODE is handled as described for OVERRIDEEXTMODE=NEVER.

OWNER

Kinds	Disk
Interrogate	When assigned
Modify	Only for files within permanent directory
Type	Character string
Range	Not applicable
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

The OWNER attribute returns the usercode value or a null string that corresponds to the file owner.

For typical user files that have a title starting with a usercode in parenthesis, that usercode is returned and the attribute is read-only.

For files that do not have a usercode in the title and are not in the permanent directory namespace, the null string is returned, and the attribute is read-only. A null string indicates that the file is in effect owned by "*" processes.

For files that are within the permanent directory namespace (that is, the title does not begin with *DIR and the PERMDIRECTORIES SYSOPS option is set) the OWNER attribute can be read and modified. It returns the usercode that owns the file, or the null string if the OWNER is "*". Typically the OWNER is the user that created the file, but this is not always the case (e.g., for files created by a COPY command that does not specify the BECOMEOWNER option). The attribute can only be modified by a privileged user or task.

PACKETWRITE

Kinds	CD
Interrogate	Anytime
Modify	When closed
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Abstract Value	DEFAULT
Stored Permanently	Yes
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

Only library maintenance uses the PACKETWRITE attribute.

The PACKETWRITE attribute determines whether the MCP writes the CD-ROM volume in track-at-once mode or packet mode. Track-at-once mode (the default) is more efficient, and the resulting CD-ROM volumes can be read by CD-ROM and CD-R units. Packet mode avoids possible buffer underrun I/O errors during the write process. It is safer, but packet mode CD-ROM volumes can be read only by CD-R units.

For further information about the PACKETWRITE attribute, refer to the *Work Flow Language (WFL) Programming Manual.*

PACKNAME

PACKNAME is a nonpreferred synonym for FAMILYNAME. Refer to the description of the FAMILYNAME attribute.

PAGE

Kinds	Printer
Interrogate	Anytime
Modify	Anytime
Type	Integer
Range	0 through 65535
Default	0
Abstract Value	DEFAULT, MINIMUM, MAXIMUM
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Use the PAGE attribute to determine or specify the number of the current logical page of a printer file. Changing the PAGE value has no effect upon the output of the current physical page and is only meaningful when the PAGESIZE attribute value is greater than 0 (zero). If the PAGESIZE value is not 0 when a printer file is opened, PAGE is initialized to 1. As long as PAGESIZE is not 0 and is less than 255, whenever the LINENUM value becomes greater than or equal to the PAGESIZE value, the PAGE value is incremented by 1, LINENUM is reinitialized to 1, and the WRITE statement returns an end-of-page result. The following is returned in such a situation:

- The I/O result ENDOFPAGE (49) is in the result enumeration field ([26:10]).
- The exception ([0:1]) and end-of-page ([9:1]) bits are set.

The PAGE value is not altered by skip-to-channel actions and cannot be altered by a file equation.

PAGECOMP

Kinds	Disk and printer
Interrogate	Anytime
Modify	When closed
Type	Character string
Range	Not applicable
Default	Null string
Abstract Value	DEFAULT
Stored Permanently	Backup file
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Use the PAGECOMP attribute to specify formatting options for a printer. Printers configured with a transform in the DEVICESUPPORT library support the use of PAGECOMP. For information about the many options available through this file attribute, refer to the *Print System Guide*.

Two examples of valid strings are:

```
LINESPERPAGE=88 CHARACTERSPERLINE=132

LPP=60 CLP=80 PORTRAIT PAPERSIZE=LETTER
```

Lengthy or frequently used PAGECOMP declarations can be stored in a disk file. You can include comments with any record by adding the comment after a percent sign (%).

The PAGECOMP attribute is ignored if the value of the PRINTDISPOSITION attribute is DIRECTDLP or DIRECTPS.

PAGESIZE

Kinds	Printer
Interrogate	Anytime
Modify	Anytime
Type	Integer
Range	0 through 255
Default	0
Abstract Value	DEFAULT, MINIMUM, MAXIMUM
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Restricted usage
Port Services	Not applicable

Use the PAGESIZE attribute to specify the number of lines on a logical page.

When the PAGESIZE value is not 0 (zero), the I/O subsystem maintains the values of the PAGE and LINENUM attributes. When PAGESIZE is set to the default value of 0, the PAGE and LINENUM attributes are ignored.

When the PAGESIZE value is not 0, the I/O subsystem suppresses the normal skip-to-channel-1 action that occurs after it identifies a channel 12 in the carriage control tape. Skipping to channel 1 must be done in the user program; the I/O subsystem supplies an end-of-page result, at the appropriate time, to signal to the program that it has filled a logical page.

If the PAGESIZE value is 0 when the file is opened, that value cannot be changed. Prior to opening the file, the PAGESIZE value can be altered to any value between 1 and 255.

When PAGESIZE is 255, carriage control actions are the same as for other nonzero PAGESIZE settings, except that the end-of-page result is never returned to the program, the PAGE value is not incremented, and LINENUM is counted continuously, wrapping around from 255 to 0. The I/O subsystem still suppresses the skip-to-channel-1 action.

Refer to the description of the SCREENSIZE attribute for information about specifying the page size for a remote file.

PARITY

Kinds	Tape
Interrogate	Anytime
Modify	When closed
Type	Mnemonic
Range	See following text
Default	ODD
Abstract Value	DEFAULT
Stored Permanently	Tape
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

The PARITY attribute always has a value of ODD, which means the tape has binary or odd parity.

Note: *This attribute will be deimplemented in a future release.*

The mnemonic values and meanings associated with the PARITY attribute are as follows:

Mnemonic Value	Integer Value	Meaning
EVEN	1	Alpha or even parity
ODD	0	Binary or odd parity

The value can be set to either ODD or EVEN, but when the attribute is interrogated, ODD is always returned.

PASSIVEOPEN

Kinds	Port subfile
Interrogate	Anytime
Modify	When closed
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	NETBIOSSESSIONSERVICE, TCPIPNATIVESERVICE, TCPNATIVESERVICE

Set the PASSIVEOPEN attribute to TRUE to cause the invocation of the OPEN statement on the specified subfile to behave like an AWAITOPEN statement. Any other attributes affect the OPEN statement in the manner that they affect the AWAITOPEN statement.

The YOURHOST and YOURNAME attribute values of your port file can be a null string when an OPEN operation occurs and the PASSIVEOPEN attribute is set to TRUE.

When both attribute values are null, your port file matches any calling endpoint address when the YOURNAME and YOURHOST attribute values of the calling endpoint match the MYNAME and MYHOST attributes of the port file.

Using the AWAITOPEN statement is preferable to setting the PASSIVEOPEN attribute to TRUE. Refer to the *I/O Subsystem Programming Guide* for information on the AWAITOPEN statement.

If PASSIVEOPEN is FALSE, an active OPEN operation is implied, and the value of the FILESTATE attribute of the subfile becomes OFFERED.

This attribute requires a subfile index as a parameter if the MAXSUBFILES value is greater than 1, except when it is used with a file declaration. If it is used in a file declaration or the subfile index specified is zero, then the value of PASSIVEOPEN is changed in all allocated subfiles and it becomes the new default for any subfiles that are allocated later, even after increasing MAXSUBFILES.

PATHNAME

Kinds	All
Interrogate	Anytime
Modify	Anytime (disk) When closed (other devices)
Type	Character string
Range	<absolute pathname> or <relative pathname>
Default	Equivalent of FILENAME or TITLE
Abstract Value	Not applicable
Stored Permanently	Yes
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	All

Use the PATHNAME attribute to determine or specify an external file name. The external file name is used to associate a logical file with a physical file, or to match corresponding port files. Setting PATHNAME has the same effect as setting LTITLE, except that the syntax of the input character string is different. Thus, setting PATHNAME will overwrite the current LTITLE value, and vice versa. See “LTITLE” and “TITLE” for more information. Refer to the *System Administration Guide* for more information about the SYSOPS LONGFILENAME option.

PATHNAME can be set to an <absolute pathname> or to a <relative pathname>. When read, PATHNAME returns the absolute pathname of that file if the file is assigned to a physical file. If not assigned to a physical file, PATHNAME returns the last path or title, in path syntax, assigned to the file. PATHNAME returns up to 263 characters not including the null character (4"00") some languages append to the end. This character limit is the same whether the SYSOPS LONGFILENAMES is set or reset.

Range

<absolute pathname>

```
── / ─┬──────────────────────┬─┬───────────────────┬─<base pathname>──┤
      └─<family name string>─┘ └─<usercode string>─┘
```

<family name string>

```
── – ── / ──<family name>── / ─────────────────────────────────────────┤
```

<usercode string>

```
── USERCODE ── / ──<usercode>── / ─────────────────────────────────────┤
```

<family name>

One to 17 uppercase letters or digits. A <family name> longer than 17 characters is invalid.

<usercode>

```
——<node>——————————————————————————————————————————————————————┤
```

If the SYSOPS LONGFILENAMES option is reset:

<relative pathname>

```
  ┌←—/15\— / —┐
——┴—<node>————┴———————————————————————————————————————————————┤
```

<base pathname>

```
  ┌←—/11\— / —┐
——┴—<node>————┴———————————————————————————————————————————————┤
```

<node>

One to 17 EBCDIC characters whose values are greater than or equal to space 4"40" except for the slash [/], double quote ["], and left parenthesis [(]. This set includes uppercase and lowercase letters, digits, and most special characters. A <node> longer than 17 characters is invalid.

If the SYSOPS LONGFILENAMES option is set:

<relative pathname>

```
  ┌←——/23\— / ——┐
——┴—<long node>—┴—————————————————————————————————————————————┤
```

<base pathname>

```
  ┌←——/19\— / ——┐
——┴—<long node>—┴—————————————————————————————————————————————┤
```

<long node>

One to 215 EBCDIC characters whose values are greater than or equal to 4"40" except for the slash [/], double quotation ["], and left parenthesis [(]. This set includes uppercase and lowercase letters, digits, and most special characters. A maximum of 215 characters, including the slash characters, can appear in the <base pathname>. A <base pathname> longer than 215 characters is invalid. A <relative pathname> longer than 263 characters is invalid.

A <relative pathname> is appended after the process's CURRENTDIRECTORY to form an <absolute pathname>. If an invalid <absolute pathname> is formed, an error is issued.

Note: *Blank spaces are treated as significant when included in the PATHNAME value. For example, a PATHNAME value of "/A / B"does not match a value of "/A/B".*

The syntax of the PATHNAME attribute requires a string of EBCDIC characters terminated by a null character. The value returned is also terminated by a null character.

The interpretation of PATHNAME is affected by the SEARCHRULE attribute. For further information, see the "SEARCHRULE" attribute in this manual.

If the first node in an absolute pathname is a hyphen (-), then the following node is taken as a family name. If the first node is not a hyphen, then the family name will be determined as described in the description of the SEARCHRULE attribute.

In an absolute pathname, if the node following the family name, if present, is not "USERCODE," the remainder of the pathname is interpreted as the name of an asterisk (*) file. If the node is "USERCODE," the following node is treated as a "USERCODE" value and the remainder of the pathname is interpreted as the name of a usercoded file.

The interpretation of a relative pathname is described in the description of the SEARCHRULE attribute.

If the PATHNAME attribute is interrogated for a disk or CD-ROM file, the name of the physical file is returned. This name will be null-terminated and might include a usercode and family name.

If the PATHNAME attribute is set while the file is unassigned, and the new value includes a family name specification (for example, it begins with "/-/<familyname>"), the FAMILYNAME attribute will also be set to <familyname>. If the new value does not include a family name specification, any current FAMILYNAME attribute setting will be nullified.

If the file's KIND is DISK, changing the PATHNAME while the file is assigned causes the MCP to attempt to change the file name of the physical file also. The family name specification that results from the new PATHNAME value (either explicitly or as a result of pathname resolution using the CURRENTDIRECTORY task attribute) must match the FAMILYNAME attribute of the existing file or the change will not be allowed. Only the owner of the file or a privileged user is permitted to change the name of the file.

Heritage Host Services Logical I/O does not support modification to the PATHNAME attribute of a file while the file is assigned.

Examples

The following examples show TITLE values and the equivalent PATHNAME value:

Title Value	Equivalent PATHNAME Value
(JASMITH)A/B	/USERCODE/JASMITH/A/B
*A/B ON MYFAM	/-/MYFAM/A/B
(JASMITH)A/B ON MYFAM	/-/MYFAM/USERCODE/JASMITH/A/B
A/B	A/B
"a"/"b"	a/b
(JASMITH)A/"b"	/USERCODE/JASMITH/A/b
"a"/b.c"	a/b.c

The following ALGOL statement sets the pathname attribute of file F to the absolute pathname /-/MYPACK/A/B:

```
REPLACE F.PATHNAME BY "/-/MYPACK/A/B"48"00";
```

The following ALGOL statement sets the pathname attribute of file X to the relative pathname MYPROGS/prog1.c:

```
FILE X(PATHNAME="MYPROGS/prog1.c"48"00");
```

The following WFL statement sets the pathname attribute of file G to the relative pathname a/b:

```
RUN PROG2;FILE G(PATHNAME="a/b");
```

PENDINGPROTECTEDFILE

Kinds	CD
Interrogate	Anytime
Modify	Never
Type	Character string
Default	Null string
Abstract Value	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O:	Not supported
Port Services	Not applicable

For information about creating a pending protected file and recovering such a file after a system interruption, refer to the *I/O Subsystem Programming Guide*.

PERMITTEDACTIONS

Kinds	Disk or CD
Interrogate	Anytime
Modify	See following text
Type	Word
Range	See following text
Default	See following text
Abstract Value	Not applicable
Stored Permanently	Disk
Parameters	None
FTAM File Access	Supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

When you use File Transfer, Access, and Management (FTAM) for file creation or access, use the PERMITTEDACTIONS attribute to determine or specify the set of actions that can be performed on the file. When the file is on an enterprise server system, the PERMITTEDACTIONS attribute has an effect only on those persons who are using OSI-FTAM to create or to access the file. You can give the PERMITTEDACTIONS attribute a value only when the file is a disk file and you are creating the file. You cannot modify that value after the file is created.

The actions that can be represented in the bit string of the PERMITTEDACTIONS attribute are shown in the following table:

Field	Permitted Action	Document Type
[00:01]	Read a record. Read a whole file.	FTAM-2 FTAM-1, FTAM-3, INTAP-1
[01:01]	Add records at the end of the file.	FTAM-2
[02:01]	Replace the contents of the file. The old contents are lost.	FTAM-1, FTAM-3, INTAP-1
[03:01]	Add new data at the end of the file.	FTAM-1, FTAM-3, INTAP-1
[04:01]	Delete the contents of the file.	All
[05:01]	Interrogate the attributes.	All
[06:01]	Modify the attributes.	All
[07:01]	Delete the file.	All

Field	Permitted Action	Document Type
[08:01]	Traverse the file from beginning to end using one of the following record identities: Begin First Next Last End	FTAM-2
[09:01]	Traverse the file from the end to the beginning using one of the following record identities: Begin First Previous Last End	FTAM-2
[10:01]	Traverse the file randomly by using one of the following record identities: Current Node number	FTAM-2

If the file is being created by a local user and the PERMITTEDACTIONS attribute is not specified, a default value is assigned to all permitted actions. Refer to the previous table to see what actions are allowed for each document type.

POPULATION

Kinds	Disk or remote
Interrogate	When assigned (disk) When open (remote)
Modify	Never
Type	Integer
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

Interrogate the POPULATION attribute to perform one of the following tasks:

- Determine the number of stations assigned to a remote file.
- Determine the number of functions that are using the disk header of the file.

Remote Files

For remote files, if the value of the POPULATION attribute is not equal to the value of the STATIONCOUNT attribute, the values of the STATIONSDENIED and DISPOSITION attributes can be used to discover the state of the file. Using the STATIONSALLOWED attribute is preferable to using the POPULATION attribute. For more information about remote files, refer to the *I/O Subsystem Programming Guide*.

Disk Files

The POPULATION value cannot be used to reliably determine the number of logical files assigned to a physical disk file, because the value is based on the open count in the disk file header of the file. That count includes the number of files assigned, the number of current executions, and the number of MCP functions currently using the disk file header. MCP functions that use the disk file header include directory search, the processing of a system command that requires access to a file, the asynchronous recording of a runtime warning in a code file, and the presence of a segment dictionary stack for a code file in the CODESTACKS structure cache.

PORTSEGMENTIO

Kinds	Port subfile
Interrogate	Anytime
Modify	When closed
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	OSINATIVESERVICE, OSISESSIONSERVICE

Set the PORTSEGMENTIO attribute to TRUE if you want the application program subport to send and receive partial messages. You can also interrogate this attribute to determine if message segmentation is being used by the subport.

When PORTSEGMENTIO is TRUE, the BLOCKSTRUCTURE file attribute value should be set to EXTERNAL. Otherwise, segments get blank filled.

When PORTSEGMENTIO is TRUE, the CENSUS attribute returns the number of segments that are in the input queue. You can use the MAXCENSUS attribute to indicate the maximum number of segments that can be queued before the service provider requests that the correspondent dialogue endpoint stop sending messages.

This attribute requires a subfile index as a parameter if the MAXSUBFILES value is greater than 1, except when it is used with a file declaration. If it is used in a file declaration or the subfile index specified is zero, then the value of PORTSEGMENTIO is changed in all allocated subfiles and it becomes the new default for any subfiles that are allocated later, even after increasing MAXSUBFILES.

Refer to the *I/O Subsystem Programming Guide* for information about how to program for message segmentation.

PRESENT

Kinds	All except port
Interrogate	Anytime
Modify	Never
Type	Boolean
Range	TRUE, FALSE
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

Interrogate the PRESENT attribute to determine if the file can be opened.

TRUE is returned in the following situations:

- The file was opened before the interrogation or has been opened by the interrogation.
- A new disk file is to be created.

FALSE is returned in each of the following situations:

- The file does not exist.
- The family specified by the FAMILYNAME attribute is not present, or a file with the appropriate value for the SERIALNO, CYCLE, or VERSION attribute does not exist on the specified family.
- The SECURITYUSE attribute value for a physical file is incompatible with the logical file FILEUSE attribute value, or with the logical file MYUSE attribute value if a FILEUSE value is not specified.
- The file is not resident on the disk, but a backup copy of the file is listed in the catalog or archive directory.
- Assigning a logical file to an existing physical file causes an open error. For example, blocking is incompatible or translation is invalid.

If FALSE is returned and the EXCLUSIVE attribute is TRUE or the program is run in an environment in which disk files are opened exclusively, the program can be suspended until the required disk file is unlocked. In this case, the operator is notified by a “WAITING ON” message. Use the AVAILABLE attribute to prevent this condition.

Using the OPEN statement with an AVAILABLE option is preferable to using the PRESENT attribute. An example of such an OPEN statement follows:

```
I := OPEN(F,AVAILABLE);
```

PRINTCHARGE

Kinds	Printer
Interrogate	Anytime
Modify	Anytime
Type	Character string
Range	Valid charge identifier
Default	See following text
Abstract Value	Not applicable
Stored Permanently	Backup file
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

Use the PRINTCHARGE attribute to determine or specify the charge code to be used when printing the printer backup file. The default value is inherited from the charge code of the session or job that issued the explicit print request or from the charge code of the task for which the automatic print request was generated. Otherwise, the default value is a null string.

The system rules for charge code validation are applied by the Print System through the USERDATA interface. If charge code validation is necessary, the value of the PRINTCHARGE attribute is validated when the file is opened. Only 60 characters, including slashes (/), of a charge code are validated.

PRINTCOPIES

Kinds	Printer
Interrogate	Anytime
Modify	Anytime
Type	Integer
Range	1 through 1000
Default	1
Abstract Value	DEFAULT, MINIMUM, MAXIMUM
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

Use the PRINTCOPIES attribute to determine or specify the number of copies of the file that should be printed at each destination. The PRINTCOPIES value is not used if the number of copies is recorded in the DESTINATION attribute information. If the value of the PRINTDISPOSITION attribute is DIRECTDLP or DIRECTPS, the PRINTCOPIES attribute does not apply and is not used by the system.

The default value is one copy for each destination.

PRINTDISPOSITION

Kinds	Disk and printer
Interrogate	Anytime
Modify	When closed
Type	Mnemonic
Range	See following text
Default	See following text
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Use the PRINTDISPOSITION attribute to determine or specify whether a printer backup file should be created, and when the system should issue an automatic print request for the file, if at all.

When a printer file is opened with PRINTDISPOSITION=DIRECT, the PRINTDISPOSITION attribute is changed to the value assigned to the DIRECTPRINTER option of the SYSOPS system command.

The only values allowed for direct printer files are DIRECT, DIRECTDLP, and DIRECTPS. The default for direct printer files is determined by the DIRECTPRINTER option of the SYSOPS system command.

The default value for disk files and non-direct printer files is EOJ. For disk files, the default value is changed at file open time to DONTPRINT unless explicitly assigned.

The mnemonic and integer values are defined as follows:

Mnemonic Value	Integer Value	Meaning
CLOSE or FILECLOSE	2	A print request is automatically issued by the system when the file is closed.
DIRECT	1	When a file with a PRINTDISPOSITION value of DIRECT is opened, the value is changed to DIRECTDLP or DIRECTPS, depending on the value assigned to the DIRECTPRINTER option of the SYSOPS system command. The previous behavior of the DIRECT value is now achieved by specifying DIRECTDLP.

Mnemonic Value	Integer Value	Meaning
DIRECTDLP	7	The file is printed directly on a peripheral printer; no backup file is created and no print request is sent to the Print System. Each record of the file is sent directly to the printer without being buffered to disk and without assistance from the Print System. Each file begins on a new page. Standard banner pages are printed if the LABEL attribute value is STANDARD. DIRECTDLP is valid for direct I/O and non-direct I/O files.
DIRECTPS	8	A print request is issued automatically by the system when the file is opened, but no backup file is created. Each record of the file is sent to the Print System, which then sends the file to the printer without buffering it to disk. The data can be sent to peripheral and remote printers as well as to virtual devices. DIRECTPS is valid for direct I/O files and for nondirect I/O files.
DONTPRINT	0	A backup file is created, but no request for printing is issued by the system. Backup files that are generated when DONTPRINT is specified are prefixed with the usercode of the originator.
EOJ	4	A print request is automatically issued by the system at the end of the job or session in which the file was created.
EOT	3	A print request is automatically issued by the system when the task that created the file is completed.
FILEOPEN	6	A print request is automatically issued by the system when the file is opened.
NOW	5	A print request is issued automatically by the system when the file is opened, but no backup file is created. Each record of the file is sent to the Print System, which then sends it to the printer without buffering it to disk. The data can be sent to peripheral and remote printers, as well as to virtual devices. Each file begins on a new page. The NOW value provides additional functionality from the Print System, but is valid only for non-direct I/O files.

If you use the PRINTDISPOSITION file attribute in a WFL PRINT statement, only the values EOT and EOJ are meaningful. EOT, which is the default value, causes a print request to be generated when the PRINT statement terminates. EOJ causes a print request to be generated when the job or session terminates. Any other value results in EOT being used.

PRINTERBACKUPDATA

Kinds	Disk
Interrogate	When assigned
Modify	Never
Type	Word list
Range	See following text
Default	See following text
Abstract Value	Not applicable
Stored Permanently	Disk, if the PRINTDISPOSITION attribute is set to a value other than DONTPRINT
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not Supported
Port Services	Not applicable

Interrogate the PRINTERBACKUPDATA attribute to obtain printer backup file attribute information. PRINTERBACKUPDATA preserves various file and task attribute information from the creation of a backup file.

The PRINTERBACKUPDATA attribute contains a control word, followed by attribute words, followed by attribute values. The initial control word and attribute word are in a fixed area and the attribute values are in a variable-sized area. Depending on the size of the attribute value, the attribute word either contains the value of the attribute or points to the attribute value location. Not all attributes are necessarily present.

The layout of the PRINTERBACKUPDATA control word and attribute word is as follows:

Name	Field	Description
		Control word layout
PBDVERSIONF	[47:08]	Version = 1
PBDNUMATTSF	[39:08]	Number of attributes stored
PBDLENGTHF	[15:16]	Total length
		Attribute word layout
PBDATTNOF	[47:08]	Attribute number
PBDLINKF	[39:01]	Value flag (index or value) 0 - value 1 - index
PBDVALUEF	[23:24]	Value or index

The PRINTERBACKUPDATA attributes are as follows:

Attribute Name		No.	Link	Description
PBD_ACCESSCODE		1	Yes	ACCESSCODE task attribute.
PBD_CCSINFO		2	No	See layout following.
Name	**Field**			
PBCCSVERSIONF	[15:16]			Value of CCSVERSION file attribute.
PBD_CHARGECODE		3	Yes	CHARGECODE task attribute.
PBD_RECORDINFO		4	Yes	FILEKIND = PRINTFILE total print records (1 word).
PBD_IOMASK		5	No	I/O mask for page specifications.
				A non-zero value occurs when the PAGESIZE file attribute is set. The exception ([0:1]) and end-of-page ([9:1]) bits are set.
PBD_JOBNUMBER		6	No	JOBNUMBER task attribute.
PBD_LINEINFO		7	Yes	FILEKIND = PRINTFILE total print lines (1 word).
PBD_MAXPRINTCHARS		8	No	Maximum block size in EXTMODE characters.
PBD_NAME		9	Yes	NAME task attribute.
PBD_PAGEINFO		10	Yes	FILEKIND = PRINTFILE page information.
				See layout following.
Name_	**Field**			
PBDMAXLPPF	[47:09]			Maximum lines between page skips.
PBDPAGEF	[38:39]			Total page skips.
PBD_PATHINFO		11	No	Originating unit information.
				See layout following.
Name_	**Field**			
ORGISREMOTEF	[15:01]			A value of 1 implies the originating unit is a remote unit; a value of 0 implies the originating unit is not a remote unit.
ORGUNITF	[14:15]			The number of the originating unit of the job that created the file. If the origin was a remote terminal, the number is an LSN. Otherwise, the number is a logical unit number.

Attribute Name		No.	Link	Description
PBD_SOURCENAME		12	Yes	SOURCENAME task attribute.
PBD_MISINFO		13	No	See layout following.
Name	**Field**			
PBDCCSFORMATF	[5:04]			Character set format: 1: ASCII-based characters. 2: EBCDIC-based characters. 14: UCS2 characters. 15: UCS2NT characters.
PBDJOBSUMF	[1:01]			True, if the file is a job summary.
PBDTRIMBLANKF	[0:01]			TRIMBLANKS file attribute.

The PRINTERBACKUPDATA attribute is supported by the GETHEADERATTRIBUTE procedure.

Example

The following ALGOL program demonstrates how to obtain printer backup file information with the PRINTERBACKUPDATA attribute.

```
BEGIN
  FILE  BACKUPFILE
        (TITLE = "BD/YOURFILE ON BACKUPDL.",
         DEPENDENTSPECS, DEPENDENTINTMODE)
       ;
  ARRAY  PBD_INFO[0:99]
        ,TEMP[0:49]
        ;
  DEFINE  PBDNUMATTSF     = [39:08]#
         ,PBDLENGTHF      = [15:16]#
         ,PBDATTNOF       = [47:08]#
              ,PBD_RECORDINFO  =  4#
              ,PBD_JOBNUMBER   =  6#
              ,PBD_LINEINFO    =  7#
              ,PBD_NAME        =  9#
              ,PBD_PAGEINFO    = 10#
              ,PBD_PATHINFO    = 11#
              ,PBD_SOURCENAME  = 12#
         ,PBDVALUEF       = [23:24]#
         ;
  REAL PBD_INX;
  POINTER  P
         , PT
         ;
  LIBRARY MCP (LIBACCESS=BYFUNCTION, FUNCTIONNAME="MCPSUPPORT.");
  BOOLEAN PROCEDURE STANDARDTODISPLAY (PS,PD);
       VALUE PS;
       POINTER PS, PD;
       LIBRARY MCP;

  OPEN(BACKUPFILE);
  REPLACE PBD_INFO BY BACKUPFILE.PRINTERBACKUPDATA;
  IF PBD_INFO[0].PBDLENGTHF > SIZE(PBD_INFO) THEN
       BEGIN
       RESIZE (PBD_INFO, PBD_INFO[0].PBDLENGTHF);
       REPLACE PBD_INFO BY BACKUPFILE.PRINTERBACKUPDATA;
       END;

  FOR PBD_INX := 1 STEP 1 UNTIL PBD_INFO[0].PBDNUMATTSF DO
       BEGIN
       IF PBD_INFO[PBD_INX].PBDATTNOF = PBD_RECORDINFO THEN
           DISPLAY ("PBD_RECORDINFO: " !!
                   STRING(PBD_INFO[PBD_INFO[PBD_INX].PBDVALUEF], *));
       IF PBD_INFO[PBD_INX].PBDATTNOF = PBD_JOBNUMBER THEN
           DISPLAY ("PBD_JOBNUMBER: " !!
                    STRING(PBD_INFO[PBD_INX].PBDVALUEF, *));
       IF PBD_INFO[PBD_INX].PBDATTNOF = PBD_LINEINFO THEN
```

```
           DISPLAY ("PBD_LINEINFO: " !!
                   STRING(PBD_INFO[PBD_INFO[PBD_INX].PBDVALUEF], *));
       IF PBD_INFO[PBD_INX].PBDATTNOF = PBD_PAGEINFO THEN
           BEGIN
           DEFINE PBDMAXLPPF = [47:09] #,
                  PBDPAGEF   = [38:39] #;
           DISPLAY ("PBD_PAGEINFO: " !!
                    "MAX LPP = " !!
                    STRING(
                     PBD_INFO[PBD_INFO[PBD_INX].PBDVALUEF].PBDMAXLPPF,
                     *) !!
                    ", PAGE SKIPS = " !!
                    STRING(
                     PBD_INFO[PBD_INFO[PBD_INX].PBDVALUEF].PBDPAGEF,
                     *));
           END;
       IF PBD_INFO[PBD_INX].PBDATTNOF = PBD_PATHINFO THEN
           BEGIN
           DEFINE ORGISREMOTEF = [15:01] #,
                  ORGUNITF     = [14:15] #;
           IF BOOLEAN(PBD_INFO[PBD_INX].ORGISREMOTEF) THEN
                DISPLAY ("PBD_PATHINFO: " !!
                         "ORGINATING LSN = " !!
                         STRING(PBD_INFO[PBD_INX].ORGUNITF, *))
           ELSE
                DISPLAY ("PBD_PATHINFO: " !!
                         "ORGINATING UNIT = " !!
                         STRING(PBD_INFO[PBD_INX].ORGUNITF, *));
           END;
       IF PBD_INFO[PBD_INX].PBDATTNOF = PBD_NAME THEN
           BEGIN
           REPLACE TEMP BY 0 FOR SIZE(TEMP) WORDS;
           P := POINTER(PBD_INFO[PBD_INFO[PBD_INX].PBDVALUEF]);
           REPLACE PT:TEMP BY "PBD_NAME: ";
           STANDARDTODISPLAY (P,PT);
           DISPLAY (POINTER(TEMP));
           END;
       IF PBD_INFO[PBD_INX].PBDATTNOF = PBD_SOURCENAME THEN
           BEGIN
           REPLACE TEMP BY 0 FOR SIZE(TEMP) WORDS;
           P := POINTER(PBD_INFO[PBD_INFO[PBD_INX].PBDVALUEF]);
           REPLACE PT:TEMP BY "PBD_SOURCENAME: ";
           STANDARDTODISPLAY (P,PT);
           DISPLAY (POINTER(TEMP));
           END;
  END;
END.
```

PRINTERKIND

Kinds	Disk and printer
Interrogate	Anytime
Modify	Anytime
Type	Mnemonic
Range	See following text
Default	DONTCARE
Abstract Value	DEFAULT
Stored Permanently	Backup file
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Use the PRINTERKIND attribute to determine or specify the kind of device the file is to be printed on, without concern for which specific device is to service the print request. More than one value can be specified for each device when the PS CONFIGURE system command is used to assign PRINTERKIND values.

When interrogated, the PRINTERKIND attribute returns the last value that was assigned to it.

The *PS CONFIGURE <device specifier> PRINTERKIND* command is used to inform the Print System of the page description languages (PDLs) that are recognized by the specified device or group of devices. A device configured with a particular PRINTERKIND value is eligible for printing a file with the same PRINTERKIND value. If a device is not configured with specific PRINTERKIND values, it can only print files with the PRINTERKIND value of DONTCARE. For further information about configuring printers with PRINTERKIND values, refer to the *Print System Guide*.

Some PRINTERKIND values identify a general class of printing devices, while others identify a specific PDL. The allowable mnemonic values for PRINTERKIND are as follows:

Mnemonic Value	Integer Value	Meaning
APT	3	The Unisys proprietary Advanced Printing Technology PDL
B9271COMPATIBLE or B9271	16	A printer that is capable of printing Japanese Kanji characters.
B9275	10	The Unisys B9275 non-impact printer PDL
DONTCARE	0	Any type of printer
D630	8	The Diablo 630 PDL
D630E	9	The Extended Diablo 630 PDL
EPSONFX	13	The Epson FX PDL
EPSONLQ	14	The Epson LQ PDL
EXPRESS	15	The Express PDL
HPGL	17	The Hewlett-Packard Graphics Language PDL
IMAGEPRINTER or IP	1	A class of printer that generates the image of an entire page at a time
LINEPRINTER or LP	2	A class of printer that generates output one line at a time
PCL3	4	Level III of the Hewlett-Packard Printer Command Language PDL
PCL4	5	Level IV of the Hewlett-Packard Printer Command Language PDL
PCL5	6	Level V of the Hewlett-Packard Printer Command Language PDL
PLOTTER	12	A class of device that generates output drawn with a pen
POSTSCRIPT or PS	7	The PostScript PDL
POSTSCRIPT2	21	The PostScript level 2 PDL (recognized by the Unisys AP9516 Plus, AP9512 Plus, and UDS9616 printers)
PROPRINTER	11	The IBM Proprinter PDL
PROPRINTERXL	19	The IBM Proprinter XL PDL
RTF	22	Rich Text Format
TTY	18	The Teletype (TTY) PDL (consisting of carriage return, line feed, and form feed)

For a list of Unisys printers that support particular PDLs, refer to the *Print System and Remote Print System Administration, Operations, and Programming Guide*.

The following are examples of assigning a value to the PRINTERKIND attribute in an ALGOL file declaration:

```
FILE F(KIND=PRINTER,PRINTERKIND=IMAGEPRINTER);
FILE F(KIND=PRINTER,PRINTERKIND=LINEPRINTER);
FILE F(KIND=PRINTER,PRINTERKIND=DONTCARE);
FILE F(KIND=PRINTER,PRINTERKIND=POSTSCRIPT);
```

The following are examples of changing the value of the PRINTERKIND attribute in an ALGOL program:

```
F.PRINTERKIND := VALUE(LINEPRINTER);
F.PRINTERKIND := VALUE(IMAGEPRINTER);
F.PRINTERKIND := VALUE(DONTCARE);
```

PRINTPARTIAL

Kinds	Printer
Interrogate	Anytime
Modify	When open
Type	Character string
Range	Not applicable
Default	Null string
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Use the PRINTPARTIAL attribute with the WFL *PRINT* statement to print selected portions of a file. You can select the desired portions of a file by specifying a combination of a single range of column numbers and a list of ascending line, record, page, or sequence numbers.

The following is an example of a valid string value:

```
LINE 1-END
COLUMN 1-80 LINE 30-60, 90, 120
RECORD 0-50
SEQUENCE 1000-1652344
PAGE 1-10
```

Using a WHERE selection clause, you can specify the field in the record that is to be compared to the user-specified text using a specified comparison operator (such as >, >=, <, <=, GTR, GEQ, LSS, and LEQ). In the following example, the syntax prints only those records that contain ABC in columns 1–3:

```
PRINTPARTIAL="WHERE 1û3 = 'ABC'"
```

For detailed information about using this attribute, refer to the *Print System Guide*.

PRINTREQUEST

Kinds	Printer or disk
Interrogate	Anytime
Modify	Never
Type	Integer
Range	Not applicable
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Use the PRINTREQUEST attribute to interrogate the print request number that is assigned to a file when it becomes eligible for printing.

The PRINTREQUEST attribute is valid only for open files with PRINTDISPOSITION attribute values of DIRECTPS, FILEOPEN, or NOW. An error results if PRINTREQUEST is interrogated for files with other PRINTDISPOSITION values or for files that are closed.

The PRINTREQUEST attribute returns 0 (zero) under the following circumstances:

- The file has a PRINTDISPOSITION value of DIRECTPS, FILEOPEN or NOW, but the active PRINTSUPPORT library is too old to support the PRINTREQUEST attribute.
- The file has a PRINTDISPOSITION value of DIRECTPS, FILEOPEN, or NOW, but the print request for the file was deleted.

If PRINTREQUEST is interrogated for a file with a PRINTDISPOSITION value of DIRECTPS, FILEOPEN, or NOW and the PRINTSUPPORT library is not active, then the program waits until PRINTSUPPORT is active.

If a file has a PRINTDISPOSITION value of DIRECTPS, FILECLOSE, or NOW and multiple DESTINATION values, the PRINTREQUEST attribute returns only the print request number that is used for one of the resulting requests.

If PRINTREQUEST is interrogated for a file with a PRINTDISPOSITION value of FILECLOSE and the program closes the file with the PURGE option, then the print request number assigned to the file's PRINTREQUEST attribute is omitted from the print request list displayed by the PS SHOW command, the sumlog file, and the program's job summary.

PRODUCT

Kinds	Disk
Interrogate	Anytime
Modify	See following text
Type	Character string
Range	Not applicable
Default	Null String
Abstract Value	DEFAULT
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Use the PRODUCT attribute to specify or determine that a piece of your software belongs to a product group. The PRODUCT attribute is used to associate a piece of software with a specific application, product, and component, as defined within the problem tracking and reporting system of Unisys.

The identifier for the PRODUCT can be 1 to 250 characters long and must be terminated by a period (.) or a space. The identifier also can contain embedded periods or blanks if the string value is enclosed in quotation marks (“ ”) and followed by a period or space. If you interrogate the PRODUCT attribute, the returned value is terminated by a period or space.

If the PRODUCT attribute is interrogated when the file is assigned, the physical file PRODUCT value is returned. If the PRODUCT attribute is interrogated when the file is unassigned and the attribute has been modified previously, the value of the last setting is returned. If the PRODUCT value has not been modified, a null string is returned.

If the PRODUCT value is modified when the file is closed and unassigned, the specified value is retained in the logical file for use when the file is opened. When the file is opened, a nonfatal attribute error is issued if the physical file has a PRODUCT value. If the physical file does not have a PRODUCT value, the value retained in the logical file becomes the value in the physical file.

If the PRODUCT value is modified when the file is assigned, an attribute error occurs if the physical file contains a PRODUCT value, or if you do not have write access to the file. In these cases, the value for the physical file is not changed. If the physical file does not contain a PRODUCT value and you have write access to the file, the value in the physical file is changed to the value specified. Modifying the PRODUCT attribute to null when a physical file is assigned has no effect if the physical file does not contain a PRODUCT value.

PROPAGATESECURITYTODIRS

Kinds	Disk
Interrogate	When assigned
Modify	When assigned
Type	Mnemonic
Range	See following text
Default	DONTPROPAGATE
Abstract Value	Not applicable
Stored Permanently	Yes
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

The PROPAGATESECURITYTODIRS file attribute allows file security attributes to be propagated from a containing permanent directory file to new permanent directories.

The mnemonic and integer values and meanings associated with PROPAGATESECURITYTODIRS are as follows:

Mnemonic Value	Integer Value	Meaning
DONTPROPAGATE	0	Don’t propagate file security attributes
PROPAGATE	1	Propagate file security attributes

The default value of PROPAGATESECURITYTODIRS is DONTPROPAGATE.

The PROPAGATESECURITYTODIRS attribute is meaningful only for permanent directories (FILEKIND = PERMDIR). Assigning the attribute to other disk files is not permitted.

The PROPAGATESECURITYTODIRS attribute is only used when the logical file is assigned to a physical file. Assigning or interrogating the PROPAGATESECURITYTODIRS attribute while the logical file is unassigned results in an attribute error.

If the PROPAGATESECURITYTODIRS attribute is interrogated when the file is assigned, the value for the physical file is returned. The value DONTPROPAGATE is returned if the physical file does not have the PROPAGATESECURITYTODIRS attribute set.

If the PROPAGATESECURITYTODIRS attribute is set when a logical file is assigned to a physical file, the MCP attempts to change the PROPAGATESECURITYTODIRS attribute of the physical file; this can only be done by the owner of the file or a privileged user. See "Setting the SECURITYMODE Attribute" for other special circumstances that also prevent setting the PROPAGATESECURITYTODIRS attribute.

Security Attributes

The following set of file attributes is referred to collectively as "file security attributes" within this description:

- SECURITYMODE and its subattributes
- SECURITYTYPE
- SECURITYUSE
- SECURITYGUARD
- GROUP
- ALTERNATEGROUPS

The PROPAGATESECURITYTODIRS file attribute causes these file security attributes to be propagated from a containing permanent directory file to a new permanent directory.

Security Attribute Defaults Within Permanent Directories

The default security attributes for a new permanent directory file (FILEKIND = PERMDIR) residing within a containing permanent directory depend upon the PROPAGATESECURITYTODIRS attribute of the containing permanent directory.

If PROPAGATESECURITYTODIRS = DONTPROPAGATE, then the PROPAGATESECURITYTODIRS attribute has no effect on any of the values of the new permanent directory file's security attributes. Or, if any of the file security attributes are set on the new permanent directory, the PROPAGATESECURITYTODIRS attribute has no effect on the values of the new permanent directory's file security attributes. The default value for each file security attribute is specified in the description of the attribute.

When PROPAGATESECURITYTODIRS = PROPAGATE and none of the file security attributes are set on the new permanent directory, then the values of all the file security attributes are inherited by the new permanent directory from the containing permanent directory. In addition to the file security attributes, the PROPAGATESECURITYTODIRS and PROPAGATESECURITYTOFILES attributes are also inherited.

The PROPAGATESECURITYTODIRS attribute causes propagation of all or none (not just some) of the file security attributes. For example, when creating a new permanent directory, *DIR/D/E, the permanent directory, *DIR/D, must already exist and is called the containing permanent directory. If the PROPAGATESECURITYTODIRS attribute of *DIR/D is set to PROPAGATE, and no security attributes were set on directory *DIR/D/E, the file security attributes of *DIR/D are propagated to the new permanent directory, *DIR/D/E, by default.

Refer to "Securing Files" in the *Security Features Operations Programming Guide* for more information about the use of the PROPAGATESECURITYTODIRS attribute.

PROPAGATESECURITYTOFILES

Kinds	Disk
Interrogate	When assigned
Modify	When assigned
Type	Mnemonic
Range	See following text
Default	DONTPROPAGATE
Abstract Value	Not applicable
Stored Permanently	Yes
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

The PROPAGATESECURITYTOFILES file attribute allows file security attributes to be propagated from a containing permanent directory file to new files.

The mnemonic and integer values and meanings associated with PROPAGATESECURITYTOFILES are as follows:

Mnemonic Value	Integer Value	Meaning
DONTPROPAGATE	0	Don't propagate file security attributes
PROPAGATE	1	Propagate file security attributes

The default value of PROPAGATESECURITYTOFILES is DONTPROPAGATE.

The PROPAGATESECURITYTOFILES attribute is meaningful only for permanent directories (FILEKIND = PERMDIR). Assigning the attribute to other disk files is not permitted.

The PROPAGATESECURITYTOFILES attribute is only used when the logical file is assigned to a physical file. Assigning or interrogating the PROPAGATESECURITYTOFILES attribute while the logical file is unassigned results in an attribute error.

If the PROPAGATESECURITYTOFILES attribute is interrogated when the file is assigned, the value for the physical file is returned. The value DONTPROPAGATE is returned if the physical file does not have the PROPAGATESECURITYTOFILES attribute set.

If the PROPAGATESECURITYTOFILES attribute is set when a logical file is assigned to a physical file, the MCP attempts to change the PROPAGATESECURITYTOFILES attribute of the physical file; this can only be done by the owner of the file or a privileged user. See "Setting the SECURITYMODE Attribute" for other special circumstances that also prevent setting the PROPAGATESECURITYTOFILES attribute.

Security Attributes

The following set of file attributes is referred to collectively as “file security attributes” within this description:

- SECURITYMODE and its subattributes
- SECURITYTYPE
- SECURITYUSE
- SECURITYGUARD
- GROUP
- ALTERNATEGROUPS

The PROPAGATESECURITYTOFILES file attribute causes these file security attributes to be propagated from a containing permanent directory file to a new file contained within that permanent directory.

Security Attribute Defaults Within Permanent Directories

The default security attributes for a new file (without FILEKIND = PERMDIR) created within a containing permanent directory depend upon the PROPAGATESECURITYTOFILES attribute of the containing permanent directory.

