UNISYS

A Series File Attributes Programming Reference Manual

Priced Item

July 1992
Printed in U S America
8600 0064–100

UNISYS

Product Information Announcement

○ New Release ○ Revision ● Update ○ New Mail Code

Title

A Series File Attributes Programming Reference Manual

This Product Information Announcement announces Update 1 to the July 1992 publication of the *A Series File Attributes Programming Reference Manual.* The update is relative to the Domain Name Service (DNS) Kit, dated June 1993. The kit is based on the Mark 4.0.1 System Software Support Release.

This update includes the following new file attributes:

- MYDOMAINNAME

  This attribute enables you to determine the complete domain name of the local host after the port file has been opened.

- MYIPADDRESS

  This attribute enables you to determine or specify the IP address of a subfile on the local host.

- YOURDOMAINNAME

  This attribute enables you to determine or specify a domain name for the correspondent end point.

- YOURIPADDRESS

  This attribute enables you to determine or specify the IP address for the correspondent end point.

The attributes that can be used with the new TCPIPNATIVESERVICE are identified and the service itself is briefly explained.

Various technical and editorial changes have been made to improve the quality and usability of the document.

Remove	**Insert**
iii through xviii	iii through xx
2–5 through 2–8	2–5 through 2–8
2–39 through 2–40	2–39 through 2–40
2–43 through 2–48	2–43 through 2–48
2–81 through 2–84	2–81 through 2–84
2–129 through 2–130	2–129 through 2–130
2–143 through 2–146	2–143 through 2–146
2–153 through 2–154	2–153 through 2–154
2–163 through 2–164	2–163 through 2–164B
2–177 through 2–178	2–177 through 2–178
	2–184A through 2–184B
2–185 through 2–188	2–185 through 2–188B
2–201 through 2–202	2–201 through 2–202
2–211 through 2–212	2–211 through 2–212

continued

Announcement only:

Announcement and attachments:
AS115

System: A Series
Release: Mark 4.0.1 June 1993

Part number: 8600 0064–110

Remove	Insert
2–229 through 2–230	2–229 through 2–230
2–237 through 2–238	2–237 through 2–238
2–245 through 2–246	2–245 through 2–246B
2–251 through 2–254	2–251 through 2–254
2–259 through 2–264	2–259 through 2–264B
2–269 through 2–270	2–269 through 2–270
2–273 through 2–274	2–273 through 2–274
2–285 through 2–286	2–285 through 2–286
2–291 through 2–292	2–291 through 2–292
2–301 through 2–302	2–301 through 2–302
2–305 through 2–306	2–305 through 2–306B
	2–318A through 2–318B
2–319 through 2–322	2–319 through 2–322B
A–3 through A–8	A–3 through A–8
B–1 through B–4	B–1 through B–4
D–3 through D–8	D–3 through D–8
E–13 through E–16	E–13 through E–16
F–11 through F–14	F–11 through F–14
Glossary–1 through 12	Glossary–1 through 14
Index–1 through 12	Index–1 through 12

Changes are indicated by vertical bars in the margins of the replacement pages.

Retain this Product Information Announcement as a record of changes made to the base publication.

To order additional copies of this document

- United States customers, call Unisys Direct at 1-800-448-1424.
- All other customers, contact your Unisys Sales Office.
- Unisys personnel, use the Electronic Literature Ordering (ELO) system.

UNISYS

Product Information Announcement

○ New Release ● Revision ○ Update ○ New Mail Code

Title

A Series File Attributes Programming Reference. Manual

This Product Information Announcement announces a revision of the *A Series File Attributes Programming Reference Manual*. The revision is relative to the Mark 4.0.0 System Software Release, dated July 1992.

This revision includes the following new file attributes:

- AUTOUNLOAD

  This attribute enables you to control if a tape volume is unloaded when the file is released.

- LOCKEDFILE

  This attribute enables you to prevent disk files from being removed or replaced, or their names changed. It also enables you to prevent tape files from being accidentally purged.

- NETACCESSPOINT

  This attribute enables you to specify the network access point that will be used to access the matching port application when your system is part of an A Series Host LAN Connection (HLCN) network.

- PRINTPARTIAL

  This attribute enables you to indicate that only a portion or portions of a file are to be printed.

Several attributes have new mnemonic values added or new functionality added. The following attributes have been affected:

- DOCUMENTTYPE

  The INTAP1 mnemonic has been added.