If PROPAGATESECURITYTOFILES = DONTPROPAGATE, then the PROPAGATESECURITYTOFILES attribute has no effect on any of the values of the new file's security attributes. Or, if any of the file security attributes are set on the new file, the PROPAGATESECURITYTOFILES attribute has no effect on the values of the new file’s file security attributes. The default value for each file security attribute is specified in the description of the attribute.

When PROPAGATESECURITYTOFILES = PROPAGATE and none of the file security attributes are set on the new file, then the values of all the file security attributes are inherited by the new file from the containing permanent directory.

The PROPAGATESECURITYTOFILES attribute causes propagation of all or none (not just some) of the file security attributes. For example, when creating a new file, *DIR/D/F, the permanent directory *DIR/D must already exist and is called the containing permanent directory. If the PROPAGATESECURITYTOFILES attribute of *DIR/D is set to PROPAGATE, and no security attributes were set on the file *DIR/D/F, the file security attributes of *DIR/D are propagated to the new file *DIR/D/F by default.

Refer to “Securing Files” in the *Security Features Operations Programming Guide* for more information about the use of the PROPAGATESECURITYTOFILES attribute.

PROTECTION

Kinds	Disk, tape, or CD
Interrogate	Anytime
Modify	When closed
Type	Mnemonic
Range	PROTECTED and TEMPORARY (disk, tape, or CD) SAVE (disk or CD)
Default	TEMPORARY
Abstract Value	DEFAULT
Stored Permanently	Disk, tape, or CD
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Restricted usage
Port Services	Not applicable

Use the PROTECTION attribute to determine or specify the amount of extra effort that should be exerted to preserve a file in case of a system failure.

The mnemonic values associated with the PROTECTION attribute are as follows:

Mnemonic Value	Integer Value
PROTECTED	2
SAVE	1
TEMPORARY	0

The PROTECTION value of a physical file cannot be modified once that file has been created.

If PROTECTION is interrogated when the file is assigned, the value for the physical file is returned.

If the physical file is a permanent disk file that was not created with PROTECTION set to PROTECTED, a value of SAVE is returned.

Disk Files

If the value of PROTECTION is PROTECTED, an entry is made immediately in the directory when a new file is opened. As the disk areas are allocated, the directory is updated and the areas are encoded with a pattern that makes it possible to discover the last valid block written on that area in the event of a system failure.

Whenever an existing protected disk file is opened, the file header in the directory is checked to determine the condition of the last CLOSE operation. If the most recent use of the file was concluded by an orderly or normal CLOSE operation, the end-of-file (EOF) pointer is correct and no special action is taken. If the most recent use of the file was terminated abnormally and the file might have been extended within its last row—for example, a file being written when a system interruption occurred—one of the following special actions is taken:

- If one or more areas at the end of the file have been destroyed—for example, if the KEEP option was specified in an RC (Reconfigure Disk) system command for the family member containing those areas—and the end-of-file (EOF) pointer in the file header points within those areas, the file header value is not changed even though the areas no longer exist.
- If one or more areas at the end of the file have not been destroyed, the last remaining allocated area of the file is examined, and the actual end of the file is determined by the boundary, within that area, between the user data and the special pattern that was written when the area was allocated.

In either case, the EOF position obtained is then used to update the EOF pointer in the file header, and the header is marked so that subsequent OPEN operations can assume that the file header value is correct.

The value PROTECTED should not be specified for files that are to be written synchronously. It offers no advantage when the SYNCHRONIZE attribute value is OUT; in fact, some extra overhead is incurred. The EOF restoration mechanism for files with a PROTECTION value of PROTECTED sets the end-of-file (EOF) indicator to a disk sector boundary. As a result, the EOF indicator of a file being written synchronously might be restored after a system failure to a record location slightly beyond that of the last record written, if the record size is not a multiple of the sector size of the disk.

The EOF restoration mechanism for a file with a FILESTRUCTURE value of STREAM places the EOF marker at the end of the disk sector that was last written, even though the end of that sector might not be the end of a record.

If the value of PROTECTION is SAVE, an entry in the directory is made immediately when the new file is opened. The file becomes a permanent disk file and remains after the program has finished unless the file is explicitly purged. However, no special action is taken to ensure that the correct EOF pointer is maintained across a system failure, so blocks written to the file might not be recoverable after the system failure.

If the value of PROTECTION is TEMPORARY and a disk file is being created, that disk file is discarded when the program is terminated or when the program exits the block in which the file was declared, unless the file is explicitly closed by execution of an overriding CLOSE statement—for example, LOCK or CRUNCH. If the disk file is locked or crunched, an entry is made in the directory and the file becomes permanent.

Tape Files

If the file has been updated and the value of PROTECTION is PROTECTED, a tape mark is written on the tape before it is rewound following a system failure. This action allows the file to be read after the system is reinitialized.

The SAVE value has no meaning for tape files.

If the value of PROTECTION is TEMPORARY, no special action is performed. The disposition of the tape file is determined by the value of the SAVEFACTOR attribute. Refer to the description of the SAVEFACTOR attribute.

The PROTECTION attribute is not supported by Heritage Host Services logical I/O for tape files.

PROVIDERGROUP

Kinds	Port subfile
Interrogate	Anytime
Modify	When closed
Type	Character string
Range	Not applicable
Default	Null string
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	All

Use the PROVIDERGROUP attribute to restrict the group of service providers that are to provide the service for the subfile. If PROVIDERGROUP is null, the system selects the service provider to support the service. The default value for PROVIDERGROUP is null.

The following table lists the port services and the system software that can provide them:

Service	**Provider**
BASICSERVICE	BNA Version 2, LPP, OSI
BNANATIVESERVICE	BNA Version 2, LPP
NETBIOSSESSIONSERVICE	HLCN
OSINATIVESERVICE	OSI
OSISESSIONSERVICE	OSI
TCPIPNATIVESERVICE	TCP
TCPNATIVESERVICE	TCP

A provider group is a group of one or more of the providers listed above. Provider groups are declared through the *NW NS ADD PROVIDERGROUP* system command. The following are the predeclared PROVIDERGROUP names for each provider:

- *LPP for Local Port Provider
- *HLCN for Host LAN Connection
- *TCPIP for TCP

The value of PROVIDERGROUP must be a 1- to 17-character identifier, and the first character must be an asterisk (*), a digit, or a letter. Provider group identifiers beginning with an asterisk are reserved for predeclared provider groups.

This attribute requires a subfile index as a parameter if the MAXSUBFILES value is greater than 1, except when it is used with a file declaration. If it is used in a file declaration or the subfile index specified is zero, then the value of PROVIDERGROUP is changed in all allocated subfiles and it becomes the new default for any subfiles that are allocated later, even after increasing MAXSUBFILES.

Note: *Subports with the values of *BNAV2, *BNAOSI, and *LPP can be mixed in the same port file if the SERVICE attribute of the port file is set to BASICSERVICE. The OPEN or AWAITOPEN operation fails with a PROVIDERRESTRICTEDRSLT (144) open error and a PROVIDERRESTRICTED (26) SUBFILEERROR if the service provider to be used cannot be mixed with the providers already being used by the subfiles of the port file. If PROVIDERGROUP is null, the system attempts to use a provider that is compatible with the providers already in use by the port file.*

READDATE

Kinds	Disk
Interrogate	When assigned
Modify	Never (see following exception)
Type	Integer
Range	70001 through 135365
Default	CREATIONDATE
Abstract Value	Not applicable
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not supported

Interrogate the READDATE attribute to determine the last time the physical file was closed after records were read from the file.

The value returned by READDATE is a decimal integer. When you add the integer 1900000 to the value returned, the result is the date in the form YYYYDDD, where YYYY and DDD represent the year and date respectively, in Julian form.

The READDATE value is unaffected by library maintenance and is not updated for files on a disk which is not write-enabled.

The READDATE attribute is used in conjunction with the READTIME and READTZ attributes. See “READTIME” and “READTZ” for more information.

The READDATE attribute can be modified by a valid MCS program if that program previously set the HSFILECOPY file attribute to TRUE.

READDATEUT

Kinds	Disk
Interrogate	When assigned
Modify	Never
Type	Integer
Range	70001 through 135365
Default	CREATIONDATE
Abstract Value	Not applicable
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not supported

Interrogate the READDATEUT attribute to determine the READDATE value with respect to UTC time instead of the READTZ time zone. See “READDATE” for more information.

When you interrogate the READDATEUT attribute, you receive an error if the time zone attribute, READTZ, is 0 (zero). See “READTZ” for more information.

READREVERSECAPABLE

Kinds	Tape
Interrogate	Anytime
Modify	When unassigned
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Abstract Value	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Set the READREVERSECAPABLE attribute to TRUE to indicate that the file requires a tape drive that is read-reverse capable.

Interrogating the attribute when the file is assigned indicates whether the tape drive is a read-reverse capable unit.

Interrogating the attribute when the file is unassigned returns the value assigned to the attribute.

READTIME

Kinds	Disk
Interrogate	When assigned
Modify	Never (see following exception)
Type	Integer
Range	0 through 86399999999
Default	CREATIONTIME
Abstract Value	Not applicable
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not supported

Interrogate the READTIME attribute to determine the time of day, in microseconds since midnight, that is associated with the READDATE attribute value. The READTIME value is updated when the READDATE value is updated.

The READTIME attribute can be modified by a valid MCS program if that program previously set the HSFILECOPY file attribute to TRUE.

READTIMEUT

Kinds	Disk
Interrogate	When assigned
Modify	Never
Type	Integer
Range	0 through 86399999999
Default	CREATIONTIME
Abstract Value	Not applicable
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not supported

Interrogate the READTIMEUT attribute to determine the READTIME value with respect to UTC time instead of the READTZ time zone. See “READTIME” for more information.

When you interrogate the READTIMEUT attribute, you receive an error if the time zone attribute, READTZ, is 0 (zero). See “READTZ” for more information.

READTZ

Kinds	Disk
Interrogate	When assigned
Modify	Never (see following exception)
Type	Integer
Range	0 through 255
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	Yes
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Interrogate the READTZ attribute to determine the time zone that is associated with the READDATE and READTIME attribute values. The READTZ value is updated when the READDATE value is updated.

To determine the meaning of non-zero values of READTZ, refer to the Appendix G “Predefined Time Zones.” You can use the MCP library procedure TIMEZONENAME to convert the numeric code of a time zone into the name or abbreviation for that time zone. TIMEZONENAME is documented in the *Master Control Program (MCP) System Interfaces Programming Reference Manual.*

If the integer value of READTZ is 0 (zero), then there is not a time zone associated with READDATE and READTIME.

The READTZ value is 0 if the file

- Has CREATIONTZ = 0 and READDATE and READTIME are equal to CREATIONDATE and CREATIONTIME.
- Resides on a disk family with FAMILY HEADER VERSION = 6. Refer to “Disk File Headers” in Section 1 for more information.
- At the time the READDATE and READTIME were set, the system did not have a time zone set.

The READTZ attribute can be modified by a valid MCS program if that program previously set the HSFILECOPY file attribute to TRUE.

RECEPTIONS

Kinds	Remote
Interrogate	When open
Modify	Never
Type	Integer
Range	Not applicable
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

Interrogate the RECEPTIONS attribute to determine the number of messages received from the station when the relative station number (RSN) is specified as a parameter, or the number of messages received from all stations of the remote file when the parameter is omitted.

RECORD

Kinds	All except CD, port, and remote
Interrogate	Anytime
Modify	Never
Type	Integer
Range	Not applicable
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

You can use the RECORD attribute to determine the current position in the file. For disk files, the returned value is valid only for a file that has a FILESTRUCTURE value of ALIGNED180. Use of the NEXTRECORD attribute is preferred because more meaningful values for more situations are returned. Refer to the NEXTRECORD attribute for more information.

When the file is open but no I/O operation has been performed upon the file, or if the file was opened with a READ[NO] operation, or is closed, RECORD returns the value –1.

Since the first record in a file has a record number of 0, RECORD returns the value of 0 when pointing at the first record in the file. When reading in reverse, the RECORD attribute starts at 0, the same as when reading forward.

For open direct I/O disk files, the RECORD attribute reports the NEXTRECORD value.

RECORDINERROR

Kinds	Disk, CD, or tape
Interrogate	When open
Modify	Never
Type	Integer
Range	Not applicable
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

Interrogate the RECORDINERROR attribute to determine the record number or block number of the data at the beginning of the currently used buffer. This information might help you diagnose an I/O error when it occurs. Refer to the description of the IOINERROR attribute.

A file with a FILESTRUCTURE value of STREAM and a BLOCKSTRUCTURE value of FIXED returns the first record number that has any data in the current buffer. For other files with a FILESTRUCTURE value of STREAM, the information returned is the sector number of the beginning of the buffer.

A file with a FILESTRUCTURE value of ALIGNED180 or BLOCKED and a BLOCKSTRUCTURE value of FIXED returns the record number. All other files with a FILESTRUCTURE value of ALIGNED180 or BLOCKED return the block number.

RECORDLEVELLOCK

Kinds	Disk
Interrogate	Anytime
Modify	When unassigned
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

If you are creating a new KEYEDIOII file, you can set the RECORDLEVELLOCK attribute to TRUE to indicate that record-level locking is to be enforced. This attribute cannot be changed for an existing file.

Records are locked each time one is read if the RECORDLEVELLOCK value is TRUE and the value of FILEUSE is IO, and the file is opened for updating. Other users who attempt to read a locked record must wait until the record is freed. A record is freed if one of the following actions is taken:

- The record is rewritten.
- Another record is read.
- A new record is written.
- The record is deleted.
- The file is closed.

Refer to the description of the FILEORGANIZATION attribute for information about KEYEDIOII files.

REDIRECTION

Kinds	Not applicable
Interrogate	Anytime
Modify	When unassigned
Type	Boolean
Range	TRUE
Default	Unassigned
Abstract Value	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

The REDIRECTION attribute is a pseudo-attribute whose purpose is to cause other attributes to be assigned default values. Specifically, when REDIRECTION is assigned the value TRUE, then the file is used to access a file on a remote system using the Client Access Services redirector.

Setting the REDIRECTION attribute causes the following attribute assignments:

- KIND is set to VALUE (VIRTUAL).
- IOHFUNCTIONNAME is set to “REDIRSUPPORT.”.
- IOHLIBACCESS is set to VALUE (BYFUNCTION).

When REDIRECTION is used, the program must specify the appropriate server, share, and file using the FILENAME and IOHSTRING attributes.

Interrogating the REDIRECTION attribute returns the value TRUE under the following conditions:

- KIND=VALUE (VIRTUAL)
- IOHFUNCTIONNAME=“REDIRSUPPORT.”
- IOHLIBACCESS=VALUE (BYFUNCTION)

REEL

REEL is a nonpreferred synonym for FILESECTION. Refer to the description of the FILESECTION attribute.

REINITIALIZE

Kinds	All
Interrogate	Never
Modify	When unassigned
Type	Boolean
Range	True
Default	Not applicable
Abstract Value	None
Stored Permanently	No
Parameters	None
FTAM File Access	Supported
Heritage Host Services Logical I/O	Supported
Port Services	All (see following exception)

Set the REINITIALIZE attribute to TRUE to revert a file to its originally declared condition and to apply any relevant label equation. You can use this attribute to reuse a single logical file for different physical files. When REINITIALIZE is set to TRUE, any attributes that the logical file may have obtained since it was first referenced are reinitialized to their original values.

The REINITIALIZE attribute cannot be used in a file equation nor can it be used in the FA operator command. It is not available in WFL.

An attempt to set REINITIALIZE to TRUE for a file with dynamic attribute initialization, or for a POSIX file, results in an attribute error.

RELEASEID

Kinds	Disk
Interrogate	Anytime
Modify	See following text
Type	Character string
Range	Not applicable
Default	Null string
Abstract Value	DEFAULT
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Use the RELEASEID attribute to specify or determine that a piece of your software belongs to a certain release level. The RELEASEID attribute is used to associate a piece of software with a specific Mark release. The identifier for the software release can be 1 to 250 characters long and must be terminated by a period (.) or a null character, and can contain embedded periods or blanks if the string value is enclosed in quotation marks (" ") and followed by a period or a null character. If you interrogate RELEASEID, the returned value is terminated by a period or a null character.

If the associated physical file is not a disk file, the RELEASEID value in the logical file is not used when the file is opened, and an attribute error is issued.

If the RELEASEID attribute is interrogated when the file is assigned, the physical file RELEASEID value is returned. If the RELEASEID attribute is interrogated when the file is unassigned and the attribute has previously been modified, the value of the last setting is returned. If the RELEASEID value has not been modified, a null string is returned.

If the RELEASEID value is modified when the file is closed and unassigned, the specified value is retained in the logical file for use when the file is opened. When the file is opened, a nonfatal attribute error is issued if the physical file has a RELEASEID value. If the physical file does not have a RELEASEID value, the value retained in the logical file becomes the value in the physical file.

If the RELEASEID value is modified when the file is assigned, an attribute error occurs if the physical file contains a RELEASEID value, or if the user does not have write access to the file. In these cases, the value for the physical file is not changed. If the physical file does not contain a RELEASEID value and the user has write access to the file, the value in the physical file is changed to the value specified. The value retained in the logical file is also changed to the value specified.

Modifying RELEASEID to null when a physical file is assigned has no effect if the physical file does not contain a RELEASEID value. An attribute error occurs if the physical file contains a RELEASEID value.

REQUESTEDMAXRECSIZE

Kinds	Port
Interrogate	Anytime
Modify	When closed
Type	Integer
Range	0 through 65535 if FRAMESIZE = 8 0 through 10922 if FRAMESIZE = 48
Default	See following text
Abstract Value	DEFAULT, MINIMUM, MAXIMUM
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	BNANATIVESERVICE, NETBIOSSESSIONSERVICE, OSINATIVESERVICE, OSISESSIONSERVICE, TCPNATIVESERVICE, BASICSERVICE

Use the REQUESTEDMAXRECSIZE attribute to request the maximum length, in FRAMESIZE units, to be used in READ and WRITE operations for this file.

This attribute is used by the local system to determine the ACTUALMAXRECSIZE value, the actual maximum length that is to be used in READ and WRITE operations. ACTUALMAXRECSIZE can be lower than REQUESTEDMAXRECSIZE because of network implementation restrictions or because of record length negotiation between the two dialogue endpoints.

The default is 320 words. If the value of FRAMESIZE is 8 or if the UNITS attribute value is CHARACTERS, the default is 1920 characters. If REQUESTEDMAXRECSIZE has not been assigned a value, it does not assume the default value until an OPEN or AWAITOPEN operation is invoked. Before this, REQUESTEDMAXRECSIZE has the value of 0.

TCPIPNativeService users may specify a maximum segment size (MSS) to use for a particular connection. The suggested MSS is passed to TCP by the RequestedMaxRecSize attribute. A suggested maximum transmission unit (MTU), the value the TCPIP library eventually uses must fall within certain criteria. The MTU used is the minimum of the following.

- MTU returned from IP (avoid fragmenting messages)
- The user requested MTU
- The MSS of the remote's SYN/SYNACK MSS option

Refer to the discussion of the ACTUALMAXRECSIZE attribute for more information and length restrictions for the different services.

The function of this attribute is equivalent to that of the MAXRECSIZE attribute when MAXRECSIZE is accessed without a subfile index.

RESIDENT

Kinds	All except port
Interrogate	Anytime
Modify	Never
Type	Boolean
Range	TRUE, FALSE
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

Interrogate the RESIDENT attribute to determine if a permanent file exists or a physical unit is available without causing the logical file to be opened or the program to be suspended. If the logical file is already assigned or the file exists, TRUE is returned. The RESIDENT attribute is unaffected by the EXCLUSIVE attribute value. TRUE is returned if the file exists even if the disk file is being used exclusively by another program. In general, the RESIDENT attribute tests for the existence of the identified permanent file and not for compatibility with the logical file. A RESIDENT value of TRUE does not necessarily mean that a subsequent open operation will succeed. When a new file is to be created—for example, if the NEWFILE attribute value is TRUE—the RESIDENT attribute returns TRUE if the appropriate physical unit is available whether or not a permanent file exists.

If RESIDENT returns TRUE and the DEPENDENTSPECS attribute value is TRUE or if the FILETYPE attribute value is 7 or 8, the attribute values that describe the permanent file, BLOCKSIZE, EXTMODE, FILESTRUCTURE, MAXRECSIZE, MINRECSIZE, SIZEMODE, SIZEOFFSET, SIZE2, and UNITS, are changed to reflect the values of the permanent file. The FILETYPE value is not changed. If the DEPENDENTSPECS value is TRUE, the BLOCKSTRUCTURE attribute value is also changed to reflect the value of the permanent file, and the FRAMESIZE attribute value is changed to the INTMODE attribute value. If the FILETYPE value is 7, the INTMODE value is changed to the EXTMODE value of the permanent file.

If RESIDENT returns TRUE and the DEPENDENTINTMODE attribute value is TRUE, the INTMODE value is changed to the EXTMODE value of the permanent file.

FALSE is returned in each of the following situations:

- The family specified by the FAMILYNAME attribute is not present, or a file with the appropriate SERIALNO, CYCLE, or VERSION attribute does not exist on the specified family.
- The SECURITYUSE attribute value for a physical file is incompatible with the logical FILEUSE attribute value, if specified, or with the logical MYUSE attribute value, if FILEUSE is not specified.
- The file is not resident on the disk, but a backup copy of the file is listed in the catalog or archive directory.

RESTRICTED

Kinds	Disk or CD
Interrogate	When assigned
Modify	Never
Type	Boolean
Range	TRUE, FALSE
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Interrogate the RESTRICTED attribute to determine if access to the file is restricted.

A restricted file cannot be read from, written to, or opened, except by someone with security-administrator status or, if security-administrator status is not authorized, privileged status. No one, regardless of status and including the owner of the file, can execute a file while it is restricted. A restricted file can be copied by those with privileged status or security-administrator status.

Access to a file is restricted if one of the following conditions is true:

- The unit on which the file resides was restricted by the RESTRICT (Restrictions) system command.
- The file itself was restricted by the RESTRICT system command.
- The file is an existing file on a remote host, and the HOSTSRESTRICTED option of the SECOPT system command is set on the local host.
- The file was created using Heritage Host Services logical I/O at a host with the HOSTSRESTRICTED option of the SECOPT command set.
- The file was copied from a restricted unit or volume.
- The file was copied to a host that had the HOSTSRESTRICTED option of the SECOPT command set.

TRUE indicates that the file is restricted.

In addition, when a file is restricted, the FILEKIND attribute returns the value RESTRICTEDFILE. Refer to the FILEKIND attribute for more details about what the attribute returns and to the *Security Administration Guide* for more information about restrictions on files, units, volumes, and hosts.

RESULTLIST

Kinds	All
Interrogate	When open
Modify	Never
Type	Word list
Range	Not applicable
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Supported
Heritage Host Services Logical I/O	Supported
Port Services	All

Interrogate the RESULTLIST attribute to obtain the list of results caused by the most recent logical I/O function performed on the file. The RESULTLIST attribute includes the results found in the STATE attribute plus any additional results. The information returned is a string of binary-valued words. The first word contains the length of the string in words, self-inclusive, and the subsequent words contain listed values.

A value of 1 in the first word indicates that the most recent I/O operation on the file has no results to report other than a successful completion.

The mnemonic values, integer values, and meanings for the RESULTLIST attribute are described in Appendix F, "I/O Result Enumerated Values."

For port files, this attribute is both a port file and subfile attribute. If a subfile index is specified, the value returned by the RESULTLIST attribute reports the result of the most recent I/O operation on the specified subfile. If no subfile index is specified, the result of the most recent I/O operations on the port file is returned.

If the file is not a port file and a parameter is specified, an attribute error occurs.

REWRITEABLE

Kinds	Tape
Interrogate	Anytime
Modify	When Unassigned
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Set the REWRITEABLE attribute to TRUE to indicate that an input tape should be mounted in a write-enabled state and on a unit that supports the write operation.

Interrogate the REWRITEABLE attribute while the file is assigned to determine whether the tape can be written. Interrogate the attribute while the file is unassigned to return the value assigned to the attribute.

When REWRITEABLE is set to TRUE, a tape volume which cannot be written will not be assigned to the logical file.

Library Maintenance sets REWRITEABLE to TRUE for its destination tapes. This ensures that a tape which is being written by the COPY & COMPARE command is write-enabled during the compare phase after a reel switch, so that the end-of-file label and all subsequent files can be written.

ROWADDRESS

ROWADDRESS is a nonpreferred synonym for AREAADDRESS. Refer to the description of the AREAADDRESS attribute.

ROWSINUSE

ROWSINUSE is a nonpreferred synonym for AREASINUSE. Refer to the description of the AREASINUSE attribute.

SAVEBACKUPFILE

Kinds	Printer
Interrogate	Anytime
Modify	Anytime
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

The SAVEBACKUPFILE attribute is functionally synonymous with the SAVEPRINTFILE attribute for printer backup files. SAVEPRINTFILE is the preferred attribute. The SAVEBACKUPFILE attribute will become a synonym for the SAVEPRINTFILE attribute in a future release. For more information, refer to the SAVEPRINTFILE attribute in this section.

SAVEFACTOR

Kinds	Disk or tape
Interrogate	Anytime
Modify	See following text
Type	Integer
Range	0 through 999
Default	0
Abstract Value	DEFAULT, MINIMUM, MAXIMUM
Stored Permanently	Disk or tape
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Use the SAVEFACTOR attribute to specify the expiration date in terms of the number of days past the creation date. The SAVEFACTOR value is stored with the permanent file when the file is created and cannot be modified after creation by modifying the value. The value of the SAVEFACTOR attribute for disk files can be modified by using the WFL *ALTER* statement. The SAVEFACTOR attribute returns the value for the physical file if the file is interrogated when the file is assigned.

Tape Files

A tape is considered expired if both of the following are true:

- The first file on the tape volume has a LABELKIND value of ASERIESDATA, B5500STANDARD, or B3500USASI.
- The number of days specified by the SAVEFACTOR value has passed since the day of creation.

An expired tape is considered a scratch tape if the SYSOPS TAPEEXPIRATION option is set and the tape unit is write-enabled.

When the first file on a tape volume has a LABELKIND value that indicates a library maintenance tape or printer backup file, the system never treats the tape as a scratch tape.

Because the default for SAVEFACTOR is 0 (zero) if you fail to specify a value, a write-enabled tape is considered an expired tape by the I/O subsystem when the tape label is subsequently read. This occurs even when you release a tape and leave the tape on the tape drive.

The LISTVOLUME utility reports the value of SAVEFACTOR for the first file on each tape reported.

Disk Files

The SAVEFACTOR value for disk files has the following effects:

- Determines whether a file is to be selected by rollout by the WFL *ARCHIVE ROLLOUT* statement.
- Sometimes determines whether the SORT procedure locks the file. For ALGOL programs, if the value is other than 0 (zero) or the PROTECTION file attribute value is PROTECTED, the SORT procedure locks the file. If the value is 0, the SORT procedure closes the file without locking it.

The SAVEFACTOR value is used in the FILEDATA and FILECOPY utilities in the following ways:

- The FILEDATA utility reports the SAVEFACTOR value for permanent files.
- The FILECOPY utility uses the SAVEFACTOR attribute value to determine if a file should be put in a list of expired files. Refer to the *System Software Utilities Manual* for information about the FILECOPY utility.

SAVEPRINTFILE

Kinds	Disk and printer
Interrogate	Anytime
Modify	Anytime
Type	Boolean
Range	TRUE, FALSE
Default	See following text
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

Use the SAVEPRINTFILE attribute to specify whether a file is to be saved or removed following a print request.

If you set SAVEPRINTFILE to TRUE, the file is saved after it is processed by the Print System as part of a print request.

If the value of the PRINTCOPIES attribute or the number of copies derived from the DESTINATION attribute is greater than 1, the file is removed after the last copy has been printed.

Disk Files

For disk files, the default value for SAVEPRINTFILE is TRUE.

If you set SAVEPRINTFILE to FALSE and the value of the LOCKEDFILE attribute to FALSE, the file is removed.

If you set SAVEPRINTFILE to FALSE and LOCKEDFILE to TRUE, the removal of the file is determined by the REMOVELOCKEDFILES option of the Print System and the source of the print request as follows:

- When the value of REMOVELOCKEDFILES is NEVER, the file is not removed.
- When the value of REMOVELOCKEDFILES is AUTOMATIC, the file is removed only if the print request was generated automatically and not by the WFL *PRINT* statement.

Printer Files

For printer backup files, the SAVEPRINTFILE attribute is functionally synonymous with the SAVEBACKUPFILE attribute. SAVEPRINTFILE is the preferred attribute. If you specify SAVEPRINTFILE for a printer backup file, the value of the SAVEBACKUPFILE attribute is ignored. The SAVEBACKUPFILE attribute will become a synonym for the SAVEPRINTFILE attribute in a future release.

The default value for SAVEPRINTFILE when used for printer backup files is determined by the source of the print request as follows:

- If the print request is generated by opening or closing a file, the default is FALSE.
- If the print request is generated by the WFL *PRINT* statement, the default is TRUE.

The printer backup file can be removed after being processed by the Print System as follows:

- If the result of the SAVEPRINTFILE and SAVEBACKUPFILE values is FALSE and the LOCKEDFILE value is also FALSE, the file is removed.
- If the result of the SAVEPRINTFILE and SAVEBACKUPFILE values is FALSE and the LOCKEDFILE value is TRUE, then the removal of the file is determined by the REMOVELOCKEDFILES option of the Print System (assigned with the PS DEFAULT system command) and the source of the print request:
  - When the value of REMOVELOCKEDFILES is NEVER, the file is not removed.
  - When the value of REMOVELOCKEDFILES is AUTOMATIC, the file is removed only if the print request for the file was generated automatically and not by the WFL *PRINT* statement.

The SAVEPRINTFILE attribute does not apply to printer backup files on tape.

SCRATCHPOOL

Kinds	Tape
Interrogate	Anytime
Modify	Anytime
Type	Character string
Range	See following text
Default	Null string
Abstract Value	DEFAULT
Stored Permanently	Yes
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Use the SCRATCHPOOL attribute to specify or determine the pool of a tape volume. If a value is specified before the file is opened for output, it is used to select a scratch tape from the specified scratch pool. The value of the SCRATCHPOOL attribute is an identifier of 1 to 17 EBCDIC characters, left-justified in a field of blanks (a hyphen or an underscore is not permitted as the first character).

You can place a scratch tape in a specific pool by naming that pool when you use the PG (Purge) or SN (System Number) system command.

If you request a printer backup tape that specifies a SCRATCHPOOL value, the file will be added on to any existing mounted backup tape from that pool.

If you set a new value after the file has been assigned to a physical file, the pool name is stored in the label if the tape volume is later closed with the PURGE option. If you do not want to change the pool name in the purge operation, do not specify a SCRATCHPOOL value after the file is assigned or use SETABSTRACTVALUE(DEFAULT).

If you want a tape to be put into the default pool, specify a null string (''.'').

The following table describes the values that are returned when the SCRATCHPOOL attribute is interrogated by a program:

When the Attribute is Interrogated . . .	The Returned Value is . . .
Before the file is opened	The specified value or a null string if no value is specified.
After the file is opened	The value that was specified after the file was assigned, or the SCRATCHPOOL value of the physical file.

If the SERIALNO file attribute has been specified for a particular member of a single volume or a multivolume set, the SCRATCHPOOL attribute is not taken into consideration when that volume is assigned.

SCREEN

Kinds	Remote
Interrogate	When open
Modify	Never
Type	Boolean
Range	TRUE, FALSE
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	No
Parameters	1 required
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Interrogate the SCREEN attribute to determine if the specified station is declared in the DATACOMINFO file as a screen device. The SCREEN attribute returns a value of TRUE when the station is declared as a screen device.

The SCREEN attribute requires a relative station number (RSN) as a parameter.

SCREENSIZE

Kinds	Remote
Interrogate	When open
Modify	Never
Type	Integer
Range	Not applicable
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	No
Parameters	1 required
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Interrogate the SCREENSIZE attribute to determine the number of lines on a page as defined in the DATACOMINFO file for the specified station. The SCREENSIZE attribute requires a relative station number (RSN) as a parameter.

The following ALGOL example shows how to interrogate the SCREENSIZE attribute:

```
NUMBEROFLINES := REMOTEFILEID (RSN).SCREENSIZE;
```

SEARCHRULE

Kinds	Disk
Interrogate	Anytime
Modify	When unassigned
Type	Mnemonic
Range	NATIVE or POSIX
Default	NATIVE
Abstract Value	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not supported

Use the SEARCHRULE attribute to specify whether POSIX or native A Series rules are to be used in searching for an existing file or creating a new file. SEARCHRULE is effective only if the file's KIND is DISK.

Note that native A Series rules correspond to traditional A Series behavior. POSIX rules provide a behavior that is compatible with POSIX requirements.

Effects of the SEARCHRULE Attribute

There are four conditions that might occur when using the SEARCHRULE attribute in a native or POSIX environment. The specific effects of the SEARCHRULE attribute are described in the following paragraphs.

1. Neither an asterisk (*), usercode, nor family name is specified, and the FAMILYNAME attribute is not set.

Examples

```
TITLE=A/B         FAMILYNAME = <not set>

PATHNAME=A/B      FAMILYNAME = <not set>
```

NATIVE The file will be searched for on DISK under both the process’s usercode, if any, and the asterisk (*) nodes. However, if the target family of the process’s FAMILY attribute is DISK, the file will instead be searched for on both the primary and alternate, if defined, families.

A new output file will be created under the process’s usercode node, if the process is running under a usercode, or the asterisk (*) node, on DISK (or the primary family, if the target family is DISK).

POSIX The file name will be treated as a relative pathname. The value of the process’s CURRENTDIRECTORY task attribute will be prefixed to the pathname representation of the file name, and the combined file name will be treated as described in conditions 2, 3, or 4 depending on the format of the combined pathname.

The CURRENTDIRECTORY value always begins with a slash, so the combined file name will always have a leading slash. A pathname with a leading slash is called an *absolute pathname*. For further information about the CURRENTDIRECTORY task attribute refer to the *Task Attributes Programming Reference Manual*.

2. An asterisk (*) or usercode is specified, but no family name is specified and the FAMILYNAME attribute is not set.

Examples

```
TITLE=*A/B                        FAMILYNAME= <not set>

TITLE=(JASMITH)A/B                FAMILYNAME= <not set>

PATHNAME=/A/B                     FAMILYNAME = <not set>

PATHNAME=/USERCODE/JASMITH/A/B  FAMILYNAME = <not set>
```

NATIVE The file will be searched for on DISK, or on both the primary and alternate families if the FAMILY target is DISK. A new output file will be created on DISK, or on the primary family if the target family is DISK.

POSIX The file will be searched for or created on the family defined by the DL ROOT system command. Family substitution will not apply. See the *System Commands Operations Reference Manual* for more information on the DL ROOT system command.

3. Neither an asterisk (*) nor a usercode is specified, but either a family name is specified or the FAMILYNAME attribute is set.

Examples

```
TITLE=A/B ON APACK    FAMILYNAME = <not set>

TITLE=A/B             FAMILYNAME = APACK

PATHNAME=A/B          FAMILYNAME = APACK
```

NATIVE The file will be searched for under both the process’s usercode, if any, and the asterisk (*) nodes on the specified family. However, if the specified family matches the target family of the process’s FAMILY attribute, the file will instead be searched for on both the primary and alternate, if defined, families.

A new output file will be created under the process’s usercode node, if the process is running under a usercode, or the asterisk (*) node, on the specified family (or the primary family if the specified family matches the target family).

POSIX The file will be searched for or created under the asterisk (*) node on the specified family. Family substitution will not apply.

4. An asterisk (*) or usercode is specified, and either the family name is specified or the FAMILYNAME attribute is set.

Examples

```
TITLE=*A/B ON APACK                      FAMILYNAME = <not set>

TITLE=(JASMITH) ON APACK                 FAMILYNAME = <not set>

TITLE=*A/B                               FAMILYNAME = APACK

TITLE=(JASMITH) A/B                      FAMILYNAME = APACK

PATHNAME=/A/B                            FAMILYNAME = APACK

PATHNAME=/-/APACK/A/B                    FAMILYNAME = <not set>

PATHNAME=/USERCODE/JASMITH/A/B           FAMILYNAME = APACK

PATHNAME=/-/APACK/USERCODE/JASMITH/A/B   FAMILYNAME = <not set>
```

NATIVE The file will be searched for on the specified family. However, if the specified family matches the target family of the process’s FAMILY attribute, the file will instead be searched for on both the primary and alternate if defined, families.

A new output file will be created on the specified family (or the primary family, if the specified family matches the target family).

POSIX The file will be searched for or created on the specified family. Family substitution will not apply.

Treatment of the File Name Nodes Dot (.) and Dot-Dot (..)

The SEARCHRULE attribute also controls the treatment of the file name nodes dot (.) and dot-dot (..) in the following ways:

NATIVE The file name nodes are treated as ordinary file name nodes.

POSIX The dot (.) file node refers to the directory indicated by its predecessor node in a pathname.

The dot-dot (..) file node refers to the parent of its predecessor node.

This convention enables a program to reference file name nodes that are upstream as well as downstream from its current position. A consequence of the way in which family and usercode specifications are implemented in the PATHNAME syntax is that encountering dot-dot nodes in a file name during the resolution process can cause a change in the usercode or family name specification that was originally determined when the TITLE or PATHNAME attribute was assigned.

Examples of SEARCHRULE = POSIX

Pathname	**Pathname after Evaluation of Dot and Dot-Dot**
/A/B/./C	/A/B/C
/A/B/../C	/A/C
/-/APACK/A/../../ANOTHERPACK/B	/-/ANOTHERPACK/B

SECTORSIZE

Kinds	Disk or CD
Interrogate	When assigned
Modify	Never
Type	Integer
Range	See following text
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Interrogate the SECTORSIZE attribute to determine the size in bytes of a logical sector on the family where the file resides. For some disks, the system emulates a logical sector size that is different from the physical sector size of the disk. For example, VSS-1 and VSS-2 disks emulate 180-byte sectors on physical 512-byte sectors.

SECURITYADMIN

Kinds	Disk
Interrogate	Anytime
Modify	Anytime
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Abstract Value	DEFAULT
Stored Permanently	Yes
Parameters	No
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

Set the SECURITYADMIN attribute of a logical file to TRUE to restrict access to only those processes marked with the USERDATA granulated privilege. The USERDATA privilege is automatically given to users and processes marked with SECADMIN privilege, if security administrator status is authorized, and to privileged users otherwise.

This attribute can be set only by processes marked with the USERDATA privilege.

When the value of SECURITYADMIN is TRUE, the following conditions apply:

- A process must have the USERDATA privilege in order to open the file; other processes receive a security violation.
- A user must have the USERDATA privilege in order to copy the file with library maintenance.
- Read and write access to the file is additionally controlled by the SECURITYTYPE, SECURITYUSE, and SECURITYMODE attributes.
- The ability to remove the file is unaffected.

The SECURITYADMIN attribute is automatically assigned to the SYSTEM/USERDATAFILE when the file is first opened by the USERDATA function.

The value of SECURITYADMIN can be changed with the WFL *ALTER* statement. Refer to the *Work Flow Language (WFL) Programming Reference Manual* for details.

SECURITYGUARD

Kinds	Disk, CD, and tape Printer (if backup)
Interrogate	Anytime
Modify	Anytime (disk and CD) When closed (tape)
Type	Character string
Range	Not applicable
Default	Null string
Abstract Value	DEFAULT
Stored Permanently	Disk or tape
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

If the SECURITYTYPE attribute value is GUARDED or CONTROLLED, use the SECURITYGUARD attribute to specify or determine the title of the guard file. Refer to the discussion of the GUARDFILE utility program in the *Security Features Operations and Programming Guide*.

When a permanent file with a SECURITYTYPE value of GUARDED is opened by a nonprivileged user other than the owner of the file, or when a file with the value of CONTROLLED is opened by a nonprivileged task, the usercode information of the guard file is used to determine if access to the file is permitted.

The format of the SECURITYGUARD attribute is the same as that for an external file title. For more information, refer to the description of the TITLE attribute. The SECURITYGUARD information is a property of the physical or permanent file and is associated with the physical file when it is created.

If the SECURITYGUARD attribute is not specified for the permanent file, or if the specified guard file is not on the specified disk, a SECURITYTYPE attribute value of PRIVATE is used instead of GUARDED or CONTROLLED.

Various actions are taken when the specified disk that contains the guard file is not present. The following table identifies the different situations and identifies the corresponding action:

Situation	Action Taken When The Specified Disk Is Not Present
The permanent file being opened has a KIND file attribute value of DISK or PACK, or an explicit family name in the file specification is present.	A "REQUIRES PK" message is displayed.
The family for the permanent file is unspecified, or no KIND attribute value is specified.	The OPEN operation fails with a "SECURITY VIOLATION" message.
The opening program has the privileges of a message control system (MCS).	The OPEN operation fails with a "FILE NOT AVAILABLE" message.

When the logical file is assigned to a physical file and the file is a CD-ROM file, and the SECURITYGUARD attribute is interrogated, a period (.) is always returned. For all other types of files when the logical file is assigned to a physical file and the SECURITYGUARD attribute is interrogated, the SECURITYGUARD attribute returns the title of the guard file of the physical file in each of the following cases:

- When the user is privileged
- When the user is the owner of the physical file
- When the program accessing the physical file is privileged

In all other cases, when the SECURITYGUARD attribute is interrogated, a null string is returned. In addition, if neither the user nor the program is privileged and the file is owned by some other user, a security violation is logged.

If a physical file has not been assigned to the logical file and a guard file has been specified by the SECURITYGUARD attribute for the logical file, the name of the specified guard file is returned.

The GUARDFILE attribute of a physical file can be changed only by the owner of the physical file or by a privileged user.

Disk File Security

The SECURITYGUARD value of a file can be changed before or after the file is opened. If the SECURITYGUARD attribute is set before opening a file, and a new disk file is created, the SECURITYGUARD value is propagated to the file that is created. Setting SECURITYGUARD before opening an existing disk file has no effect on the file that is opened.

Setting the SECURITYGUARD attribute while the logical file is assigned to a disk file causes the MCP to attempt to change the security of the physical file.

Tape File Security

The SECURITYGUARD attribute can be used for tape files if the Security Accountability Facility is installed, and if the TAPECHECK option of the SECOPT (Security Options) system command is set to AUTOMATIC. SECURITYGUARD is not used for tape files if the TAPECHECK option has a value of NONE.

When a tape file is accessed through Heritage Host Services logical I/O, the TAPECHECK option value on the other host controls whether the SECURITYGUARD attribute can be used for the tape. Refer to the discussion of performing I/O operations using Heritage Host Services logical I/O in the *I/O Subsystem Programming Guide*.

The SECURITYGUARD attribute is set when a tape volume family is created. On multifile tapes, the guard file used for the volume family is the one specified by the SECURITYGUARD attribute for the first file. On multireel files, the SECURITYGUARD attribute is established by the first file on the first reel of the tape volume family. The SECURITYGUARD attribute cannot be changed programmatically after the first file on the first reel is created.

If the SECURITYLABELS tape volume security option is TRUE, the value of the SECURITYGUARD attribute is stored in the file label and ANSI87 tape labels are created. Refer to the *Work Flow Language (WFL) Programming Reference Manual* for information about the SECURITYLABELS tape volume security option.

Heritage Host Services Logical I/O Security

For Heritage Host Services logical I/O, guard files control access only by usercode; that is, the access specification in the guard file must be in the following form:

```
USERCODE X;
```

If the access specification is in either of the following forms, the specified program cannot access the file:

```
PROGRAM Y;
USERCODE X USING PROGRAM Y;
```

SECURITYMODE

Kinds	Disk or tape
Interrogate	Anytime
Modify	Anytime (disk) When assigned (tape)
Type	Word
Range	See following text
Default	See following text
Stored Permanently	Disk or tape
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Set or interrogate the SECURITYMODE attribute to specify or determine the security mode of a file or tape.

The layout of the attribute is a 16-bit bit mask that matches the layout of the disk file header SECURITYMODE field. Not all bits in the mask are defined and user-specified values for undefined bits are ignored; however, be aware that setting undefined bits might result in future migration problems.

Note: *Prior to the 44.2 release, any attempt to set the SECURITYMODE attribute while the ENABLEPOSIX system option is RESET results in a READONLY attribute value. It is no longer necessary to set or reset the ENABLEPOSIX option to control any functionality. The ENABLEPOSIX option will be deimplemented in a future release.*

If you have not already read the section “Securing Files Using the POSIX Security Model” in the I/O Subsystem Programming Guide, do so before attempting to use the SECURITYMODE attribute.

The ALTERNATEGROUPS attribute can be used in conjunction with the SECURITYMODE attribute to specify specific permissions for multiple groups of users.

SECURITYMODE Subattributes

Several subattributes can be used to set and interrogate portions of the SECURITYMODE mask. Setting a subattribute only affects the field in the SECURITYMODE value in which the specified attribute is stored; others remain unchanged. Multiple attributes are applied cumulatively in the order in which they are specified.