- PRINTERKIND

  The ability to identify the type of page description language (PDL) that should be used when printing a file has been added.

- SAVEBACKUPFILE

  The FALSE value no longer always indicates that the printer backup file will be removed. The FALSE value can now mean that the printer backup file will not be removed if the file to be printed has a LOCKEDFILE value of TRUE and the REMOVELOCKEDFILES option of the Print System is AUTOMATIC.

Various technical and editorial changes have been made to improve the quality and usability of the document.

Retain this Product Information Announcement as a record of changes made to the base publication.

continued

Announcement only:

Announcement and attachments:
AS115

System: A Series
Release: Mark 4.0.0 July 1992

Part number: 8600 0064–100

To order additional copies of this document

- United States customers call Unisys Direct at 1-800-448-1424
- All other customers contact your Unisys Subsidiary Librarian
- Unisys personnel use the Electronic Literature Ordering (ELO) system

UNISYS

A Series File Attributes

Programming Reference Manual

Release Mark 4.0.0

Priced Item

July 1992
Printed in U S America
8600 0064–100

The names, places, and/or events used in this publication are not intended to correspond to any individual, group, or association existing, living, or otherwise. Any similarity or likeness of the names, places, and/or events with the names of any individual, living or otherwise, or that of any group or association, is purely coincidental and unintentional.

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.

RESTRICTED RIGHTS LEGEND. Use, reproduction, or disclosure is subject to the restrictions set forth in DFARS 252.227-7013 and FAR 52.227-14 for commercial computer software.

Correspondence regarding this publication may be forwarded using the Product Information card at the back of the manual, or may be addressed directly to Unisys, Technical Publications, 25725 Jeronimo Road, Mission Viejo, CA 92691-2792.

Page Status

Page	Issue
iii through v	–110
vi	Blank
vii through xvii	–110
xviii	Blank
xix	–110
xx	Blank
1–1 through 1–4	–100
2–1 through 2–4	–100
2–5 through 2–8	–110
2–9 through 2–39	–100
2–40	–110
2–41 through 2–43	–100
2–44 through 2–47	–110
2–48 through 2–81	–100
2–82 through 2–83	–110
2–84 through 2–128	–100
2–129 through 2–130	–110
2–131 through 2–142	–100
2–143 through 2–145	–110
2–146 through 2–153	–100
2–154	–110
2–155 through 2–162	–100
2–163 through 2–164A	–110
2–164B	Blank
2–165 through 2–176	–100
2–177	–110
2–178 through 2–184	–100
2–184A	–110
2–184B	Blank
2–185	–110
2–186 through 2–187	–100
2–188 through 2–188A	–110
2–188B	Blank
2–189 through 2–201	–100
2–202	–110
2–203 through 2–210	–100
2–211	–110
2–212 through 2–228	–100
2–229 through 2–230	–110
2–231 through 2–237	–100
2–238	–110
2–239 through 2–245	–100

continued

continued

Page	Issue
2–246 through 2–246A	–110
2–246B	Blank
2–247 through 2–251	–100
2–252 through 2–254	–110
2–255 through 2–258	–100
2–259 through 2–264A	–110
2–264B	Blank
2–265 through 2–268	–100
2–269	–110
2–270 through 2–273	–100
2–274	–110
2–275 through 2–285	–100
2–286	–110
2–287 through 2–290	–100
2–291	–110
2–292 through 2–301	–100
2–302	–110
2–303 through 2–304	–100
2–305 through 2–306A	–110
2–306B	Blank
2–307 through 2–318	–100
2–318A	–110
2–318B	Blank
2–319 through 2–320	–110
2–321	–100
2–322 through 2–322A	–110
2–322B	Blank
2–323 through 2–324	–100
3–1 through 3–10	–100
4–1 through 4–20	–100
A–1 through A–3	–100
A–4 through A–8	–110
B–1 through B–4	–110
C–1 through C–5	–100
C–6	Blank
D–1 through D–3	–100
D–4 through D–7	–110
D–8	Blank

continued

continued

Page	Issue
E–1 through E–13	–100
E–14 through E–15	–110
E–16	–100
F–1 through F–10	–100
F–11 through F–13	–110
F–14	Blank
Glossary–1 through 14	–110
Bibliography–1 through 2	–100
Index–1 through 11	–110
Index–12	Blank

Unisys uses an 11-digit document numbering system. The suffix of the document number (1234 5678–*xyz*) indicates the document level. The first digit of the suffix (*x*) designates a revision level; the second digit (*y*) designates an update level. For example, the first release of a document has a suffix of –000. A suffix of –130 designates the third update to revision 1. The third digit (*z*) is used to indicate an errata for a particular level and is not reflected in the page status summary.

About This Manual

Purpose

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

Scope

This manual is a reference manual and does not explain how to use file attributes in a program or how to modify file attribute values through the Work Flow Language (WFL). For information about how to use file attributes in a program, refer to the *A Series I/O Subsystem Programming Guide* and the appropriate Volume 1 language manual. For information about how to modify a file attribute value using a WFL program, refer to the *A Series Work Flow Language (WFL) Programming Reference Manual.*

Audience

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

Prerequisites

If you are an ALGOL programmer, you should know how to describe files in a program using file attributes before you read this manual. Refer to the *I/O Subsystem Programming Guide* for that information. If you are a programmer of another language, refer to the appropriate language manual for instructions about describing files.

If you are an operator, you should know how to modify the value of a file attribute through the WFL. Refer to the *WFL Reference Manual* for information about modifying file attribute values.

How to Use This Manual

This manual is a reference tool in which each section can be used independently of the other sections.

File attributes are deimplemented when they are no longer needed. The announcement of such deimplementations are made in the Mark Release Capabilities Overview for the release. During the deimplementation warning period, the system issues a warning message whenever you interrogate or modify the affected attribute. You can suppress the warning message by using the SUPPRESSWARNING task attribute or system command.

When an attribute value contains no characters, the phrase *null string* is used in this manual.

Documents relevant to this product are listed with complete titles in the "Related Product Information" portion of this preface and in the Bibliography. In text, documents may be referred to by their complete titles or by a shortened version of their titles. Unless otherwise noted, documents referred to in the text of this manual are for A Series systems.

Organization

The information in this manual is divided into four sections and six appendixes. In addition, a glossary, a bibliography, and an index appear at the end of this manual.

Section 1. Understanding File Attributes

This section describes how file attributes function in the A Series environment.

Section 2. General File Attributes

This section includes detailed descriptions of the general file attributes.

Section 3. User-Defined Disk File Attributes (UDDFAs)

This section describes how to use library procedures to create, set or modify, interrogate, or remove user-defined file attributes for a disk file.

Section 4. Direct I/O Buffer Attributes

This section includes descriptions of the buffer attributes that are used with direct I/O files.

Appendix A. Device Types and Associated File Attributes

This appendix contains a table that identifies the file attributes that can be used with each device type.

Appendix B. Port File Attributes and Associated Services

This appendix contains a table that identifies file attributes that are unique to port files and the services available for any given file attribute. Another table identifies file attributes that can be used by all devices, but are not the preferred attributes for port files.

Appendix C. Host Services Logical I/O and FTAM File Attributes

This appendix contains a table that identifies the file attributes that can be used with Host Services logical I/O and File Transfer, Access, and Management (FTAM).

Appendix D. Attribute Numbers

This appendix contains two lists. The first list presents the attribute numbers numerically and the second list presents the attribute numbers alphabetically by attribute name.

Appendix E. OPEN, CLOSE, and RESPOND Results

This appendix contains two tables. The first table lists the open, close, and respond results and the second table lists the open and close warnings.

Appendix F. I/O Result Enumerated Values

This appendix lists the mnemonic values and integer values returned by specific attributes, and explains their meanings.

Related Product Information

A Series 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. The *A Series File Attributes Programming Reference Manual* is a companion manual.

A Series Print System (PrintS/ReprintS) Administration, Operations, and Programming Guide **(8600 1039)**

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.

A Series Security Administration Guide **(8600 0973)**

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.

A Series 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.

A Series System Operations Guide **(8600 0387)**

This guide describes the basic concepts and procedures required to operate Micro A through A 6 systems and, more generally, all A Series systems. This guide is written for A Series operators, especially those with little or no experience.

A Series 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.

Contents

Tables

Examples

Tables

Section 1
Understanding File Attributes

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 open 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.