The following table lists the subattributes used to modify the SECURITYMODE attribute value. Although you can modify the value of SECURITYMODE directly by applying an operand with the appropriate bits set, direct modification of the attribute is not recommended.

Name	Type	Field	Description
OWNERRWX	Mnemonic	[8:3]	The read, write, and execute permissions that are granted to the owner of the file.
OWNERR	Boolean	[8:1]	If TRUE, the owner is granted read access to the file.
OWNERW	Boolean	[7:1]	If TRUE, the owner is granted write access to the file.
OWNERX	Boolean	[6:1]	If TRUE, the owner is granted execute access to the file.
GROUPRWX	Mnemonic	[5:3]	The read, write, and execute permissions that are granted to the group associated with the file.
GROUPR	Boolean	[5:1]	If TRUE, the members of the group are granted read access to the file.
GROUPW	Boolean	[4:1]	If TRUE, the members of the group are granted write access to the file.
GROUPX	Boolean	[3:1]	If TRUE, the members of the group are granted execute access to the file.
OTHERRWX	Mnemonic	[2:3]	The read, write and execute permissions that are granted to all other users of the system.
OTHERR	Boolean	[2:1]	If TRUE, other users are granted read access to the file.
OTHERW	Boolean	[1:1]	If TRUE, the members of the group are granted write access to the file.
OTHERX	Boolean	[0:1]	If TRUE, the other users are granted execute access to the file.
SETGROUPCODE	Boolean	[10:1]	If TRUE for a code file, the code file executes with a GROUPCODE of the group of the file, not the USERCODE of who executed it.
SETUSERCODE	Boolean	[11:1]	If TRUE for a code file, the code file executes with a USERCODE of the owner of the file, not the USERCODE of who executed it.
USEGUARDFILE	Boolean	[12:1]	If TRUE, a guard file also defines access privileges to the file.
GUARDOWNER	Boolean	[13:1]	If TRUE, when USEGUARDFILE is TRUE, the guard file also applies to the owner of the file.

***Note:** The SETUSERCODE and SETGROUPCODE flags can cause programs to behave unexpectedly because they execute under a USERCODE and/or GROUPCODE that might differ from the USERCODE and/or GROUPCODE of the initiator. Refer to the* Security Features Operations Programming Guide *for more information about the use of these flags.*

The following table lists the mnemonics used for the mnemonic subattributes.

Mnemonic	Security Mode	Value	Mask
RWX	Read, Write, Execute	7	1'111'
RW	Read, Write	6	1'110'
RX	Read, Execute	5	1'101'
R	Read	4	1'100'
WX	Write, Execute	3	1'011'
W	Write	2	1'010'
X	Execute	1	1'001'
NO	None	0	1'000'

A value of 1 for a bit grants permission. For example, an OWNERRWX value of 1'100' allows the owner of the file to read it but not to write to it or execute it (if it is a code file).

SECURITYMODE Defaults

The default SECURITYMODE value for a file created by a usercoded task is PRIVATE, that is OWNERRWX = RWX, OTHERRWX = NO, GROUPRWX = NO, and 0 for all other SECURITYMODE fields.

The default SECURITYMODE value for a file created by an unusercoded task is dependent upon the setting of the system NONUSERFILES security option (SECOPT). If the NONUSERFILES setting is PRIVATE, the default value of SECURITYMODE for a file created by an unusercode task is PRIVATE, that is OWNERRWX = RWX, OTHERRWX = NO, GROUPRWX = NO, and 0 for all other SECURITYMODE attributes. If the NONUSERFILES setting is PUBLIC, the default value of SECURITYMODE for a file created by a unusercoded task is PUBLIC, that is OWNERRWX = RWX, OTHERRWX = RWX, GROUPRWX = RWX, and 0 for all other SECURITYMODE fields.

The default SECURITYMODE value for a new permanent directory (file with FILEKIND = PERMDIR) is OWNERRWX = RWX, OTHERRWX = X, GROUPRWX = X, and 0 for all other SECURITYMODE fields.

The SECURITYMODE value (and other file security attributes) of a new file or directory residing within a permanent directory may be inherited from the parent directory instead of being set to the default if the PROPAGATESECURITYTOFILES or PROPAGATESECURITYTODIRS file attributes of the containing permanent directory is set to PROPAGATE.

Interrogating the SECURITYMODE Attribute

Interrogating the SECURITYMODE attribute of a file returns one of the following values:

- If the SECURITYMODE attribute or any of its related attributes is interrogated while the logical file is assigned to a physical file, the appropriate field from the security mode value of the physical file is returned.
- If it is interrogated while the logical file is unassigned, the appropriate field from any user specified or default security mode value is returned. For example, assuming the following file declaration:

  ```
  FILE F(KIND = DISK, OTHERRWX = RW)
  ```

  Interrogating F.OTHERR would return TRUE.

SECURITYMODE Interdependencies

If you set the SECURITYTYPE or SECURITYUSE attributes, the value of SECURITYMODE might change.

The following table shows how the SECURITYTYPE and SECURITYUSE attributes are related to the SECURITYMODE attribute. Any related SECURITYMODE fields not explicitly listed in the table are 0 (FALSE).

SECURITYTYPE Setting	SECURITYUSE Setting	SECURITYMODE Settings

PUBLIC	IN	OWNERRWX = RWX
		GROUPRWX = RX
		OTHERRWX = RX

	OUT	OWNERRWX = RWX
		GROUPRWX = WX
		OTHERRWX = WX

	IO	OWNERRWX = RWX
		GROUPRWX = RWX
		OTHERRWX = RWX

	SECURED	OWNERRWX = RWX
		GROUPRWX = X
		OTHERRWX = X

PRIVATE	Not applicable	OWNERRWX = RWX

GUARDED	Not applicable	OWNERRWX = RWX
		GROUPRWX = RWX
		OTHERRWX = RWX
		USEGUARDFILE = TRUE

SECURITYTYPE Setting	SECURITYUSE Setting	SECURITYMODE Settings

CONTROLLED	Not applicable	OWNERRWX = RWX
		GROUPRWX = RWX
		OTHERRWX = RWX
		USEGUARDFILE = TRUE GUARDOWNER = TRUE

If you change the SECURITYMODE attribute, the SECURITYTYPE or SECURITYUSE attributes, or both, might change. The value returned when interrogating the SECURITYTYPE attributes is determined by the SECURITYMODE attribute as follows:

```
if F.USEGUARDFILE is TRUE then
  if F.GUARDOWNER is TRUE then
     F.SECURITYTYPE is VALUE (CONTROLLED)
  else
     F.SECURITYTYPE is VALUE (GUARDED)
else
    if F.GROUPRWX is VALUE (NO) and
      F.OTHERRWX is VALUE (NO) then
      F.SECURITYTYPE is VALUE (PRIVATE)
    else
       F.SECURITYTYPE is VALUE (PUBLIC)
```

Avoid mixing the use of the SECURITYTYPE and SECURITYUSE attributes with the SECURITYMODE attribute, because unexpected behavior can result. Certain values of SECURITYMODE do not map exactly back to SECURITYTYPE. For example, the SECURITYMODE value of a file might have GROUPR and GROUPW set (granting read and write permissions to the group), but no permissions set in OTHERRWX. Interrogating the SECURITYTYPE attribute for this file would return a value of PUBLIC even though the file is not accessible to all users of the system. In addition, the value of SECURITYUSE that is returned would be IO even though execute permission is not granted.

Setting the SECURITYMODE or ALTERNATEGROUPS Attribute

The restrictions specified in this subsection for setting the SECURITYMODE attribute also apply to setting the ALTERNATEGROUPS attribute.

When a logical file is assigned to a physical file, if you set SECURITYMODE or any of its subattributes, the MCP attempts to change the security of the physical file; but only the owner of the file or a privileged user can change the security of an existing file. However, if both the useguardfile and the guardowner subattributes are set in the SECURITYMODE attribute of the physical file, the owner (or the program being used) can only change the security if write access to the file is attainable through the guard file of the physical file.

When a new disk file is created as part of the file open process, any user specified SECURITYMODE value is propagated to the physical file.

When an existing disk file is opened, any previously specified SECURITYMODE value of the logical file has no effect on the SECURITYMODE attribute of the physical file. In other words, the SECURITYMODE attribute is not propagated from the logical file to an existing physical file during the file open process.

For example, to set the OTHERRWX field in a newly created file, use the following declaration:

```
      FILE F(KIND = DISK, NEWFILE=TRUE, SECURITYMODE = 0, OTHERRWX = RW)
```

If the USEGUARDFILE subattribute permission is used (for example, for GUARDED or CONTROLLED files), the file access permissions granted by the guard file are combined with the permissions granted by the SECURITYMODE value to guarantee that the permissions granted by the guard file are never more liberal than that which is specified by the SECURITYMODE value. For example, if the guard file grants read permission to a process but the SECURITYMODE attribute does not, the process does not grant read access to the file.

If the guard file is to completely control access to the file, all of the file permission flags within the SECURITYMODE need to be set. That is, set OWNERRWX, GROUPRWX, and OTHERRWX to RWX. In addition, the ALTERNATEGROUPS attribute should not be used in this case.

Whenever the SECURITYMODE attribute (or any of its subattributes) of an existing disk file is modified, the timestamp attribute of the file is updated.

Tape Information

The SECURITYMODE attribute can only be used for tape files if the system SECOPT TAPECHECK security option has the value AUTOMATIC. The SECURITYMODE value is assigned when a tape volume family is created and cannot subsequently be changed programmatically.

Using the FILEMASK task attribute

The FILEMASK task attribute can be used to further restrict the access privileges defined in the SECURITYMODE attribute for newly created disk files. Refer to the *Task Attribute Programming Reference Manual* for more information.

Interdependencies

Refer to the *Security Features Operations and Programming Guide* for more information about the use of the SECURITYMODE attribute.

Exception Handling

If the SECURITYMODE attribute is set while the file is unassigned and that file is subsequently used to open a PORT file, an error is generated.

Additional Information

The ALTERNATEGROUPS attribute can be used to assign additional GROUPS to the file and file access permissions for those GROUPS. The permissions specified by the ALTERNATEGROUPS attribute and the GROUPRWX subattribute of SECURITYMODE are additive. Refer to the ALTERNATEGROUPS attribute for additional information.

The typical usage of the SECURITYMODE, GROUP, and ALTERNATEGROUPS attributes is to give the most access rights to the file owner, perhaps less access rights to users in the GROUP or ALTERNATEGROUPS, and least access rights to other users (using OTHERRWX). However, the GROUP and ALTERNATEGROUPS attributes can also be used to block access to files for specific groups of users by giving those groups fewer access rights than are given to other users.

SECURITYTYPE

Kinds	Disk, CD, port, or tape Printer (if backup)
Interrogate	Anytime
Modify	Anytime (disk or CD) When closed (port or tape)
Type	Mnemonic
Range	See following text
Default	See following text
Abstract Value	DEFAULT
Stored Permanently	Disk, CD, or tape
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Restricted values
Port Services	BNANATIVESERVICE, NETBIOSSESSIONSERVICE, OSINATIVESERVICE, OSISESSIONSERVICE, TCPNATIVESERVICE, BASICSERVICE

Use the SECURITYTYPE attribute to determine or specify the class of users that can access the file. All privileged users have access to all files regardless of the SECURITYTYPE value. The mnemonic and integer values and meanings associated with SECURITYTYPE for nonprivileged users are as follows:

Mnemonic Value	Integer Value	Meaning
CONTROLLED	3	Access by all users including the owner is controlled by a guard file specified by the SECURITYGUARD attribute. This value is invalid for port files.
GUARDED	2	Access by nonowner users is controlled by a guard file. This value is invalid for port files.
PRIVATE	0	The owner can access the file.
PUBLIC	1	Access for a nonowner is controlled by the SECURITYUSE attribute.

CLASSA and CLASSB can be used in place of PUBLIC and GUARDED, but PUBLIC and GUARDED are the preferred mnemonics.

If SECURITYTYPE is interrogated when the file is assigned, the value for the physical file is returned.

The default value of SECURITYTYPE is PRIVATE if the task that owns the file is running with a usercode. If the Security Accountability Facility is installed, physical files created and port files opened by tasks running without a usercode default to the value set in the NONUSERFILES option of the SECOPT (Security Options) system command; otherwise, they default to PUBLIC. The value of NONUSERFILES can be either PUBLIC or PRIVATE.

Disk File Security

The SECURITYTYPE value of the file can be changed before or after the file is opened. If the SECURITYTYPE attribute is set before opening a file and a new disk file is created, the SECURITYTYPE value is propagated to the file that is created. Setting SECURITYTYPE before opening an existing disk file has no effect on the file that is opened.

Setting the SECURITYTYPE attribute while the logical file is assigned to a disk file causes the MCP to attempt to change the security of the physical file.

Port File Security

The only valid values of SECURITYTYPE for port files are PRIVATE and PUBLIC. If the value of SECURITYTYPE is PRIVATE, the value of the YOURUSERCODE attribute for the port subfile must match the usercode of the task of the correspondent endpoint. Refer to the description of the YOURUSERCODE attribute.

When the SECURITYTYPE value is PUBLIC, usercodes are not used when matching the dialogues. OSINATIVESERVICE, OSISESSIONSERVICE, TCPNATIVESERVICE, and BASICSERVICE port files must have the SECURITYTYPE value equal to PUBLIC. The fixed default setting of SECURITYTYPE can be changed by the NW NS SET MIGRATETOBASICSERVICE system command.

SECURITYTYPE is valid only for TCPNATIVESERVICE.

For information about opening port files, refer to the *I/O Subsystem Programming Guide*.

Tape File Security

The SECURITYTYPE attribute can be used for tape files if the Security Accountability Facility is installed, and if the TAPECHECK option of the SECOPT (Security Options) system command has a value of AUTOMATIC. SECURITYTYPE is not used for tape files if the TAPECHECK option value is NONE.

The SECURITYTYPE value is assigned when a tape volume family is created. On multifile tapes, the class of users that can access the file is the value specified in the SECURITYTYPE attribute for the first file. On multireel files, the SECURITYTYPE attribute value is established by the first file on the first reel of the tape volume family. The SECURITYTYPE attribute cannot be changed programmatically after the first file on the first reel is created.

When you access a tape file through Heritage Host Services logical I/O, the TAPECHECK option value at the host where the tape volume is located controls the SECURITYTYPE value the tape file can have. Refer to the discussion of performing I/O operations using Heritage Host Services logical I/O in the *I/O Subsystem Programming Guide*.

If the SECURITYLABELS tape volume security option is TRUE, the value of the SECURITYTYPE attribute is stored in the file label and ANSI87 tape labels are created.

Refer to the *Work Flow Language (WFL) Programming Reference Manual* for information about the SECURITYLABELS tape volume security option.

Heritage Host Services Logical I/O Security

GUARDED, PRIVATE, and PUBLIC are supported by Heritage Host Services logical I/O; CONTROLLED is not supported by Heritage Host Services logical I/O.

Interdependencies

Mixing the use of the SECURITYTYPE and SECURITYUSE attributes with the SECURITYMODE or ALTERNATEGROUPS attribute can produce unexpected results and should be avoided. See the SECURITYMODE option in this manual for more information.

SECURITYUSE

Kinds	Disk, CD, or tape Printer (if backup)
Interrogate	Anytime
Modify	Anytime (disk) Never (CD) When closed (tape)
Type	Mnemonic
Range	See following text
Default	IO
Abstract Value	DEFAULT (disk and tape)
Stored Permanently	Disk or tape
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Use the SECURITYUSE attribute to determine or specify how a physical file that is protected by security can be accessed by nonprivileged users using nonprivileged programs.

If a nonprivileged task owns a file, the task usercode matches the file usercode, and the SECURITYTYPE value is CONTROLLED, the information about who can use the file is retrieved from the guard file. If SECURITYTYPE has any other value, file access is not restricted.

If a nonprivileged task does not own a file and the SECURITYTYPE value is PUBLIC, the SECURITYUSE attribute is interrogated. If the SECURITYTYPE value is GUARDED or CONTROLLED, the guard file is interrogated.

The SECURITYUSE attribute is not interrogated when a usercode or program is privileged.

The mnemonic and integer values associated with SECURITYUSE follow. Their mnemonic meanings are defined in the following text:

Mnemonic Value	Integer Value
IN	1
IO	3
OUT	2
SECURED	0

Disk File Security

The SECURITYUSE attribute value can be changed before or after the file is opened. If the SECURITYUSE attribute is set prior to opening a file and a new disk file is created, the SECURITYUSE value is propagated to the file that is created. Setting SECURITYUSE prior to opening an existing disk file has no effect on the file that is opened.

Setting the SECURITYUSE attribute while the logical file is assigned to a disk file causes the MCP to attempt to change the security of the physical file.

The IN, IO, OUT, and SECURED values for disk security are described as follows:

Value	Meaning
IN	Specifies read-only access to source files, data files, and code files. Also, code files can be executed.
IO	Specifies read and write access to source files, data files, and code files. Also, code files can be executed. The default value of SECURITYUSE is IO for all disk files.
OUT	Specifies write-only access to source files, data files, and code files. Also, code files can be executed.
SECURED	Specifies that nonprivileged users do not have access to source or data files, but code files can be executed. For example, a nonprivileged user can execute a secured code file but cannot copy it.

If the SECURITYUSE attribute is interrogated when the file is assigned, the value of the physical file is returned.

CD-ROM File Security

When an existing CD-ROM file is opened for input, the SECURITYUSE value returns IN.

Tape File Security

The SECURITYUSE attribute can be used for tape files if the Security Accountability Facility is installed, and the TAPECHECK option value of the SECOPT (Security Options) system command is AUTOMATIC. If the TAPECHECK option value is NONE, the SECURITYUSE attribute cannot be used.

The IN, IO, and OUT values for tape volume security are described as follows:

Value	Meaning
IN	Specifies that the process can read all files on a tape volume, but cannot write, rewrite, or purge any file on a tape volume.
IO	Specifies that the process can do the following: Read all files on the tape volume. Purge all files on the tape volume, after each file is opened—for example, with an ALGOL *CLOSE (<file name>, PURGE)* statement. Rewrite any file on the tape volume in one of the following ways: Change the MYUSE value to IO, open the file, and write to the file. Open the tape file for input with the NEWFILE attribute value set to FALSE and the MYUSE attribute set to IN, close the tape with retention by using a *CLOSE (<file name>, *)* or *CLOSE (<file name>, REWIND)* statement, change the MYUSE value to OUT, and reopen the file for writing. Write new files on the tape volume with NEWFILE set to TRUE and MYUSE set to OUT if one of the following is true: The process usercode is the owner of the tape volume. The tape is temporarily owned because the PERMANENTLYOWNED tape volume attribute is FALSE.
OUT	Specifies that a process has the following characteristics: Cannot read or purge any file on the tape volume. Can rewrite files on the tape with NEWFILE set to FALSE and MYUSE set to OUT. Can write new files on the tape volume with NEWFILE set to TRUE and MYUSE set to OUT if one of the following is true: The process usercode is the owner of the tape volume because the FAMILYOWNER attribute value is equal to the task usercode. The tape is temporarily owned because the PERMANENTLYOWNED tape volume attribute is FALSE.
SECURED	The SECURED value is not applicable to tape files.

The SECURITYUSE attribute is set when a tape volume family is created. On multifile tapes, the SECURITYUSE attribute used for the volume family is the one specified by the SECURITYUSE attribute for the first file. On multireel files, the SECURITYUSE attribute is established by the first file on the first reel of the tape volume family. The SECURITYUSE attribute cannot be changed programmatically after the first file on the first reel is created.

When a tape file is accessed through Heritage Host Services logical I/O, the TAPECHECK option setting at the host where the tape volume is located controls the use of the SECURITYUSE attribute, if specified, on the tape. Refer to the discussion of performing

I/O operations using Heritage Host Services logical I/O in the *I/O Subsystem Programming Guide*.

If the SECURITYLABELS tape volume security option is TRUE, the value of the SECURITYUSE attribute is stored in the file label and ANSI87 tape labels are created. Refer to the *Work Flow Language (WFL) Programming Reference Manual* for information about the SECURITYLABELS tape volume security option.

Interdependencies

Mixing the use of the SECURITYTYPE and SECURITYUSE attributes with the SECURITYMODE or ALTERNATEGROUPS attribute can produce unexpected results and should be avoided. See the SECURITYMODE option in this manual for more information.

SENSITIVEDATA

Kinds	Disk or CD
Interrogate	Anytime
Modify	When closed
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Abstract Value	DEFAULT
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Use the SENSITIVEDATA attribute to determine or specify if the disk or pack areas assigned to the file are to be overwritten with an arbitrary pattern before the disk space is returned to the system for reallocation. This action prevents the data in the file from being seen by a program that reuses the disk space for another file. If you have the Security Accountability Facility installed, this task can be accomplished more efficiently by using security disk scrubbing. When they are created, files with the SENSITIVEDATA attribute set to TRUE are entered into the directory as if the PROTECTION attribute were equal to SAVE. Refer to the description of the PROTECTION attribute.

SERIALNO

Kinds	Disk or tape
Interrogate	Anytime
Modify	When unassigned
Type	Word
Range	See following text
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	Disk or tape
Parameters	1 optional
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Restricted usage
Port Services	Not applicable

Use the SERIALNO attribute to determine or specify the serial number of the labeled tape or base member of the disk family to which the logical file is assigned. The serial number of a tape is established at the ODT through the SN (Serial Number) system command. The RC (Reconfigure Disk) or LB (Relabel Pack) system command can be used to establish a serial number for a disk peripheral. Refer to the *System Operations Guide* for information about using the SN and RC command. Refer to the *System Commands Reference Manual* for a description of the LB command.

A serial number is one word of data consisting of an alphanumeric string of up to six EBCDIC characters left-justified in a field of blanks, or an integer in the range of 1 through 999999.

The SERIALNO attribute can be modified only when the file is closed and unassigned.

The optional FILESECTION number parameter can be used to create a list of serial numbers of the members of a multivolume tape volume family. For example, the following ALGOL statements would specify the serial numbers of the volumes of a three-member tape volume family:

```
F(1).SERIALNO := "FIRST ";
F(2).SERIALNO := "SECOND";
F(3).SERIALNO := "THIRD ";
```

Note: *If the parameter is not specified in a program, the serial number for FILESECTION 1 is modified. However, if the FA (File Attributes) system command is used to respond to a waiting message, the file section whose serial number is modified is the file section number that appears in the waiting message.*

You can also specify serial number lists by using a file equation. For example, the preceding tape volume family can be defined with the following file equation:

```
FILE F(SERIALNO=("FIRST ","SECOND","THIRD "));
```

A list can be specified with missing elements; this list overrides any program-declared SERIALNO value. The following file equation has a missing element:

```
FILE F(SERIALNO=("FIRST ","SECOND",,"FOURTH"));
```

The system option SERIALNUMBER (option 27) controls the assignment of scratch tapes when the SERIALNO and SCRATCHPOOL file attribute values are not specified. The assignment of scratch tapes is controlled in the following ways:

- If the option is set, no file assignment automatically takes place and the "<file name> REQUIRES" message is displayed. The operator can respond with a DS (Discontinue) or OU (Output Unit) system command, or can specify a serial number or scratch pool name with the FA (File Attribute) system command.
- If the option is reset, the system uses any available scratch tape that does not have a scratch pool name.

For further information about using the OP (Options), DS, OU, and FA system commands, refer to the *System Operations Guide*.

If the SERIALNO attribute is specified when a tape file is created, the file is assigned when the serial numbers match, even if the tape is considered a scratch tape. The tape must have a write ring and cannot be locked, saved, assigned, or not ready. If the tape is not a scratch tape, the tape is rewound so that new labels can be written at the beginning of the volume, in effect purging the tape. If you use a compiler other than WFL and assign null characters to the SERIALNO value, where all bits are equal to 0 (zero), the compiler does not consider the attribute during file assignment.

The SERIALNO attribute can be modified while the logical file is assigned if the element in the serial number list being modified is not the FILESECTION currently in use.

The SERIALNO attribute can be interrogated at any time. When the logical file is open or closed with retention—that is, when the logical file is assigned to a physical file—and you are using a disk file, the SERIALNO of the base pack is returned. When the physical file is not assigned to the logical file, the first serial number in the SERIALNO list is returned. You can use the optional index to interrogate a particular serial number in the SERIALNO list.

If the logical file is assigned to a nonlabeled tape file and is positioned beyond the last file of a multifile tape, or is unassigned and the SERIALNO attribute has not been set for the particular element of the list, an attribute error is issued.

When a new nonlabeled tape file is created, interrogating the SERIALNO attribute returns the serial number of the tape that was assigned for output.

When assigned a value, the SERIALNO attribute is used in file assignment. When the I/O subsystem searches for a permanent disk file or selects a mass-storage family on which to create a disk file, it uses the first element in a serial number list, if a value is assigned, to identify the base unit of the family. Because the FAMILYNAME attribute is sufficient to identify a disk family, specifying a SERIALNO value is not necessary for disk files.

Specifying the SERIALNO attribute for a particular tape volume indicates that checking for serial numbers is requested when the file is assigned. For permanent tape files, the serial number of the physical tape must match the value of the SERIALNO attribute.

If a permanent tape file is found that meets all requirements except for the serial number, a “NO FILE” message is displayed. The operator can respond by making a tape with the matching serial number available or by using the IL (Ignore Label), OF (Optional File), FA (File Attribute), or DS system commands. Refer to the *System Operations Guide* for information about using these commands.

When Heritage Host Services logical I/O is used, if the SERIALNO attribute is specified before the file is opened, serial numbers for volumes other than the first volume are ignored.

When serial numbers do not match, the RESIDENT and PRESENT attributes return FALSE and the SERIALNOMISMATCHRSLT (6) open error is returned.

The SERIALNO attribute can be assigned a value by means of a program statement or a file declaration; the value can be one of the following:

- An integer value from 1 through 999999
- One to six EBCDIC, uppercase alphanumeric characters, left-justified with blank fill
- A 0 (zero), the null value for SERIALNO

The following ALGOL and WFL texts specify the same SERIALNO values:

ALGOL	WFL	SERIALNO
F.SERIALNO := 123	F(SERIALNO = 123)	[000123]
F.SERIALNO := "AB "	F(SERIALNO = "AB")	[AB]
F.SERIALNO := "AB"	F(SERIALNO = 49602)	[049602]
F.SERIALNO := "000000"	F(SERIALNO = 0) or F(SERIALNO = "000000")	[000000]
F.SERIALNO := 0	F(SERIALNO="")	Not specified

The third line illustrates a mistake that is easy to make because ALGOL short strings are right-justified with NULL fill and thus have integer values. "AB" has a hexadecimal value of 00000000C1C2, which has a decimal value of 49602.

SERVICE

Kinds	Port and disk
Interrogate	Anytime
Modify	When closed
Type	Mnemonic
Range	See following text
Default	See following table
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Supported
Heritage Host Services Logical I/O	Supported
Port Services	All

The SERVICE attribute has two functionalities: one for port files and one for File Transfer, Access, and Management (FTAM) and Heritage Host Services logical I/O files.

Port Files

Use the SERVICE attribute to specify the interprocess communication service type for a port file. The default value of this attribute can be modified through the NW NS SET MIGRATETOBASICSERVICE system command.

The value of SERVICE for a port file determines its behavior and functionality. The service of a port file is dependent upon the underlying network environment or environments that are to be used to support the subfile dialogues of the file. There is one SERVICE value associated with a subfile dialogue. The service and network environment used by the dialogue cannot change during the life of the dialogue.

The following values for SERVICE are available:

Mnemonic Value	Integer Value	Description
BASICSERVICE	5	Offers a port service that provides basic dialogue establishment and termination, and read and write capabilities.
BNANATIVESERVICE	0	Offers a port service that is native to a BNA network environment.
NETBIOSSESSIONSERVICE	9	Offers a message-oriented port service that enables enterprise server applications to communicate with PC applications on Novell NetWare local area networks (LANs).
OSINATIVESERVICE	3	Offers a port service that is native to an OSI network environment.

Mnemonic Value	Integer Value	Description
OSISESSIONSERVICE	4	Offers a subset of the service available on an OSI network environment up to the OSI session layer only. This subset is used for communication with OSI implementations that support up to the session layer only.
TCPIPNATIVESERVICE	12	Offers a port service that is native to a Transmission Control Protocol (TCP) network environment. This service offers byte-stream data transfer. TCPIPNATIVESERVICE is the native service of TCPIP ***Note:*** *This service will replace TCPNATIVESERVICE in a future release.*
TCPNATIVESERVICE	2	Offers a port service that is native to a TCP network environment. This service is not message-oriented; it offers byte-stream data transfer. TCPNATIVESERVICE is the native service of TCPIP. ***Note:*** *This service will be deimplemented in a future release.*

Refer to the *I/O Subsystem Programming Guide* for a complete definition of each port service.

To specify a particular system software or group of software that is to provide the service of a port subfile, use the PROVIDERGROUP subfile attribute. By default, the software provider of a subfile is automatically selected by the system for the program. Refer to the PROVIDERGROUP attribute for a description.

The following table illustrates the selection of SERVICE if the application does not specify the SERVICE attribute:

Provider Group	**No. of providers in Provider Group**	**Selected Service**
Not specified	Not applicable	BNANATIVESERVICE or BASICSERVICE, depending on the value of VICE.
Specified	One	Native service of the provider in the Provider Group. (Native service is TCPIPNATIVESERVICE).
Specified	More than one	If all providers in the Provider Group have the same native service, that service is selected. Otherwise, a service common to all providers in the Provider Group is selected. However, if providers in the Provider Group don't have a common service, then the service defaults to BNANATIVESERVICE or BASICSERVICE, depending on MIGRATETOBASICSERVICE.

FTAM and Heritage Host Services Logical I/O Files

Use the SERVICE attribute to specify whether Heritage Host Services logical I/O or FTAM should be used to access a file on another host. If SERVICE is not assigned a value, the operating system selects the service based on which service has been purchased and if the service is currently available to the remote host.

The following values for SERVICE are available:

Mnemonic Value	**Integer Value**	**Description**
FTAM	8	Specifies that FTAM should be used to access the file.
HOSTSERVICES	7	Specifies that Heritage Host Services logical I/O should be used to access the file.

SINGLEPACK

SINGLEPACK is a nonpreferred synonym for SINGLEUNIT. Refer to the description of the SINGLEUNIT attribute.

SINGLEUNIT

Kinds	Disk
Interrogate	Anytime
Modify	When closed
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Abstract Value	DEFAULT
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Set the SINGLEUNIT attribute to TRUE to indicate that areas for the disk file are to be allocated on a single family member selected by the system. The default value for SINGLEUNIT is FALSE, which provides for the distribution of areas over the entire family. If additional members are added to the family subsequent to the opening of the logical file, they are inserted into the rotational scheme of allocation.

When space is unavailable on the original family member, the system displays the following message:

```
PK uu (FAMILYINDEX nnn) sss SECTORS REQUIRED; OK TO IGNORE FAMILYINDEX
```

An operator can perform one of three tasks after receiving the preceding message:

- Discontinue the program by entering the DS (Discontinue) system command.
- Allow the file to overflow to another unit in the family by entering the OK (Reactivate) system command.
- Take some action to free up space on that family member. Possible actions include the following:
  - If the required pack is online and write-enabled, issue the system command SQUASH. Refer to the *System Commands Manual* for details.
  - If the required pack is online and write-enabled, remove some files.
  - If the required pack is online and write-enabled, issue the WFL statement ARCHIVE ROLLOUT. Refer to the *System Operations Guide* for details.

If the OK command is used, the following message is displayed to indicate where the extra space was found:

```
PKnn SECTORS ALLOCATED ON FAMILYINDEX nnn ON <family name>
```

SIZEMODE

Kinds	All except port
Interrogate	Anytime
Modify	When closed
Type	Mnemonic
Range	See following text
Default	SINGLE
Abstract Value	DEFAULT
Stored Permanently	Disk or tape
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

If a file has a BLOCKSTRUCTURE value of VARIABLEOFFSET, use the SIZEMODE attribute to determine or specify the recording mode of the record length information contained in the field defined by the SIZEOFFSET and SIZE2 attributes.

The mnemonic values and meanings associated with SIZEMODE are as follows:

Mnemonic Value	Integer Value	Meaning
ASCII	5	8-bit ASCII numeric characters
EBCDIC	4	8-bit EBCDIC numeric characters
HEX	2	4-bit, packed decimal
SINGLE	0	Word mode, binary, or 48-bit

Although the SIZEMODE value can be specified for all devices, a value different from the INTMODE value or an INTMODE value of SINGLE is meaningful only on devices that can support a complex record format, such as disks and tapes.

SIZEOFFSET

Kinds	All except port
Interrogate	Anytime
Modify	When closed
Type	Integer
Range	0 through 65535
Default	0
Abstract Value	DEFAULT, MINIMUM, MAXIMUM
Stored Permanently	Disk or tape
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

If the file has a BLOCKSTRUCTURE value of VARIABLEOFFSET, use the SIZEOFFSET attribute to determine or specify the zero-relative offset from the beginning of the record where the record length information is contained. The recording mode of the SIZEOFFSET attribute is specified in SIZEMODE units.

SIZEVISIBLE

Kinds	All except port
Interrogate	Anytime
Modify	When unassigned
Type	Boolean
Range	TRUE, FALSE
Default	TRUE
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	None
FTAM File Access	Supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

If a file has a BLOCKSTRUCTURE value of VARIABLE and you want the operating system to maintain the size field of the record, set the SIZEVISIBLE attribute to FALSE. If the file is a port file and has a BLOCKSTRUCTURE value of VARIABLE, the SIZEVISIBLE attribute must be set to FALSE.

If the SIZEVISIBLE attribute is set to TRUE, the first four characters of each record must contain a decimal number indicating the number of INTMODE-sized characters and size field characters in the record. The number itself is encoded using the INTMODE character set. For example, if a record is 50 characters long and the INTMODE attribute value is EBCDIC, the size field would contain 0050 in EBCDIC characters. If the INTMODE value is SINGLE, the size is stored in the first word of the record in binary. You are responsible for maintaining the size field.

If the SIZEVISIBLE attribute is set to FALSE, the operating system maintains the sizes of the records and your program does not need to be concerned with the size field. When your program invokes a WRITE operation, the size of the record is set to the length of the data written. When your program invokes a READ operation, the size of the record read is returned in the logical result descriptor (LRD). Refer to the description of the STATE attribute for details.

Attempting to write a record whose size cannot be represented in the size field results in a data error result. Any positive integer can be represented when INTMODE is set to SINGLE. For other values of INTMODE, the maximum value that can be represented in the size field is 9999 if SIZEVISIBLE is TRUE, and 9995 if SIZEVISIBLE is FALSE. The minimum value that can be represented is 4 if SIZEVISIBLE is TRUE, and 0 (zero) if SIZEVISIBLE is FALSE.

When all of the following conditions exist, the system is unable to correctly maintain the size of the records and the "SIZEVISIBLE/FRAMESIZE/INTMODE CONFLICT" error message is displayed when an attempt is made to open the file:

- SIZEVISIBLE is FALSE.
- The value of the FRAMESIZE attribute is 48 or the value of UNITS is WORDS.
- The INTMODE value is not SINGLE.

The SIZEVISIBLE value affects the values of the MINRECSIZE, MAXRECSIZE, and BLOCKSIZE attributes. When SIZEVISIBLE is FALSE at file creation time, the MINRECSIZE and MAXRECSIZE values do not include the length of the system-maintained record size field, and are adjusted upward accordingly before being stored in the physical file header. If this adjustment results in the MAXRECSIZE value being greater than the BLOCKSIZE value, the BLOCKSIZE value is also adjusted upward. When you interrogate the MINRECSIZE and MAXRECSIZE attributes, the logical values originally specified before the adjustments are returned. When you interrogate BLOCKSIZE, the actual size of the block being used is returned.

Files can be created with SIZEVISIBLE set to TRUE and then read or updated with SIZEVISIBLE set to FALSE, and vice versa.

SIZE2

Kinds	All except port
Interrogate	Anytime
Modify	When closed
Type	Integer
Range	0 through 23
Default	0
Abstract Value	DEFAULT, MINIMUM, MAXIMUM
Stored Permanently	Disk or tape
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

If the file has a BLOCKSTRUCTURE value of VARIABLEOFFSET, use the SIZE2 attribute to determine or specify the length, in SIZEMODE units, of the field in the variable-length record pointed to by the SIZEOFFSET attribute. If the value of the SIZEMODE attribute is SINGLE, a value of 1 is used for SIZE2.

A message control system (MCS) can assign to a file a value of up to 65535 for the SIZE2 attribute and use this value in any way that the MCS program chooses. If this technique is used, the program must not set the BLOCKSTRUCTURE file attribute value of the file to VARIABLEOFFSET.

STATE

Kinds	All
Interrogate	When assigned
Modify	Never
Type	Word
Range	Not applicable
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Supported
Heritage Host Services Logical I/O	Restricted usage/values
Port Services	All

Interrogate the STATE attribute to determine the result of the last I/O operation performed on the logical file. A copy of the logical I/O result descriptor is returned. While the STATE attribute returns a bit mask of results and a single enumerated result, the RESULTLIST attribute returns a list of all results of the I/O operation whether or not the results have corresponding bit positions in the STATE attribute.

For port files, a subfile index can be used as a parameter. When accessed as a port subfile attribute, the STATE attribute contains the result of the last I/O operation on the specified subfile. When no subfile index is specified, the STATE attribute contains the result of the last I/O operation on the port file.

This attribute is not valid for direct I/O files. Refer to the description of the IORESULT buffer attribute in Section 7 for information that relates to direct I/O files.

As shown in the following table, the fields of the descriptor contain information about the last I/O operation performed on the file:

Field	Contents
[0:1]	If this bit is on, an error has occurred. If the bits of field [16:16] are zeros, the error information is found only in the enumerated result field [26:10]. If an implicit OPEN operation on the file has failed, the value IMPLICITOPENFAILED (91) is found in that field. This bit is not on if bits [26:10] and the RESULTLIST attribute contain only warnings and indications but no error results.
[16:16]	This field contains the following error information:

Field	Contents
[2:1]	The I/O operation was canceled.
[3:1]	For port files only. An invalid subfile index was specified.
[4:1]	A length error occurred; this is a size error for variable-length records.

Field **Contents**

[6:1] For files with a FILEORGANIZATION value of INDEXED or INDEXEDNOTRESTRICTED, a primary key error occurred for sequential WRITE operations. This bit indicates that the primary keys are not in sequential order. For REWRITE and DELETE operations, this bit indicates that the primary keys are not equal.

[7:1] If this bit is on, a parity error has occurred.

[8:1] If the file is a port file and this bit is on, an I/O operation failed for one of the following reasons:

- A broadcast WRITE operation failed for one or more subfiles.
- A WRITE operation with the DONTWAIT option failed because no buffer was available.
- A READ operation with the DONTWAIT option failed because no data was available.

If the file is a disk file and this bit is on, a deleted disk area was referenced.

[9:1] If this bit is on, end of file or end of page occurred.

[10:1] If this bit is on, a short block has been read, or if the FILESTRUCTURE attribute value is STREAM, a short record at end of file has been read. This information is reported without the exception field ([0:1]) being turned on because this condition is not necessarily an error.

[13:1] If this bit is on, the remote file has encountered a break on output.

[15:1] If this bit is on, the I/O time limit has been exceeded. For remote files, refer to the description of the TIMELIMIT attribute. For all other files, this information indicates a possible hardware failure.

[16:1] If this bit is on, a security violation has been attempted.

[26:10] If the attention bit [1:1] is not on, this field contains an I/O result enumeration. See Appendix F, “I/O Result Enumerated Values,” for a detailed description.

One of the following situations can result in this field containing the value 1023:

- A discontinued process—for example, one that is invoking code in an EPILOG procedure—attempted to gain access to a logical file that was being used by another process.
- A process that cannot be discontinued was discontinued while it was attempting to gain access to a logical file that was being used by another process.

[27:1] If this bit is on, there are results stored in the RESULTLIST attribute value.

[47:20] This field contains the size of the last logical record read or written in FRAMESIZE units. This value is undefined if the previous I/O operation was a SEEK or DELETE operation. This field corresponds to the value of the CURRENTRECORDLENGTH attribute. When the value of DUMMYFILE is TRUE, this field is 0 (zero).

The logical result returned by a READ operation always reports any errors that occurred during the READ operation.

The logical result returned by a WRITE operation that specified a SYNCHRONIZE value of OUT always reports any errors that occurred during that WRITE operation. The result also reports any errors that occurred on prior WRITE operations, but have not been reported.

The logical result returned by a WRITE operation that did not specify a SYNCHRONIZE value of OUT usually reports any error that might have prevented the current WRITE operation from being issued, for example, an end-of-file condition. In general, however, errors that occur during the WRITE operation are not reported until some subsequent WRITE, READ, SEEK, or CLOSE operation occurs.

STATIONCOUNT

Kinds	Remote
Interrogate	When open
Modify	Never
Type	Integer
Range	Not applicable
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Interrogate the STATIONCOUNT attribute to determine the number of stations in the station list of the remote file. The initial value of STATIONCOUNT is 1, unless the file is declared in the DATACOMINFO file, in which case the initial STATIONCOUNT value is the number of stations specified in the DATACOMINFO description of the file. The STATIONCOUNT value changes only if stations are added to or subtracted from the file. If stations are subtracted from the file, the largest valid relative station number (RSN) can be larger than the STATIONCOUNT value. The STATIONLIST attribute can be used to add or subtract stations.

For further information regarding RSNs, refer to the *I/O Subsystem Programming Guide*.

STATIONLIST

Kinds	Remote
Interrogate	Never
Modify	When open
Type	Character string
Range	Not applicable
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Use the STATIONLIST attribute to specify the name of a station or set of stations that are to be dynamically added or subtracted to or from a remote file. The station list of the file is not changed if one of the following conditions occurs:

- The file or station to be added or subtracted is not described in the DATACOMINFO file.
- The file is already in the station list if the file is being added.
- The file is not in the station list if the file is being subtracted.

The syntax for the use of the STATIONLIST attribute is as follows:

ALGOL

```
REPLACE FILEID.STATIONLIST BY * + "DC1.";

REPLACE FILEID.STATIONLIST BY * - "L1011.";
```

COBOL

```
CHANGE ATTRIBUTE STATIONLIST OF FILEID UP BY "AAA1.".

CHANGE ATTRIBUTE STATIONLIST OF FILEID DOWN BY "BBB2.".
```

Attempting to add or subtract a station or file not found in the DATACOMINFO file causes an attribute error. Subtracting all stations from the file does not close the file; however, performing subsequent I/O statements while STATIONCOUNT is 0 (zero) causes end-of-file (EOF) action.

When a station is subtracted from a file, the STATIONCOUNT value and all other relevant attribute values are modified to reflect the change, the controlling message control system (MCS) of the station is sent a “FILE CLOSE” message for that station, and all knowledge of that station from the point of view of the logical file is lost. When a station is added to a family of the remote file, the value of the STATIONCOUNT attribute is increased, and the controlling MCS is sent a “FILE OPEN” message for that station.

When a station is subtracted from the station list, the relative station numbers (RSNs) for all other stations in the list remain the same. If a WRITE operation is directed to an RSN corresponding to a station that has been subtracted from the list, an end-of-file (EOF) result is returned. For a full explanation of RSNs, refer to the *I/O Subsystem Programming Guide*.

When a station is added to the station list of a remote file, the LASTSUBFILE attribute is updated to be the RSN of the newly added station, if no attribute error occurred during the addition. In this way, a user program can determine the RSN of the newly added station for future use.

The value of STATIONLIST cannot be altered by a file equation.

STATIONNAME

Kinds	Remote
Interrogate	When open
Modify	Never
Type	Character string
Range	Not applicable
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	No
Parameters	1 required
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Interrogate the STATIONNAME attribute to determine the name of the station, as defined in the DATACOMINFO file, that is being used for communication. You must use a relative station number (RSN) as a parameter. If the specified RSN is invalid, a null string is returned.

The following ALGOL example shows how to interrogate STATIONNAME:

```
REPLACE POINTER (A) BY REMOTFYL(THIS_RSN).STATIONNAME;
```

STATIONSALLOWED

Kinds	Remote
Interrogate	When open
Modify	Never
Type	Integer
Range	Not applicable
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Interrogate the STATIONSALLOWED attribute to determine the number of stations assigned to a remote file. If the STATIONSALLOWED value for a remote file is not equal to the STATIONCOUNT value of the file, the STATIONSDENIED and DISPOSITION attributes can be used to determine the state of the file. Refer to the descriptions of these attributes for further information.

STATIONSDENIED

Kinds	Remote
Interrogate	When open
Modify	Never
Type	Integer
Range	Not applicable
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Interrogate the STATIONSDENIED attribute to determine the number of the stations in a family that have been denied assignment to the file by their controlling message control system (MCS).

Because remote file assignment is an asynchronous process requiring the cooperation of an MCS, there is a period when the POPULATION attribute can return a value of 0 (zero) because the MCS has not yet reacted to the file OPEN request or requests. To determine if the POPULATION attribute is 0 because of such an interim period or because all stations have been denied assignment, interrogate the values of the STATIONSDENIED and STATIONCOUNT attributes. Refer to the descriptions of the POPULATION and STATIONCOUNT attributes.