The process of opening a logical file involves either searching for a device or a place on a device on which to create the 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 should be taken with the physical file after the association is broken. For example, a disk file can be discarded 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 the tape to 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 multiple files on a multifile tape, for example. The ALGOL statements CLOSE(F,*) and CLOSE(F,REWIND) and the COBOL or COBOL74 statement CLOSE FILEID leave the logical file assigned to the physical file.

Thus, a logical file can be in one of four basic states: open and assigned, open and unassigned, closed and assigned, or closed and unassigned.

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 logical attribute 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 the 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 logical frame sizes.

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.

Example

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=9Ø,BLOCKSIZE=18Ø,NEWFILE=TRUE,             (1)
      FILENAME="TEST.",FRAMESIZE=8,AREALENGTH=36ØØ);
ARRAY
    A[Ø: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 9Ø;
REPLACE POINTER(A) BY "THIS FILE'S FILENAME IS:  ", F.FILENAME;
F.SYNCHRONIZE :=VALUE(OUT);                                            (6)
WRITE(F,9Ø,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=18Ø,FILENAME=TEST2);                             (2)
RUN OBJECT/FILE/PROGRAM;
  FILE F(BLOCKSIZE=252Ø,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) superceding the values of BLOCKSIZE and FILENAME assigned in (1) and the values of BLOCKSIZE, FILENAME, and NEWFILE assigned in (3) superceding 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.

continued

continued

Notation Number	Explanation
(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 *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.

Section 2
General File Attributes

This section presents file attribute descriptions alphabetically. 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.

The following characteristics are identified in the formatted list:

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.
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.
Host Services Logical I/O	Indicates whether 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.

Tables 2-1 through 2-10 describe the values that are used with the preceding characteristics.

Table 2–1. Possible Values for Kinds

Value	Meaning
All	Relevant to all KIND values.
CD	Relevant to CD-ROM and Write-Once-Read-Many (WORM) files.
Direct I/O disk	Relevant to direct I/O disk files.
Disk (DISK and PACK)	Relevant to a disk file.
Duplicated disk	Relevant to disks files that are duplicated.
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.
Punch	Relevant to a punch card file.
Reader	Relevant to a card reader file.
Remote	Relevant to a remote file.
Tape	Relevant to a tape file.

Table 2–2. Possible Values for 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.

Table 2–3. Possible Values for 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.
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.

Table 2–4. Possible Values for Type

Value	Meaning
Boolean	The attribute value is a Boolean.
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 (.).
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.

Table 2–5. Possible Values for 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

Table 2–6. Possible Values for Stored Permanently

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

Table 2–7. Possible Values for 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

Table 2–8. Possible Values for 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.

continued

Table 2–8. **Possible Values for FTAM File Access** (cont.)

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

Table 2–9. Possible Values for Host Services Logical I/O

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

Table 2–10. Possible Values for 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.
TCPPUSHEDMSGSERVICE	This attribute is available for the TCPPUSHEDMSGSERVICE.

continued

Table 2–10. Possible Values for Port Services (cont.)

Value	Meaning
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.*

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
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
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 and TCPPUSHEDMSGSERVICE used with the 3.1 and later releases of TCP/IP, interrogate ACTUALMAXRECSIZE to determine the maximum segment size, in FRAMESIZE units, transmitted by the TCP provider to its peer.

For TCPNATIVESERVICE and TCPPUSHEDMSGSERVICE used with the 1.2 and 3.0 releases of TCP/IP, the ACTUALMAXRECSIZE value is equal to the REQUESTEDMAXRECSIZE value.

The ACTUALMAXRECSIZE attribute requires a subfile index as a parameter if MAXSUBFILES is greater than 1.

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 network implementation restrictions or because of record size negotiation between the two dialogue endpoints.

The following length restrictions apply to the network implementations:

Network Implementation	Range
BNA Version 1	1 through 65487
BNA Version 2 local dialogue	1 through 65513
BNA Version 2 remote dialogue	1 through 20000
HLCN	1 through 65435
OSI	1 through 64512
Transmission Control Protocol (TCP) remote dialogue for the 1.2 and 3.0 releases	1 through 20000
TCP remote dialogue for the 3.1 and later releases	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 A Series system might have a REQUESTEDMAXRECSIZE default value that is different from that of the A Series 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 A Series 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 and TCPPUSHEDMSGSERVICE used with the 3.1 and later releases of TCP/IP, 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.