SUBFILEERROR

Kinds	Port subfile
Interrogate	Anytime
Modify	Never
Type	Mnemonic
Range	See following text
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	All

Interrogate the SUBFILEERROR attribute to determine the error, if any, that occurred on the specified subfile during dialogue establishment and termination. SUBFILEERROR is also used to return the reason for deactivation of a subfile.

To determine errors on a READ or WRITE operation, interrogate the result of the READ or WRITE operation and the RESULTLIST attribute rather than interrogating the SUBFILEERROR attribute.

The following lists contains mnemonic values and meanings associated with the SUBFILEERROR attribute:

0 (NOERROR)

No error occurred.

1 (DISCONNECTED)

Communication with the corresponding port subfile was severed due to network failure.

2 (DATALOST)

All data might not have been transmitted before the correspondent port subfile was deactivated. This error is also returned if a CLOSE statement with the ABORT option is invoked on the subfile and there is still data in the subfile read queue. The CLOSE ABORT operation does not fail if this error occurs.

3 (NOBUFFER)

A WRITE operation with the DONTWAIT option failed because no buffer space was available.

4 (NOFILEFOUND)

Dialogue establishment failed because a matching dialogue endpoint was not found.

5 (UNREACHABLEHOST)

The host specified in the YOURHOST attribute describes a foreign host that was unreachable during dialogue establishment evaluation or became unreachable during dialogue establishment.

It is possible that the Port Provider does not support dialogues with foreign hosts; this is different from the host being unreachable. An example of this is the Local Port Provider (LPP). When dialogues with foreign hosts are not supported by the Port Provider, then UNSUPPORTEDFUNCTION (6) is set. It is also possible that the Port Provider does not support dialogues within the local host (for example, local ports). An example of this is the BNA 4.3 provider. When this is the case, then UNSUPPORTEDFUNCTION (6) is set.

6 (UNSUPPORTEDFUNCTION)

An attempted OPEN or AWAITOPEN operation on this port subfile resulted in a request for an unsupported function. The unsupported functions include the following cases:

- The YOURHOST attribute specifies the local host and local ports are not supported by the Port Provider.
- The YOURHOST attribute specifies a foreign host and foreign hosts are not supported by the Port Provider.

7 (UNSUPPORTEDPROTOCOLTYPE)

An attempt to establish a dialogue on this port subfile required an unsupported protocol type.

8 (PROTOCOLERROR)

A protocol error occurred.

9 (LACKOFRESOURCES)

An attempt to establish a dialogue on this port subfile failed due to a lack of resources.

10 (HOSTNOTINHOSTGROUP)

An attempt to establish a dialogue on this port subfile failed because the YOURHOST value is not a legal member of the specified YOURHOSTGROUP. This error applies to BNANATIVESERVICE BNA port files only.

11 (UNAUTHORIZEDFORAPPLICATIONGROUP)

An attempt to establish a dialogue on this port subfile failed because the user is not an authorized user of the specified application group.

12 (IPCUNAVAILABLE)

An attempt to establish a dialogue on this port subfile failed because the system software that supports ports is presently unavailable.

13 (BADATTRIBUTESFOROPEN)

An attempted OPEN operation on this port subfile failed because one or more of the attributes affecting the OPEN process either has illegal syntax or has an invalid value.

14 (UNAVAILABLEFUNCTION)

An attempt to establish a dialogue on this port subfile failed because the service requested is unavailable.

15 (UNSUPPORTEDINTMODE)

An attempt to establish a dialogue on this port subfile failed because the INTMODE value specified is unsupported on the remote host specified by YOURHOST.

16 (NETWORKINGNOTSUPPORTED)

An attempted OPEN operation on this port subfile failed because the networking software on the local host is not in network mode.

17 (DISCONNECTEDDURINGOPEN)

The matching process was aborted and the subfile is closed. This error can be returned anytime during dialogue establishment.

18 (UNSUPPORTEDPARAMETER)

The attempted OPEN, AWAITOPEN, CLOSE, or RESPOND statement was invoked with parameters or options not supported by the service of the subfile.

19 (LOCALIPCNOTSUPPORTED)

A local dialogue was requested by setting the value of YOURHOST to the value of MYHOST, and such a dialogue is not supported by the service of the subfile.

20 (UNSUPPORTEDTRANSLATION)

An attempt to establish a dialogue failed because a system translation that the service could not provide was requested.

21 (OPENREJECTED)

An attempted OPEN operation failed because the correspondent endpoint rejected the request made by the port subfile.

23 (ENDPOINTUNKNOWN)

The attempt to establish a dialogue failed because the local system operator failed to register the endpoint or the endpoint is unknown at the correspondent host.

24 (INVALIDRESPOND)

The attempted RESPOND operation failed because an invalid RESPOND option was provided. Either an ACCEPTOPEN or REJECTOPEN operation was specified and the value of the FILESTATE attribute was CLOSERESPONSEPLEASE, or an ACCEPTCLOSE or REJECTCLOSE operation was specified and the value of FILESTATE was OPENRESPONSEPLEASE.

25 (INVALIDPROVIDER)

An attempt to establish a dialogue failed because the PROVIDERGROUP value does not include a service provider that can provide the service requested for the subfile.

26 (PROVIDERRESTRICTED)

An attempt to establish a dialogue failed because providers currently supporting the subfiles of the port file cannot be mixed with the provider of the service of the subfile.

27 (TIMELIMITEXCEEDED)

The attempted OPEN or AWAITOPEN operation failed because the value of CONNECTTIMELIMIT was exceeded.

28 (UNSUPPORTEDAPPLCONTEXT)

An attempt to establish a dialogue failed because the correspondent rejected the request since it did not support the application context. The error applies to OSINATIVESERVICE only.

31 (BADASSOCIATEDDATA)

The ASSOCIATEDDATA was unreadable. For a description of ASSOCIATEDDATA, refer to the *I/O Subsystem Programming Guide*.

32 (ASSOCIATEDDATATOOLONG)

The ASSOCIATEDDATA was too long. For a description of ASSOCIATEDDATA, refer to the *I/O Subsystem Programming Guide*.

33 (BADATTRIBUTESFORRESPOND)

The attempted RESPOND operation failed because one or more attributes were set to invalid values.

34 (BADATTRIBUTESFORAWAITOPEN)

The attempted AWAITOPEN operation failed because one or more attributes were set to invalid values.

36 (WARNPORTATTIGNORED)

One or more port attributes were set to values invalid for the service of the subfile and are being ignored. This warning does not cause the OPEN or AWAITOPEN operation to fail.

37 (WARNCONTEXTCHANGED)

The APPLICATIONCONTEXT attribute changed this negotiation. This warning does not cause the OPEN operation to fail.

38 (WARNCLOSECOLLINITIATOR)

The ORDERLY CLOSE operation cannot complete because a collision of CLOSE requests occurred. This endpoint is the initiator of the dialogue. This error applies to OSINATIVESERVICE and OSISESSIONSERVICE services only.

39 (WARNCLOSECOLLRESPONDER)

The ORDERLY CLOSE operation cannot complete because a collision of CLOSE requests occurred. This endpoint is the responder of the dialogue. This error applies to OSINATIVESERVICE and OSISESSIONSERVICE services only.

40 (ENDPOINTINCOMPWITHSERVICE)

The dialogue endpoint cannot be accessed by this service.

41 (UNSUPPORTEDPRIMITIVE)

A statement not supported by the service of the subfile was invoked.

42 (BASICSERVICEVIOLATION)

A protocol message from the correspondent system or endpoint was received that used a feature that BASICSERVICE service does not support. The dialogue is closed.

43 (INVALIDFILESTATE)

The RESPOND operation was invoked while the subfile was in an invalid file state.

44 (TRANSPARENTNOTSUPPORTED)

The correspondent system or endpoint does not support the abstract syntax NIST ABSTRACT SYNTAX 3, which is the message format used by this OSI subfile for data transfer.

45 (OPENABORTBYCORRESPONDENT)

The dialogue establishment failed because an abort was sent by the correspondent endpoint.

46 (OPENREJECTEDTRANSIENT)

The OPEN operation failed because the correspondent endpoint rejected the request for transient reasons.

47 (WARNCONTEXTIGNORED)

The APPLICATIONCONTEXT attribute was not null when the AWAITOPEN operation was invoked and is being ignored. The responder can only participate in dialogue parameter negotiation if the AWAITOPEN statement is invoked with the PARTICIPATE option, and the subfile is in the OPENRESPONSEPLEASE file state. This AWAITOPEN operation does not fail.

48 (WARNABORTDATAIGNORED)

The associated data parameter specified with the CLOSE ABORT statement was not sent with the abort because it is too long. The CLOSE ABORT operation does not fail.

49 (PROVIDERGROUPUNDEFINED)

The OPEN or AWAITOPEN operation failed because the provider group specified by the PROVIDERGROUP attribute has not been defined through the NW NS ADD PROVIDERGROUP operator command.

50 (WARNDEFINEPROVIDERGROUP)

The control option specified on the OPEN or AWAITOPEN operation was DONTWAIT or OFFER and the provider group specified by the PROVIDERGROUP attribute is undefined. The OPEN or AWAITOPEN operation does not fail, but the program is warned that the dialogue establishment process cannot continue until the provider group is defined through the NW NS ADD PROVIDERGROUP operator command.

51 (WARNINITIATINGPROVIDER)

The control option specified on the OPEN or AWAITOPEN operation was DONTWAIT or OFFER and the provider of the port service is not available. The system automatically initiates the provider only if a halt/load has just occurred and if the provider was running just before the halt/load took place. In all other cases, the operator must manually initiate the provider. The OPEN or AWAITOPEN operation does not fail, but the program is warned that the dialogue establishment process cannot continue until the provider initializes successfully. If the provider fails to initialize, the dialogue establishment fails with a NETWORKINGNOTSUPPORTED (16) SUBFILEERROR and the subfile moves to the CLOSED file state.

54 (NETBIOSNAMEINUSE)

Another application on the same local area network (LAN) is attempting to open a port subfile with the same MYNAME value as this subfile. For further information about this error, refer to the discussion of NETBIOSSESSIONSERVICE in the *I/O Subsystem Programming Guide*.

55 (ALLSEGMENTSNOTSENT)

The PORTSEGMENTIO file attribute value is TRUE and all segments of data were not sent when the program attempted a CLOSE ORDERLY operation.

56 (RESOLVERNOTAVAILABLE)

The Resolver is not running. The Resolver must be running to resolve the YOURDOMAINNAME file attribute value.

57 (NAMESERVICENOTAVAILABLE)

The Resolver has not been configured with any known name server.

58 (NAMESERVICEUNREACHABLE)

The Resolver cannot contact any name server to process the open request.

59 (CONNECTIONINUSE)

Another application on the same local host is attempting to open or support a subfile that has the same MYNAME, MYIPADDRESS, YOURNAME, and YOURIPADDRESS values as this subfile. Each subfile must have a unique combination of these four attributes on the local host.

SYNCHRONIZE

Kinds	Disk, tape, or CD
Interrogate	Anytime
Modify	Anytime
Type	Mnemonic
Range	See following text
Default	NO (nondirect I/O) OUT (direct I/O)
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

Use the SYNCHRONIZE attribute to determine if or specify that each WRITE operation to a logical file also forces a WRITE operation to the physical file and waits for the operation to finish before control is returned to the program.

For tape devices that have local data buffers, the use of the SYNCHRONIZE attribute also controls the buffering mode of the device. If the SYNCHRONIZE value is NO for a file assigned to such a device, the unit is placed in buffered mode; otherwise, the unit is placed in the unbuffered mode. Operating in an unbuffered mode can have a negative impact on performance, but can be used to insure that data has physically been written to tape and not just placed in the buffer. The effect on performance when using unbuffered mode might be severe for tape drives having long repositioning times (such as the HS8500 tape drive).

The SYNCHRONIZE attribute is valid only for files with a FILEORGANIZATION attribute value of NOTRESTRICTED.

If the FILESTRUCTURE attribute value is STREAM or BLOCKED and the UPDATEFILE attribute value is TRUE, the SYNCHRONIZE value must be NO.

The mnemonic values and their meanings that are associated with the SYNCHRONIZE attribute are as follows:

Mnemonic Value	Integer Value	Meaning
NO	0	Logical and physical output operations are asynchronous. A program issuing a WRITE operation can continue execution as soon as the record has been moved to one of the buffers of the file. When the buffer is full, it is written to the physical file while the program can continue writing to another buffer. The result of the physical WRITE operation is returned to the program when a buffer is reused.
OUT	2	Each WRITE operation to the logical file also forces a WRITE operation to the physical file. A program performing a WRITE operation remains suspended until the completion of the WRITE operation to the physical file. The result of the physical WRITE operation is returned to the program when processing resumes. For disk files, the physical record blocking of the file is unaffected on synchronous output. For tape files, however, each logical record is written to the physical file as a separate block. Records in a blocked tape file, therefore, are written as short blocks. SYNCHRONIZE can also be directly specified in WRITE statements. A SYNCHRONIZE specification in a WRITE statement overrides any setting of the SYNCHRONIZE attribute for that particular WRITE operation. However, when multiple, consecutive WRITE operations are to be done in synchronous mode, it is more efficient to set the attribute directly.

The following examples show how to request synchronization in a WRITE statement:

ALGOL

```
WRITE (FILENAME [KEY, SYNCHRONIZE], 30, ARRAYNAME);
```

COBOL74

```
WRITE RECORDNAME SYNCHRONIZE
```

An occasional synchronous WRITE operation that follows one or more asynchronous WRITE operations can be used to ensure that all outstanding WRITE operations to the file are completed before the program continues execution.

If a WRITE operation to a permanent file is directed past the current end of file, the synchronous WRITE operation also forces an update of the end-of-file (EOF) pointer in the disk directory. As a consequence, nothing is gained by assigning the PROTECTION attribute a value of PROTECTED for files that are being written synchronously. The extra disk directory update for each synchronous WRITE operation can be avoided through a synchronous, random WRITE operation to a record location that is larger than the current EOF location before any intervening records are written. Applications that utilize this technique must be prepared to deal with a situation following a system failure where some records prior to the end of file might never have been written.

Direct I/O provides full programmatic control of I/O synchronization. Every direct I/O WRITE statement causes an I/O operation to be issued to the physical file. A program using direct I/O can wait for the I/O operation to finish and can check the result before continuing. For this reason, the SYNCHRONIZE attribute for direct I/O affects only the buffering mode of buffered tape devices.

Section 5
General File Attributes T through Z

This section presents file attribute descriptions for attributes with names starting with the letter T through names starting with the letter Z. Each attribute description begins with a formatted list of attribute characteristics, followed by a narrative discussion of the use of the attribute. Where appropriate, this discussion includes a list of the various combinations of values, mnemonics, and meanings associated with that attribute.

Refer to Tables 1–1 through 1–12 for descriptions of the formatted lists.

TANKING

Kinds	Remote
Interrogate	Anytime
Modify	When closed
Type	Mnemonic
Range	See following text
Default	UNSPECIFIED
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

Use the TANKING attribute to indicate whether or not output is to be tanked and whether the message control system (MCS) can override your specification. The mnemonic values associated with the TANKING attribute are as follows:

Mnemonic Value	Integer Value	Meaning
ASYNC	3	The output is tanked. When the file closes, the task can proceed and even go to end of task. The output continues to be transmitted until there is no more output.
NONE	1	The output is not tanked.
SYNC	2	The output is tanked. When the file closes, the task cannot proceed until all tanked output has been transmitted.
UNSPECIFIED	0	The output is not tanked unless the MCS specifies tanking at the time the file is assigned.

The MCS can override only a value of UNSPECIFIED.

The currently tanked output for the file is discarded if a break-on-output request is made.

TAPEREELRECORD

Kinds	Tape
Interrogate	When assigned
Modify	Never
Type	Integer
Range	Not applicable
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	Tape
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

If a tape file is labeled, interrogate the TAPEREELRECORD attribute to determine the record number of the last record read or written, relative to the first record on the current volume of the tape file. For example, if the first record of the second volume of a tape file is read, TAPEREELRECORD returns 0 (zero). The value returned before the first I/O operation is -1.

A CLOSE REEL statement on an input tape invalidates the value returned by TAPEREELRECORD.

TIMELIMIT

Kinds	Remote
Interrogate	Anytime
Modify	Anytime
Type	Integer
Range	0 through 549755813887
Default	0
Abstract Value	DEFAULT, MINIMUM, MAXIMUM
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Use the TIMELIMIT attribute to determine or specify how long, in seconds, the I/O subsystem is to wait for an I/O operation to occur. The value of the attribute is either a positive integer or 0 (zero). If specified as REAL, the value is rounded off to the nearest integer. You can modify the attribute in a READ or a WRITE statement in ALGOL by using the *[TIMELIMIT <arithmetic expression>]* form of the *[<record number or carriage control>]* part of the I/O statement.

When the value of the TIMELIMIT attribute is 0 (zero)—after rounding, if necessary—I/O operations wait indefinitely. If the value is 1 or greater, the action is as follows:

- On a READ operation, if no input is received within the number of seconds specified by TIMELIMIT, the READ operation is terminated with a TIMELIMIT error.
- On a WRITE operation, if the data cannot be buffered for output within the number of seconds specified by TIMELIMIT, the WRITE operation is terminated with a TIMELIMIT error.

If the TIMELIMIT value is exceeded during an I/O operation, the logical I/O result descriptor has bit [0:1] and bit [15:1] turned on, and the TIMEOUT (15) enumeration is present. Refer to the description of the STATE attribute.

TIMESTAMP

Kinds	Disk or CD
Interrogate	When assigned
Modify	Never (See following exception)
Type	Word
Range	Not applicable
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	Disk or CD
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

Interrogate the TIMESTAMP attribute to obtain the disk file header timestamp of the associated disk file in TIME(6) format.

In general, the purpose of the disk-file-header timestamp is to note whether the contents or attributes of the file have been changed since the last time the file was backed up by using library maintenance. The value of the time stamp does not necessarily reflect the most recent changes to either the contents or the attributes. The exact circumstances under which the timestamp is updated can change without notice from release to release.

For more information about the TIME(6) format, refer to the TIME intrinsic function in the *ALGOL Reference Manual, Volume 1*.

The TIMESTAMP attribute can be modified by a valid MCS program if that program has already set the HSFILECOPY file attribute to TRUE.

TITLE

Kinds	All
Interrogate	Anytime
Modify	Anytime (disk) When closed (other devices)
Type	Character string
Range	Not applicable
Default	FILENAME
Abstract Value	DEFAULT
Stored Permanently	Disk, CD, or tape
Parameters	1 optional
FTAM File Access	Restricted Use
Heritage Host Services Logical I/O	Supported
Port Services	All

Use the TITLE attribute to determine or specify an external file name. The external file name is used to associate a logical file with a physical file or to match correspondent port files. The default value of TITLE is the value of the FILENAME attribute.

Note: *The FILENAME, LFILENAME, TITLE, LTITLE and PATHNAME file attributes are all used for naming a file. Each of these attributes has its own unique characteristics that makes it the right interface to use in certain circumstances. However, setting any one of these attributes causes a change in the value of all of the others. For disk files, the interpretation of the FILENAME attribute is also affected by the SEARCHRULE file attribute. For further information, see the "PATHNAME" and "SEARCHRULE" file attributes in this manual.*

If the TITLE attribute is interrogated for a disk, CD-ROM, or tape file, the name of the physical file is returned whenever one is assigned to the logical file. A null string is returned if the file is open, but not assigned to a physical file. For example, a null string is returned if an OF (Optional File) command was entered when an optional file was opened, if the DUMMYFILE attribute value is TRUE, or if the file is a tape file that is positioned beyond the last file on the tape. If the TITLE attribute is interrogated when a disk file is assigned to a logical file, a string containing an asterisk (*) or a usercode and an *ON <family name>* phrase can be returned.

The TITLE attribute can be modified while the file is unassigned by using the form *<file name> ON <family name>*. When TITLE is modified using this form, the FILENAME, FAMILYNAME, and KIND attributes are affected as follows:

- The FILENAME attribute is set to the <file name> value.
- The FAMILYNAME attribute is set to the <family name> value.
- The KIND attribute is set to PACK.

When the TITLE of a file is modified to a value that does not include the *ON <family name>* phrase, only the FILENAME is changed; the FAMILYNAME and KIND attributes are not affected.

The TITLE value can be changed while the file is assigned only if the file is assigned to a disk file.

Disk Files

Changing the TITLE while the file is assigned causes the FILENAME value of the physical disk file to be changed also. If a family name is supplied in this case, it must match the FAMILYNAME value of the physical file. If you want to change the file name, using the FILENAME attribute is preferable to using the TITLE attribute. Only the owner of the file or a privileged user is permitted to change the name of a file.

Heritage Host Services logical I/O does not support modification to the TITLE attribute of a file while the file is assigned, but the FILENAME attribute can be modified.

FTAM supports interrogating the title and setting the title except for the <on part> variable. If you set the <on part> variable, you must enclose the entire title in apostrophes ('). It is better to use the FILENAME attribute rather than the TITLE attribute for files accessed through FTAM.

In SSR 44.2 and subsequent releases, the MCP supports two forms of file names: “traditional” file names and “long” file names. A title specifying a long file name that does not conform to the traditional file name limits can be assigned to the LTITLE attribute, but not to the TITLE attribute. When read, the TITLE and LTITLE attributes always return the same value, even when that value is a long file name that cannot be assigned to the TITLE attribute by a user program. See the “Valid File Names in the MCP Environment” subsection of the FILENAME attribute description for additional information.

Printer Files

Refer to the USERBACKUPNAME attribute for information about the title of a file when the USERBACKUPNAME value is TRUE.

The use of the file identifier SUMMARY should be avoided for printer files. If a printer file is declared in WFL with a file identifier of SUMMARY, and a job summary is generated with its default title of SUMMARY, the printer file declared in WFL is lost. The printer file is lost because the job summary file overwrites the file declared in WFL.

To keep the WFL job from creating a job summary file with the file identifier SUMMARY, either use the JOBSUMMARY task attribute to suppress the job summary or use the JOBSUMMARYTITLE task attribute to assign a different name to the job summary file.

When a printer backup file is open and assigned to a physical file, the TITLE attribute returns the title of the physical file. When the file is closed, the TITLE attribute returns the value that has been assigned by the user program. For example, the following ALGOL program fragment returns “PRINTFILE.” when the file is closed:

```
FILE LINE(KIND=PRINTER,TITLE="PRINTFILE.")
ARRAY A[0:44];

REPLACE POINTER(A) BY LINE.TITLE;
```

When the file is open and assigned to backup on pack, the program fragment above responds in the following form:

```
*BD/<7 digit job #>/<7 digit mix #>/<print file count>PRINTFILE ON PACK.
```

For more information about the backup disk file title, see the information about the structure of backup files in the *Print System Guide*.

Tape Files

The *<file name> ON TAPE* qualifier can be used to access tape files. When this qualifier is used, the I/O subsystem changes the value of the KIND attribute to TAPE.

For information about a valid tape file name, refer to the FILENAME attribute.

Port Files

Using the FILENAME attribute is preferable to using the TITLE attribute. A port file title uses the port file name format. For information about valid port file names, refer to the FILENAME attribute.

Remote Files

If the STATION task attribute value is 0 (zero), you can use this attribute to identify the name of the DATACOMINFO file that is to be assigned to the remote file. If a file with the TITLE value is not found in the DATACOMINFO file, the TITLE value becomes the first station in the station list.

The STATIONNAME and STATIONLIST file attributes are used to determine the name of a station in the station name list, and change the contents of the station name list, respectively.

TOTALSECTORS

Kinds	Disk or CD
Interrogate	When assigned
Modify	Never
Type	Integer
Range	Not applicable
Default	None
Abstract Value	Not applicable
Stored Permanently	Disk or CD
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Interrogate the TOTALSECTORS attribute to determine the number of sectors currently assigned to the file.

For disk files, this value changes when the disk space allocated to a file changes. The value increases by the value of the AREASECTORS attribute when a new area is allocated due to I/O activity in a previously unallocated area. The value can decrease when a file is crunched. Additionally, the value can either increase or decrease by the AREASECTORS value as a result of an EXCHANGE intrinsic call.

TRAINID

Kinds	Disk and printer
Interrogate	Anytime
Modify	When closed
Type	Mnemonic
Range	See following text
Default	NOID
Abstract Value	DEFAULT
Stored Permanently	Backup file
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Use the TRAINID attribute to determine or specify the print train for a train printer. At the time the file is printed, if this attribute has a value other than NOID, the translate table is loaded, as needed, from the file SYSTEM/TRAINTABLES and the operating system waits until the selected print train is made available.

The mnemonic and integer values, and meanings associated with TRAINID are as follows:

Mnemonic Value	Integer Value	Print Train
NOID	0	Any available printer
EBCDIC18	1	EBCDIC18
FORTRAN48	2	FORTRAN48
B300B500	3	B300B500
EBCDIC48	4	EBCDIC48
EBCDIC72	5	EBCDIC72
UK3500	6	UK3500
UK6500	7	UK6500
LATINPORTUGAL	8	LATINPORTUGAL
LATINSPAIN3	9	LATINSPAIN3
SWEDENFINLAND3	10	SWEDENFINLAND3
DENMARK	11	DENMARK
BCL64	12	BCL64
TURKEY	13	TURKEY
SWEDENOCRB	14	SWEDENOCRB

Mnemonic Value	Integer Value	Print Train
ASCII72	15	ASCII72
EBCDIC96	16	EBCDIC96
KATAKANA	17	KATAKANA
ALPHAEBCDIC	18	ALPHAEBCDIC
NUMERICEBCDIC	19	NUMERICEBCDIC
RPG48	20	RPG48
OCRANUMERIC	21	OCRANUMERIC
OCRBNUMERIC	22	OCRBNUMERIC
OCRAALPHANUMERIC	23	OCRAALPHANUMERIC
UK	24	UK
GERMANYAUSTRIA	25	GERMANYAUSTRIA
ITALY	26	ITALY
SWEDENFINLAND2	27	SWEDENFINLAND2
LATINSPAIN2	28	LATINSPAIN2
ASCII64A	29	ASCII64A
BRAZIL	30	BRAZIL
DENMARKNORWAY	31	DENMARKNORWAY
YUGOSLAVIA	32	YUGOSLAVIA
EBCDIC64A	33	EBCDIC64A
ASCII96A	34	ASCII96A
EBCDIC64B	35	EBCDIC64B
ASCII64B	36	ASCII64B
ASCII96B	37	ASCII96B
UKNETHITALY	38	UKNETHITALY
GREECE	39	GREECE
UKHOLLITALY64	40	UKHOLLITALY64
FRBELGIUM64	41	FRBELGIUM64
FRBELGIUM96	43	FRBELGIUM96
LATINSPAIN96	44	LATINSPAIN96
LATINPORTUGAL96	45	LATINPORTUGAL96
POSTNET64	52	POSTNET64
ARABIC1A96	58	ARABIC1A96
EBCDIC256	82	EBCDIC256

Mnemonic Value	Integer Value	Print Train
ASCII256	83	ASCII256
USERTRAINID1	90	BNA 90
USERTRAINID2	91	BNA 91
USERTRAINID3	92	BNA 92
USERTRAINID4	93	BNA 93
USERTRAINID5	94	BNA 94
USERTRAINID6	95	BNA 95

TRANSFORM

Kinds	Disk and printer
Interrogate	Anytime
Modify	When closed
Type	Character string
Range	Not applicable
Default	Null string
Abstract Value	DEFAULT
Stored Permanently	Backup file
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Use the TRANSFORM attribute to determine or specify a file transform function. Transform functions can be used to manipulate the data of a source file before printing the data. For more information, refer to the *Print System Guide*.

If the value of the PRINTDISPOSITION attribute is DIRECTDLP or DIRECTPS, the TRANSFORM attribute is ignored.

When this attribute is set, the string value must conform to the following syntax:

```
──<transform name>─┬────────────────────┬──────────────────────────────→
                   └─ ( ──<string>── ) ─┘

→─┬──────────────────────────────────────────────────────────────────────┤
  └─ IN ─┬─<library title>───────┤
         └─ SL ──<function name>─┘
```

When the library title is specified,

- The transform name must be an object within the specified library.
- If the SECOPT SLTRANSFORMS option is set, access to the transform library on remote printers is restricted.

When SL is specified, the library object must reside in the library specified by the function name that was created with the SL (Support Library) system command.

If a library title or SL <function name> is not specified, the transform name must reside in the PRINTSUPPORT library.

If a parenthetical string of characters is specified, it is passed to the transform function when it is invoked by the Print System. This unquoted string can be used to pass parameters or initialization information to the transform function.

TRANSLATE

Kinds	All
Interrogate	Anytime
Modify	When closed
Type	Mnemonic
Range	See following text
Default	NOTRANS (port) FULLTRANS (all others)
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Restricted values
Port Services	BNANATIVESERVICE

Use the TRANSLATE attribute to determine or specify situations under which translation can occur. The system default is satisfactory for most programs. In general, translation is required and is performed when the INTMODE attribute value does not equal the EXTMODE attribute value, and neither attribute has a value of SINGLE. For further information regarding translation, refer to the *I/O Subsystem Programming Guide*.

The mnemonic values and meanings associated with the TRANSLATE attribute are as follows:

Mnemonic Value	Integer Value	Meaning
DEFAULTTRANS	0	Software translation can occur when translation is required and hardware translation is not provided. Under some circumstances, translation is not performed when required. Refer to the discussion of migrating to an enterprise server system in the *I/O Subsystem Programming Guide*. The DEFAULTTRANS value is not supported by Heritage Host Services logical I/O. Any request for DEFAULTTRANS is changed to FULLTRANS.
FORCESOFT	3	Software translation takes place whether or not translation is required, as long as software translation is not specifically disallowed. If translate tables are provided by the program, those translate tables are used. Translation is not permitted while binary I/O operations are being performed. If translation would not otherwise be required, the program must supply the translate tables or the program receives a fatal error on the first I/O operation. Refer also to the descriptions of the INPUTTABLE and OUTPUTTABLE attributes.

Mnemonic Value	Integer Value	Meaning
FULLTRANS	1	Software translation occurs when translation is required and hardware translation is not provided.
NOSOFT	4	An invalid translation open error occurs whenever translation is required and hardware translation is not provided.
NOTRANS	5	An invalid translation open error occurs whenever hardware or software translation is required.
SOFTONLY	2	Only software translation takes place whenever translation is required. No hardware translation takes place.
USERTRANS	6	If translation is required, it is done by the user. Translation is required when the INTMODE value is not equal to the EXTMODE value and the TRANSLATING value is FALSE.

Port Files

Only the NOTRANS, USERTRANS, and FULLTRANS values are supported.

TRANSLATING

Kinds	All
Interrogate	When assigned
Modify	Never
Type	Boolean
Range	TRUE, FALSE
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	BNANATIVESERVICE

Interrogate the TRANSLATING attribute to determine if software translation is being performed on the records of the file. If the value is TRUE, software translation is being performed.

For files other than port files, the TRANSLATING value is determined during file assignment, taking into consideration the following attribute values: BLOCKSTRUCTURE or FILETYPE, EXTMODE, INTMODE, and TRANSLATE. Refer to the descriptions of these attributes for related information.

Port Files

The TRANSLATING value is determined during the matching process. Refer to the discussion of opening port files in the *I/O Subsystem Programming Guide*.

TRANSLATING is TRUE if the value of INTMODE is not equal to EXTMODE and the value of TRANSLATE is FULLTRANS. TRUE indicates that translation is being performed for the application. FALSE indicates that no translation is being performed, or that the correspondent endpoint is doing the translation.

TRANSMISSIONNO

Kinds	Remote
Interrogate	When open
Modify	Never
Type	Integer
Range	Not applicable
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	No
Parameters	1 required
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Interrogate the TRANSMISSIONNO attribute to determine the transmission number of the input last received by the program from the station specified by the relative station number (RSN) parameter. If the description of the station in the DATACOMINFO file does not require the data communications subsystem to assign transmission numbers to the messages of the station, the value returned is –1.

TRANSMISSIONO is a nonpreferred synonym for *TRANSMISSIONNO*.

TRANSMISSIONS

Kinds	Remote
Interrogate	When open
Modify	Never
Type	Integer
Range	Not applicable
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

Interrogate the TRANSMISSIONS attribute to determine the total number of output messages sent to all stations in the file or to a specific station. To obtain information for a specific station, include the relative station number (RSN) parameter for the station.

TRIMBLANKS

Kinds	Printer backup, or remote
Interrogate	Anytime
Modify	When closed
Type	Boolean
Range	TRUE, FALSE
Default	TRUE for printer backup FALSE for remote
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

Use the TRIMBLANKS attribute to determine if or specify that trailing blank characters are to be removed when the records of the file are written. Blank trailing characters are removed only when the file is not a direct I/O file and the values of the INTMODE and EXTMODE attributes for the file are EBCDIC.

When TRIMBLANKS is set to TRUE, blanks are removed and your program gains the following advantages:

- Printer backup files can be much smaller, and the I/O time of programs that produce the files can be considerably reduced.
- The data communication line time of programs that write to remote files can be reduced, especially when data communication lines are slow, or when the program writes one line at a time.

However, when TRIMBLANKS is set to TRUE, the processor time required to perform each WRITE operation is increased.

The following statements are true if you do not specify a value for the TRIMBLANKS attribute:

- If you interrogate the value while the file is closed, TRUE is returned.
- If the file is assigned to a remote file, or to a file for which removal of blanks is not valid, the value is changed to FALSE while the file is open.

The following statements are true if you specify a value for the TRIMBLANKS attribute:

- If you interrogate the value while the file is closed, the value you specified is returned.
- When the file is assigned, the value you specified is used, unless you have specified TRUE and the removal of blanks is not valid for the file that was assigned. In that case, FALSE is returned.

Printer Backup Files

If the value of TRIMBLANKS is TRUE, the trailing blank characters are not stored in the backup file.

When blank characters are removed from a print line, the characters are removed to the word boundary of the first word that is not all blank. The last partial EBCDIC word containing less than six blank characters does not have blank characters removed.

Remote Files

If the value of TRIMBLANKS is TRUE, the trailing blank characters of the output message are not transmitted or tanked.

If the file has a UNITS attribute value of CHARACTERS, all trailing blank characters are removed. If the file has a UNITS value of WORDS, all trailing blank words are removed. I/O statements of the ALGOL form *WRITE(F[STOP],...)* are not affected.

UNIQUETOKEN

Kinds	Disk
Interrogate	Anytime
Modify	When closed
Type	Character string
Range	Any displayable character
Default	Null string
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

When you are creating a disk file and you want the file name to change every time the file is opened, use the UNIQUETOKEN attribute to specify the displayable character that identifies the changeable part of the file name.

The displayable character must be followed by a period.

The FILENAME value must identify the system-supplied character strings that must be inserted into the file name before the file is assigned. This string is identified by a keyword preceded by the displayable character that is specified with the UNIQUETOKEN attribute. For valid keywords, refer to the FILENAME attribute.

Examples

The following ALGOL statement directs the system to create a file name with the current job number and mix number in the third and fourth nodes:

```
FILE
   ACCTSRECEIVABLE(KIND=DISK,
                       NEWFILE=TRUE,
                       UNIQUETOKEN="@.",

FILENAME="ACCOUNTS/RECEIVABLES/"""@JOBNO"""/"""@MIXNO""".",
                       .
                       .
                       .
```

The following ALGOL statement directs the system to create a file name with the Julian date and the time in the second node, and the job number and the task number in the fourth node:

```
FILE
   PAYROLL(KIND=DISK,
           NEWFILE=TRUE,
           UNIQUETOKEN="\.",

FILENAME="PAYROLL/"""TD\JDATETM\TIME"""/BPL/"""\JOBNO\TASKNO"""."",
           .
           .
           .
```

Interrogate the UNIQUETOKEN attribute to determine if the file name of the created disk file might contain system-supplied character strings.

UNITNO

Kinds	All except port and remote
Interrogate	Anytime
Modify	When unassigned
Type	Integer
Range	Not applicable
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

Interrogate the UNITNO attribute to determine the assigned hardware unit number or the hardware unit number that is about to be assigned. If the file is an assigned disk file or a printer file with a BACKUPKIND value of DISK or PACK, the UNITNO attribute returns the header index of the disk file header rather than the hardware unit number.

Tape files requiring additional reels cause the operating system to select reels on any available tape drive. Scratch tapes are selected automatically if the system option SERIALNUMBER (option 27) is not set.

Modify the UNITNO attribute to assign a unit number only before the logical file is assigned to a physical file. If the values of the KIND and FORMID attributes of the file are not the same as those of the assigned device or if the device is not ready, a waiting message is issued. For more information, refer to the FORMID attribute. If there is no physical unit with the UNITNO value, and the KIND value is PRINTER or TAPE, the UNITNO attribute has no effect. If the UNITNO attribute has been set to a value for a file with a KIND attribute value of DISK or PACK, an attribute error occurs. For a file with a KIND attribute value of CD, an attribute error does not occur.

Modifying the UNITNO attribute for disk files has no effect.

If a file is closed without retention, any specified UNITNO value is discarded.

If the system security option NONPRIVUNITNO value is NOTOK, assignment to the unit specified by the UNITNO value requires operator intervention if the task is not privileged.

UNITS

Kinds	All
Interrogate	When unassigned (Heritage Host Services logical I/O) Anytime (local host)
Modify	When closed
Type	Mnemonic
Range	See following text
Default	WORDS
Abstract Value	DEFAULT
Stored Permanently	Disk or tape
Parameters	None
FTAM File Access	Supported
Heritage Host Services Logical I/O	Restricted usage
Port Services	All

You can use the UNITS attribute to determine or specify whether or not the transfer of data in the file is to be word- or character-oriented. However, since the FRAMESIZE attribute overrides and can change the value of UNITS, the use of FRAMESIZE is preferred over the use of the UNITS attribute. The UNITS value does affect the FRAMESIZE value if a FRAMESIZE value is not specified.

The mnemonic values and meanings associated with the UNITS attribute are as follows:

Mnemonic Value	Integer Value	Meaning
CHARACTERS	1	INTMODE-defined units
WORDS	0	48-bit units

If the value of UNITS is CHARACTERS and the INTMODE value is a value other than SINGLE, the file is character-oriented.

Heritage Host Services logical I/O does not support interrogation of the UNITS attribute when the file is assigned. Use the FRAMESIZE attribute instead.

UPDATEFILE

Kinds	Disk
Interrogate	Anytime
Modify	When closed
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Restricted values
Port Services	Not applicable

Set the UPDATEFILE attribute to TRUE to specify that the disk file is to use the update I/O accessing method.

When the update I/O access method is in use, a serial, or nonrandom WRITE operation following a READ operation rewrites the record just read. When the update I/O access method is not used, a serial WRITE operation following a READ operation writes the record to the next record in the file.

If you specified an UPDATEFILE value of TRUE, the value is changed to FALSE when the logical file is assigned to a physical file that is not a disk file.

For the update I/O access method to operate efficiently, the default number of buffers is 3 when UPDATEFILE is TRUE. A smaller number of buffers would require the operating system I/O routines to wait for look-ahead READ operations to complete. For related information, refer to the description of the BUFFERS attribute.

For files with a FILESTRUCTURE attribute value of STREAM or BLOCKED, synchronous update I/O is not supported. If the UPDATEFILE value is TRUE, an open error occurs if the SYNCHRONIZE attribute value is OUT and a synchronous WRITE operation is treated as an asynchronous WRITE operation.

If the UPDATEFILE attribute is interrogated when the file is closed, the last value assigned is returned.

Heritage Host Services logical I/O supports only the FALSE value.

USECATALOG

Kinds	Disk or tape
Interrogate	Anytime
Modify	When closed
Type	Boolean
Range	TRUE, FALSE
Default	See following text
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

If the system uses cataloging, set the USECATALOG attribute to TRUE to indicate that one of the following tasks should be performed:

- Permanent disk or tape file searches should use the information in the system catalog.
- New files should be entered into the system catalog when they are created, and will be opened on volumed pack families and volumed tapes only.

The default value of USECATALOG is the value of the system option USECATDEFAULT if cataloging is enabled; otherwise, the default value is FALSE. For more information about setting system options, refer to the *System Operations Guide*.

For more information about cataloging, refer to the *System Administration Guide*.

USEDATE

USEDATE is a nonpreferred synonym for ACCESSDATE. Refer to the description of the ACCESSDATE attribute.

USERBACKUPNAME

Kinds	Printer
Interrogate	Anytime
Modify	When closed
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

Set the USERBACKUPNAME attribute to TRUE to override the system naming conventions for backup files. Additionally, use the FILENAME attribute to specify the name of the printer backup file, and the FAMILYNAME attribute to select the disk family on which the backup file is to be created. If you do not specify these values, the system sets these values as follows:

- If the FAMILYNAME attribute is not used, the backup file is created on the default backup family that was designated by the DL BACKUP form of the DL (Disk Location) system command.
- If the FILENAME attribute is not used, the name associated with the INTNAME attribute is used to name the printer backup file.

If USERBACKUPNAME is TRUE, the file name by default includes the usercode of the task creating the file. A backup file that includes a usercode in its name by default causes the SECURITYTYPE attribute value to change to PRIVATE.

Note: *If you use the TITLE attribute with the ON clause, the KIND value is changed to PACK even when the USERBACKUPNAME value is TRUE.*

Regardless of the USERBACKUPNAME value, the TITLE file attribute returns the title of the backup file when a printer file is open. The backup file title is returned whether the title was constructed by the system or supplied by the user.

USERCODE

Kinds	Disk
Interrogate	Anytime
Modify	When closed
Type	Character string
Range	Not applicable
Default	See following text
Abstract Value	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

When you are creating or accessing a File Transfer, Access, and Management (FTAM) file on a remote host, use the USERCODE attribute to pass remote log-on information to the remote host. The log-on information is made up of three optional pieces of information: the log-on usercode, the log-on password, and the log-on account information. Each piece of information is separated by a slash (/).

Each part of the log-on information can contain the following:

- Two quotation marks ('' '') to indicate that no item is being sent.
- Two apostrophes (' ') to indicate that a string with a length of 0 (zero) is being sent.
- An identifier that is from 1 to 249 characters long and contains any combination of EBCDIC characters other than an apostrophe or a character that has a hexadecimal code less than 4'40'.
  - To send an apostrophe to the remote host, the identifier must be enclosed in apostrophes. A single apostrophe must be represented by two apostrophes in a row.
  - To send lowercased characters, the identifier must be enclosed in apostrophes.
  - The enclosing apostrophes in each of these cases are removed before the characters are sent to the remote host.

  The following example sends only usercode information in uppercase characters:

  xyzcompany

  The following example sends all three pieces of information in uppercase characters:

  ours/mine/347

The following example sends only usercode and account information. The USERCODE characters are sent as uppercase characters and the account information is sent as lowercase characters with embedded apostrophes, (a'b'c'd).

ours/""/'a''b''c''d'

The following example sends only a password in lowercase letters and an account with a length of 0:

" "/'mine'/''

For more information about using the USERCODE attribute when using FTAM, refer to the *I/O Subsystem Programming Guide*.

For port subfiles, USERCODE is a nonpreferred synonym for YOURUSERCODE.

USERINFO

Kinds	Disk or CD
Interrogate	When assigned
Modify	When assigned and the program has write access to the file (disk)
Type	Word
Range	Not applicable
Default	0
Abstract Value	DEFAULT
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

Use the USERINFO attribute to save site- or application-specific information.

If the USERINFO attribute is interrogated, the value of the physical file is returned.

When USERINFO is modified, the timestamp of the disk file header is also updated.

The values that the USERINFO attribute returns are unaffected by library maintenance.

Note: *The USERINFO attribute can be used by Unisys released software or files created by Unisys software. Therefore, it is recommended that you do not attempt to modify this attribute on such software.*

USETIME

USETIME is a nonpreferred synonym for ACCESSTIME. Refer to the description of the ACCESSTIME attribute.

VERSION

Kinds	Disk or tape
Interrogate	Anytime
Modify	When not in directory (disk) When closed (tape)
Type	Integer
Range	0 through 99
Default	0
Abstract Value	DEFAULT, MINIMUM, MAXIMUM
Stored Permanently	Disk or tape
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Use the VERSION attribute, in conjunction with the CYCLE attribute, to distinguish one copy of a file with the same name from another within a given cycle.

If the VERSION attribute is interrogated when the file is assigned, the value for the physical file is returned.

Refer also to the description of the CYCLE attribute.

WARNINGS

Kinds	Disk
Interrogate	When assigned
Modify	When assigned
Type	Character string
Range	Not applicable
Default	Null string
Abstract Value	DEFAULT
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Interrogate the WARNINGS attribute to determine what warnings have been logged for a code file. The warning numbers are expressed as EBCDIC decimal numbers separated by commas (,). The last number is followed by a period (.).

The operating system modifies the WARNINGS value of a code file when the code file encounters run-time warnings. Compilers modify the WARNINGS value of a code file when the compilation encounters deimplementation warnings. With each modification, the new warning is added to the current list of warnings.

Modification of the WARNINGS attribute can happen only when the file is assigned for output or I/O and the program has write access to the file. If the program does not have write access, the program is terminated and a security error is issued.

Warning numbers range from 1 through 29,999, as defined in the following table:

Range	Purpose
1 through 9,999	Reserved for the operating system and compilers
10,000 through 24,999	Reserved for Unisys Corporation
25,000 through 29,999	Reserved for installation use

WIDTH

Kinds	Remote
Interrogate	When open
Modify	Never
Type	Integer
Range	Not applicable
Default	Not applicable
Abstract Value	Not applicable
Stored Permanently	No
Parameters	1 required
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Supported
Port Services	Not applicable

Interrogate the WIDTH attribute to determine the logical line length, in characters, for the station specified by the relative station number (RSN) parameter.

The following example shows how to interrogate the WIDTH attribute in ALGOL:

```
I := DCOM(RSN).WIDTH;
```

YOURDOMAINNAME

Kinds	Port subfile
Interrogate	Anytime
Modify	When closed
Type	Character string
Range	Valid domain name
Default	Null string
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	TCPIPNATIVESERVICE, TCPNATIVESERVICE

Use the YOURDOMAINNAME attribute to determine or specify a domain name. When a domain name is specified, that name indicates the host on which the correspondent endpoint exists. The correspondent host can be the local host.

The domain name can be 1 to 255 characters long. The string must be enclosed in quotation marks ('' ''), so that periods (.) can be used in the domain name.

A null string allows the subfile to match any host when the dialogue is established.

This attribute requires a subfile index as a parameter if the MAXSUBFILES value is greater than 1, except when it is used with a file declaration. If it is used in a file declaration or the subfile index specified is zero, then the value of YOURDOMAINNAME is changed in all allocated subfiles and it becomes the new default for any subfiles that are allocated later, even after increasing MAXSUBFILES.

ALGOL Example

```
REPLACE DOMAIN_NAME BY 48"7F","YOUR.DOMAIN.NAME",48"7F",".";

REPLACE TCPPORT.YOURDOMAINNAME BY DOMAIN_NAME;
```

YOURHOST

Kinds	All
Interrogate	Anytime
Modify	When closed
Type	Character string
Range	Not applicable
Default	MYHOST
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Supported
Heritage Host Services Logical I/O	Supported
Port Services	BNANATIVESERVICE, OSINATIVESERVICE, OSISESSIONSERVICE, TCPIPNATIVESERVICE, TCPNATIVESERVICE, BASICSERVICE

Use the YOURHOST attribute to specify the host name of the physical file or the correspondent host of the port subfile. The name can be 1 to 17 characters long and can contain uppercase alphanumeric characters.

Heritage Host Services Logical I/O Files

YOURHOST is a nonpreferred synonym for *HOSTNAME*. Heritage Host Services logical I/O does not support modification to the YOURHOST attribute using the FA (File Attribute) system command once the file is open. For information about using the FA command, refer to the *System Operations Guide*.

If the file is closed and unassigned and YOURHOST has never been set, YOURHOST returns the name of the local host. If the file is assigned, YOURHOST returns the name of the host on which the file resides.

Port Files

The value of the YOURHOST attribute identifies the host on which the correspondent application resides. The correspondent host can be the local host. In this case, the YOURHOST value for that subfile equals the MYHOST value. By default, the value of YOURHOST is the value of MYHOST unless the value of YOURHOSTGROUP is set, in which case YOURHOST is implicitly set to null.

A null value enables the subfile to match any host during dialogue establishment. If YOURHOST has a null value, an OPEN operation is allowed only if the port service is BNANATIVESERVICE; otherwise, the OPEN operation returns a BADATTRIBUTESRSLT (66) open error and a BADATTRIBUTESFOROPEN (13) SUBFILEERROR. The YOURHOST value can be a null string when an AWAITOPEN operation is invoked.

This attribute requires a subfile index as a parameter if the MAXSUBFILES value is greater than 1, except when it is used with a file declaration. If it is used in a file declaration or the subfile index specified is zero, then the value of YOURHOST is changed in all allocated subfiles and it becomes the new default for any subfiles that are allocated later, even after increasing MAXSUBFILES.

Specific service information follows:

Service	**Information**
BNANATIVESERVICE	When two subfiles on different hosts both have a null YOURHOST value, the subfiles can match only if the DONOTSEARCHNETWORK attribute is FALSE for at least one subfile, and Directory Services is running.
TCPIPNATIVESERVICE and TCPNATIVESERVICE	Local dialogue is supported.

YOURHOSTGROUP

Kinds	Port subfile
Interrogate	Anytime
Modify	When closed
Type	Character string
Range	Not applicable
Default	Null string
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	BNANATIVESERVICE

Use the YOURHOSTGROUP attribute to determine or specify the name of the host group in which the correspondent application resides. Setting the YOURHOSTGROUP attribute implicitly sets the value of YOURHOST to a null string.

This attribute requires a subfile index as a parameter if the MAXSUBFILES value is greater than 1, except when it is used with a file declaration. If it is used in a file declaration or the subfile index specified is zero, then the value of YOURHOSTGROUP is changed in all allocated subfiles and it becomes the new default for any subfiles that are allocated later, even after increasing MAXSUBFILES.

YOURIPADDRESS

Kinds	Port subfile
Interrogate	Anytime
Modify	When closed
Type	Character string
Range	Valid IP address
Default	Null string
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	TCPIPNATIVESERVICE, TCPNATIVESERVICE

Use the YOURIPADDRESS attribute to determine or specify the IP address. When an IP address is specified, the address is used during file assignment to indicate the host on which the correspondent end point exists.

The IP address can be from 1 to 15 characters long and must be enclosed in quotation marks ('' '').

If you specify a value of 127.0.0.1 for YOURIPADDRESS, a local port dialogue is opened. Another method of opening a local port is to specify one of the valid IP addresses for the local host.

This attribute requires a subfile index as a parameter if the MAXSUBFILES value is greater than 1, except when it is used with a file declaration. If it is used in a file declaration or the subfile index specified is zero, then the value of YOURIPADDRESS is changed in all allocated subfiles and it becomes the new default for any subfiles that are allocated later, even after increasing MAXSUBFILES.

ALGOL Example

```
REPLACE IP_ADDRESS BY 48"7F","192.9.200.4",48"7F",".";

REPLACE TCPPORT.YOURIPADDRESS BY IP_ADDRESS;
```

YOURNAME

Kinds	Port subfile
Interrogate	Anytime
Modify	When closed
Type	Character string
Range	Not applicable
Default	Null string
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	All

Use the YOURNAME attribute to specify the correspondent endpoint of the subfile. The YOURNAME attribute is a string of up to 100 characters and defaults to a null string.

If the YOURNAME value for the subfile is null, the subfile is allowed to match to any endpoint on the host specified by the YOURHOST attribute of the subfile. All port services allow a null YOURNAME value on AWAITOPEN operations. However, BNANATIVESERVICE is the only service that also allows a null YOURNAME value on an OPEN operation. When a process uses any other service, a null YOURNAME value on an OPEN operation results in a BADATTRIBUTESRSLT (66) open error and a BADATTRIBUTESFOROPEN (13) SUBFILEERROR.

This attribute requires a subfile index as a parameter if the MAXSUBFILES value is greater than 1, except when it is used with a file declaration. If it is used in a file declaration or the subfile index specified is zero, then the value of YOURNAME is changed in all allocated subfiles and it becomes the new default for any subfiles that are allocated later, even after increasing MAXSUBFILES.

For information about using the YOURNAME attribute with TCP files, refer to the PASSIVEOPEN attribute in this manual.

NETBIOSSESSIONSERVICE and TCPNATIVESERVICE impose limits on the range of values accepted for the YOURNAME attribute. Refer to the descriptions of these port services in the *I/O Subsystem Programming Guide*.

YOURNSAPA

Kinds	Port subfile
Interrogate	Anytime
Modify	When closed
Type	Character string
Range	Not applicable
Default	Null string
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	OSINATIVESERVICE, OSISESSIONSERVICE

Use the YOURNSAPA attribute to determine or specify the network service access point address of the network endpoint of the correspondent application. The length of the string can be 1 to 20 bytes.

The string of bytes can include non-EBCDIC characters such as all-zero bytes, EBCDIC quotation marks (HEX 7F), EBCDIC periods (HEX 4B), and other values. Enclose the string in quotation marks ('' '') if the string contains all-zero bytes, EBCDIC quotation marks, or EBCDIC periods; follow the last quotation mark with a period. If you want the string to contain quotation marks, you must enter two quotation marks for each quotation mark.

Refer to the *I/O Subsystem Programming Guide* for information about using the YOURNSAPA attribute with OSINATIVESERVICE and OSISESSIONSERVICE.

This attribute requires a subfile index as a parameter if the MAXSUBFILES value is greater than 1, except when it is used with a file declaration. If it is used in a file declaration or the subfile index specified is zero, then the value of YOURNSAPA is changed in all allocated subfiles and it becomes the new default for any subfiles that are allocated later, even after increasing MAXSUBFILES.

ALGOL Example

The following ALGOL code shows how to represent a string that contains a hex byte string of 00. The actual value of the attribute is the HEX string 2F0000F1010102D4D7C1F1F5C101.

```
BEGIN
FILE PORTFILE (KIND=PORT);
ARRAY TEMPATTR[0:3];
REPLACE TEMPATTR BY 48"7F2F0000F1010102D4D7C1F1F5C1017F4B";
REPLACE PORTFILE(1).YOURNSAPA BY POINTER (TEMPATTR[0]);
END.
```

YOURPRESENTATIONSEL

Kinds	Port subfile
Interrogate	Anytime
Modify	When closed
Type	Character string
Range	Not applicable
Default	Null string
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	OSINATIVESERVICE

Use the YOURPRESENTATIONSEL attribute to determine or specify the Presentation Selector of the network endpoint of the correspondent application. The length of the string can be 1 to 4 bytes.

The string of bytes can include non-EBCDIC characters such as all-zero bytes, EBCDIC quotation marks (HEX 7F), EBCDIC periods (HEX 4B), and other values. Enclose the string in quotation marks ('' '') if the string contains all-zero bytes, EBCDIC quotation marks, or EBCDIC periods; follow the last quotation mark with a period. If you want the string to contain quotation marks, you must enter two quotation marks for each quotation mark.

Refer to the *I/O Subsystem Programming Guide* for information about using the YOURPRESENTATIONSEL attribute with OSINATIVESERVICE and OSISESSIONSERVICE.

This attribute requires a subfile index as a parameter if the MAXSUBFILES value is greater than 1, except when it is used with a file declaration. If it is used in a file declaration or the subfile index specified is zero, then the value of DIALOGPRIORITY is changed in all allocated subfiles and it becomes the new default for any subfiles that are allocated later, even after increasing MAXSUBFILES.

ALGOL Example

The following ALGOL code shows how to represent a string that contains a hex byte string of 00. The actual value of the attribute is the HEX string 2F0000F1.

```
BEGIN
FILE PORTFILE (KIND=PORT);
ARRAY TEMPATTR[0:3];
REPLACE TEMPATTR BY 48"7F2F0000F17F4B";
REPLACE PORTFILE(1).YOURPRESENTATIONSEL BY POINTER (TEMPATTR[0]);
END.
```

YOURSESSIONSEL

Kinds	Port subfile
Interrogate	Anytime
Modify	When closed
Type	Character string
Range	Not applicable
Default	Null string
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	OSINATIVESERVICE, OSISESSIONSERVICE

Use the YOURSESSIONSEL attribute to determine or specify the Session Selector of the network endpoint of the correspondent application. The length of the string can be 1 to 16 bytes.

The string of bytes can include non-EBCDIC characters such as all-zero bytes, EBCDIC quotation marks (HEX 7F), EBCDIC periods (HEX 4B), and other values. Enclose the string in quotation marks ('' '') if the string contains all-zero bytes, EBCDIC quotation marks, or EBCDIC periods; follow the last quotation mark with a period. If you want the string to contain quotation marks, you must enter two quotation marks for each quotation mark.

Refer to the *I/O Subsystem Programming Guide* for information about using the YOURSESSIONSEL attribute with OSINATIVESERVICE and OSISESSIONSERVICE.

This attribute requires a subfile index as a parameter if the MAXSUBFILES value is greater than 1, except when it is used with a file declaration. If it is used in a file declaration or the subfile index specified is zero, then the value of YOURSESSIONSEL is changed in all allocated subfiles and it becomes the new default for any subfiles that are allocated later, even after increasing MAXSUBFILES.

ALGOL Example

The following ALGOL code shows how to represent a string that contains a hex byte string of 00. The actual value of the attribute is the HEX string 2F0000F1010102D4D7C1F1F5C101.

```
BEGIN
FILE PORTFILE (KIND=PORT);
ARRAY TEMPATTR[0:3];
REPLACE TEMPATTR BY 48"7F2F0000F1010102D4D7C1F1F5C1017F4B";
REPLACE PORTFILE(1).YOURSESSIONSEL BY POINTER (TEMPATTR[0]);
END.
```

YOURTRANSPORTSEL

Kinds	Port subfile
Interrogate	Anytime
Modify	When closed
Type	Character string
Range	Not applicable
Default	Null string
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	OSINATIVESERVICE, OSISESSIONSERVICE

Use the YOURTRANSPORTSEL attribute to determine or specify the Transport Selector of the network endpoint of the correspondent application. The length of the string can be 1 to 32 bytes.

The string of bytes can include non-EBCDIC characters such as all-zero bytes, EBCDIC quotation marks (HEX 7F), EBCDIC periods (HEX 4B), and other values. Enclose the string in quotation marks ('' '') if the string contains all-zero bytes, EBCDIC quotation marks, or EBCDIC periods; follow the last quotation mark with a period. If you want the string to contain quotation marks, you must enter two quotation marks for each quotation mark.

Refer to the *I/O Subsystem Programming Guide* for information about using the YOURTRANSPORTSEL attribute with OSINATIVESERVICE and OSISESSIONSERVICE.

This attribute requires a subfile index as a parameter if the MAXSUBFILES value is greater than 1, except when it is used with a file declaration. If it is used in a file declaration or the subfile index specified is zero, then the value of YOURTRANSPORTSEL is changed in all allocated subfiles and it becomes the new default for any subfiles that are allocated later, even after increasing MAXSUBFILES.

ALGOL Example

The following ALGOL code shows how to represent a string that contains a hex byte string of 00. The actual value of the attribute is the HEX string 2F0000F1010102D4D7C1F1F5C101.

```
BEGIN
FILE PORTFILE (KIND=PORT);
ARRAY TEMPATTR[0:3];
REPLACE TEMPATTR BY 48"7F2F0000F1010102D4D7C1F1F5C1017F4B";
REPLACE PORTFILE(1).YOURTRANSPORTSEL BY POINTER (TEMPATTR[0]);
END.
```

YOURUSERCODE

Kinds	Port subfile
Interrogate	Anytime
Modify	When closed
Type	Character string
Range	Not applicable
Default	Null string
Abstract Value	DEFAULT
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	BNANATIVESERVICE

When the SECURITYTYPE attribute value of the subfile is PRIVATE, use the YOURUSERCODE attribute to specify the usercode under which the correspondent endpoint of the subfile must be running. The value of YOURUSERCODE is a 1- to 17-character usercode. The default value of the YOURUSERCODE attribute for the subfile is the usercode of the task that opened that subfile.

When the value of SECURITYTYPE is PUBLIC, use the YOURUSERCODE attribute to determine the usercode of the corresponding task that declared the complementary subfile. This process enables you to develop a more flexible security mechanism. The YOURUSERCODE value reverts to its previous value when the subfile is closed.

If the value of the SECURITYTYPE attribute for the port file is PRIVATE, the usercode under which the correspondent task is running must equal the value of the YOURUSERCODE attribute of the subfile. The correspondent task is the task that is offering the correspondent endpoint. To ignore the use of usercodes when matching the subfile, set the SECURITYTYPE value to PUBLIC. Refer to the SECURITYTYPE attribute description and to the discussion of ports in the *I/O Subsystem Programming Guide*.

Port files with PRIVATE security do not match if one task has a usercode and the other task does not have a usercode unless the value of YOURUSERCODE is a null string.

This attribute requires a subfile index as a parameter if the MAXSUBFILES value is greater than 1, except when it is used with a file declaration. If it is used in a file declaration or the subfile index specified is zero, then the value of YOURUSERCODE is changed in all allocated subfiles and it becomes the new default for any subfiles that are allocated later, even after increasing MAXSUBFILES.

For files that are not port files, YOURUSERCODE is a nonpreferred synonym for USERCODE.

Section 6
User-Defined Disk File Attributes (UDDFAs)

This section explains how to use the user-defined disk file attributes (UDDFAs). UDDFAs have the following characteristics:

Kinds	Disk
Interrogate	When open
Modify	When assigned
Type	As specified by user
Range	See following text
Default	As specified by user
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
Heritage Host Services Logical I/O	Not supported
Port Services	Not applicable

UDDFAs are a class of attributes in which both the attribute name and its value or values are defined by the user. These attributes (also known as named attributes) are stored in the disk file header and can be modified or interrogated only when the file is assigned.

UDDFAs have no system-defined meaning; the operating system checks only for consistency of data types. Therefore, UDDFAs are used only as a means of communication between the creator of a file and its users.

UDDFAs are stored in the disk file header along with other file attributes. A maximum of 4800 bytes of UDDFAs can be stored, which include the control information stored with each UDDFA. An error is returned if an attempt is made to exceed this limit. For information about the format in which UDDFAs are stored in the header, refer to the discussion of disk file headers in the *I/O Subsystem Programming Guide*.

The MCPSUPPORT Library includes the entry points for UDDFAs. Entry points create, set or modify, interrogate, and destroy UDDFAs. In addition to using the interrogate function, you can see UDDFAs as they are stored in the header by using a GETSTATUS call. Refer to the Version 6 header information in the *I/O Subsystem Programming Guide* and to information on GETSTATUS in the *GETSTATUS/SETSTATUS Programming Reference Manual*. The four entry points or library procedures do not require security privileges for access, but they do require read or write access, as appropriate, to the file.

The library entry points are described as follows:

Library Entry Point	Description
NAMEDATTRIBUTECOPY	Copies all UDDFAs from one file to another. If the destination file already has one or more UDDFAs then the error “operation not allowed” is returned.
NAMEDATTRIBUTECOPYDIRECT	Similar to NAMEDATTRIBUTECOPY, however, NAMEDATTRIBUTECOPYDIRECT is used for direct files.
NAMEDATTRIBUTECREATE	Adds a UDDFA to a file and assigns a default value. The attribute must not already exist. To assign a value other than the default value, call this library entry point and then call the NAMEDATTRIBUTESET.
NAMEDATTRIBUTECREATEDIRECT	Adds a UDDFA to a direct file and assigns a default value. The attribute must not already exist. To assign a value other than the default value, call this library entry point and then call the NAMEDATTRIBUTESETDIRECT.
NAMEDATTRIBUTESET	Changes the value of a UDDFA. The attribute must already exist.
NAMEDATTRIBUTESETDIRECT	Changes the value of a UDDFA for a direct file. The attribute must already exist.
NAMEDATTRIBUTEINTERROGATE	Returns the value of a UDDFA. The attribute must already exist. The information is returned in the format that the UDDFA was created in. If no value was assigned to the UDDFA, the default value is returned if such a value exists; otherwise, the error “operation not allowed” is returned.
NAMEDATTRIBUTEINTERROGATEDIRECT	Returns the value of a UDDFA for a direct file. The attribute must already exist. The information is returned in the format in which the UDDFA was created. If no value was assigned to the UDDFA, the default value is returned if such a value exists; otherwise, the error “operation not allowed” is returned.
NAMEDATTRIBUTEDESTROY	Deletes a UDDFA. The attribute must already exist.
NAMEDATTRIBUTEDESTROYDIRECT	Deletes a UDDFA for a direct file. The attribute must already exist.

The parameters to each procedure include at least a file name and the attribute name. Some procedures also require an attribute data type, access privileges, and other parameters. These library entry points are real procedures with required parameters as shown in the following examples:

```
REAL PROCEDURE NAMEDATTRIBUTECOPY (FIN, FOUT);
     FILE FIN, FOUT;

REAL PROCEDURE NAMEDATTRIBUTECOPYDIRECT (FIN, FOUT);
     FILE FIN, FOUT;

REAL PROCEDURE NAMEDATTRIBUTECREATE (FYLE,NAME,TYPE,VALR,VALA);
     VALUE TYPE, VALR;
     FILE  FYLE;
     REAL  TYPE, VALR;
     ARRAY NAME[*], VALA[*];

REAL PROCEDURE NAMEDATTRIBUTESET (FYLE,NAME,TYPE,VALR,VALA);
     VALUE TYPE, VALR;
     FILE  FYLE;
     REAL  TYPE, VALR;
     ARRAY NAME[*], VALA[*];

REAL PROCEDURE NAMEDATTRIBUTEINTERROGATE(FYLE,NAME,TYPE,VALR,VALA);
     VALUE TYPE;
     FILE  FYLE;
     REAL  TYPE, VALR;
     ARRAY NAME[*], VALA[*];

REAL PROCEDURE NAMEDATTRIBUTEDESTROY (FYLE,NAME);
     FILE  FYLE;
     ARRAY NAME[*];

REAL PROCEDURE NAMEDATTRIBUTECOPY (FIN, FOUT);
     FILE FIN, FOUT;

REAL PROCEDURE NAMEDATTRIBUTECOPYDIRECT (FIN, FOUT);
     DIRECT FILE FIN, FOUT;

REAL PROCEDURE NAMEDATTRIBUTECREATEDIRECT (FYLE,NAME,TYPE,VALR,VALA);
     VALUE TYPE, VALR;
     DIRECT FILE  FYLE;
     REAL  TYPE, VALR;
     ARRAY NAME[*], VALA[*];

REAL PROCEDURE NAMEDATTRIBUTESETDIRECT (FYLE,NAME,TYPE,VALR,VALA);
     VALUE TYPE, VALR;
     DIRECT FILE  FYLE;
     REAL  TYPE, VALR;
     ARRAY NAME[*], VALA[*];
```

```
REAL PROCEDURE NAMEDATTRIBUTEINTERROGATEDIRECT(FYLE,NAME,TYPE,VALR,VALA);
     VALUE TYPE;
     DIRECT FILE  FYLE;
     REAL  TYPE, VALR;
     ARRAY NAME[*], VALA[*];

REAL PROCEDURE NAMEDATTRIBUTEDESTROYDIRECT (FYLE,NAME);
     DIRECT FILE  FYLE;
     ARRAY NAME[*];
```

The parameters are described as follows:

Parameter	Description
FIN	An open source disk file. This parameter is required.
FOUT	An open or assigned destination disk file. This parameter is required.
FYLE	An open or assigned disk file. This parameter is required.
NAME	A real array containing the attribute name. This array is processed as a string of bytes, with the name starting in the first byte and ending with a period (.). This parameter is required.
TYPE	A real value specifying the characteristics of the attribute—for example, data type, read or write access, or the default value. During attribute creation, the access type must be specified. The fields in the TYPE parameter are as follows:

Field	Name	Description
[8:2]	ReadAccess	Must be defined as 0 (always), 1 (once), or 2 (never). This restriction is applied on top of any file access restriction—for example, you cannot write to read-only files. This field is examined only at the time the UDDFA is created.
[6:2]	WriteAccess	Must be defined as 0 (always), 1 (once), or 2 (never). This restriction is applied on top of any file access restriction—for example, you cannot write to read-only files. This field is examined only at the time the UDDFA is created.
[4:1]	HasDefault	Must be defined as 1 to create a UDDFA with a default, or as 0 otherwise. This field is examined only at the time the UDDFA is created.

[3:4] DataType Specifies the type of data in the file. This parameter is stored from the time the attribute is created until it is destroyed. When a UDDFA is set or interrogated, the type of data must match the DataType value used when the UDDFA was created. The following are the DataType values:

Value	Meaning
0	Boolean values 0 or 1
1	Integer values 0 through 255 (stored in 8 bits, with no sign bit)
2	Integer values –32767 through 32767 (stored in 16 bits; field [15:1] is the sign bit)
3	48-bit value (word)
4	Array of 8-bit values (bytes) with 6 UDDFA array elements per word
5	Array of 16-bit values (longbytes) with 3 UDDFA array elements per word
6	Array of 48-bit values (words)
7 through 15	Reserved. Note that the internal representation of all DataTypes values, except 4 and 5, is transparent to the user. The user is encouraged to use the DataType value that uses the least number of bits and meets the value range requirement. For example, for a UDDFA whose maximum value will be 123, type 1 is recommended over type 2, because type 1 uses only 8 bits in the header. Refer to Table 6–1 for more information about the relationship of the TYPE, VALR, and VALA parameters

VALR A real value for the attribute (or the length if the attribute type is array). It is used for Boolean, 8-bit, 16-bit, or 48-bit (word) data. If the UDDFA type is array, array VALA is used to hold the data, and VALR is the length of this array (in UDDFA array elements).

VALA A real array, if the attribute is an array type. It is used for data types 4, 5, and 6.

The following table summarizes the contents of the VALR and VALA parameters when a specific TYPE parameter is selected and gives the size of information that is stored in the header:

TYPE	VALR	VALA	Stored in Header
0	Least significant bit is interpreted as a Boolean value; for example, VALR=1	Not used	1 bit
1	Integers from 0 to 255; for example, VALR=12	Not used	8 bits (unsigned)
2	Integers from –32767 to 32767; for example, VALR=--12345	Not used	16 bits including sign and magnitude
3	Real value; for example, VALR=4.5	Not used	48 bits
4	Number of values in the UDDFA; for example, to pass 3 values, VALR=3	Real array packed with 8-bit values; for example, to pass 9, 30, and 19, VALA[0]= 0 & 9 [47:8] & 30 [39:8] & 19 [31:8]	8 bits per value
5	Number of values in the UDDFA; for example, to pass 2 values, VALR=2	Real array packed with 16-bit values (high-order bit of each value is interpreted as sign); for example, to pass 299 and–12, VALA[0]= 0 & 0 [47:1] & 299 [46:15] & 1 [31:1] & 12 [30:15]	16 bits per value
6	Number of values in the UDDFA; for example, to pass 2 values, VALR=2	Real array; for example, to pass –2 and 5.9, VALA[0]=–2 and VALA[1]=5.9	48 bits per value

Any of the preceding procedures perform the requested function, if possible, and return a result value that consists of the following:

- Error flag (HasError = [0:1])
- Error specification (WhichError = [5:5])
- Status specification ([17:12])

The status specification reflects the status of the UDDFA, if it can be determined, such as “was written,” “read only,” or a null if the requested UDDFA could not be located.

If there is an error, the result value might help diagnose the problem because the status specification, if the UDDFA name has been matched, reflects information about the UDDFA stored in the header.

The fields for status specification, error specification, and error flag are described as follows:

Field	Name	Description
[17:1]	WasRead	Indicates that the UDDFA has been read since its creation.
[16:2]	ReadAccess	Indicates if read access is allowed. **Value** — **Meaning** 0 — Always 1 — Once 2 — Never
[14:1]	WasWritten	Indicates that the UDDFA has been changed since its creation.
[13:2]	WriteAccess	Indicates if write access is allowed. **Value** — **Meaning** 0 — Always 1 — Once 2 — Never
[11:1]	HasDefault	Indicates that the default value was specified when the UDDFA was created.
[10:4]	DataType	Specifies the type of data that was stored. Its values are the same as the values defined for the DataType field under the TYPE parameter described previously.
[06:1]		Reserved for future expansion.
[05:5]	WhichError	Shows which (if any) of the following errors occurred: **Value** — **Meaning** 0 — No error 1 — Unknown attribute 2 — File not open 3 — Operation not allowed 4 — Data type mismatch 5 — Out of space in the header 6 — Invalid name 7 — System error 8 — Value out of range (for example, > 255 for a type-1 attribute)

Field	Name	Description	
		9	VALA too small
		10	File not assigned
		11 through 31	Reserved
[00:1]	HasError	If any error occurs, the error flag equals 1 and the error specification equals a nonzero value (see WhichError); otherwise, the error flag equals 0 and the error specification equals 0. If an error has occurred, and a UDDFA exists for the attribute specified by NAME, the fields in the status specification hold data as it is stored in the header	

Example

The following example is an ALGOL example that demonstrates the use and manipulation of UDDFAs. Four UDDFAs are used as follows:

- A1 is a 16-bit UDDFA. It is created with a default, initialized with a value, and later interrogated.
- BLUE is a Boolean UDDFA created with a default and later initialized to a value.
- HHHH4 is a 48-bit (word) UDDFA created with a default and later initialized, interrogated, and destroyed.
- TEST1 is an array of 8-bit values. This UDDFA is created with a default and later initialized.

```
BEGIN
   REAL     A, TYPE, VALR, RESULT, EXPECTED;
   ARRAY    NAME[0:4], VALA[0:249];
   FILE F   (KIND=DISK,NEWFILE=TRUE);
   POINTER  P;
   LIBRARY  MCP(LIBACCESS=BYFUNCTION,FUNCTIONNAME="MCPSUPPORT");

 REAL PROCEDURE NAMEDATTRIBUTECREATE (F,NAME,TYPE,VALR,VALA);
   VALUE    TYPE, VALR;
   FILE     F;
   ARRAY    NAME[*], VALA[*];
   REAL     TYPE, VALR;
   LIBRARY  MCP;

 REAL PROCEDURE NAMEDATTRIBUTESET (F,NAME,TYPE,VALR,VALA);
   VALUE    TYPE, VALR;
   FILE     F;
   ARRAY    NAME[*], VALA[*];
   REAL     TYPE, VALR;
   LIBRARY  MCP;

 REAL PROCEDURE NAMEDATTRIBUTEINTERROGATE (F,NAME,TYPE,VALR,VALA);
   VALUE    TYPE;
   FILE     F;
   ARRAY    NAME[*], VALA[*];
   REAL     TYPE, VALR;
   LIBRARY  MCP;

 REAL PROCEDURE NAMEDATTRIBUTEDESTROY (F,NAME);
   FILE     F;
   ARRAY    NAME[*];
   LIBRARY  MCP;

 DEFINE
   DATA_TYPEF   = [3:4] #,       % of "TYPE" parameter
      BOOLEAN_TYPE =        0 #,
      BYTE_TYPE =           1 #, %  8-bit data
      LONGBYTE_TYPE =       2 #, % 16-bit data
      WORD_TYPE =           3 #, % 48-bit data
      BYTE_ARRAY_TYPE =     4 #,
      LONGBYTE_ARRAY_TYPE = 5 #,
      WORD_ARRAY_TYPE =     6 #,
   HAS_DEFAULTF = [4:1] #,       % of "TYPE" parameter
   READ_ACCESSF = [8:2] #,       % of "TYPE" parameter
   WRITE_ACCESSF= [6:2] #,       % of "TYPE" parameter
      ALWAYS =              0 #,
      ONCE =                1 #,
      NEVER =               2 #,
   CHECK_RSLT(RSLT,INT,VALR,EXP,NAME) =
             BEGIN
             STRING MSG;
             POINTER PN;
             INTEGER L;
             IF BOOLEAN(RSLT) THEN
                  BEGIN
                  DISPLAY("**** ERROR ****");
                  PROGRAMDUMP(FILES,ARRAYS);
                  END
             ELSE
                  IF INT          AND  % Check integrity of data
                     VALR NEQ EXP THEN % after interrogation.
                       BEGIN
                       PN := POINTER(NAME);
                       WHILE STRING(PN,1) NEQ "." DO
                            BEGIN
                            L := * + 1;
                            PN := * + 1;
                            END;
                       MSG:= "*** DATA ERROR *** "  !!
                             " attribute "          !!
                             STRING(POINTER(NAME),L)!!
                             " expected "           !!
                             STRING(EXPECTED,*)     !!
                             ", found "             !!
                             STRING(VALR,*);
                       DISPLAY(MSG);
                       END
                  ELSE;
             END   #;

    %========================
OPEN(F);

REPLACE POINTER(NAME) BY "BLUE.";
TYPE := 0 &  BOOLEAN_TYPE  DATA_TYPEF
          &  ALWAYS        READ_ACCESSF
          &  ALWAYS        WRITE_ACCESSF
          &  1             HAS_DEFAULTF;
VALR := 1;
RESULT := NAMEDATTRIBUTECREATE (F,NAME,TYPE,VALR,VALA);
CHECK_RSLT (RESULT,FALSE,VALR,EXPECTED,NAME);
```

```
REPLACE POINTER(NAME) BY "A1.";
TYPE := 0 &  LONGBYTE_TYPE  DATA_TYPEF
          &  ALWAYS         READ_ACCESSF
          &  ALWAYS         WRITE_ACCESSF
          &  1              HAS_DEFAULTF;
VALR := 12345;
RESULT := NAMEDATTRIBUTECREATE (F,NAME,TYPE,VALR,VALA);
CHECK_RSLT (RESULT,FALSE,VALR,EXPECTED,NAME);
VALR := -12345;
RESULT := NAMEDATTRIBUTESET (F,NAME,TYPE,VALR,VALA);
CHECK_RSLT (RESULT,FALSE,VALR,EXPECTED,NAME);
EXPECTED := VALR;
RESULT := NAMEDATTRIBUTEINTERROGATE (F,NAME,TYPE,VALR,VALA);
CHECK_RSLT (RESULT,TRUE,VALR,EXPECTED,NAME);

REPLACE POINTER(NAME) BY "BLUE.";
TYPE := 0 &  BOOLEAN_TYPE  DATA_TYPEF;
VALR := 0;
RESULT := NAMEDATTRIBUTESET (F,NAME,TYPE,VALR,VALA);
CHECK_RSLT (RESULT,FALSE,VALR,EXPECTED,NAME);

REPLACE POINTER(NAME) BY "HHHH4.";
TYPE := 0 &  WORD_TYPE  DATA_TYPEF
          &  ALWAYS     READ_ACCESSF
          &  ALWAYS     WRITE_ACCESSF
          &  1          HAS_DEFAULTF;
VALR := 12345;
RESULT := NAMEDATTRIBUTECREATE (F,NAME,TYPE,VALR,VALA);
CHECK_RSLT (RESULT,FALSE,VALR,EXPECTED,NAME);
EXPECTED := VALR;
RESULT := NAMEDATTRIBUTEINTERROGATE (F,NAME,TYPE,VALR,VALA);
CHECK_RSLT (RESULT,TRUE,VALR,EXPECTED,NAME);
TYPE := 0 &  WORD_TYPE  DATA_TYPEF;
VALR := 6789;
RESULT := NAMEDATTRIBUTESET (F,NAME,TYPE,VALR,VALA);
CHECK_RSLT (RESULT,FALSE,VALR,EXPECTED,NAME);
EXPECTED := VALR;
RESULT := NAMEDATTRIBUTEINTERROGATE (F,NAME,TYPE,VALR,VALA);
CHECK_RSLT (RESULT,TRUE,VALR,EXPECTED,NAME);
RESULT := NAMEDATTRIBUTEDESTROY (F,NAME);
CHECK_RSLT (RESULT,FALSE,VALR,EXPECTED,NAME);
```

```
   REPLACE POINTER(NAME) BY "TEST1.";
   TYPE := 0 &  BYTE_ARRAY_TYPE  DATA_TYPEF
             &  ALWAYS           READ_ACCESSF
             &  ALWAYS           WRITE_ACCESSF
             &  1                HAS_DEFAULTF;
   VALR := 15;
   P := POINTER(VALA);
   FOR A := 14 STEP -1 UNTIL 0 DO
        REPLACE P:P BY A.[7:48] FOR 1;
   RESULT := NAMEDATTRIBUTECREATE (F,NAME,TYPE,VALR,VALA);
   CHECK_RSLT (RESULT,FALSE,VALR,EXPECTED,NAME);
   TYPE := 0 &  BYTE_ARRAY_TYPE  DATA_TYPEF;
   FOR A := 44 STEP -3 UNTIL 0 DO
        REPLACE P:P BY A.[7:48] FOR 1;
   RESULT := NAMEDATTRIBUTESET (F,NAME,TYPE,VALR,VALA);
   CHECK_RSLT (RESULT,FALSE,VALR,EXPECTED,NAME);

   LOCK(F);
END.
```

Section 7
Direct I/O Buffer Attributes

Buffer attributes are provided to aid you in controlling a direct I/O operation and to describe the results of the operation. Buffer attributes are accessible in ALGOL and COBOL.

This section presents buffer attribute descriptions alphabetically. Each buffer attribute description begins with a formatted list of attribute characteristics, followed by a narrative discussion of the use of the buffer attribute.

The following characteristics are identified in the formatted list:

Characteristic	**Meaning**
Interrogate	Indicates whether the value of the attribute can be read, and, if so, under what restrictions.
Modify	Indicates whether the value of the attribute can be modified and, if so, under what restrictions.
Type	Indicates the data type required for the attribute.
Range	Indicates a list of possible values associated with the attribute. These values vary according to the type of the given attribute.
Default	Indicates the value that the system uses when the user does not specify a value for the attribute. Depending on the attribute, the entry for this item is usually either a value that is a member of the defined range for that attribute or *Not applicable*. If no default exists for the attribute, the entry for this item is *No default*.

The following tables describe the values that are used with the preceding characteristics:

Value	Meaning
Never	The attribute can never be interrogated. In other words, it is a write-only attribute.
When closed	The file must be closed in order for the attribute to be interrogated.
When open	The file must be open in order for the attribute to be interrogated.
When assigned	The logical file must be assigned to a physical file in order for the attribute to be interrogated.
When unassigned	The logical file must not be assigned to any physical file in order for the attribute to be interrogated.
Anytime	The attribute can be interrogated under any circumstances.

Value	Meaning
Never	The attribute can never be modified. In other words, it is a read-only attribute.
When closed	The file must be closed in order for the attribute to be modified.
When open	The file must be open in order for the attribute to be modified.
When assigned	The logical file must be assigned to a physical file for the attribute to be modified.
When unassigned	The logical file must not be assigned to any physical file for the attribute to be modified.
Anytime	The attribute can be modified under any circumstances.

Value	Meaning
Boolean	Boolean-valued
Integer	Integer-valued
Word	48 bits of information as described in the explanation of the attribute

HYCOMMAND

Interrogate	Anytime
Modify	Anytime
Type	Integer
Range	See following text
Default	Not applicable

If you are using an HY file, modify the HYCOMMAND buffer attribute with the following values to indicate the action to be taken on a direct array buffer:

Value	Meaning
READ DATA (4'16')	The direct array buffer is read.
WRITE DATA (4'17')	The direct array buffer is written.
–1	The HYCOMMAND has not been initialized. The action taken is dependent on the I/O operation. READ DATA is used if a READ operation is invoked, and WRITE DATA is used if a WRITE operation is invoked.

Once the HYCOMMAND attribute is set for any given direct array buffer, it does not require subsequent changes if the selected operation does not change.

If a READ or WRITE statement is invoked for a direct array with the HYCOMMAND attribute set to the opposite I/O operation, an IOERRORTYPE value of 4 is issued and the requested I/O operation is not initiated.

This buffer attribute can be used with a HYPERchannel file. For more information about using HYPERchannel files, refer to the *I/O Subsystem Programming Guide*.

HYMPLENGTH

Interrogate	Anytime
Modify	Anytime
Type	Integer
Range	See following text
Default	12 (write only)

If you are using an HY file and are writing a message proper, modify the HYMPLENGTH attribute to indicate the length in bytes of the message proper. If you assign the value of 0 (zero), 12 bytes are written.

Interrogate the HYMPLENGTH attribute after a READ operation to determine the length in bytes of the received message proper.

This buffer attribute can be used with a HYPERchannel file. For more information about using HYPERchannel files, refer to the *I/O Subsystem Programming Guide*.

IOCANCEL

Interrogate	Anytime
Modify	Anytime
Type	Boolean
Range	TRUE, FALSE
Default	Not applicable

Interrogate the IOCANCEL attribute to determine if the last I/O operation attempted on the current buffer was canceled. If TRUE is returned, the I/O operation was canceled and the IORESULT attribute returns the value 1.

If you set IOCANCEL to TRUE for a buffer while an I/O operation is in process for that buffer, the following two actions are taken:

- The I/O queue for the device is purged.
- All the outstanding I/O operations are canceled that were initiated using direct arrays that were declared in the same stack as the direct array for which the IOCANCEL is being specified.

The IOCANCEL attribute for a disk file is supported on IOM systems only (for example, A14, A18, and A19).

The following actions outline the effect of setting the IOCANCEL attribute to TRUE for a disk file buffer while an I/O is in process for that buffer:

- If the IOCANCEL attribute is set for a direct I/O request issued to a nonmirrored unit
  - The unit queue affected by the I/O request is suspended and recalled.
  - The recalled queue is searched for “qualifying user-cancel masked I/O requests.” By definition, the IOCANCEL target I/O request always qualifies. By qualifying, the I/O request is returned and its IORESULT attribute is set to the User Cancel Result.
- If the IOCANCEL attribute is set for a direct I/O request issued to a mirrored unit
  - For each member of the mirrored set, the unit queue is suspended and recalled.
  - Each recalled queue is searched for “qualifying user-cancel masked I/O requests.” By definition, the IOCANCEL target I/O request always qualifies.

  If an online set member is determined to have no qualifying I/O control block (IOCB) queued, online members with a qualifying IOCB queued are decommitted to audit.

  If each on the online members has a qualifying IOCB queued, only the copy determined to be best will remain online; the remaining copies are decommitted to audit. A copy is determined best if its topmost qualifying IOCB has been queued for the shortest amount of time.

Among the qualifying user operations, a mirrored WRITE operation is returned successfully if completion occurs without exception to any set member remaining online. Uncompleted qualifying mirror READ operations, and qualifying mirror WRITE operations that do not complete successfully to any member left online are returned with a User Cancel Result.

To restore set integrity as soon as possible, the MCP immediately begins to apply audited WRITE operations to the decommitted members. The MCP uses a visible MIRROR_CREATE process for each member being restored.

- For each member of the mirrored set, the unit queue is restarted.

Once processing of the attribute is completed, I/O activity resumes on the affected logical unit, with the exception of any mirror that was placed in the offline mirror state. I/O activity to these mirrors is automatically restarted after the audit is applied by the MCP.

Note: *If you interrogate IOCANCEL while an I/O operation is in process, an invalid result could be returned.*

This buffer attribute can be used with a HYPERchannel file.

For information on the I/O timer handling feature, refer to the *I/O Subsystem Programming Guide*.

IOCHARACTERS

Interrogate	Anytime
Modify	Never
Type	Integer
Range	Not applicable
Default	Not applicable

Interrogate the IOCHARACTERS attribute to determine the number of characters read into the current buffer. The value returned after a WRITE operation is undefined.

If you interrogate IOCHARACTERS after a forward READ operation on a tape file, the value returned indicates the actual size of the tape block. For SCSI devices that do not support reverse READ operations, interrogating the IOCHARACTERS after a forward READ operation returns the desired results.

Note: *If you interrogate this attribute while an I/O operation is in process, an invalid result could be returned.*

This buffer attribute can be interrogated for a HYPERchannel file.

IOCOMPLETE

Interrogate	Anytime
Modify	Never
Type	Boolean
Range	TRUE, FALSE
Default	Not applicable

Interrogate the IOCOMPLETE attribute to determine if the I/O action for the current buffer is complete. A value of TRUE indicates the I/O action is complete.

This buffer attribute can be interrogated for a HYPERchannel file.

IOCW

Interrogate	Anytime
Modify	Anytime
Type	Word
Range	Valid IOCWs
Default	Device-dependent

Use the IOCW attribute to determine or specify the I/O control word (IOCW) for the buffer. Specify a value of 0 (zero) to change the IOCW value back to the system-supplied default.

Modifying the DIRECTION attribute can affect the value of IOCW.

The IOCW attribute is ignored for host control files.

This buffer attribute cannot be used with a HYPERchannel file.

Note: *You can no longer set field [35:12] in the IOCW. If your program previously set this field to specify the space or skip values for a printer device, your program should be modified to provide carriage control information, such as SPACE, SKIP, and STACKER through an I/O statement.*

IOEOF

Interrogate	Anytime
Modify	Never
Type	Boolean
Range	TRUE, FALSE
Default	Not applicable

Interrogate the IOEOF attribute to determine if an end-of-file (EOF) condition was encountered as a result of the last I/O operation on the current buffer. A value of TRUE indicates the EOF condition has been reached.

Note: *If you interrogate this attribute while an I/O operation is in process, an invalid result could be returned.*

This buffer attribute can be interrogated for a HYPERchannel file, but TRUE is never returned.

IOERRORTYPE

Interrogate	Anytime
Modify	Never
Type	Integer
Range	See following text
Default	Not applicable

Interrogate the IOERRORTYPE attribute to identify the error, if any, that occurred as a result of the last I/O operation on this buffer. If IOPENDING is TRUE, which indicates that the I/O operation has not completed, the value of IOERRORTYPE is undefined. The mnemonic and integer values and their meanings are described in Appendix F, "I/O Result Enumerated Values."