AFTER

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

Use the AFTER attribute to indicate when printing or punching of the printer or punch backup file should start. The string value must conform to the syntax for the <starttime spec> variable, which is defined in the *A Series 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 *A Series Print System (PrintS/ReprintS) Administration, Operations, and Programming Guide.*

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

The following are examples of valid string values:

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

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

ALIGNFILE

Kinds	Printer
Interrogate	Anytime
Modify	When closed
Type	Character string
Range	Not applicable
Default	Null string
Stored Permanently	Backup file
Parameters	None
FTAM File Access	Not supported
Host Services Logical I/O	Not 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 used only for printer backup files, and is not valid when the PRINTDISPOSITION attribute has the value DIRECT.

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 A Series disk file name.

ALIGNMENT

Kinds	Printer
Interrogate	Anytime
Modify	When closed
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Stored Permanently	Backup file
Parameters	None
FTAM File Access	Not supported
Host Services Logical I/O	Not 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 prespecified 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 used only for printer backup files, and is not valid when the PRINTDISPOSITION attribute has the value DIRECT.

ALTERDATE

Kinds	Disk or CD
Interrogate	When assigned
Modify	Never
Type	Integer
Range	0 through 99999
Default	Not applicable
Stored Permanently	Disk or CD
Parameters	None
FTAM File Access	Supported
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 file following some alteration to the data in the file.

The value is returned as a decimal integer in the form YYDDD, where YY and DDD represent the year and day, respectively, in Julian form. ALTERDATE is unaffected by library maintenance. Changing a permanent file attribute – for example, FILENAME or SECURITYTYPE – does not cause the ALTERDATE of the file to change.

ALTERTIME

Kinds	Disk or CD
Interrogate	When assigned
Modify	Never
Type	Integer
Range	0 through 86400000000
Default	Not applicable
Stored Permanently	Disk or CD
Parameters	None
FTAM File Access	Supported
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 changed only when the ALTERDATE value is changed.

ANYSIZEIO

Kinds	Disk or CD
Interrogate	Anytime
Modify	When closed or closed with retention
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
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 and the length of the file. 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 FILESTRUCTURE attribute value is not STREAM, or the BLOCKSTRUCTURE attribute value is not FIXED, or 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 in this section for more information.

APL

Kinds	Disk
Interrogate	When open
Modify	When closed
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
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 value of APL cannot be altered by file equation.

The APL attribute can also be set to TRUE by using the MA (May Access) system command, described in the *A Series System Commands Operations Reference Manual.*

APPLICATIONCONTEXT

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

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 counterproposing application context during dialogue establishment. Refer to the *I/O Subsystem Programming Guide* for a discussion about negotiation of application context.

The APPLICATIONCONTEXT attribute requires a subfile index as a parameter if the MAXSUBFILES value is greater than 1.

APPLICATIONGROUP

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

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.

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.

The APPLICATIONGROUP attribute is not supported by the BNA Version 1 implementation.

AREAALLOCATED

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

Interrogate the AREAALLOCATED attribute to determine whether or not a specific area of a disk file has been allocated. At least one parameter, the area number, is required. The copy number parameter is also required if the DUPLICATED attribute value of the physical file is TRUE. Area numbers begin at 0; copy numbers begin at 1.

The following ALGOL examples show how to interrogate the AREAALLOCATED attribute:

```
BOOL := F(AREANO).AREAALLOCATED;          % FOR NONDUPLICATED FILES

BOOL := F(AREANO,COPYNO).AREAALLOCATED;   % FOR DUPLICATED FILES
```

AREALENGTH

Kinds	Disk or CD
Interrogate	Anytime
Modify	When unassigned
Type	Integer
Range	0 through 4294967295
Default	See following text
Stored Permanently	Disk or CD
Parameters	None
FTAM File Access	Not supported
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.

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, the AREALENGTH value is rounded up so that it is divisible by the BLOCKSIZE value. The default value of AREALENGTH is the multiple of BLOCKSIZE closest to but not exceeding 184320 bytes, or BLOCKSIZE if the BLOCKSIZE value exceeds 184320 bytes. If the AREALENGTH value is less than the BLOCKSIZE value, the AREALENGTH value is changed to the BLOCKSIZE value.