An end-of-tape condition, which occurs when the tape is positioned beyond the end-of-tape marker, is returned by the IORESULT and IOERRORTYPE attributes. End-of-tape conditions are system-specific, as follows:

- On Gamma and Delta systems, end-of-tape conditions are returned when they are encountered on tape READ operations only if the appropriate bit is set in the IOMASK attribute; otherwise, they are not reported.
- On A7, A2100, and NX4200 systems, end-of-tape conditions are always returned when they are encountered on tape READ operations.

On all systems, if an end-of-file (EOF) condition occurs in conjunction with an end-of-tape condition, IOERRORTYPE reports only the EOF condition.

This buffer attribute can be interrogated for a HYPERchannel file and the IOERRORTYPE values can be found in the discussion of a HYPERchannel network in the *I/O Subsystem Programming Guide*.

IOMASK

Interrogate	Anytime
Modify	Anytime
Type	Word
Range	Not applicable
Default	0

Use the IOMASK attribute to determine or modify the contents of the I/O Error Mask field of the I/O control block (IOCB). The IOMASK attribute allows you to specify the error conditions that the system is to overlook. When an I/O exception is masked out, the I/O subsystem does not initiate recovery or logging procedures and assumes that your program will handle these errors programmatically. Once the I/O Error Mask Field is modified, it is used instead of the system-supplied default. The I/O Error Mask Field reverts to the system-supplied default when the IOMASK attribute is modified to the value 0 (zero).

On all systems, you can set bit 9 to a value of 1 to suppress the normal system end-of-file (EOF) action. This action is particularly valuable for tape files because suppression of the EOF action of the system circumvents automatic volume switching.

On Gamma and Delta systems, when you set certain bits of the IOMASK attribute, certain conditions that are otherwise suppressed are reported through the IORESULT attribute. For example, if you set bit 8 to a value of 1, end-of-tape conditions are reported through the IORESULT attribute when they are encountered on tape READ operations.

Due to the fact that disk units are shared, and can be critical units for the operating system, direct I/O masking on disk units is limited to the behaviors documented in the following paragraph.

On resource management module (RMM) and I/O module (IOM) systems, if you set bit 3 and an I/O operation is directed to a disk that is not ready, the I/O system reports the condition through the IORESULT attribute. The following bits of the IORESULT attribute are set: 7 (Parity Error), 3 (Not Ready), 2 (I/O Canceled), and 0 (Exception).

On IOM systems, if you set bit 13 and a READ or WRITE operation is directed to a disk, the MCP is instructed to user-cancel the I/O in the event that another direct I/O buffer sets the IOCANCEL attribute. In order for the masked I/O to be returned as user-canceled, the following conditions must apply:

- The direct I/O buffer for the IOCANCEL target I/O and the direct I/O buffer for the masked I/O must have been declared by the same (owner) stack.
- Both I/Os must be in process to the same disk unit.
- The masked I/O request cannot have been previously completed.

When these conditions apply, the user cancel masked I/O will be returned to its requester with a User Cancel logical result. The following IORESULT attribute bits will be set:

- 13 (User Cancel)
- 2 (Cancel)
- 1 (Exception)

The meaning of the mask bits is documented in the IORESULT attribute logical result descriptor error field [16:17].

The IOMASK attribute is ignored for host control files.

This buffer attribute cannot be used with a HYPERchannel file.

For information on the I/O timer handling feature, refer to the *I/O Subsystem Programming Guide*.

IOPENDING

Interrogate	Anytime
Modify	Never
Type	Boolean
Range	TRUE, FALSE
Default	Not applicable

Interrogate the IOPENDING attribute to determine if an I/O operation is queued or in process on this buffer. A value of TRUE indicates an I/O operation is queued or in process.

This buffer attribute can be interrogated for a HYPERchannel file.

IORECORDNUM

Interrogate	Anytime
Modify	Never
Type	Integer
Range	Not applicable
Default	Not applicable

Interrogate the IORECORDNUM attribute to determine the logical record number of the last disk I/O operation performed on this buffer or the value of the LASTSUBFILE attribute for direct I/O remote files. If the FILESTRUCTURE value is STREAM or the DIOFILESTRUCTURE value is SECTORSTREAM, the sector number is returned.

This buffer attribute can be interrogated for a HYPERchannel file, but 0 (zero) is always returned.

IORESULT

Interrogate	Anytime
Modify	Never
Type	Word
Range	Not applicable
Default	Not applicable

Interrogate the IORESULT attribute to determine the logical result for the last I/O operation on this buffer. The logical result is described in the following text.

Note: *If you interrogate this attribute while an I/O operation is in process, an invalid result could be returned.*

The format of the logical result descriptor returned by the IORESULT attribute is the same as that returned by a WAIT statement on the buffer, except as described in field [47:20]. The following table defines the logical result descriptor error fields:

Field	Description
[47:20]	Number of units transferred. This value is in characters for the IORESULT attribute, but in words when the logical result descriptor is returned by a WAIT statement on the buffer.
[27:11]	For host control files: • [27:04] Hubindex of the partner • [23:07] Unused For other files: • [27:01] Unused • [26:10] Result enumeration. See Appendix F for a further description.
[16:17]	Logical result descriptor error field.
[16:01]	Unused.
[15:01]	Timed out.
[14:01]	Unit in rewind. HYPERchannel remote ABORT operation.
[13:01]	Break on output (for remote files). Host control exception was “partnered.” HYPERchannel message in queue. User-cancel (for disk files).
[12:01]	HYPERchannel command rejected. Image printer page printed.

Field	Description
[11:01]	Card reader control card READ operation. HYPERchannel channel check. HYPERchannel A223 deadman timer (RUNTIMER).
[10:01]	Short block READ operation. Direct disk I/O is truncated at the area boundary. HYPERchannel command retry.
[09:01]	End-of-file (EOF) or end-of-page.
[08:01]	End-of-tape. HYPERchannel ABORT operation.
[07:01]	Parity error detected.
[06:01]	WRITE operation lockout.
[05:01]	Undefined error.
[04:01]	Descriptor error.
[03:01]	Not ready.
[02:01]	I/O was canceled.
[01:01]	Attention.
[00:01]	Exception. Implicit OPEN operation failed.

This buffer attribute can be interrogated for a HYPERchannel file and a comprehensive description of the IORESULT values for HYPERchannel operations is included in the discussion of a HYPERchannel network in the *I/O Subsystem Programming Guide*.

An end-of-tape (EOT) condition, which occurs when the tape is positioned beyond the end-of-tape marker, is returned by the IORESULT and IOERRORTYPE attributes. End-of-tape conditions are system-specific, as follows:

- On Gamma and Delta systems, end-of-tape conditions are returned when they are encountered on tape READ operations only if the appropriate bit is set in the IOMASK attribute; otherwise, they are not reported.
- On A7, A2100, and NX4200 systems, end-of-tape conditions are always returned when they are encountered on tape READ operations.

On all systems, if an end-of-file (EOF) condition occurs in conjunction with an end-of-tape condition, IOERRORTYPE returns only the end-of-file condition.

For host control file READ operations that complete without errors or have IOERRORTYPE results of PARITYERRORTYPE (2), SHORTRECORD (9), or WRONGHUBINDEX (21)—that is, READ operations in which data transfer occurs---field [27:4] contains the hubindex from which the data was written. Otherwise, field [24:8] of the IORESULT attribute is 0 (zero).

The User Cancel bit is set for a disk file when an I/O is returned as a result of a direct I/O buffer setting its IOCANCEL attribute. The following IORESULT attribute bits are set:

- 13 (User Cancel)
- 2 (Cancel)
- 1 (Exception)

This result applies to the IOCANCEL target I/O and any qualifying user-cancel masked I/O.

For information on the I/O timer handling feature, refer to the *I/O Subsystem Programming Guide*.

IOTIME

Interrogate	Anytime
Modify	Never
Type	Integer
Range	Not applicable
Default	Not applicable

Interrogate the IOTIME attribute to determine the time, in 2.4-microsecond units, for the last I/O operation on the current buffer. The I/O time is a measured value that is used as the basis for system I/O accounting. Its precise definition is system-specific and peripheral-dependent.

Note: *If you interrogate this attribute while an I/O operation is in process, an invalid result could be returned.*

This buffer attribute can be interrogated for a HYPERchannel file.

IOWORDS

Interrogate	Anytime
Modify	Never
Type	Integer
Range	Not applicable
Default	Not applicable

Interrogate the IOWORDS attribute to determine the number of words read into this buffer. The value returned is inclusive; that is, if the record is *n* words long plus *m* characters, where *m* is greater than 0 (zero), IOWORDS returns the value *n + 1*. The value returned after a WRITE operation is undefined.

If you interrogate IOWORDS after a forward READ operation on a tape file, the value returned indicates the actual size of the tape block. For SCSI devices that do not support reverse READ operations, interrogating the IOWORDS after a forward READ operation returns the desired results.

Note: *If you interrogate this attribute while an I/O operation is in process, an invalid result could be returned.*

This buffer attribute can be interrogated for a HYPERchannel file.

OVERLAYABLE

Interrogate	Anytime
Modify	Anytime
Type	Boolean
Range	TRUE, FALSE
Default	FALSE

Set the OVERLAYABLE attribute to TRUE to indicate that the direct array can be overlaid.

This buffer attribute can be used with a HYPERchannel file.

READPARTNER

Interrogate	Anytime
Modify	Anytime
Type	Integer
Range	Not applicable
Default	–1

If the KIND value of the file is HC, use the READPARTNER attribute to determine or specify the hubindex from which the input is expected during host control READ operations.

Because 0 (zero) is a valid READPARTNER specification, a value of –1 is used to indicate an unspecified I/O partner.

A READPARTNER should be specified at the direct array level only for operations on host control files that do not have installation-imposed I/O partner restrictions in the HUBMAP. If a program specifies READPARTNER and attempts an I/O operation to an HC that has a READPARTNER specified at the HUBMAP level, a BAD IOPARTNER SPECIFICATION (IOERRORTYPE 19) error results.

This buffer attribute can be used with HC files. For more information about HC files, refer to the *I/O Subsystem Programming Guide*.

WRITEPARTNER

Interrogate	Anytime
Modify	Anytime
Type	Integer
Range	Not applicable
Range	Not applicable
Default	–1

If the KIND value of a file is HC, use the WRITEPARTNER attribute to determine or specify the target hubindex to which host control WRITE operations are directed.

Because 0 (zero) is a valid WRITEPARTNER specification, a value of –1 is used to indicate an unspecified I/O partner.

A WRITEPARTNER must be specified at the direct array level only for operations on host control files that do not have an installation-specified WRITEPARTNER in the HUBMAP. If a program specifies WRITEPARTNER and attempts an I/O operation to an HC that has a WRITEPARTNER specified at the HUBMAP level, a BAD IOPARTNER SPECIFICATION (IOERRORTYPE 19) error results. If WRITEPARTNER for a host control file is not specified at the HUBMAP level, an attempt to write with a direct array that does not have WRITEPARTNER specified results in a NO WRITE PARTNER SPECIFICATION (IOERRORTYPE 20) error.

This buffer attribute can be used with HC files. For more information about HC files, refer to the *I/O Subsystem Programming Guide*.

Appendix A
Device Types and Associated File Attributes

Use the following table to identify the general file attributes that can be used with each device type:

Attribute	Disk	CD	Printer	Remote	Tape	Other	Port
ACCESSDATE	X	X					
ACCESSDATEUT	X						
ACCESSTIME	X	X					
ACCESSTIMEUT	X						
ACCESSTZ	X						
ACTUALMAXRECSIZE							X
ADAPTABLE	X						
AFTER			X				
ALIGNFILE			X				
ALIGNMENT			X				
ALLOWSPECIALFILE	X						
ALTERDATE	X	X					
ALTERDATEUT	X						
ALTERNATEGROUPS	X						
ALTERTIME	X	X					
ALTERTIMEUT	X						
ALTERTZ	X						
ANYSIZEIO	X	X					
APL	X	X					
APPEND	X						
APPLICATIONCONTEXT							X
APPLICATIONGROUP							X
AREAADDRESS	X	X					

Attribute	Disk	CD	Printer	Remote	Tape	Other	Port
AREAALLOCATED	X	X					
AREALENGTH	X	X					
AREAS	X	X					
AREASECTORS	X	X					
AREASINUSE	X	X					
AREASIZE	X	X					
ASSIGNTIME				X			
ATTERR	X	X	X	X	X	X	X
ATTMODIFYDATE	X						
ATTMODIFYDATEUT	X						
ATTMODIFYTIME	X						
ATTMODIFYTIMEUT	X						
ATTMODIFYTZ	X						
ATTVALUE	X	X	X	X	X	X	X
ATTYPE	X	X	X	X	X	X	X
AUTOUNLOAD					X		
AVAILABLE	X	X	X	X	X	X	X
AVAILABLEONLY							X
BACKUPDATE	X						
BACKUPDATEUT	X						
BACKUPKIND			X				
BACKUPTIME	X						
BACKUPTIMEUT	X						
BACKUPTZ	X						
BANNER			X				
BLANK	X	X		X	X	X	X
BLOCK	X		X		X	X	
BLOCKEDTIMEOUT							X
BLOCKSIZE	X		X	X	X	X	
BLOCKSTRUCTURE	X	X	X	X	X	X	X
BUFFERS	X	X	X	X	X	X	X
BUFFERSHARING	X						
BUFFERSIZE	X	X	X		X	X	

Attribute	Disk	CD	Printer	Remote	Tape	Other	Port
CARRIAGECONTROL			X				
CCSVERSION	X						
CDCOPIES		X					
CENSUS				X			X
CHANGEDSUBFILE							X
CHANGEEVENT							X
CHECKPOINT			X				
CLEARAREAS	X						
COMPRESSING					X		X
COMPRESSION							X
COMPRESSIONCONTROL					X		X
COMPRESSIONREQUESTED					X		X
COPYDESTDATE	X						
COPYDESTDATEUT	X						
COPYDESTTIME	X						
COPYDESTTIMEUT	X						
COPYDESTTZ	X						
COPYSOURCEDATE	X						
COPYSOURCEDATEUT	X						
COPYSOURCETIME	X						
COPYSOURCETIMEUT	X						
COPYSOURCETZ	X						
CREATEPASSWORD	X						
CREATIONDATE	X	X			X		
CREATIONDATEUT	X						
CREATIONTIME	X	X					
CREATIONTIMEUT	X						
CREATIONTZ	X						
CRUNCHED	X	X					
CURRENTBLOCKLENGTH	X		X	X	X	X	
CURRENTRECORDLENGTH	X	X	X	X	X	X	X
CYCLE	X				X		
DENSITY					X		

Attribute	Disk	CD	Printer	Remote	Tape	Other	Port
DEPENDENTINTMODE	X	X	X	X	X	X	
DEPENDENTSPECS	X	X	X	X	X	X	
DESTINATION			X				
DIALOGCHECKINTERVAL							X
DIALOGPRIORITY							X
DIOFILESTRUCTURE	X	X					
DIRECTION					X		
DISPOSITION				X			
DOCUMENTTYPE	X	X					
DONOTSEARCHNETWORK							X
DONTCLEARAREASBYDEFAULT	X						
DUMMYFILE	X	X	X		X	X	
ENABLEINPUT				X			
EOF					X		
ESTIMATEDRECORDS	X		X				
EXCLUSIVE	X						
EXECUTEDATE	X						
EXECUTEDATEUT	X						
EXECUTETIME	X						
EXECUTETIMEUT	X						
EXECUTETZ	X						
EXTDELIMITER	X	X	X	X	X	X	X
EXTMODE	X	X	X	X	X	X	X
FAMILYINDEX	X	X					
FAMILYNAME	X	X	X				
FAMILYOWNER					X		
FILECLASS	X	X	X	X	X	X	X
FILEEQUATED	X	X	X	X	X	X	X
FILEKIND	X	X			X		
FILELENGTH	X	X					
FILENAME	X	X	X	X	X	X	X
FILEORGANIZATION	X	X					
FILESECTION					X		

Attribute	Disk	CD	Printer	Remote	Tape	Other	Port
FILESTATE	X	X	X	X	X	X	X
FILESTRUCTURE	X	X	X				
FILETYPE	X	X	X	X	X	X	X
FILEUSE	X	X	X	X	X	X	
FLEXIBLE	X						
FORMID			X				
FRAMESIZE	X	X	X	X	X	X	X
FRAMESIZECENSUS							X
GENERATION	X				X		
GROUP	X				X		
HOSTNAME	X	X	X	X	X	X	X
HSFILECOPY	X						
INPUTEVENT				X			X
INPUTTABLE	X	X	X	X	X	X	
INTERACTIVEFILE	X	X	X	X	X	X	X
INTMODE	X	X	X	X	X	X	X
INTNAME	X	X	X	X	X	X	X
IOCLOCKS	X	X	X	X	X	X	X
IOHFUNCTIONNAME						X	
IOHINTERFACENAME						X	
IOHLIBACESS						X	
IOHLIBPARAMETER						X	
IOHPREFIX						X	
IOHSTRING						X	
IOHTITLE						X	
IOINERROR	X	X			X		X
IPTYPEOFSERVICE							
KERBEROSACCESS	X						
KIND	X	X	X	X	X	X	X
LABEL	X	X	X	X	X	X	
LABELKIND					X		
LASTACCESSIBLEAREA	X						
LASTRECORD	X	X					

Attribute	Disk	CD	Printer	Remote	Tape	Other	Port
LASTSUBFILE							X
LFILENAME	X	X	X	X	X	X	X
LIBMAINTAPPEND					X		
LIBMAINTDIR					X		
LICENSEKEY	X						
LINENUM			X				
LOCATECAPABLE					X		
LOCKEDFILE	X		X		X		
LTITLE	X	X	X	X	X	X	X
MAXCENSUS							X
MAXFRAMESIZECENSUS							X
MAXRECORDNUMBER	X						
MAXRECSIZE	X	X	X	X	X	X	X
MAXSUBFILES							X
MINRECSIZE	X	X	X	X	X	X	
MULTIVOLUME		X					
MYDOMAINNAME							X
MYHOST							X
MYHOSTGROUP							X
MYIPADDRESS							X
MYNAME							X
MYUSE	X	X	X	X	X	X	
NETACCESSPOINT							X
NEWFILE	X	X	X	X	X	X	
NEXTRECORD	X	X	X		X	X	
NONBLOCK	X						
NORECOVERY	X						
NORESOURCEWAIT	X						
NOTE	X		X				
ODDBLOCKSIZE					X		
OFFSITE					X		
OFNOTIFICATION	X	X		X	X	X	
OPEN	X	X	X	X	X	X	X

Attribute	Disk	CD	Printer	Remote	Tape	Other	Port
OPTIONAL	X	X		X	X	X	
OUTPUTEVENT							X
OUTPUTTABLE	X	X	X	X	X	X	
OWNER	X						
OVERRIDEEXTMODE	X	X			X		
PACKETWRITE		X					
PAGE			X				
PAGECOMP			X				
PAGESIZE			X				
PARITY					X		
PASSIVEOPEN							X
PATHNAME	X	X	X	X	X	X	X
PENDINGPROTECTEDFILE		X					
PERMITTEDACTIONS	X	X					
POPULATION	X			X			
PORTSEGMENTIO							X
PRESENT	X	X	X	X	X	X	
PRINTCHARGE			X				
PRINTCOPIES			X				
PRINTDISPOSITION	X		X				
PRINTERBACKUPDATA	X						
PRINTERKIND			X				
PRINTPARTIAL			X				
PRINTREQUEST	X		X				
PRODUCT	X						
PROPAGATESECURITYTODIRS	X						
PROPAGATESECURITYTOFILES	X						
PROTECTION	X	X			X		
PROVIDERGROUP							X
READDATE	X						
READDATEUT	X						
READREVERSECAPABLE					X		
READTIME	X						

Attribute	Disk	CD	Printer	Remote	Tape	Other	Port
READTIMEUT	X						
READTZ	X						
RECEPTIONS				X			
RECORD	X		X		X	X	
RECORDINERROR	X	X			X		
RECORDLEVELLOCK	X						
REDIRECTION						X	
REINITIALIZE	X	X	X	X	X	X	X
RELEASEID	X						
REQUESTEDMAXRECSIZE							X
RESIDENT	X	X	X	X	X	X	
RESTRICTED	X	X					
RESULTLIST	X	X	X	X	X	X	X
REWRITEABLE						X	
ROWADDRESS	X	X					
ROWSINUSE	X	X					
SAVEBACKUPFILE			X				
SAVEFACTOR	X				X		
SAVEPRINTFILE	X		X				
SCRATCHPOOL					X		
SCREEN				X			
SCREENSIZE				X			
SEARCHRULE	X						
SECTORSIZE	X	X					
SECURITYADMIN	X						
SECURITYGUARD	X	X	X		X		
SECURITYMODE	X				X		
SECURITYTYPE	X	X	X		X		X
SECURITYUSE	X	X	X		X		
SENSITIVEDATA	X	X					
SERIALNO	X				X		
SERVICE	X						X
SINGLEUNIT	X						

Attribute	Disk	CD	Printer	Remote	Tape	Other	Port
SIZEMODE	X	X	X	X	X	X	
SIZEOFFSET	X	X	X	X	X	X	
SIZEVISIBLE	X	X	X	X	X	X	
SIZE2	X	X	X	X	X	X	
STATE	X	X	X	X	X	X	X
STATIONCOUNT				X			
STATIONLIST				X			
STATIONNAME				X			
STATIONSALLOWED				X			
STATIONSDENIED				X			
SUBFILEERROR							X
SYNCHRONIZE	X	X			X		
TANKING				X			
TAPEREELRECORD					X		
TIMELIMIT				X			
TIMESTAMP	X	X					
TITLE	X	X	X	X	X	X	X
TOTALSECTORS	X	X					
TRAINID			X				
TRANSFORM			X				
TRANSLATE	X	X	X	X	X	X	X
TRANSLATING	X	X	X	X	X	X	X
TRANSMISSIONNO				X			
TRANSMISSIONS				X			
TRIMBLANKS			X	X			
UNIQUETOKEN	X						
UNITNO	X	X	X		X	X	
UNITS	X	X	X	X	X	X	X
UPDATEFILE	X						
USECATALOG	X				X		
USEDATE	X	X					
USERBACKUPNAME			X				
USERCODE	X						

Attribute	Disk	CD	Printer	Remote	Tape	Other	Port
USERINFO	X	X					
USETIME	X	X					
VERSION	X				X		
WARNINGS	X						
WIDTH				X			
YOURDOMAINNAME							X
YOURHOST	X	X	X	X	X	X	X
YOURHOSTGROUP							X
YOURIPADDRESS							X
YOURNAME							X
YOURNSAPA							X
YOURPRESENTATIONSEL							X
YOURSESSIONSEL							X
YOURTRANSPORTSEL							X
YOURUSERCODE							X

Note: *SECURITYGUARD, SECURITYTYPE, and SECURITYUSE attributes apply to backup files only.*

Appendix B
Port File Attributes and Associated Services

Use the following table to determine the port file attributes that are supported by specific port services. The table includes attributes that can be used by port files.

The following legend defines the column headings used in the table:

- BASIC—BASICSERVICE
- BNA—BNANATIVESERVICE
- NETBIOS—NETBIOSSESSIONSERVICE
- OSIN—OSINATIVESERVICE
- OSIS—OSISESSIONSERVICE
- TCPIPN—TCPIPNATIVESERVICE
- TCPN—TCPNATIVESERVICE

Attribute	**BASIC**	**BNA**	**NETBIOS**	**OSIN**	**OSIS**	**TCPIPN**	**TCPN**
ACTUALMAXRECSIZE	X	X	X	X	X	X	X
APPLICATION CONTEXT				X			
APPLICATIONGROUP ***Note:*** *For BNA Version 2 and BNAOSI only.*	X	X		X	X		
ATTERR	X	X	X	X	X	X	X
ATTVALUE	X	X	X	X	X	X	X
ATTYPE	X	X	X	X	X	X	X
AVAILABLE	X	X	X	X	X	X	X
AVAILABLEONLY	X	X	X	X	X		
BLANK		X					
BLOCKEDTIMEOUT	X	X		X	X	X	X
BLOCKSTRUCTURE	X	X	X	X	X		
BUFFERS		X	X				
CENSUS	X	X	X	X	X		

Attribute	BASIC	BNA	NETBIOS	OSIN	OSIS	TCPIPN	TCPN
CHANGEDSUBFILE	X	X	X	X	X	X	X
CHANGEEVENT	X	X	X	X	X	X	X
COMPRESSING	X	X	X	X	X		
COMPRESSION	X	X					
COMPRESSIONCONTROL	X	X					
COMPRESSIONREQUESTED	X	X					
CURRENTRECORDLENGTH	X	X	X	X	X	X	X
DIALOGCHECKNTERVAL		X				X	X
DIALOGPRIORITY		X				X	
DONOTSEARCHNETWORK		X				X	
EXTDELIMITER	X	X	X	X	X	X	X
EXTMODE		X					
FILECLASS	X	X	X	X	X	X	X
FILEEQUATED	X	X	X	X	X	X	X
FILENAME	X	X		X	X		
FILESTATE	X	X	X	X	X	X	X
FRAMESIZE	X	X	X	X	X	X	X
FRAMESIZECENSUS						X	X
INPUTEVENT	X	X	X	X	X	X	X
INTERACTIVEFILE	X	X	X	X	X	X	X
INTMODE		X					
INTNAME	X	X	X	X	X	X	X
KIND	X	X	X	X	X	X	X
LASTSUBFILE	X	X	X	X	X	X	X
LFILENAME	X	X		X	X		
LTITLE	X	X	X	X	X	X	X
MAXCENSUS	X	X	X	X	X		X
MAXFRAMESIZECENSUS						X	X
MAXSUBFILES	X	X	X	X	X	X	X
MYDOMAINNAME						X	X
MYHOST	X	X		X	X	X	X
MYHOSTGROUP		X					
MYIPADDRESS						X	X

Attribute	BASIC	BNA	NETBIOS	OSIN	OSIS	TCPIPN	TCPN
MYNAME	X	X	X	X	X	X	X
NETACCESSPOINT			X				
OUTPUTEVENT	X	X	X	X	X	X	X
PASSIVEOPEN			X			X	X
PATHNAME	X	X	X	X	X	X	X
PORTSEGMENTIO				X	X		
PROVIDERGROUP	X	X	X	X	X	X	X
REINITIALIZE	X	X	X	X	X	X	X
REQUESTEDMAXRECSIZE	X	X	X	X	X	X	X
RESULTLIST	X	X	X	X	X	X	X
SECURITYTYPE	X	X	X	X	X		X
SERVICE	X	X	X	X	X	X	X
STATE	X	X	X	X	X	X	X
SUBFILEERROR	X	X	X	X	X	X	X
TRANSLATE		X					
TRANSLATING		X					
YOURDOMAINNAME						X	X
YOURHOST	X	X		X	X	X	X
YOURHOSTGROUP		X					
YOURIPADDRESS						X	X
YOURNAME	X	X	X	X	X	X	X
YOURNSAPA				X	X		
YOURPRESENTATIONSEL				X			
YOURSESSIONSEL				X	X		
YOURTRANSPORTSEL				X	X		
YOURUSERCODE		X					

The following table lists attributes that are applicable to all devices but provide no additional functionality for a port file or are not the preferred attribute for port files:

The following legend defines the column headings used in the table:

- BASIC—BASICSERVICE
- BNA—BNANATIVESERVICE
- NETBIOS—NETBIOSSESSIONSERVICE
- OSIN—OSINATIVESERVICE
- OSIS—OSISESSIONSERVICE
- TCPIPN—TCPIPNATIVESERVICE
- TCPN—TCPNATIVESERVICE

Attribute	BASIC	BNA	NETBIOS	OSIN	OSIS	TCPIPN	TCPN
FILETYPE	X	X	X	X	X		X
HOSTNAME	X	X		X	X	X	X
IOCLOCKS	X	X	X	X	X	X	X
MAXRECSIZE	X	X	X	X	X	X	X
OPEN	X	X	X	X	X	X	X
TITLE	X	X	X	X	X	X	X
UNITS	X	X	X	X	X	X	X

Appendix C

Heritage Host Services Logical I/O and FTAM File Attributes

Use the following table to determine the file attributes that are valid for Heritage Host Services logical I/O files and FTAM files:

Attribute	**Heritage Host Services**	**FTAM**
ACCESSDATE	Supported	Not supported
ACCESSDATEUT	Supported	Not supported
ACCESSTZ	Supported	Not supported
ADAPTABLE	Supported	Not supported
AFTER	Supported	Not supported
ALIGNFILE	Supported	Not supported
ALIGNMENT	Supported	Not supported
ALTERDATE	Supported	Supported
ALTERDATEUT	Supported	Supported
ALTERNATEGROUPS	Supported	Not supported
ALTERTIME	Supported	Supported
ALTERTIMEUT	Supported	Supported
ALTERTZ	Supported	Not supported
APPEND	Supported	Not supported
AREAADDRESS	Not supported	Not supported
AREAALLOCATED	Supported	Not supported
AREALENGTH	Supported	Not supported
AREAS	Supported	Not supported
AREASECTORS	Supported	Not supported
AREASINUSE	Not supported	Not supported
AREASIZE	Restricted usage	Not supported
ATTERR	Supported	Supported

Attribute	Heritage Host Services	FTAM
ATTMODIFYDATE	Supported	Not supported
ATTMODIFYDATEUT	Supported	Not supported
ATTMODIFYTIME	Supported	Not supported
ATTMODIFYTIMEUT	Supported	Not supported
ATTMODIFYTZ	Supported	Not supported
ATTVALUE	Supported	Supported
ATTYPE	Not supported	Supported
AUTOUNLOAD	Supported	Not supported
AVAILABLE	Supported	Not supported
BACKUPDATE	Supported	Not supported
BACKUPDATEUT	Supported	Not supported
BACKUPKIND	Supported	Not supported
BACKUPTIME	Supported	Not supported
BACKUPTIMEUT	Supported	Not supported
BACKUPTZ	Supported	Not supported
BLOCK	Supported	Not supported
BLOCKSIZE	Restricted values	Not supported
BLOCKSTRUCTURE	Restricted values	Restricted values
BUFFERS	Supported	Not supported
BUFFERSIZE	Supported	Not supported
CARRIAGECONTROL	Supported	Not supported
CCSVERSION	Supported	Not Supported
CENSUS	Supported	Not supported
CHECKPOINT	Supported	Not supported
CLEARAREAS	Supported	Not supported
COMPRESSING	Supported	Not supported
COMPRESSIONCONTROL	Supported	Not supported
COMPRESSIONREQUESTED	Supported	Not supported
COPYDESTDATE	Supported	Not supported
COPYDESTDATEUT	Supported	Not supported
COPYDESTTIME	Supported	Not supported
COPYDESTTIMEUT	Supported	Not supported
COPYDESTTZ	Supported	Not supported

Attribute	Heritage Host Services	FTAM
COPYSOURCEDATE	Supported	Not supported
COPYSOURCEDATEUT	Supported	Not supported
COPYSOURCETIME	Supported	Not supported
COPYSOURCETIMEUT	Supported	Not supported
COPYSOURCETZ	Supported	Not supported
CREATEPASSWORD	Not supported	Supported
CREATIONDATE	Supported	Supported
CREATIONDATEUT	Supported	Supported
CREATIONTIME	Supported	Supported
CREATIONTIMEUT	Supported	Supported
CREATIONTZ	Supported	Not supported
CRUNCHED	Supported	Not supported
CURRENTBLOCKLENGTH	Supported	Not supported
CURRENTRECORDLENGTH	Supported	Supported
CYCLE	Supported	Not supported
DENSITY	Supported	Not supported
DEPENDENTINTMODE	Supported	Supported
DEPENDENTSPECS	Supported	Supported
DIRECTION	Supported	Not supported
DISPOSITION	Supported	Not supported
DOCUMENTTYPE	Supported	Supported
DUMMYFILE	Supported	Not supported
ESTIMATEDRECORDS	Supported	Not supported
EXCLUSIVE	Not supported	Supported
EXECUTEDATE	Supported	Not supported
EXECUTEDATEUT	Supported	Not supported
EXECUTETIME	Supported	Not supported
EXECUTETIMEUT	Supported	Not supported
EXECUTETZ	Supported	Not supported
EXTDELIMITER	Supported	Not supported
EXTMODE	Restricted values	Restricted values
FAMILYINDEX	Supported	Not supported
FAMILYNAME	Supported	Not supported

Attribute	Heritage Host Services	FTAM
FAMILYOWNER	Supported	Not supported
FILECLASS	Supported	Not supported
FILEEQUATED	Supported	Supported
FILEKIND	Restricted values	Restricted values
FILELENGTH	Not supported	Supported
FILENAME	Supported	Supported
FILEORGANIZATION	Restricted values	Restricted values
FILESECTION	Supported	Not supported
FILESTATE	Supported	Supported
FILESTRUCTURE	Restricted values	Restricted values
FILETYPE	Restricted values	Restricted values
FILEUSE	Supported	Supported
FLEXIBLE	Supported	Not supported
FORMID	Supported	Not supported
FRAMESIZE	Restricted values	Restricted values
GENERATION	Supported	Not supported
GROUP	Supported	Not supported
HOSTNAME	Supported	Supported
INPUTTABLE	Supported	Not supported
INTERACTIVEFILE	Supported	Not supported
INTMODE	Restricted values	Restricted values
INTNAME	Supported	Supported
IOHFUNTIONNAME	Not supported	Not supported
IOHINTERFACENAME	Not supported	Not supported
IOHLIBACCESS	Not supported	Not supported
IOHLIBPARAMETER	Not supported	Not supported
IOHPREFIX	Not supported	Not supported
IOHSTRING	Not supported	Not supported
IOHTITLE	Not supported	Not supported
KERBEROSACCESS	Not supported	Not supported
KIND	Restricted values	Restricted values
LABEL	Supported	Not supported
LABELKIND	Supported	Not supported

Attribute	Heritage Host Services	FTAM
LASTACCESSIBLEAREA	Supported	Not supported
LASTRECORD	Supported	Not supported
LASTSUBFILE	Supported	Not supported
LICENSEKEY	Restricted usage/values	Not supported
LINENUM	Supported	Not supported
LFILENAME	Supported	Supported
LOCATECAPABLE	Supported	Not supported
LOCKEDFILE	Supported	Not supported
LTITLE	Supported	Restricted values
MAXRECSIZE	Restricted values	Restricted values
MINRECSIZE	Supported	Supported
MYUSE	Restricted usage	Not supported
NEWFILE	Supported	Supported
NEXTRECORD	Supported	Not supported
ODDBLOCKSIZE	Not supported	Not supported
OFFSITE	Not supported	Not supported
OFNOTIFICATION	Supported	Supported
OPEN	Supported	Supported
OPTIONAL	Supported	Supported
OUTPUTTABLE	Supported	Not supported
OVERRIDEEXTMODE	Supported	Not supported
PAGE	Supported	Not supported
PAGECOMP	Supported	Not supported
PAGESIZE	Restricted usage	Not supported
PARITY	Supported	Not supported
PATHNAME	Supported	Not supported
PERMITTEDACTIONS	Supported	Supported
PRINTDISPOSITION	Supported	Not supported
PRINTERBACKUPDATA	Not supported	Not supported
PRINTERKIND	Supported	Not supported
PRINTPARTIAL	Supported	Not supported
PRINTREQUEST	Supported	Not Supported

Attribute	Heritage Host Services	FTAM
PRODUCT	Supported	Not supported
PROTECTION	Restricted usage	Not supported
READDATE	Supported	Not supported
READDATEUT	Supported	Not supported
READREVERSECAPABLE	Supported	Not supported
READTIME	Supported	Not supported
READTIMEUT	Supported	Not supported
READTZ	Supported	Not supported
REDIRECTION	Not supported	Not supported
REINITIALIZE	Supported	Supported
RELEASEID	Supported	Not supported
RESTRICTED	Supported	Not supported
RESULTLIST	Supported	Supported
SAVEFACTOR	Supported	Not supported
SAVEPRINTFILE	Not supported	Not supported
SCRATCHPOOL	Supported	Not supported
SCREEN	Supported	Not supported
SCREENSIZE	Supported	Not supported
SEARCHRULE	Supported	Not supported
SECTORSIZE	Supported	Not supported
SECURITYADMIN	Not supported	Not supported
SECURITYGUARD	Supported	Not supported
SECURITYTYPE	Restricted values	Not supported
SECURITYUSE	Supported	Not supported
SENSITIVEDATA	Supported	Not supported
SERIALNO	Restricted usage	Not supported
SERVICE	Supported	Supported
SINGLEUNIT	Supported	Not supported
SIZEVISIBLE	Not supported	Supported
STATE	Restricted values	Supported
STATIONCOUNT	Supported	Not supported
STATIONLIST	Supported	Not supported
STATIONNAME	Supported	Not supported

Attribute	Heritage Host Services	FTAM
STATIONSALLOWED	Supported	Not supported
STATIONSDENIED	Supported	Not supported
TAPEREELRECORD	Supported	Not supported
TIMELIMIT	Restricted values	Not supported
TITLE	Supported	Restricted use
TOTALSECTORS	Supported	Not supported
TRAINID	Supported	Not supported
TRANSFORM	Supported	Not supported
TRANSLATE	Restricted values	Not supported
TRANSLATING	Supported	Not supported
TRANSMISSIONNO	Supported	Not supported
UNIQUETOKEN	Supported	Not supported
UNITS	Restricted usage	Supported
UPDATEFILE	Restricted values	Not supported
USECATALOG	Supported	Not supported
USEDATE	Supported	Not supported
USERCODE	Not supported	Supported
VERSION	Supported	Not supported
WARNINGS	Supported	Not supported
WIDTH	Supported	Not supported
YOURHOST	Supported	Supported

Appendix D
Attribute Numbers

This appendix contains two lists. One lists the file attributes by their attribute number and the other lists the attributes alphabetically and identifies each attribute number.

Numerical List

The following list organizes the attribute numbers numerically:

0	TITLE
1	FILESECTION REEL
2	CREATIONDATE DATE
3	CYCLE
4	VERSION
5	SAVEFACTOR
6	DENSITY
7	PARITY
8	KIND
9	LABEL
10	EXTMODE
11	OPTIONAL
12	PROTECTION
13	FILETYPE
14	BLOCKSIZE
15	MAXRECSIZE
16	MINRECSIZE
17	AREASIZE
18	AREAS
19	APL
20	MYUSE
22	FLEXIBLE
26	BUFFERS
27	DIRECTION
28	CURRENTBLOCKLENGTH, CURRENTBLOCK
29	INTMODE
30	OPEN
31	PRESENT
32	SIZEMODE
33	SIZEOFFSET
34	SIZE2
35	STATE
36	EOF
39	FAMILYNAME, PACKNAME
40	SINGLEUNIT, SINGLEPACK
42	RESIDENT
43	CREATEPASSWORD
44	BLOCK
45	CARRIAGECONTROL
46	RECORD
47	UPDATEFILE
48	AVAILABLE

49	LASTRECORD
50	PAGESIZE
51	PAGE
52	LINENUM
54	FORMID, FORMMESSAGE
55	REINITIALIZE
56	NETACCESSPOINT
58	FILEKIND
59	CRUNCHED
60	AREASINUSE, ROWSINUSE
61	ACCESSDATE, LASTACCESSDATE, USEDATE
62	An internal representation for the SERIALNO attribute
64	AREAADDRESS, ROWADDRESS
70	ESTIMATEDRECORDS
71	PRINTREQUEST
72	INTNAME
73	UNITS
74	ATTERR
75	ATTYPE
76	ATTVALUE
77	IOCLOCKS
78	UNITNO
79	SECURITYGUARD
80	SECURITYTYPE
81	SECURITYUSE
84	RECORDINERROR
87	IOINERROR
88	TAPEREELRECORD
90	EXCLUSIVE
91	TRANSLATE
92	TRANSLATING
93	INPUTTABLE
94	OUTPUTTABLE
95	USECATALOG
96	GENERATION
97	SENSITIVEDATA
98	FAMILYINDEX
99	STATIONCOUNT, FAMILYSIZE
100	POPULATION
101	STATIONLIST, FAMILY
102	ENABLEINPUT
103	STATIONSDENIED
104	TRAINID
105	TIMELIMIT
106	LASTSUBFILE, LASTSTATION
107	RECEPTIONS
108	CENSUS
109	NORESOURCEWAIT
111	DISPOSITION
112	TRANSMISSIONS
113	ASSIGNTIME
114	TANKING
116	SCREEN
117	TRANSMISSIONNO
118	WIDTH
119	SERIALNO
120	IOHLIBACCESS
121	IOHFUNCTIONNAME
122	IOHINTERFACENAME
123	IOHTITLE
124	IOHPREFIX
125	IOHLIBPARAMETER
126	IOHSTRING
127	REDIRECTION

135	BACKUPKIND
136	NEWFILE
140	MYNAME
141	YOURNAME
142	FILEUSE
143	HOSTNAME, YOURHOST
144	FILEORGANIZATION
146	TIMESTAMP
147	HSFILECOPY
150	MAXSUBFILES
151	MYHOST, MYHOSTNAME
152	INPUTEVENT
153	OUTPUTEVENT
154	CHANGEEVENT
155	YOURUSERCODE, USERCODE
156	COMPRESSION
157	MAXCENSUS
158	CHANGEDSUBFILE
159	AREALENGTH
160	AREAALLOCATED
161	FRAMESIZE
162	FILENAME
164	STATIONNAME
165	BLOCKSTRUCTURE
166	DEPENDENTSPECS
167	SCREENSIZE
168	CURRENTRECORDLENGTH, CURRENTRECORD
169	SUBFILEERROR
170	FILESTATE
171	NEXTRECORD
172	LABELKIND
173	STATIONSALLOWED
175	ACCESSTIME, USETIME
176	USERINFO
177	ALTERDATE
178	ALTERTIME
179	CREATIONTIME
180	BLANK
181	PRINTERKIND
184	AFTER
185	ALIGNFILE
186	ALIGNMENT
187	BANNER
188	PRINTCHARGE
190	CHECKPOINT
191	DESTINATION
192	NOTE
193	PRINTCOPIES
194	PRINTDISPOSITION
195	SAVEBACKUPFILE
196	TRANSFORM
197	USERBACKUPNAME
198	DUMMYFILE
199	TRIMBLANKS
201	ACTUALMAXRECSIZE
202	APPLICATIONGROUP
203	AVAILABLEONLY
204	BLOCKEDTIMEOUT
205	COMPRESSING
206	COMPRESSIONCONTROL
207	COMPRESSIONREQUESTED
208	DIALOGCHECKINTERVAL
209	DIALOGPRIORITY
210	DONOTSEARCHNETWORK

211	MYHOSTGROUP
212	REQUESTEDMAXRECSIZE
213	YOURHOSTGROUP
214	WARNINGS
215	FILEEQUATED
216	RESTRICTED
218	FAMILYOWNER
220	LICENSEKEY
221	RELEASEID
224	SYNCHRONIZE
227	RECORDLEVELLOCK
228	NORECOVERY
230	ANYSIZEIO
231	BUFFERSIZE
232	FILELENGTH
233	FILESTRUCTURE
234	PASSIVEOPEN
235	SERVICE
236	DIOFILESTRUCTURE
237	APPLICATIONCONTEXT
240	PROVIDERGROUP
241	DOCUMENTTYPE
242	PERMITTEDACTIONS
243	RESULTLIST
244	OFNOTIFICATION
245	FRAMESIZECENSUS
246	MAXFRAMESIZECENSUS
249	AUTOUNLOAD
256	AREASECTORS
257	AUTOUNLOAD
258	TOTALSECTORS
264	DEPENDENTINTMODE
265	CCSVERSION
267	PAGECOMP
353	ATTMODIFYTIME
354	ATTMODIFYTIMEUT
268	PRINTPARTIAL
269	LOCKEDFILE
281	PENDINGPROTECTEDFILE
282	SCRATCHPOOL
283	YOURDOMAINNAME
284	MYDOMAINNAME
285	YOURIPADDRESS
286	MYIPADDRESS
289	PORTSEGMENTIO
290	APPEND
294	YOURPRESENTATIONSEL
295	YOURSESSIONSEL
296	YOURTRANSPORTSEL
297	YOURNSAPA
298	CLEARAREAS
299	BUFFERSHARING
334	OWNER
336	GROUP
337	ACCESSDATEUT
338	ATTMODIFYDATE
339	ATTMODIFYDATEUT
340	ALTERDATEUT
341	BACKUPDATE
342	BACKUPDATEUT
343	CREATIONDATEUT
344	COPYDESTDATE
345	COPYDESTDATEUT
346	COPYSOURCEDATE
347	COPYSOURCEDATEUT
348	EXECUTEDATE
349	EXECUTEDATEUT
350	READDATE
351	READDATEUT
352	ACCESSTIMEUT
390	OTHERRWX
392	OTHERW

355	ALTERTIMEUT
356	BACKUPTIME
357	BACKUPTIMEUT
358	COPYDESTTIME
359	COPYDESTTIMEUT
360	COPYSOURCETIME
361	COPYSOURCETIMEUT
362	EXECUTETIME
363	EXECUTETIMEUT
364	CREATIONTIMEUT
365	READTIME
366	READTIMEUT
367	ACCESSTZ
368	ALTERTZ
369	ATTMODIFYTZ
370	BACKUPTZ
371	CREATIONTZ
372	EXECUTETZ
373	READTZ
374	COPYDESTTZ
375	COPYSOURCETZ
376	UNIQUETOKEN
379	PATHNAME
380	SEARCHRULE
381	SECURITYMODE
382	OWNERRWX
383	OWNERR
384	OWNERW
385	OWNERX
386	GROUPRWX
393	OTHERX
394	USEGUARDFILE
395	GUARDDOWNER
396	SETUSERCODE
397	SETGROUPCODE
401	DONTCLEARAREASBYDEFAULT
402	READREVERSECAPABLE
403	NONBLOCK
404	ALLOWSPECIALFILE
405	ADAPTABLE
406	MAXRECORDNUMBER
407	PRODUCT
409	OVERRIDEEXTMODE
411	LOCATECAPABLE
412	LIBMAINTDIRECTORY
414	KERBEROSACCESS
416	FILECLASS
418	EXTDELIMITER
427	INTERACTIVEFILE
428	LIBMAINTAPPEND
429	LTITLE
430	LFILENAME
431	LASTACCESSIBLEAREA
432	ALTERNATEGROUPS
433	PROPAGATESECURITYTOFILES
434	PROPAGATESECURITYTODIRS
435	PRINTERBACKUPDATA
436	SAVEPRINTFILE
437	OFFSITE
438	ODDBLOCKSIZE

387	GROUPR
388	GROUPW
389	GROUPX
391	OTHERR
439	CDCOPIES
442	SECURITYADMIN
444	REWRITEABLE
445	ASSOCIATEDFILENAME

Note: *See SECURITYMODE for information on the subattributes SIZEOFFSET, OTHERRWX, OTHERW, OWNERRWX, OWNERR, OWNERW, OWNERX, GROUPRWX, GROUPR, GROUPW, GROUPX, OTHERR, USEGUARDFILE, GUARDDOWNER, SETUSERCODE, and SETGROUPCODE.*

Alphabetical List

The following list orders the attributes alphabetically and identifies the attribute numbers:

ACCESSDATE (61)

ACCESSDATEUT (337)

ACCESSTIME (175)

ACCESSTIMEUT (352)

ACCESSTZ (367)

ACTUALMAXRECSIZE (201)

ADAPTABLE (405)

AFTER (184)

ALIGNFILE (185)

ALIGNMENT (186)

ALLOWSPECIAL (405)

ALTERDATE (177)

ALTERDATEUT (340)

ALTERNATEGROUPS (432)

ALTERTIME (178)

ALTERTIMEUT (355)

ALTERTZ (368)

APPLICATIONGROUP (202)

ANYSIZEIO (230)

APL (19)

APPEND (290)

APPLICATIONCONTEXT (237)

AREAADDRESS (64)

AREAALLOCATED (160)

AREALENGTH (159)

AREAS (18)

AREASECTORS (256)

AREASINUSE (60)

AREASIZE (17)

ASSIGNTIME (113)

ASSOCIATEDFILENAME (445)

ATTERR (74)

ATTMODIFYDATE (338)

ATTMODIFYDATEUT (339)

ATTMODIFYTIME (353)

ATTMODIFYTIMEUT (354)

ATMODIFYTZ (369)

ATTVALUE (76)

ATTYPE (75)

AUTOUNLOAD (249)

AVAILABLE (48)

AVAILABLEONLY (203)

BACKUPDATE (341)

BACKUPDATEUT (342)

BACKUPKIND (135)

BACKUPTIME (356)

BACKUPTIMEUT (357)

BACKUPTZ (370)

BANNER (187)

BLANK (180)

BLOCK (44)

BLOCKEDTIMEOUT (204)

BLOCKSIZE (14)

BLOCKSTRUCTURE (165)

BUFFERS (26)

BUFFERSHARING (299)

BUFFERSIZE (231)

CARRIAGECONTROL (45)

CCSVERSION (265)

CDCOPIES (439)

CENSUS (108)

CHANGEDSUBFILE (158)

CHANGEEVENT (154)

CHECKPOINT (190)

CLEARAREAS (298)

COMPRESSING (205)

COMPRESSION (156)

COMPRESSIONCONTROL (206)

COMPRESSIONREQUESTED (207)

COPYDESTDATE (344)

COPYDESTDATEUT (345)

COPYDESTTIME (358)

COPYDESTIMEUT (361)

COPYDESTTZ (374)

COPYSOURCEDATE (346)

COPYSOURCEDATEUT (347)

COPYSOURCETIME (360)

COPYSOURCETIMEUT (361)

COPYSOURCETZ (375)

CREATEPASSWORD (43)

CREATIONDATE (2)

CREATIONDATEUT (343

CREATIONTIME (179)

CREATIONTIMEUT (364)

CREATIONTZ (371)

CRUNCHED (59)

CURRENTBLOCKLENGTH (28)

CURRENTBLOCK (28)

CURRENTRECORDLENGTH (168)

CURRENTRECORD (168)

CREATIONDATE (2)

DATE (2)

DEPENDENTINTMODE (264)

DENSITY (6)

DESTINATION (191)

DEPENDENTSPECS (166)

DIALOGPRIORITY (209)

DIALOGCHECKINTERVAL (208)

DIRECTION (27)

DIOFILESTRUCTURE (236)

DISPOSITION (111)

ESTIMATEDRECORDS (70)

EXCLUSIVE (90)

EXECUTEDATE (348)

EXECUTEDATEUT (349)

EXECUTETIME (362)

EXECUTETIMEUT (363)

EXECUTETZ (372)

EXTDELIMITER (418)

EXTMODE (10)

FAMILY (101)

FAMILYINDEX (98)

FAMILYNAME (39)

FAMILYOWNER (218)

FAMILYSIZE (99)

FILECLASS (416)

FILEEQUATED (215)

FILENAME (162)

FILESTATE (170)

FILESTRUCTURE (233)

FILETYPE (13)

FILEUSE (142)

FLEXIBLE (22)

GENERATION (96)

GROUP (336)

GROUPR (387)

GROUPRWX (386)

GROUPW (388)

GROUPX (389)

GUARDOWNER (395)

HOSTNAME (143)

HSFILECOPY (147)

INPUTEVENT (152)

INPUTTABLE (93)

INTERACTIVEFILE (427)

INTMODE (29)

INTNAME (72)

IOCLOCKS (77)

IOHFUNCTIONNAME (121)

IOHINTERFACENAME (122)

IOHLIBACCESS (120)

IOHLIBPARAMETER (125)

IOHPREFIX (124)

IOHSTRING (126)

IOHTITLE (123)

IOINERROR (87)

KERBEROSACCESS (414)

KIND (8)

LABEL (9)

LABELKIND (172)

LASTACCESSDATE (61)

LASTACCESSIBLEAREA (431)

LASTRECORD (49)

LASTSTATION (106)

LASTSUBFILE (106)

LFILENAME (430)

LIBMAINTAPPEND (428)

LIBMAINTDIR (412)

LICENSEKEY (220)

LINENUM (52)

LOCATECAPABLE (411)

LOCKEDFILE (269)

LTITLE (429)

MAXCENSUS (157)

MAXFRAMESIZECENSUS (246)

MAXRECORDNUMBER (406)

MAXRECSIZE (15)

MAXSUBFILES (150)

MINRECSIZE (16)

MYDOMAINNAME (284)

MYHOST (151)

MYHOSTGROUP (211)

MYHOSTNAME (151)

MYIPADDRESS (286)

MYNAME (140)

MYUSE (20)

NETACCESSPOINT (56)

NEWFILE (136)

NEXTRECORD (171)

NONBLOCK (403)

NORECOVERY (228)

NORESOURCEWAIT (109)

NOTE (192)

ODDBLOCKSIZE (438)

OFFSITE (437)

OPEN (30)

OPTIONAL (11)

OTHERR (391)

OTHERRW (390)

OTHERW (392)

OTHERX (393)

OUTPUTEVENT (153)

OUTPUTTABLE (94)

OVERRIDEEXTMODE (409)

OWNER (334)

OWNERR (383)

OWNERRWX (382)

OWNERW (384)

OWNERX (385)

PACKNAME (39)

PAGE (51)

PAGECOMP (267)
PAGESIZE (50)
PARITY (7)
PASSIVEOPEN (234)
PATHNAME (379)
PENDINGPROTECTEDFILE (281)
PERMITTEDACTIONS (242)
POPULATION (100)
PORTSEGMENTIO (289)
PRESENT (31)
PRINTCHARGE (188)
PRINTCOPIES (193)
PRINTDISPOSITION (194)
PRINTERBACKUPDATA (435)
PRINTERKIND (181)
PRINTPARTIAL (268)
PRINTREQUEST (71)
PRODUCT (407)
PROPAGATESECURITYTODIRS (434)
PROPAGATESECURITYTOFILES (433)
PROTECTION (12)
PROVIDERGROUP (240)
READDATE (350)
READDATEUT (351)
READREVERSECAPABLE (402)
READTIME (365)
READTIMEUT (366)
READTZ (373)
RECEPTIONS (107)
RECORD (46)
RECORDINERROR (84)
RECORDLEVELLOCK (227)
REDIRECTION (127)
REEL (1)
REINITIALIZE (55)
RELEASEID (221)
REQUESTEDMAXRECSIZE (212)
RESIDENT (42)
RESTRICTED (216)
RESULTLIST (243)
REWRITEABLE (444)
ROWADDRESS (64)
ROWSINUSE (60)
SAVEBACKUPFILE (195)
SAVEFACTOR (5)
SAVEPRINTFILE (436)
SCRATCHPOOL (282)
SCREEN (116)
SCREENSIZE (167)
SEARCHRULE (380)
SECTORSIZE (257)
SECURITYADMIN (442)
SECURITYGUARD (79)
SECURITYMODE (381)
SECURITYTYPE (80)
SECURITYUSE (81)
SENSITIVEDATA (97)
SERIALNO (119)
SERVICE (235)
SETUSERCODE (396)
SETGROUPCODE (397)
SINGLEPACK (40)
SINGLEUNIT (40)
SIZEMODE (32)
SIZEOFFSET (33)
SIZEVISIBLE (183)
SIZE2 (34)
STATE (35)
STATIONCOUNT (99)
STATIONLIST (101)
STATIONNAME (164)

STATIONSALLOWED (173)

STATIONSDENIED (103)

SUBFILEERROR (169)

SYNCHRONIZE (224)

TANKING (114)

TAPEREELRECORD (88)

TIMELIMIT (105)

TIMESTAMP (146)

TITLE (0)

TOTALSECTORS (258)

TRAINID (104)

TRANSFORM (196)

TRANSLATE (91)

TRANSLATING (92)

TRANSMISSIONNO (117)

TRANSMISSIONS (112)

TRIMBLANKS (199)

UNIQUETOKEN (376)

UNITNO (78)

UNITS (73)

UPDATEFILE (47)

USECATALOG (95)

USEDATE (61)

USEGUARDFILE (395)

USERBACKUPNAME (197)

USERCODE (155)

USERINFO (176)

USETIME (175)

VERSION (4)

WARNINGS (214)

WIDTH (118)

YOURDOMAINNAME (283)

YOURHOST (143)

YOURIPADDRESS (285)

YOURNAME (141)

YOURNSAPA (297)

YOURPRESENTATIONSEL (294)

YOURSESSIONSEL (295)

YOURTRANSPORTSEL (296)

YOURUSERCODE (155)

Note: *See SECURITYMODE for information on the subattributes OTHERR, OTHERRW, OTHERX, OWNERRWX, OWNERR, OWNERW, OWNERX, GROUPRWX, GROUPR, GROUPW, GROUPX, USEGUARDFILE, GUARDDOWNER, SETUSERCODE, and SETGROUPCODE.*

Appendix E
OPEN, CLOSE, and RESPOND Results

This appendix contains two lists. The first list contains the OPEN, CLOSE, and RESPOND results that might be returned during an OPEN, CLOSE, or RESPOND operation, and the second list contains the OPEN and CLOSE warnings that might be returned during an OPEN or CLOSE operation.

Note: *These OPEN and CLOSE results are different from the open and close error message numbers that are found in the system log.*

A result that is an odd number—that is, a Boolean value of TRUE—indicates that the operation was successful. A result that is an even number—that is, a Boolean value of FALSE—indicates that the operation was not successful.

Note that values normally associated with closing a file can be returned during an OPEN operation and values normally associated with opening a file can be returned during a CLOSE operation if one of the following conditions is true and the OPEN or CLOSE operation causes a volume switch to occur:

- The KIND value is TAPE.
- The KIND value is PRINTER and the BACKUPKIND value is TAPE.

OPEN, CLOSE, and RESPOND Results List

This list contains all the OPEN, CLOSE, and RESPOND results that might be returned during an OPEN, CLOSE, or RESPOND operation. It lists the number of the result and the meaning of the returned result.

0 (NOTAVAILABLERSLT)

The permanent file exists but is not available; as a result the program is locked out. This condition occurs when another program has opened the file with an EXCLUSIVE attribute value of TRUE, or when the program is attempting to open a file with an EXCLUSIVE attribute value of TRUE and another program has the file open.

This result is returned only when the open type is AVAILABLE.

1 (OKRSLT)

The operation was successful. If the control option specified is DONTWAIT or OFFER, this result indicates only that the process was successfully started.

2 (NOFILEFOUNDRSLT)

The permanent file does not exist or cannot be opened because the logical file and the permanent file are incompatible. If the OPEN statement contained a WAIT option, this result indicates that an NF (No File) system command was entered in response to a NO FILE message and the file has been left unassigned and in the closed state.

For port files, this error is returned to indicate that a match was not found.

3 (OFRSLT)

This result is returned only if the OFNOTIFICATION attribute is TRUE. The result indicates that an optional file did not exist and an OF (Optional File) system command was entered. As a result of this operator intervention, the file is now unassigned and in the open state.

4 (GENEALOGYMISMATCHRSLT)

The genealogy of the file does not match.

6 (SERIALNOMISMATCHRSLT)

The serial number does not match.

7 (WARNCONTEXTCHANGEDRSLT)

See “OPEN and CLOSE Warnings List.”

8 (FILEOFFLINERSLT)

The file is not on disk, but there is a backup copy of the file listed in the catalog or archive directory.

9 (WARNPORTATTIGNOREDRSLT)

See “OPEN and CLOSE Warnings List.”

10 (NOTAPERESOURCESRSLT)

No resources are available for a tape file.

11 (WARNCLOSECOLLINITIATORRSLT)

See “OPEN and CLOSE Warnings List.”

12 (NOFAMILYFOUNDRSLT)

A required disk is not online.

13 (WARNCLOSECOLLRESPONDERRSLT)

See “OPEN and CLOSE Warnings List.”

14 (ACCESSCODEMISMATCHRSLT)

The file is not available due to ACCESSCODE protection in the directory of the user.

15 (WARNCONTEXTIGNOREDRSLT)

See “OPEN and CLOSE Warnings List.”

17 (WARNABORTDATAIGNOREDRSLT)

See “OPEN and CLOSE Warnings List.”

18 (INVALIDKINDLISTRSLT)

An invalid KIND list or no KIND list is specified on a foreign host.

19 (WARNDEFINEPROVIDERGROUPRSLT)

See “OPEN and CLOSE Warnings List.”

20 (DIALOGFAILURERSLT)

An error occurred during initiation of a Heritage Host Services Logical I/O dialogue when the host was specified by the HOSTNAME attribute. This error can occur when the HOSTNAME attribute specifies an unavailable or unknown host.

21 (WARNINITIATINGPROVIDERRSLT)

See “OPEN and CLOSE Warnings List.”

23 (WARNFTAMFILEUSERSLT)

See “OPEN and CLOSE Warnings List.”

25 (WARNPURGELOCKEDFILETAPERSLT)

See “OPEN and CLOSE Warnings List.”

27 (WARNPURGEWRITEPROTECTEDTAPERSLT)

See “OPEN and CLOSE Warnings List.”

28 (BADHSFILERSLT)

This result indicates that the file uses attribute values that are not supported by Heritage Host Services Logical I/O. Look at the preceding attribute error message to determine which attribute was in error.

29 (WARNPURGEUNLABELLEDTAPERSLT)

See "OPEN and CLOSE Warnings List."

30 (FILENOTOPENRSLT)

An attempt was made to close a closed subfile.

32 (IOERRORRSLT)

All data might not have been transmitted before the subfile was closed. This error is also returned if a CLOSE statement with the ABORT option is invoked on the subfile and there is still data in the subfile read queue. The CLOSE ABORT operation does not fail.

34 (RECORDCOUNTMISMATCHRSLT)

The record counts in the EOF1 and EOV1 labels for a tape file do not match the number of records read.

36 (BLOCKCOUNTMISMATCHRSLT)

The block counts in the EOF1 and EOV1 labels for a tape file do not match the number of blocks read.

38 (UNAVAILABLEHOSTRSLT)

For port files, AVAILABLEONLY is TRUE or the OPEN statement with the AVAILABLE option was specified and the YOURHOST attribute names an unreachable or unknown host. If AVAILABLE or AVAILABLEONLY was not specified, this error is returned if the OPEN statement with the WAIT option is specified and the host named by the YOURHOST attribute becomes unreachable during the process of opening the subfile.

It is possible that the Port Provider does not support dialogues with foreign hosts; this is different from the host being unreachable. An example of this is the Local Port Provider (LPP). When dialogues with foreign hosts are not supported by the Port Provider, then UNSUPPORTEDFUNCTION will be set. It is also possible that the Port Provider does not support dialogues within the local host (for example, local ports). An example of this is the BNA 4.3 provider. When this is the case, then UNSUPPORTEDFUNCTION will be set.

This result can also occur when a TCPIPNATIVESERVICE subport is in the process of closing and the associated host is no longer available.

40 (FILENOTCLOSEDRSLT)

The value of the FILESTATE attribute for the specified subfile is not CLOSED.

42 (BADSUBFILEINDEXRSLT)

The file is a port file with the MAXSUBFILES value greater than 1, and either an operation was attempted that did not explicitly specify a subfile index or the subfile index specified is not within the range of 0 (zero) to the value of MAXSUBFILES.

44 (ERRORINOPENRSLT)

An unknown error occurred during an OPEN operation. To obtain more information about the error, refer to the OPEN information in the SUMLOG file for your system.

For port files, either an unknown error occurred during the dialogue establishment, or the OPEN or AWAITOPEN operation was performed on all subfiles and an error occurred on at least one subfile.

46 (ERRORINCLOSERSLT)

An unknown error occurred during the file CLOSE operation. To obtain more information about the error, refer to the CLOSE information in the SUMLOG file for your system.

For port files, either an unknown error occurred during the file CLOSE operation, or the CLOSE operation was performed on all subfiles and an error occurred while at least one subfile was being closed.

48 (UNSUPPORTEDFUNCTIONRSLT)

An attempted OPEN or AWAITOPEN operation on this port subfile resulted in a request for an unsupported function. The unsupported functions include the following cases:

- The YOURHOST attribute specifies the local host and local ports are not supported by the Port Provider.
- The YOURHOST attribute specifies a foreign host and foreign hosts are not supported by the Port Provider.

52 (BADFILEORGANIZATIONRSLT)

The logical file cannot be assigned to a physical file; the FILEORGANIZATION value is incompatible.

54 (UNSUPPORTEDPROTOCOLTYPERSLT)

A request was made to use an unsupported protocol type.

56 (PROTOCOLERRORRSLT)

A protocol error was detected.

58 (LACKOFRESOURCESRSLT)

The OPEN or AWAITOPEN operation failed due to a lack of resources.

60 (HOSTNOTINHOSTGROUPRSLT)

The OPEN request on this subfile failed because the YOURHOST value is not a legal member of the specified YOURHOSTGROUP.

62 (UNAUTHORIZEDFORAGRSLT)

The OPEN or AWAITOPEN operation failed because the user is not authorized to use the specified application group.

64 (SUPPORTLIBUNAVAILABLERSLT)

For port files, the OPEN or AWAITOPEN operation on this subfile failed because the network software was not running.

For disk files with FILEORGANIZATION equal to INDEXED or INDEXEDNOTRESTRICTED, the linkage to the KEYEDIOSUPPORT library failed because the library was not available or was incompatible with the operating system version.

66 (BADATTRIBUTESRSLT)

The OPEN request on this file or subfile failed because attributes required for the OPEN operation were specified with invalid values. If the file is a tape file, this result indicates that the FAMILYOWNER attribute is incompatible with the usercode portion of the FILENAME or TITLE attribute. Refer to the FAMILYOWNER attribute description for more information.

68 (UNAVAILABLEFUNCTIONRSLT)

The OPEN or AWAITOPEN operation on this subfile failed because the function sought was not available for this port service.

70 (UNSUPPORTEDINTMODERSLT)

The OPEN request on this subfile failed because the specified INTMODE value is not supported on the host specified by YOURHOST.

72 (NETWORKINGNOTSUPPORTEDRSLT)

The OPEN or AWAITOPEN operation on this subfile failed because the networking feature is not supported on this host.

74 (DISCONNECTEDDURINGOPENRSLT)

The matching process for an OPEN or AWAITOPEN operation was aborted by the provider.

76 (ACCESSRESTRICTEDRSLT)

The file can be opened only by a user having a usercode that is designated as a security administrator. This condition occurs only if the system SECADMIN option is TRUE. For more information on the system SECADMIN option, refer to the ??SECAD primitive command in the *System Commands Reference Manual*.

78 (CANNOTCREATERESTRICTEDFILERSLT)

The operation required the creation of a restricted file. The user is not assigned an alias to a privileged user or a security administrator at the host on which the file was to be created.

80 (UNSUPPORTEDDEVICETYPERSLT)

A device type was specified that logical I/O does not support.

82 (INCOMPATIBLEWITHMCPRSLT)

The file cannot be opened because it is incompatible with the current operating system version. A later version of the operating system has created the file, and the current operating system version cannot process it.

86 (SETMODEFAILEDRSLT)

The file cannot be opened because an error occurred when an attempt to change the mode of the tape drive was made. Depending on the type of tape drive, the mode change operation is used to set one or more of the following: buffered mode, density, speed, and block size.

88 (UNSUPPORTABLEBLOCKSIZERSLT)

The file cannot be opened because the tape drive cannot support the BLOCKSIZE value of the physical or logical file.

90 (FAMILYLIMITRSLT)

A family-limit error occurred during an attempt to close the file. A family-limit error occurs when an attempt is made to exceed the family limit specified for a usercode or a family in the USERDATAFILE while the disk resource control system is active. This error can occur when you are trying to close a temporary file that has been made permanent. Refer to the *System Operations Guide* for information on the disk resource control system.

92 (INTEGRALLIMITRSLT)

A family integral-limit error occurred during an attempt to close the file. A family integral-limit error occurs when the family integral limit specified for a usercode or a family in the USERDATAFILE is exceeded while the disk resource control system is active. This error can occur on a CLOSE operation when a temporary file is made permanent. Refer to the *System Operations Guide* for information on the disk resource control system.

94 (ILLEGALMYUSERSLT)

The MYUSE attribute value is incompatible with the device.

96 (ILLEGALNEWFILERSLT)

The NEWFILE attribute value is incompatible with the device.

98 (OPENAFTERLOCKRSLT)

The file cannot be opened because it was previously closed with a CLOSE statement that does not allow the file to be reopened. An example of this type of statement is the COBOL74 *CLOSE WITH LOCK* statement.

100 (INCOMPATIBLEBLOCKINGRSLT)

The BLOCKSIZE attribute value is incompatible with other attributes of the file.

102 (SHORTTAPEBLOCKINGRSLT)

The BLOCKSIZE attribute value is less than the minimum value required for a tape file. Refer to the topic of tape files in the *I/O Subsystem Programming Guide*.

104 (LOGICALPHYSICALMISMATCHRSLT)

No physical file can match the attribute values of the logical file.

106 (FRAMESIZEINCOMPATIBLERSLT)

The FRAMESIZE and INTMODE attribute values are incompatible. Refer to the description of the FRAMESIZE attribute.

108 (SIZEVISIBLECONFLICTRSLT)

The SIZEVISIBLE, FRAMESIZE, and INTMODE attribute values create a situation in which the record sizes cannot be maintained. Refer to the description of the SIZEVISIBLE attribute.

110 (INVALIDANYSIZEIOSPECRSLT)

The specified value of the ANYSIZEIO attribute is incompatible with other attributes of the file.

112 (AREALENGTHTOOBIGRSLT)

The size requested for an area requires more sectors than the maximum number of sectors allowed in an area.

114 (INTMODEFRAMESIZECONFLICTRSLT)

The value of the INTMODE attribute is incompatible with the FRAMESIZE attribute value that would be assumed from the permanent file. Refer to the description of the INTMODE attribute.

116 (INCOMPATIBLEFILESTRUCTURERSLT)

The value of FILESTRUCTURE for the logical file is incompatible with the value of FILESTRUCTURE for the permanent file.

118 (MUSTBESTREAMRSLT)

The value of FILESTRUCTURE for the permanent file is not STREAM.

120 (INCONSISTENTMAXRECSIZERSLT)

The MAXRECSIZE values for the permanent and logical files are not equal.

122 (INCONSISTENTBLOCKSIZERSLT)

The BLOCKSIZE value for the logical file is inconsistent with the BLOCKSIZE value of the permanent file.

124 (MAXRECSIZEGTRAREALENGTHRSLT)

The MAXRECSIZE value of a file with a FILESTRUCTURE value of STREAM exceeds the value of the AREALENGTH attribute.

126 (FRAMESIZEAREALENGTHCONFLICTRSLT)

The length of an area of a file with a FILESTRUCTURE value of STREAM is not an integral number of the FRAMESIZE units.

128 (UNSUPPORTEDPARAMETERRSLT)

The statement was invoked with parameters or parameter values not supported by the service of the port subfile.

130 (LOCALIPCNOTSUPPORTEDRSLT)

A local dialogue was requested by making the YOURHOST value equal to the MYHOST value, and local dialogue is not supported by the service of the subfile.

132 (UNSUPPORTEDTRANSLATIONRSLT)

The service of the subfile does not provide the translation requested.

134 (OPENREJECTEDRSLT)

The OPEN request failed because the correspondent application rejected the request made by the subfile

138 (ENDPOINTUNKNOWNRSLT)

The OPEN request failed because the local system operator failed to register the endpoint or the endpoint is unknown at the correspondent host.

140 (INVALIDRESPONDRSLT)

The respond request failed because an invalid RESPOND option was provided. Either ACCEPTOPEN or REJECTOPEN was specified and the value of FILESTATE is CLOSERESPONSEPLEASE, or ACCEPTCLOSE or REJECTCLOSE was specified and the value of FILESTATE is OPENRESPONSEPLEASE.

142 (INVALIDPROVIDERRSLT)

The provider group specified does not provide the port service requested for the subfile.

144 (PROVIDERRESTRICTEDRSLT)

Providers currently support subfiles of the port file that cannot be mixed with the provider of the service of this subfile.

146 (TIMELIMITEXCEEDEDRSLT)

The OPEN or AWAITOPEN operation failed because the value of CONNECTTIMELIMIT was exceeded.

148 (UNSUPPORTEDAPPLCONTEXTRSLT)

The correspondent rejected the request because it did not support the application context.

154 (BADASSOCIATEDDATARSLT)

The ASSOCIATEDDATA was unreadable.

156 (ASSOCIATEDDATATOOLONGRSLT)

The ASSOCIATEDDATA was too long.

158 (BADATTRIBUTESFORRESPONDRSLT)

One or more attributes were set to invalid values on a RESPOND operation.

160 (BADATTRIBUTESFORAWAITOPENRSLT)

One or more attributes were set to invalid values on an AWAITOPEN operation.

162 (SECTORSTREAMREQDEPSPECSRSLT)

An attempt was made to open an existing file using direct I/O when the DIOFILESTRUCTURE value is SECTORSTREAM and neither the DEPENDENTSPECS value is TRUE nor the FILETYPE value is 7 or 8.

164 (FILESTRUCTURENOTSETRSLT)

An attempt was made to create a new file using direct I/O when the DIOFILESTRUCTURE value is DEPENDENT or SECTORSTREAM and the FILESTRUCTURE attribute was not set.

166 (ENDPTINCOMPATIBLESERVICESLT)

The dialogue endpoint cannot be accessed by this service.

168 (UNSUPPORTEDPRIMITIVERSLT)

A statement not supported by the service of the port file was invoked.

170 (BADCONNECTTIMELIMITRSLT)

The OPEN or AWAITOPEN operation failed because the value of the CONNECTTIMELIMIT attribute is invalid.

172 (BASICSERVICEVIOLATIONRSLT)

A protocol message from the correspondent system or endpoint was received that used a feature that BASICSERVICE does not support. The dialogue is closed.

174 (INVALIDFILESTATERSLT)

The RESPOND operation was invoked while the subfile was in an invalid file state.

176 (TRANSPARENTNOTSUPPORTEDRSLT)

The correspondent system or endpoint does not support the abstract syntax {ISO(1) IDENTIFIER-ORGANIZATION(3) OIW(14) ULSIG(8) ABSTRACT-SYNTAX(2) OCTETSTRING(1)}, which is the message format used by this OSI subfile for data transfer.

178 (CANNOTACCESSFILEKINDRSLT)

The permanent file cannot be opened because of the value of its FILEKIND attribute.

180 (KEYEDOPENFAILURERSLT)

The KEYEDIO or KEYEDIOII library indicated an error condition, which caused the OPEN request to fail.

182 (UNSUPDMSIIFSRSLT)

This value of FILESTRUCTURE is not supported when DMIO is TRUE. Information about the DMIO attribute is found in the *DMALGOL Programming Reference Manual*.

184 (INSUFFICIENTSPACERSLT)

While the buffers were being flushed, a no-space condition occurred, and the NORESOURCEWAIT attribute is TRUE.

186 (UNMATCHEDDIOFILESTRUCTURERSLT)

For direct files, the value of DIOFILESTRUCTURE does not match either the specified value of the FILESTRUCTURE attribute for a new file, or the value of the FILESTRUCTURE attribute for the permanent file.

188 (INCONSISTENTMINRECSIZERSLT)

The MINRECSIZE values for the permanent and logical files are not equal.

190 (OPENREJECTEDTRANSIENTRSLT)

The OPEN operation failed because the correspondent endpoint rejected the request for transient reasons.

192 (PROVIDERGROUPUNDEFINEDRSLT)

The OPEN or AWAITOPEN operation failed because the provider group specified by the PROVIDERGROUP attribute has not been defined through the *NW NS ADD PROVIDERGROUP* operator command.

194 (OPENABORTBYCORRESPONDENTRSLT)

The OPEN operation failed because an abort was sent by the correspondent endpoint.

196 (NOUSERCODERSLT)

The task is running without a USERCODE value, and an attempt was made to open a file through Heritage Host Services Logical I/O.

198 (ASSIGNEDTAPECHGDRSLT)

An operator physically replaced a tape on a drive that was in use.

200 (NOTRANSLATETABLERSLT)

One or both translate tables are missing. If only one table is missing, it is the required table.

228 (NETBIOSNAMEINUSERSLT)

Another application on the same local area network (LAN) is attempting to open a port subfile with the same MYNAME value as this subfile. For further information about this error, refer to the discussion of NETBIOSSESSIONSERVICE in the *I/O Subsystem Programming Guide*.

230 (REOPENSTOPPEDBYTMRSLT)

The TAPEMANAGER library intervened to prevent a file that had previously been closed with retention from being reopened.

232 (CREATIONSTOPPEDBYTSRSLT)

The TAPESERVER library intervened to prevent a new file from being created.

234 (LIBCANNOTOPENFILERSLT)

This file cannot be opened by the library to which it was passed.

236 (ASSIGNMENTSTOPPEDBYTMRSLT)

The TAPEMANAGER library intervened to prevent file assignment.

238 (ALLSEGMENTSNOTSENTRSLT)

The PORTSEGMENTIO file attribute value is TRUE and all segments of data were not sent when the program attempted a CLOSE ORDERLY operation.

240 (RESOLVERNOTAVAILABLERSLT)

The enterprise server Resolver is not running. The Resolver must be running to resolve the YOURDOMAINNAME file attribute value.

242 (NAMESERVICENOTAVAILABLERSLT)

The enterprise server Resolver has been configured with an unknown name server.

244 (NAMESERVICEUNREACHABLERSLT)

The enterprise server Resolver cannot contact any name server to process the open request.

246 (CONNECTIONINUSERSLT)

Another application on the same local host is attempting to open or support a subfile that has the same MYNAME, MYIPADDRESS, YOURNAME, and YOURIPADDRESS values as this subfile. Each subfile of TCP/IP must have a unique combination of these four attributes on the local host.

248 (INVALIDBUFFERSHARINGRSLT)

One of the following conditions is present and the file is being opened with a BUFFERSHARING value of SHARED or EXCLUSIVELYSHARED:

- The FILESTRUCTURE attribute value is ALIGNED180 and the BLOCKSIZE attribute value of the logical file does not equal the block size of the physical file.
- The BLOCKSTRUCTURE attribute is not FIXED.
- The MAXRECSIZE attribute value is different than the stored MAXRECSIZE value in the disk file header.
- The UPDATEFILE file attribute value is TRUE.
- The FILEORGANIZATION file attribute value is other than NOTRESTRICTED.
- The file is a direct I/O file.

250 (OPENWITHBUFFERSHARINGRSLT)

An OPEN operation was invoked with the AVAILABLE option for a file that has a BUFFERSHARING value of NONE, and the physical file is already being used by a file that has a BUFFERSHARING value of EXCLUSIVELYSHARED.

252 (OPENWITHOUTBUFFERSHARINGRSLT)

An OPEN operation was invoked with the AVAILABLE option for a file that has a BUFFERSHARING value of EXCLUSIVELYSHARED, and the physical file is already being used by a file that has a BUFFERSHARING value of NONE.

254 (COMPRESSIONVSLABELFORMATRSLT)

The LABELFORMAT file attribute value of the file is ANSI69, but the ANSI87 label format is required because the COMPRESSIONCONTROL file attribute value is USER and the COMPRESSIONREQUESTED file attribute value is TRUE.

272 (UNIQUETOKENEXPANSIONRSLT)

A node of a file name that was generated by using the UNIQUETOKEN feature is greater than 17 characters.

274 (INVALIDREBLOCKRSLT)

The disk file cannot be opened because the BLOCKSIZE value is different from that of the permanent file and would cause interblock gaps to become accessible. This occurs if either or both the logical and permanent file blocks do not contain an integral number of sectors.

276 (INVALIDAPPENDRSLT)

An attempt has been made to open a file with APPEND TRUE, but the value of another file attribute conflicted with this usage.

278 (INVALIDFIFORSLT)

Either a direct file attempted to open a FIFO file or an attempt was made to create or open a FIFO file that violated one of the following restrictions:

- The value of BLOCKSTRUCTURE must be FIXED.
- The value of FILESTRUCTURE must be STREAM.
- The value of MAXRECSIZE must be 1 (one).
- The EXTMODE and INTMODE attributes must both have values that reflect 8 bit characters. (SINGLE, HEX, and BCL are not allowed.)
- The value of FRAMESIZE must be 8.
- ALLOWSPECIALFILE must be set to TRUE.
- UPDATEFILE must be FALSE.
- FILEORGANIZATION must be NOTRESTRICTED.
- The value of ANYSIZEIO must be TRUE for any logical file opening a FIFO file.

280 (NOFIFOREADERRSLT)

An attempt was made to open a FIFO file for write-only access with the NONBLOCK attribute set to TRUE, but the FIFO file was not open for reading.

282 (FIFOINUSERSLT)

An attempt was made to close a FIFO logical file that still had a pending read or write request.

284 (FILEEXISTSRSLT)

Either an attempt was made to open a new file with the MUSTBENEW open option and a file of the same name exists, or an attempt was made to create a PERMDIR file when a file of the same name exists with a FILEKIND value that does not equal PERMDIR.

286 (PERMDENIEDRSLT)

Insufficient permission exists at some level of the file name.

290 (BADTRUNCATERSLT)

Invalid use of the TRUNCATE option in an OPEN statement.

292 (RETAINEDINVALIDRSLT)

The OPEN type is incompatible with a file that was closed with retention.

296 (PERMDIRMISSINGRSLT)

An attempt was made to create a new permanent directory or open an existing permanent directory, but the parent directories do not exist.

298 (INCOMPATIBLEEXTMODESRSLT)

The EXTMODE values of the logical and physical files are incompatible. An attempt was made to open an existing file with an OVERRIDEEXTMODE file attribute value of ALWAYS but the value of EXTMODE stored in the physical file is incompatible with the value of EXTMODE stored in the logical file.

300 (HDRVERSIONRSLT)

The program attempted to open an existing disk file but the file could not be accessed by programs running on this SSR of the MCP. Programs running on this SSR of the MCP cannot access disk files that are declared with more than 1000 areas or that were declared with a row size greater than 16,777,215 sectors.

302 (FILEUSECONFLICTRSLT)

The user does not have sufficient privileges to open the specified disk file.

304 (INVALIDSPECIALFILERSLT)

There were incompatible attributes on an OPEN when a file with a FILEKIND attribute value of DEVNULL was opened. This error is due to violating one of the same list of attribute restrictions specified above for open result 278 (INVALIDFIFOSLT).

306 (PERMDIRNOTEMPTYRSLT)

An attempt was made to remove a non-empty permanent directory.

308 (PERMDIREXISTSRSLT)

An attempt was made to create a new permanent directory when a permanent directory of the same name already exists.

310 (NAMETOOLONGRSLT)

The combination of the CURRENTDIRECTORY task attribute value and a relative pathname produced a resolved pathname containing more than 14 nodes (including usercode and family name).

312 (NAMEINVALIDRSLT)

The pathname resolution process produced an invalid name or a pathname with a trailing slash (indicating a directory) was used in an inappropriate context.

332 (PRINTSPROBLEMRSLT)

An error occurred when PrintS attempted to create the printer backup file.

334 (INVALIDTRANSLATIONRSLT)

The file could not be opened because translation between the INTMODE and EXTMODE of the file is unavailable.

336 (DSEDINFINDV)

An open attempt by a non-discontinuable task was discontinued (DS system command) while trying to find the file.

338 (INVALIDPERMDIRRSLT)

An attempt was made to create or open a PERMDIR file that violated one of the following restrictions:

- The value of BLOCKSTRUCTURE must be FIXED.
- The value of FILESTRUCTURE must be STREAM.
- The value of MAXRECSIZE must be 1 (one).
- The EXTMODE and INTMODE attributes must both have values that reflect 8 bit characters. (SINGLE, HEX, and BCL are not allowed.)
- The value of FRAMESIZE must be 8.
- DIRECTFILE must be FALSE.
- UPDATEFILE must be FALSE.
- FILEORGANIZATION must be NOTRESTRICTED.
- The value of ANYSIZEIO must be TRUE for any logical file opening a PERMDIR file.

342 (MISSINGCCSFILERSLT)

The SYSTEM/CCSFILE is not accessible or is missing.

344 (COMPLEXTRANSRSLT)

User-supplied translate tables cannot be used when complex translation is required. MCP complex translation is only done through the CENTRALSUPPORT library. Complex translation occurs for those translations in which one or both character sets include 16-bit characters. Translate tables deal solely with character sets that include 8-bit characters only.

346 (EXTINTMODERSLT)

The complex input translation attempted (EXTMODE to INTMODE) is not supported by the CENTRALSUPPORT library. However, the complex translation in the opposite direction (INTMODE to EXTMODE) is supported.

348 (INTEXTMODERSLT)

The complex output translation attempted (INTMODE to EXTMODE) is not supported by the CENTRALSUPPORT library. However, the complex translation in the opposite direction (EXTMODE to INTMODE) is supported.

350 (UNKNOWNCCSTOCCSRSLT)

An unknown error was encountered in the CENTRALSUPPORT library. Please contact your customer support.

352 (COMPTRANSNOTSUPRSLT)

Complex translation is not allowed or supported for this file. Please check the complex translation restrictions against the attributes of the file. Complex translation occurs for those translations in which one or both character sets include 16-bit characters. Refer to "Double-byte and Mixed Multibyte Character Sets" in the *I/O Subsystem Programming Guide* for the restrictions on complex translation.

354 (CCSTOCCSSERRORRSLT)

An internal software error or a fault was encountered in the CENTRALSUPPORT library. Please contact your customer support.

356 (COMPCHARSNOTSUPRSLT)

Complex character sets are not allowed or supported for this file. Please check the complex characters restrictions against the attributes of the file. Complex character sets are all those which include 16-bit characters. Refer to "Double-byte and Mixed Multibyte Character Sets" in the *I/O Subsystem Programming Guide* for the restrictions on complex translation.

358 (UNSUPMODESRSLT)

Complex translation is not supported in either direction for the INTMODE and EXTMODE values provided.

360 (PACKNOTPRESORVOLUMEDRSLT)

The target family for the new permanent directory is not volumed, but the USECATALOG file attribute is TRUE or the USECATDEFAULT system option is set.

362 (SHARENEWOPENRSLT)

An attempt was made to open a new file on a disk family that is marked as shared and the share level does not support this action.

366 (IMPROPEREXTDELIMRSLT)

A value of the EXTDELIMITER attribute was specified that is incompatible with the other attributes of the file.

368 (IOHEXCEPTIONRSLT)

A fault occurred during the OPEN or CLOSE operation of a virtual file.

370 (IOHLINKAGERSLT)

The MCP was unable to link to the IOHANDLER library supplied for the virtual file.

372 (IOHOPENCLOSERSLT)

An error occurred during the OPEN operation of a virtual file.

374 (IOHVERSIONRSLT)

A version mismatch was reported by the IOHANDLER library during the OPEN operation of a virtual file.

376 (IOHSTRINGERRRSLT)

The IOHANDLER library detected an error in the IOHSTRING parameter set on the virtual file and passed to the IOHANDLER library OPEN operation.

378 (REDIRSMBERRRSLT)

The REDIRECTOR component encountered a protocol error processing the OPEN request for a redirected file.

380 (REDIRNETBIORSLT)

The REDIRECTOR component was unable to establish a NetBIOS session with the server specified in the OPEN request for a redirected file.

382 (REDIRNEGPROTRSLT)

The server specified in the OPEN request for a redirected file does not support the protocols used by the MCP for the REDIRECTION attribute.

384 (REDIRCREDENTIALSRSLT)

The server could not validate the credentials supplied in the OPEN request of a redirected file.

386 (REDIRNORESOURCERSLT)

Resources are not available on the server to handle the OPEN request of a redirected file.

388 (REDIRTCONERRORRSLT)

An error was encountered by the REDIRECTOR component while connecting to the SHARE specified for the redirected file.

390 (REDIRFILENAMERSLT)

The file name supplied during the OPEN operation of a redirected file was invalid for the REDIRECTOR component.

392 (UNCFILENAMERSLT)

The file name supplied during the OPEN operation of a redirected file indicated that it was formatted as a UNC (Uniform Naming Convention) file name, but was incorrectly formatted.

394 (INCOMPATIBLEBLOCKSTRUCTRSLT)

The value of BLOCKSTRUCTURE for the logical file is incompatible with the value of BLOCKSTRUCTURE for the permanent file.

396 (LIBCANNOTCLOSEFILERSLT)

The file cannot be closed by the library to which it was passed.

OPEN and CLOSE Warnings List

This list identifies OPEN and CLOSE results that are warnings. The OPEN or CLOSE operation does not fail.

7 (WARNCONTEXTCHANGEDRSLT)

The APPLICATIONCONTEXT attribute changed this negotiation.

9 (WARNPORTATTIGNOREDRSLT)

One or more port attributes were set to values invalid for the service of the subfile and are being ignored.

11 (WARNCLOSECOLLINITIATORRSLT)

The ORDERLY CLOSE operation of the dialogue cannot complete because a collision of CLOSE requests occurred. This endpoint is the initiator of the dialogue.

13 (WARNCLOSECOLLRESPONDERRSLT)

The ORDERLY CLOSE operation cannot complete because a collision of CLOSE requests occurred. This endpoint is the responder of the dialogue.

15 (WARNCONTEXTIGNOREDRSLT)

The APPLICATIONCONTEXT attribute was not null when the AWAITOPEN statement was invoked and is being ignored. The responder can only participate in dialogue parameter negotiation if the AWAITOPEN statement is invoked with the PARTICIPATE option, and the subfile is in the OPENRESPONSEPLEASE file state.

17 (WARNABORTDATAIGNOREDRSLT)

The associated data parameter specified with the CLOSE ABORT statement was not sent with the abort because it is too long.

19 (WARNDEFINEPROVIDERGROUPRSLT)

The control option specified in the OPEN or AWAITOPEN statement was DONTWAIT or OFFER, and the provider group specified by the PROVIDERGROUP attribute is undefined. The program is warned that the dialogue establishment process cannot continue until the provider group is defined through the *NW NS ADD PROVIDERGROUP* operator command.

21 (WARNINITIATINGPROVIDERRSLT)

The control option specified in the OPEN or AWAITOPEN statement was DONTWAIT or OFFER, and the provider of the port service is not available. The system will be initiating the provider for this port file. The program is warned that the dialogue establishment process cannot continue until the provider initializes successfully. If the provider fails to initialize, the dialogue establishment fails with a NETWORKINGNOTSUPPORTED (16) SUBFILEERROR and the subfile transits to the CLOSED file state.

23 (WARNFTAMFILEUSERSLT)

This warning appears when you attempt to open an FTAM-1 or an FTAM-3 file through FTAM with a FILEUSE value of I/O. It is illegal to switch from read to write except at the beginning or the end of the file, and any attempt to do so causes the program to be discontinued.

25 (WARNPURGELOCKEDFILETAPERSLT)

A CLOSE operation with a disposition of purge was unable to purge the specified tape file because, even though the tape unit was write-enabled, the LOCKEDFILE file attribute of the tape file is TRUE.

27 (WARNPURGEWRITEPROTECTEDTAPERSLT)

A CLOSE operation with a disposition of purge was unable to purge the specified tape because the tape was write-protected.

29 (WARNPURGEUNLABELLEDTAPERSLT)

A CLOSE operation with a disposition of PURGE was unable to purge the tape because the tape was unlabeled.

Appendix F
I/O Result Enumerated Values

This appendix lists the possible result enumerations returned by the direct I/O buffer attribute IOERRORTYPE, in the file attribute RESULTLIST, and in field [26:10] of the following:

- The STATE attribute
- The result of a READ, WRITE, or SEEK operation on a file that is not a direct I/O file
- The result of a WAIT operation on a direct array

The following list shows the mnemonic value of the result, the integer value of the result, and the meaning of the result.

< 0 (No mnemonic)

Valid only for IOERRORTYPE. Reported as UNDEFINEDERROR by other attributes. This result indicates an undefined error occurred. The error field of the hardware result descriptor is returned as a negative number.

0 (NOERROR)

No error occurred.

1 (NOTREADYTYPE)

Valid only for IOERRORTYPE. Reported as NOTREADY by other attributes. This result indicates that a peripheral unit is not ready.

1 (ASSIGNMENTDENIED)

Not valid for IOERRORTYPE. This result indicates that remote file assignment is rejected. Refer to the discussion of opening remote files in the *I/O Subsystem Programming Guide*.

2 (PARITYERRORTYPE)

Valid only for IOERRORTYPE. Reported as PARITYERROR by other attributes. This result indicates that a parity error occurred. For a host control I/O operation, this result indicates a coordinated error—that is, an exception result has been returned to both the READ side and the WRITE side of an attempted intersystem control (ISC) transfer.

2 (ASSIGNMENTPOSTPONED)

Not valid for IOERRORTYPE. This result indicates that the remote file assignment is postponed. Refer to the discussion of opening remote files in the *I/O Subsystem Programming Guide*.

3 (REWINDINGTYPE)

Valid only for IOERRORTYPE. Reported as REWINDING by other attributes. This result indicates the tape drive is rewinding.

3 (ILLEGALFILEUSE)

Not valid for IOERRORTYPE. This result indicates that an illegal FILEUSE value was used for a remote file. Refer to the discussion of opening remote files in the *I/O Subsystem Programming Guide*.

4 (DESCRIPTORERROR)

A descriptor error occurred (direct I/O).

4 (DATAERROR)

A data error occurred (normal I/O).

5 (BEGINNINGORENDOFTAPE)

Valid only for IOERRORTYPE. Reported as ENDOFTAPE, DELETEDDISKAREA, or INSUFFICIENTSPACE by other attributes. For tape I/O operations, this result indicates an end-of-tape or beginning-of-tape condition occurred. For disk I/O operations, this result indicates either a no-space condition occurred and the NORESOURCEWAIT attribute is TRUE, or the I/O operation referenced a deleted disk area.

6 (WLOOREOF)

Valid only for IOERRORTYPE. Reported as ENDOFFILE or WRITELOCKOUT by other attributes. This result indicates an end-of-file (EOF) condition or write lockout occurred. For a host control file WRITE operation, this result indicates that write access is denied, which means the destination host control file, specified by the WRITEPARTNER attribute, is not marked as input-capable.

6 (WRITELOCKOUT)

Not valid for IOERRORTYPE. Reported as WLOOREOF by IOERRORTYPE. This result indicates a write lockout occurred.

7 (READPASTROW)

Valid only for IOERRORTYPE. This result indicates a READ or WRITE operation attempted to cross an area boundary or to exceed the disk area, or a READ operation attempted to access data beyond the end of the file.

8 (CANCELED)

I/O was canceled. If returned by IOERRORTYPE, this result includes the conditions reported as IMPLICITOPENFAILED, UNSPECIFIEDEXCEPTION, and OUTOFRANGE by other attributes.

9 (SHORTRECORD)

Valid only for IOERRORTYPE. This result indicates a short record was read. For a host control file READ operation, this result indicates a long block, which means that the host control file that is writing sent more data than the reading direct array could hold. This error is not a coordinated error; that is, the corresponding host control file WRITE operation does not get an exception result.

10 (BREAKONOUTPUT)

This result indicates a break on output occurred (remote files only).

11 (PACKPROGRAMOROPERATORERROR)

Valid only for IOERRORTYPE. This result indicates a disk pack program or operator error occurred.

12 (PACKHARDWAREERROR)

Valid only for IOERRORTYPE. This result indicates a disk pack hardware error occurred.

13 (PACKHARDWAREORDATAERROR)

Valid only for IOERRORTYPE. This result indicates a disk pack hardware or data error occurred.

14 (PACKDATAERROR)

Valid only for IOERRORTYPE. This result indicates a disk pack data error occurred.

15 (TIMEOUT)

A disk storage unit timeout, blank tape timeout, or time limit was exceeded for remote files.

16 (DMSROWLOCKOUT)

Valid only for IOERRORTYPE. This result indicates an area WRITE operation locked out by Enterprise Database Server. The program has read an area that cannot be written to.

18 (HCWRITETIMEOUT)

Valid only for IOERRORTYPE. This result indicates that a host control file WRITE operation specified an input-capable WRITEPARTNER, but there was no corresponding READ operation at the destination host control file within the allowed 30 seconds.

19 (BADIOPARTNER)

Valid only for IOERRORTYPE. This result indicates that a host control file user attempted to override a HUBMAP level READPARTNER or WRITEPARTNER specification by modifying the corresponding direct array attribute.

20 (NOWRITEPARTNER)

Valid only for IOERRORTYPE. This result indicates that a host control file WRITE operation was attempted when there was no HUBMAP level WRITEPARTNER specification and no direct buffer attribute WRITEPARTNER was specified.

21 (WRONGHUBINDEX)

Valid only for IOERRORTYPE. This result indicates that a host control file READ operation with the READPARTNER specified received input from a different hubindex.

22 (OPERATIONABORT)

Valid only for IOERRORTYPE. Refer to the HY file IOERRORTYPE values for direct I/O HY files in the *I/O Subsystem Programming Guide*.

23 (REMOTEOPERATIONABORT)

Valid only for IOERRORTYPE. Refer to the HY file IOERRORTYPE values for direct I/O HY files in the *I/O Subsystem Programming Guide*.

24 (DEADMANTIMEOUT)

Valid only for IOERRORTYPE. Refer to the HY file IOERRORTYPE values for direct I/O HY files in the *I/O Subsystem Programming Guide*.

25 (DATACONFLICT)

Valid only for IOERRORTYPE. Refer to the HY file IOERRORTYPE values for direct I/O HY files in the *I/O Subsystem Programming Guide*.

26 (INPROGRESS)

Valid only for IOERRORTYPE. An I/O operation is not complete.

27 (NOTREADY)

Not valid for IOERRORTYPE. Reported as NOTREADYTYPE by IOERRORTYPE. This result indicates that the peripheral unit was no ready.

28 (PARITYERROR)

Not valid for IOERRORTYPE. Reported as PARITYERRORTYPE by IOERRORTYPE. This result indicates that a parity error has occurred.

29 (REWINDING)

Not valid for IOERRORTYPE. Reported as REWINDINGTYPE by IOERRORTYPE. This result indicates that the tape drive is rewinding.

30 (ADDNONDELCAPABLE)

For KEYEDIOII files created with the RPG extended file system (EFS), an attempt was made to add a record randomly in a file that is not delete-capable.

31 (DELETENONDELCAPABLE)

For KEYEDIOII files created with the RPG extended file system (EFS), an attempt was made to delete a record in a file that is not delete-capable.

32 (UPDATENOREAD)

For KEYEDIOII files created with the RPG extended file system (EFS), an attempt was made to update a record that was not the last record read.

33 (BEYONDEXTENDAREA)

For KEYEDIOII files created with the RPG extended file system (EFS), an attempt was made to read or write a record beyond the extend area.

34 (UPDATENOTLOCKED)

For KEYEDIOII files created with the RPG extended file system (EFS), an attempt was made to update an unlocked record.

35 (POSITIONOCCUPIED)

For KEYEDIOII files created with the RPG extended file system (EFS), an attempt was made to add a record randomly to a record position that contains a valid record.

36 (SEQWRITETODIRECT)

For KEYEDIOII files created with the RPG extended file system (EFS), an attempt was made to perform a sequential WRITE operation on an EFS direct file.

39 (FAMILYLIMITERR)

A FAMILYLIMITERR result indicates that the FAMILYLIMIT of the FAMILYLIST group in the USERDATAFILE for a family or usercode was exceeded. Refer to the *Security Administration Guide* for information on FAMILYLIMIT.

40 (INTEGRALLIMITERR)

An INTEGRALLIMITERR result indicates that the FAMILYINTLIMIT of the FAMILYLIST group in the USERDATAFILE for a family or usercode was exceeded. Refer to the *Security Administration Guide* for information about FAMILYINTLIMIT and FAMILYINTEGRAL.

41 (TEMPFILEEXCEEDED)

This result indicates that an attempt was made to exceed the temporary file limit.

42 (URGENTFLAG)

For port files, this result indicates that the message read contains data marked URGENT.

43 (BADWRITEOPTION)

For port files, a WRITE option that is inappropriate to the value of the SERVICE attribute was specified.

44 (UNDEFINEDERROR)

This result indicates that an undefined error occurred.

45 (ENDOFTAPE)

Not valid for IOERRORTYPE. Reported as BEGINNINGORENDOFTAPE by IOERRORTYPE. This result indicates that an end-of-tape condition occurred.

45 (DELETEDDISKAREA)

Not valid for IOERRORTYPE. Reported as BEGINNINGORENDOFTAPE by IOERRORTYPE. The I/O operation referenced a deleted disk area.

46 (ENDOFFILE)

Not valid for IOERRORTYPE. Reported as WLOOREOF by IOERRORTYPE. An end-of-file (EOF) condition was encountered.

47 (SHORTBLOCK)

One of the following actions has occurred:

- A short block was read.
- A direct disk I/O was truncated at an area boundary.
- If the FILESTRUCTURE attribute value is STREAM, a short record at the end of the file was read.

48 (CONTROLCARD)

A control card was read.

49 (ENDOFPAGE)

An end-of-page condition was encountered.

50 (SECURITYERROR)

A security violation has been attempted.

51 (DSSABORT)

DSS Management detected an error while communicating with the remote host.

52 (INVALIDSUBFILE)

For port files, an invalid subfile index was specified.

53 (NODATAFORREAD)

For port files, a READ operation with the DONTWAIT option failed because no data was available.

54 (BROADCASTWRITEFAILED)

For port files, a broadcast WRITE operation failed for one or more subfiles.

55 (NOBUFFERFORWRITE)

For port files, a WRITE operation with the DONTWAIT option failed because no buffer was available.

Buffers become unavailable when the CENSUS or the FRAMESIZECENSUS attribute of the remote subfile has reached the MAXCENSUS or MAXFRAMESIZECENSUS attribute value of the subfile.

Other flow control situations in the underlying network protocol can also result in buffers being unavailable. For example, ICP can run out of memory for remote dialogues.

56 (DELETEDORDUPLICATERECORD)

For files with a FILEORGANIZATION value of RELATIVE, an attempt was made to read a deleted record or to write a duplicate record.

57 (ENDOFMEDIUM)

No space is available on the medium. For disk files, an area could not be allocated.

71 (INSUFFICIENTSPACE)

Not valid for IOERRORTYPE. Reported as BEGINNINGORENDOFTAPE by IOERRORTYPE. A no-space condition occurred and the NORESOURCEWAIT attribute is TRUE.

72 (OPENDATA)

Indicates that the message read was user data on an incoming OPEN request.

73 (OPENRESPONSEDATA)

Indicates that the message read was user data on an incoming OPEN response.

74 (CLOSEREQUESTDATA)

Indicates that the data read was user data on an incoming ORDERLY CLOSE request.

75 (CLOSEABORTDATA)

Indicates that the data read was user data on an incoming ABORT indication.

76 (CLOSERESPONSEDATA)

Indicates that the message read was user data on an incoming CLOSE response.

77 (ENCODEDDATAERROR)

Indicates that the READ or WRITE operation failed because encoded data was truncated.

78 (MOREDATA)

The PORTSEGMENTIO file attribute of the application program is TRUE, and there are more segments of the message to be read.

79 (MESSAGETRUNCATEDWARNING)

Indicates that the message read or written was truncated.

91 (IMPLICITOPENFAILED)

Not valid for IOERRORTYPE. Reported as CANCELED by IOERRORTYPE. The attempt to open the file so that the I/O operation could be performed was unsuccessful.

92 (UNSPECIFIEDEXCEPTION)

Not valid for IOERRORTYPE. Reported as CANCELED by IOERRORTYPE. An exception occurred, but there is no information about its nature. This can occur in Heritage Host Services logical I/O when the enumerated result at the file host does not correspond to any error bits in a logical result descriptor (LRD).

93 (SETMODEFAILED)

A change-of-mode operation failed for the tape unit. The buffering mode of a buffered tape unit is changed for a synchronized WRITE operation when the SYNCHRONIZE attribute has the value NO.

94 (DISKALLOCATIONFAILED)

An unspecified failure occurred during an attempt to allocate disk space for a file.

95 (RECORDNOTFOUND)

For KEYEDIO and KEYEDIOII files, no record with a matching key was found.

96 (PRIMARYKEYSNOTEQUAL or PRIMARYKEYNOTEQUAL)

For KEYEDIO and KEYEDIOII files, an attempt was made to change the key of a record when that record was rewritten.

97 (DUPLICATEKEYS or DUPLICATEKEY)

For KEYEDIO and KEYEDIOII files, an attempt was made to write a record with a duplicate key when duplicate keys are not allowed.

98 (KEYISINVALID)

For KEYEDIO and KEYEDIOII files, the key information supplied is not valid.

99 (LASTIOMUSTBEREAD)

For KEYEDIO files, KEYEDIOII files, and all COBOL85 files, an attempt was made to use a sequential operation to rewrite or delete the current record without first reading it. For COBOL85 files, to rewrite or delete the current record, the read must have been successful.

100 (DIFFERENTLENGTHRECORDS or DIFFERENTLENGTHRECORD)

For KEYEDIO and KEYEDIOII files, the MAXRECSIZE value or record size or record length supplied by the program does not match the MAXRECSIZE attribute of the file.

101 (PRIMARYKEYSOUTOFORDER or PRIMARYKEYOUTOFORDER)

For KEYEDIO and KEYEDIOII files, a sequential WRITE operation was attempted with a key value that did not have the proper relation to the key of the previous record.

106 (KIOIIWRITEERROR)

For KEYEDIOII files, an error occurred when the KEYEDIOII library attempted a WRITE operation. Each program using the file should close so that recovery can proceed.

107 (NOTCLOSERETAINED)

The file must be in a closed retained state—CLOSE (<FILE>, *)—to perform the requested operation.

108 (VALIDONLYFORDISK)

The file must be a disk file.

110 (NODISKHEADER)

There is no disk file header associated with the logical file.

111 (NOTONLYUSEROFFILE)

ERASE requires that the file not be in use by any user but the caller.

114 (DATAMAYBECORRUPTED)

FTAM performed the requested I/O operation but received subsequent protocol messages that indicate an error. The data associated with the I/O operation might be good, but the program should use the data with caution.

115 (RESPSEEKFAIL)

The foreign system returned an error when a SEEK operation was performed through FTAM.

116 (RESPREADFAIL)

The foreign system returned an error when a READ operation was performed through FTAM.

117 (RESPWRITEFAIL)

The foreign system returned an error when a WRITE operation was performed through FTAM.

118 (WRITEONINPUT)

A WRITE operation to a file failed because the FILEUSE attribute value of the file was IN.

119 (READONOUTPUT)

A WRITE operation to a file failed because the FILEUSE attribute value of the file was OUT.

120 (HYDATALOST)

Valid only for IOERRORTYPE. Refer to the HY file IOERRORTYPE values for direct I/O HY files in the *I/O Subsystem Programming Guide*.

121 (HYMESSAGEOVERFLOW)

Valid only for IOERRORTYPE. Refer to the HY file IOERRORTYPE values for direct I/O HY files in the *I/O Subsystem Programming Guide*.

122 (NOTATEND)

The file was not positioned at the end of the file when a WRITE operation was performed through FTAM. FTAM specifies that records can be written only at the end of this file. This result indicates that either the file is not positioned at the end or that FTAM cannot determine if the file is positioned at the end. Opening a file with an open option of ATEND or AVAILATEND (OPEN [AVAILABLE] AT END in COBOL74) causes the file to be positioned at the end. After the file is opened in this manner, FTAM allows write operations only if no intervening read or seek operations are performed. If intervening read or seek operations are performed, FTAM cannot guarantee that the file is positioned at the end of the file.

125 (BUFFERLESSTHANASEGMENT)

The PORTSEGMENTIO file attribute of the application program is TRUE, and the buffer passed by the application program for the port file READ statement is less than a single segment.

126 (DATALENGTHEXCEEDSBUFFERSIZE)

The PORTSEGMENTIO file attribute of the application program is TRUE, and the data length specified by the application program is greater than the buffer size passed in the WRITE statement.

127 (DATALENGTHEXCEEDED)

The PORTSEGMENTIO file attribute of the application program is TRUE, and the data length specified in the WRITE statement is greater than 63K.

128 (ALLSEGMENTSMUSTBESENT)

The PORTSEGMENTIO file attribute of the application program is TRUE, and the program attempted to issue session primitives before sending all segments of the message.

129 (PORTSEGMENTIONOTSET)

The application program attempted to send a partial message, but the PORTSEGMENTIO attribute was not set.

130 (DEADLOCKEMBRACE)

The process encountered a deadly embrace condition for the specified file.

131 (DEADLOCKTIMEOUT)

A KIOII timeout was detected for the specified file.

132 (INSUFFICIENTBUFFERS)

For port files, a WRITE operation with the DONTWAIT option requested more data be sent than the provider could send at the moment, but some data was sent. Interrogate Field [47:20] of the STATE file attribute value or the CURRENTRECORDLENGTH value to determine how much data was actually sent.

133 (OUTOFRANGE)

Not valid for IOERRORTYPE. Reported as CANCELED by IOERRORTYPE. The MCP run-time option IORANGECHECK and compile-time option DIAGNOSTICS are set, and an I/O operation was attempted outside of any area of the file. This indicates a system software problem. The preceding IORANGECHECK dump can be used in diagnosing the problem. The I/O operation was canceled.

134 (ENDOFDATAONTAPE)

The end-of-data mark was detected while reading or spacing on a tape. This result might occur when reading or spacing on either an improperly created tape that does not have an end-of-file mark, or on a blank tape.

136 (READFIRSTDUPLICATE)

For COBOL85 KEYEDIOII files, a READ statement was successfully executed, but a duplicate key was detected. This condition indicates a warning only. The key value for the current key of reference is equal to the value of the same key in the next record within the current key of reference.

137 (CREATEDDUPLICATEALTERNATEKEY)

For COBOL85 KEYEDIOII files, a WRITE or REWRITE was successfully executed, but a duplicate alternate key was created. This condition indicates a warning only. The record just written created a duplicate key value for at least one alternate record key for which duplicates are allowed.

138 (SEQUENTIALAFTEREOF)

For COBOL85 files, a sequential READ or WRITE statement was attempted after the previous I/O request returned an end-of-file condition. This result is not produced for PORT or REMOTE files.

139 (KEYTOOSMALL)

For COBOL85 sequential and relative files, a sequential READ statement was attempted, but the number of significant digits in the relative record number is larger than the size of the actual key or relative key data item described for the file; or a sequential WRITE statement was attempted and the number is larger than the size of the actual key or relative key data item described for the file.

158 (APPENDRANDOMWRITE)

A random write operation has been issued to a file with the APPEND attribute set to TRUE. The operation has been rejected.

159 (NOREADERS)

A WRITE operation was attempted on a FIFO file that had not also been opened for reading.

162 (MTVOLUMEUNCERTAIN)

Not valid for IOERRORTYPE. Reported as DISCRIPTORERROR by IOERRORTYPE. A tape has undergone unexpected manual manipulation during use. The tape has been repositioned at the beginning of the tape (BOT) due to a manual unload and reload. Further IO requests connected with the same logical use of the unit are canceled.

163 (SEQUENTIALIOAFTERBADSEEK)

For COBOL85 relative files, a sequential READ was attempted after an unsuccessful previous START or SEEK statement. An unsuccessful START or SEEK statement is one that resulted in an error causing the exception bit to be ON.

169 (VSSREADMODIFYWRITE)

Valid only for direct I/O files. A write operation has required the I/O subsystem to perform a read-modify-write operation.

1023 (DSEDLOGICALFILEINUSE)

One of two possible conditions has occurred. Either a discontinued process attempted to gain access to a logical file that was being used by another process, or a process that cannot be discontinued was discontinued while that process was attempting to gain access to a logical file that was being used by another process. In either case, the STATE result has all bits turned on.

The numbered I/O results found in the previous list are used to provide additional or more specific information than the bits in the logical result descriptor (LRD). Some bits of the existing LRD have two or more meanings.

The Heritage Host Services logical I/O protocol does not exchange the numbered I/O results between hosts. Results at the file host that have a corresponding error bit are returned to the program with the primary meaning for the bit. Exception results that do not have a corresponding error bit are returned as the UNSPECIFIEDEXCEPTION (92) I/O result. Results that are not exceptions are returned as the NOERROR (0) I/O result. The value DSSABORT (51) is generated at the process host when appropriate.

The following table shows the correspondence between the error bits of the LRD and their primary and secondary I/O result meanings (not all bits have secondary meanings).

Bit	Primary I/O Result Meaning	Secondary I/O Result Meaning
[16:1]	SECURITYERROR (50)	
[15:1]	TIMEOUT (15)	ENDOFDATAONTAPE (134)
[14:1]	REWINDING (29)	
[13:1]	BREAKONOUTPUT (10)	
[12:1]	ENDOFSHEET (90)	
[11:1]	CONTROLCARD (48)	
[10:1]	SHORTBLOCK (47)	
[9:1]	ENDOFFILE (46)	ASSIGNMENTDENIED (1) ASSIGNMENTPOSTPONED (2) ILLEGALFILEUSE (3) ENDOFPAGE (49) ENDOFMEDIUM (57) KEYISINVALID (98) NOREADERS (159) RECORDNOTFOUND (95) SEQUENTIALIOAFTEREOF (138)
[8:1]	ENDOFTAPE (45)	NODATAFORREAD (53) BROADCASTWRITEFAILED (54) NOBUFFERFORWRITE (55) INSUFFICIENTSPACE (71)
[7:1]	PARITYERROR (28)	DSSABORT (51) APPENDRANDOMWRITE (158)
[6:1]	WRITELOCKOUT (6)	PRIMARYKEYSNOTEQUAL (96) PRIMARYKEYSOUTOFORDER (101)
[5:1]	UNDEFINEDERROR (44)	DELETEDORDUPLICATERECORD (56) DUPLICATEKEY (97)
[4:1]	DESCRIPTORERROR (direct I/O) (4) DATAERROR (nondirect I/O) (4)	MTVOLUMEUNCERTAIN (162) DIFFERENTLENGTHRECORDS (100)
[3:1]	NOTREADY (27)	INVALIDSUBFILE (52)
[2:1]	CANCELED (8)	OUTOFRANGE (133)

Appendix G
Predefined Time Zones

The following table lists the predefined time zones.

Number	Name	Abbreviation	Offset
1	Alaska Standard Time	(AKST)	– 9 : 00
2	Alaska Daylight Time	(AKDT)	– 8 : 00
3	Pacific Standard Time	(PST)	– 8 : 00
4	Pacific Daylight Time	(PDT)	– 7 : 00
5	Mountain Standard Time	(MOST)	– 7 : 00
6	Mountain Daylight Time	(MODT)	– 6 : 00
7	Central Standard Time	(CST)	– 6 : 00
8	Central Daylight Time	(CDT)	– 5 : 00
9	Eastern Standard Time	(EST)	– 5 : 00
10	Eastern Daylight Time	(EDT)	– 4 : 00
11	Atlantic Standard Time	(AST)	– 4 : 00
12	Atlantic Daylight Time	(ADT)	– 3 : 00
13	Greenland Standard Time	(GST)	– 3 : 00
14	Greenland Daylight Time	(GDT)	– 2 : 00
15	Newfoundland Standard Time	(NST)	– 3 : 30
16	Newfoundland Daylight Time	(NDT)	– 2 : 30
17	Hawaii-Aleutian Standard Time	(HST)	– 10 : 00
18	Hawaii-Aleutian Daylight Time	(HDT)	– 9 : 00
19	Greenwich Mean Time	(GMT) or (GCT)	+ 0 : 00
20	British Summer Time	(BST)	+ 1 : 00
21	Central European Standard Time	(CEST)	+ 1 : 00
22	Central European Daylight Time	(CEDT)	+ 2 : 00
23	Eastern European Standard Time	(EEST)	+ 2 : 00
24	Eastern European Daylight Time	(EEDT)	+ 3 : 00
25	Indian Standard Time	(IST)	+ 5 : 30

Number	Name	Abbreviation	Offset
26	Hong Kong Standard Time	(HKST)	+ 8 : 00
27	Chinese Standard Time	(CHST)	+ 8 : 00
28	Japanese Standard Time	(JST)	+ 9 : 00
29	Eastern Australian Standard Time	(EAST)	+ 10 : 00
30	Eastern Australian Daylight Time	(EADT)	+ 11 : 00
31	Central Australian Standard Time	(CAST)	+ 9 : 30
32	Central Australian Daylight Time	(CADT)	+ 10 : 30
33	Western Australian Standard Time	(WAST)	+ 8 : 00
34	Western Australian Daylight Time	(WADT)	+ 9 : 00
35	UT–11:30	(–11:30)	– 11 : 30
36	UT–11:00	(–11:00)	– 11 : 00
37	UT–10:30	(–10:30)	– 10 : 30
38	UT–10:00	(–10:00)	– 10 : 00
39	UT–9:30	(–9:30)	– 9 : 30
40	UT–9:00	(–9:00)	– 9 : 00
41	UT–8:30	(–8:30)	– 8 : 30
42	UT–8:00	(–8:00)	– 8 : 00
43	UT–7:30	(–7:30)	– 7 : 30
44	UT–7:00	(–7:00)	– 7 : 00
45	UT–6:30	(–6:30)	– 6 : 30
46	UT–6:00	(–6:00)	– 6: 00
47	UT–5:30	(–5:30)	– 5 : 30
48	UT–5:00	(–5:00)	– 5 : 00
49	UT–4:30	(–4:30)	– 4 : 30
50	UT–4:00	(–4:00)	– 4 : 00
51	UT–3:30	(–3:30)	– 3 : 30
52	UT–3:00	(–3:00)	– 3 : 00
53	UT–2:30	(–2:30)	– 2 : 30
54	UT–2:00	(–2:00)	– 2 : 00
55	UT–1:30	(–1:30)	– 1 : 30
56	UT–1:00	(–1:00)	– 1 : 00
57	UT–0:30	(–0:30)	– 0 : 30
58	UT + 0:00	(+0:00)	+ 0 : 00

Number	Name	Abbreviation	Offset
59	UT + 0:30	(+0:30)	+ 0 : 30
60	UT + 1:00	(+1:00)	+ 1 : 00
61	UT + 1:30	(+1:30)	+ 1 : 30
62	UT+ 2:00	(+2:00)	+ 2 : 00
63	UT + 2:30	(+2:30)	+ 2 : 30
64	UT+ 3:00	(+3:00)	+ 3 : 00
65	UT+ 3:30	(+3:30)	+ 3 : 30
66	UT+ 4:00	(+4:00)	+ 4 : 00
67	UT+ 4:30	(+4:30)	+ 4 : 30
68	UT+ 5:00	(+5:00)	+ 5 : 00
69	UT+ 5:30	(+5:30)	+ 5 : 30
70	UT+ 5:45	(+5:45)	+ 5 : 45
71	UT+ 6:00	(+6:00)	+ 6 : 00
72	UT+ 6:30	(+6:30)	+ 6 : 30
73	UT+ 6:45	(+6:45)	+ 6 : 45
74	UT+ 7:00	(+7:00)	+ 7 : 00
75	UT+ 7:30	(+7:30)	+ 7 : 30
76	UT+ 8:00	(+8:00)	+ 8 : 00
77	UT+ 8:30	(+8:30)	+ 8 : 30
78	UT+ 9:00	(+9:00)	+ 9 : 00
79	UT+ 9:30	(+9:30)	+ 9 : 30
80	UT+ 10:00	(+10:00)	+ 10 : 00
81	UT+ 10:30	(+10:30)	+ 10 : 30
82	UT+ 11:00	(+11:00)	+ 11 : 00
83	UT+ 11:30	(+11:30)	+ 11 : 30
84	UT+ 12:00	(+12:00)	+ 12 : 00
85	UT+ 12:30	(+12:30)	+ 12 : 30
86	UT+ 12:45	(+12:45)	+ 12 : 45
87	UT+ 13:00	(+13:00)	+ 13 : 00
88	UT+ 13:45	(+13:45)	+ 13 : 45

Appendix H
Understanding Railroad Diagrams

This appendix explains railroad diagrams, including the following concepts:

- Paths of a railroad diagram
- Constants and variables
- Constraints

The text describes the elements of the diagrams and provides examples.

Railroad Diagram Concepts

Railroad diagrams are diagrams that show you the standards for combining words and symbols into commands and statements. These diagrams consist of a series of paths that show the allowable structures of the command or statement.

Paths

Paths show the order in which the command or statement is constructed and are represented by horizontal and vertical lines. Many commands and statements have a number of options so the railroad diagram has a number of different paths you can take.

The following example has three paths:

```
── REMOVE ─┬──────────┬─────────────────────────────────┤
           ├─ SOURCE ─┤
           └─ OBJECT ─┘
```

The three paths in the previous example show the following three possible commands:

- REMOVE
- REMOVE SOURCE
- REMOVE OBJECT

A railroad diagram is as complex as a command or statement requires. Regardless of the level of complexity, all railroad diagrams are visual representations of commands and statements.

Railroad diagrams are intended to show

- Mandatory items
- User-selected items
- Order in which the items must appear
- Number of times an item can be repeated
- Necessary punctuation

Follow the railroad diagrams to understand the correct syntax for commands and statements. The diagrams serve as quick references to the commands and statements.

The following table introduces the elements of a railroad diagram:

The diagram element . . .	**Indicates an item that . . .**
Constant	Must be entered in full or as a specific abbreviation
Variable	Represents data
Constraint	Controls progression through the diagram path

Constants and Variables

A constant is an item that must be entered as it appears in the diagram, either in full or as an allowable abbreviation. If part of a constant appears in boldface, you can abbreviate the constant by

- Entering only the boldfaced letters
- Entering the boldfaced letters plus any of the remaining letters

If no part of the constant appears in boldface, the constant cannot be abbreviated.

Constants are never enclosed in angle brackets (< >) and are in uppercase letters.

A variable is an item that represents data. You can replace the variable with data that meets the requirements of the particular command or statement. When replacing a variable with data, you must follow the rules defined for the particular command or statement.

In railroad diagrams, variables are enclosed in angle brackets.

In the following example, BEGIN and END are constants, whereas <statement list> is a variable. The constant BEGIN can be abbreviated, since part of it appears in boldface.

```
── BEGIN ──<statement list>── END ──────────────────────────────────┤
```

Valid abbreviations for BEGIN are

- BE
- BEG
- BEGI

Constraints

Constraints are used in a railroad diagram to control progression through the diagram. Constraints consist of symbols and unique railroad diagram line paths. They include

- Vertical bars
- Percent signs
- Right arrows
- Required items
- User-selected items
- Loops
- Bridges

A description of each item follows.

Vertical Bar

The vertical bar symbol (I) represents the end of a railroad diagram and indicates the command or statement can be followed by another command or statement.

```
── SECONDWORD ── ( ──<arithmetic expression>── ) ─────────────────────┤
```

Percent Sign

The percent sign (%) represents the end of a railroad diagram and indicates the command or statement must be on a line by itself.

```
── STOP ───────────────────────────────────────────────────────────────%
```

Right Arrow

The right arrow symbol (>)

- Is used when the railroad diagram is too long to fit on one line and must continue on the next
- Appears at the end of the first line, and again at the beginning of the next line

```
── SCALERIGHT ── ( ──<arithmetic expression>── , ─────────────────────→

→─<arithmetic expression>── ) ────────────────────────────────────────┤
```

Required Item

A required item can be

- A constant
- A variable
- Punctuation

If the path you are following contains a required item, you must enter the item in the command or statement; the required item cannot be omitted.

A required item appears on a horizontal line as a single entry or with other items. Required items can also exist on horizontal lines within alternate paths, or nested (lower-level) diagrams.

In the following example, the word EVENT is a required constant and <identifier> is a required variable:

```
── EVENT ──<identifier>─────────────────────────────────────────┤
```

User-Selected Item

A user-selected item can be

- A constant
- A variable
- Punctuation

User-selected items appear one below the other in a vertical list. You can choose any one of the items from the list. If the list also contains an empty path (solid line) above the other items, none of the choices are required.

In the following railroad diagram, either the plus sign (+) or the minus sign (–) can be entered before the required variable <arithmetic expression>, or the symbols can be disregarded because the diagram also contains an empty path.

```
──┬─────┬─<arithmetic expression>───────────────────────────────┤
  ├─ + ─┤
  └─ – ─┘
```

Loop

A loop represents an item or a group of items that you can repeat. A loop can span all or part of a railroad diagram. It always consists of at least two horizontal lines, one below the other, connected on both sides by vertical lines. The top line is a right-to-left path that contains information about repeating the loop.

Some loops include a return character. A return character is a character—often a comma (,) or semicolon (;)—that is required before each repetition of a loop. If no return character is included, the items must be separated by one or more spaces.

```
   ┌←────── ; ─────┐
───┴─<field value>─┴──────────────────────────────────────────┤
```

Bridge

A loop can also include a bridge. A bridge is an integer enclosed in sloping lines (/ \) that

- Shows the maximum number of times the loop can be repeated
- Indicates the number of times you can cross that point in the diagram

The bridge can precede both the contents of the loop and the return character (if any) on the upper line of the loop.

Not all loops have bridges. Those that do not can be repeated any number of times until all valid entries have been used.

In the first bridge example, you can enter LINKAGE or RUNTIME no more than two times. In the second bridge example, you can enter LINKAGE or RUNTIME no more than three times.

```
   ┌←──────── , ────────┐
───┴─/2\─┬─ LINKAGE ─┬──┴─────────────────────────────────────┤
         └─ RUNTIME ─┘

   ┌←─/2\─────────┐
───┴─┬─ LINKAGE ─┬┴───────────────────────────────────────────┤
     └─ RUNTIME ─┘
```

In some bridges an asterisk (*) follows the number. The asterisk means that you must cross that point in the diagram at least once. The maximum number of times that you can cross that point is indicated by the number in the bridge.

```
   ┌←──────── , ─────────┐
───┴─┬─/2*\─ LINKAGE ─┬──┴────────────────────────────────────┤
     └─ RUNTIME ──────┘
```

In the previous bridge example, you must enter LINKAGE at least once but no more than twice, and you can enter RUNTIME any number of times.

Following the Paths of a Railroad Diagram

The paths of a railroad diagram lead you through the command or statement from beginning to end. Some railroad diagrams have only one path; others have several alternate paths that provide choices in the commands or statements.

The following railroad diagram indicates only one path that requires the constant LINKAGE and the variable <linkage mnemonic>:

```
── LINKAGE ──<linkage mnemonic>──────────────────────────────────┤
```

Alternate paths are provided by

- Loops
- User-selected items
- A combination of loops and user-selected items

More complex railroad diagrams can consist of many alternate paths, or nested (lower-level) diagrams, that show a further level of detail.

For example, the following railroad diagram consists of a top path and two alternate paths. The top path includes

- An ampersand (&)
- Constants that are user-selected items

  These constants are within a loop that can be repeated any number of times until all options have been selected.

The first alternative path requires the ampersand and the required constant ADDRESS. The second alternative path requires the ampersand followed by the required constant ALTER and the required variable <new value>.

```
         ┌←────── , ──────┐
── & ─┬──┴─┬─ TYPE ───────┼──┬───────────────────────────────────┤
      │    ├─ ASCII ──────┤  │
      │    ├─ BCL ────────┤  │
      │    ├─ DECIMAL ────┤  │
      │    ├─ EBCDIC ─────┤  │
      │    ├─ HEX ────────┤  │
      │    └─ OCTAL ──────┘  │
      ├─ ADDRESS ────────────┤
      └─ ALTER ──<new value>─┘
```

Railroad Diagram Examples with Sample Input

The following examples show five railroad diagrams and possible command and statement constructions based on the paths of these diagrams.

Example 1

<lock statement>

```
── LOCK ── ( ── <file identifier> ── ) ─────────────────────────────┤
```

Sample Input	Explanation
LOCK (FILE4)	LOCK is a constant and cannot be altered. Because no part of the word appears in boldface, the entire word must be entered. The parentheses are required punctuation, and FILE4 is a sample file identifier.

Example 2

<open statement>

```
── OPEN ─┬───────────┬─<database name>──────────────────────────────┤
         ├─ INQUIRY ─┤
         └─ UPDATE ──┘
```

Sample Input	Explanation
OPEN DATABASE1	The constant OPEN is followed by the variable DATABASE1, which is a database name. The railroad diagram shows two user-selected items, INQUIRY and UPDATE. However, because an empty path (solid line) is included, these entries are not required.
OPEN INQUIRY DATABASE1	The constant OPEN is followed by the user-selected constant INQUIRY and the variable DATABASE1.
OPEN UPDATE DATABASE1	The constant OPEN is followed by the user-selected constant UPDATE and the variable DATABASE1.

Example 3

<generate statement>

```
── GENERATE ──<subset>── = ─┬─ NULL ─────────────────────┬──────────┤
                            └<subset>─┬──────────────────┤
                                      ├─ AND ─┬─<subset>─┘
                                      ├─ OR ──┤
                                      ├─ + ───┤
                                      └─ – ───┘
```

Sample Input	Explanation
GENERATE Z = NULL	The GENERATE constant is followed by the variable Z, an equal sign (=), and the user-selected constant NULL.
GENERATE Z = X	The GENERATE constant is followed by the variable Z, an equal sign, and the user-selected variable X.
GENERATE Z = X AND B	The GENERATE constant is followed by the variable Z, an equal sign, the user-selected variable X, the AND command (from the list of user-selected items in the nested path), and a third variable, B.
GENERATE Z = X + B	The GENERATE constant is followed by the variable Z, an equal sign, the user-selected variable X, the plus sign (from the list of user-selected items in the nested path), and a third variable, B.

Example 4

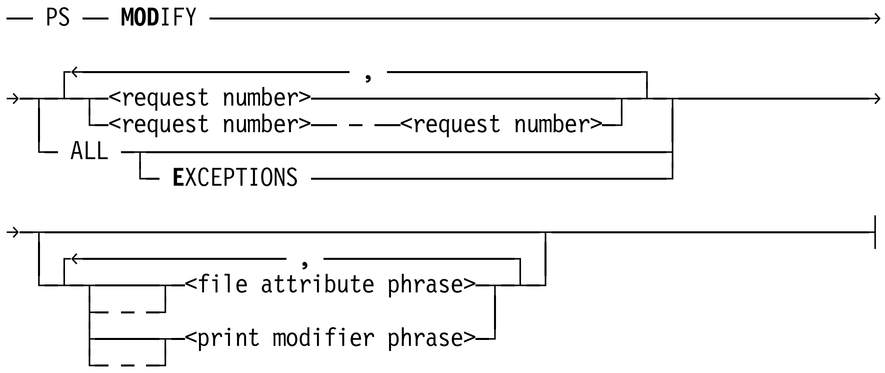
<entity reference declaration>

```
                     ┌←───────────────── , ──────────────────┐
── ENTITY REFERENCE ─┴─<entity ref ID>── ( ──<class ID>── ) ─┴──────────┤
```

Sample Input	Explanation
ENTITY REFERENCE ADVISOR1 (INSTRUCTOR)	The required item ENTITY REFERENCE is followed by the variable ADVISOR1 and the variable INSTRUCTOR. The parentheses are required.
ENTITY REFERENCE ADVISOR1 (INSTRUCTOR), ADVISOR2 (ASST_INSTRUCTOR)	Because the diagram contains a loop, the pair of variables can be repeated any number of times.

Example 5

<PS MODIFY command>



Sample Input	Explanation
PS MODIFY 11159	The constants PS and MODIFY are followed by the variable 11159, which is a request number.
PS MODIFY 11159,11160,11163	Because the diagram contains a loop, the variable 11159 can be followed by a comma, the variable 11160, another comma, and the final variable 11163.
PS MOD 11159–11161 DESTINATION = "LP7"	The constants PS and MODIFY are followed by the user-selected variables 11159–11161, which are request numbers, and the user-selected variable DESTINATION = "LP7", which is a file attribute phrase. Note that the constant MODIFY has been abbreviated to its minimum allowable form.
PS MOD ALL EXCEPTIONS	The constants PS and MODIFY are followed by the user-selected constants ALL and EXCEPTIONS.

Appendix I
Related Product Information

The following documents provide information that is directly related to the primary subject of this manual.

***MCP/AS Security Features Operations and Programming Guide* (8600 0528)**

This guide describes the security features available to users and provides instructions for their use. This guide is written for users who are responsible for maintaining the security of their individual programs and data.

***MCP/AS Work Flow Language (WFL) Programming Reference Manual* (8600 1047)**

This manual presents the complete syntax and semantics of WFL. WFL is used to construct jobs that compile or run programs written in other languages and that perform library maintenance such as copying files. This manual is written for individuals who have some experience with programming in a block-structured language such as ALGOL and who know how to create and edit files using CANDE or the Editor.

***Unisys e-@ction ClearPath Enterprise Servers I/O Subsystem Programming Guide* (8600 0056)**

This guide contains information about how to program for various types of peripheral files and how to program for interprocess communication, using port files. This guide is written for programmers who need to understand how to describe the characteristics of a file in a program. This manual is a companion manual to the *File Attributes Programming Reference Manual*.

***Unisys e-@ction ClearPath Enterprise Servers Print System and Remote Print System Administration, Operations, and Programming Guide* (8600 1039)**

The shortened title for this document is the *Print System Guide*.

This guide describes the features of the Print System and provides a complete description of its command syntax. This guide is written for programmers, operators, system administrators, and other interactive users of Menu-Assisted Resource Control (MARC) and CANDE.

***Unisys e-@ction ClearPath Enterprise Servers Security Administration Guide* (8600 0973)**

The shortened title for this document is the *Security Administration Guide*.

This guide describes system-level security features and suggests how to use them. It provides administrators with the information necessary to set and implement effective security policy. This guide is written for system administrators, security administrators, and those responsible for establishing and implementing security policy.

***Unisys e-@ction ClearPath Enterprise Servers System Commands Operations Reference Manual* (8600 0395)**

This manual gives a complete description of the system commands used to control system resources and work flow. This manual is written for systems operators and administrators.

***Unisys e-@ction ClearPath Enterprise Servers System Operations Guide* (8600 0387)**

This guide describes concepts and procedures required to operate most Unisys systems. Sections 1 and 2 contain information and procedures that can be done by novice operators. Section 3 contains operations and procedures that require more advanced operations experience. This guide is written for operators responsible for operating the enterprise server, especially operators with little or no experience.

***Unisys e-@ction ClearPath Enterprise Servers Task Attributes Programming Reference Manual* (8600 0502)**

This manual describes all the available task attributes. It also gives examples of statements for reading and assigning task attributes in various programming languages. The *Task Management Programming Guide* is a companion manual.

***Unisys e-@ction ClearPath Enterprise Servers Task Management Programming Guide* (8600 0494)**

This guide explains how to initiate, monitor, and control processes on an enterprise server. It describes process structures and process family relationships, introduces the uses of many task attributes, and gives an overview of interprocess communication techniques. The *Task Attributes Programming Reference Manual* is a companion manual.

Index

86000064-508