When a file with a FILESTRUCTURE value of STREAM is created, the AREALENGTH value is not adjusted unless the AREALENGTH value is less than the MAXRECSIZE value. In that case, the AREALENGTH value is changed to the MAXRECSIZE value. The default value of AREALENGTH is the multiple of MAXRECSIZE closest to but not exceeding 184320 bytes, or the MAXRECSIZE value if the MAXRECSIZE value exceeds 184320 bytes.

The length of an area cannot exceed 16777215 sectors.

WORM and CD-ROM files always have an AREAS attribute value of 1.

When a WORM file is opened for output, the AREALENGTH value is the maximum possible capacity of the file in FRAMESIZE units. When an existing WORM or CD-ROM file is opened for input, the AREALENGTH value is the actual length of the file in FRAMESIZE units.

If you specify an AREALENGTH attribute value when a WORM file is opened for output, that value is ignored. The WORM media supports only one file opened for output at a time. The opened file is assigned an area containing the remaining data space on the WORM media or the result of the calculation 2**32–1 bytes, whichever is smaller. The data must be written contiguously from the beginning of the assigned area. When the file is closed, the AREALENGTH value is updated to the actual number of FRAMESIZE units in the file.

AREAS

Kinds	Disk or CD
Interrogate	Anytime
Modify	When unassigned
Type	Integer
Range	0 through 1000
Default	20
Stored Permanently	Disk or CD
Parameters	None
FTAM File Access	Not supported
Host Services Logical I/O	Supported
Port Services	Not applicable

Use the AREAS attribute to specify or determine the number of areas that can be allocated for a disk file or determine the number of areas currently allocated. 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 unless the FLEXIBLE attribute value is FALSE. When areas are added, the AREAS value is increased appropriately.

The length of an area cannot exceed 16777215 sectors.

WORM and CD-ROM files always have an AREAS value of 1. If you specify a value for the AREAS attribute when a WORM file is opened for output, the value is ignored. For information about programming for WORM or CD-ROM files, refer to the *I/O Subsystem Programming Guide.*

AREASECTORS

Kinds	Disk or CD
Interrogate	When assigned
Modify	Never
Type	Integer
Range	1 through 16777215
Default	Not applicable
Stored Permanently	Disk or CD
Parameters	None
FTAM File Access	Not supported
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, unoccupied disk space in the last area is released for reallocation. This last area then contains 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.

WORM and CD-ROM files always have an AREAS attribute value of 1.

When a WORM file is opened for output, the AREASECTORS value is the maximum possible number of sectors for the area. When an existing WORM or CD-ROM file is opened for input, the AREASECTORS value is the total number of sectors for the file.

If you specify an AREASECTORS attribute value when a WORM file is opened for output, that value is ignored. The WORM media supports only one file opened for output at a time. The opened file is assigned an area containing the remaining data space on the WORM media or the result of the calculation 2**32–1 bytes, whichever is smaller. The data must be written contiguously from the beginning of the assigned area. When the file is closed, the AREASECTORS value is updated to the actual number of sectors in the file.

AREASIZE

Kinds	Disk or CD
Interrogate	Anytime (local host)
	When unassigned (remote host)
Modify	When unassigned
Type	Integer
Range	0 through 1048575
Default:	See following text
Stored Permanently	Disk or CD
Parameters	None
FTAM File Access	Not supported
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.

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.

The length of an area cannot exceed 16777215 sectors.

WORM and CD-ROM files always have an AREAS attribute value of 1.

When a WORM file is opened for output, the AREASIZE value is the maximum number of records that the file can contain. When an existing WORM or CD-ROM file is opened for input, the AREASIZE attribute value is the actual number of records in the file.

If you specify a value for the AREASIZE attribute when the WORM file is opened for output, the value is ignored. The WORM media supports only one file opened for output at a time. The opened file is assigned an area containing the remaining data space on the WORM media or the result of the calculation 2**32–1 bytes, whichever is smaller. The data must be written contiguously from the beginning of the assigned area. When the file is closed, the AREASIZE value is updated to 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.

When a file with a FILESTRUCTURE value of BLOCKED is created, the AREASIZE value is rounded up so that it is a multiple of number of records in a block. The default value of AREASIZE is the AREALENGTH default value divided by the BLOCKSIZE value and multiplied by the number of records in a block. Because AREASIZE is expressed in blocks when the BLOCKSTRUCTURE value is other than FIXED, the number of records per block is considered 1 for such BLOCKSTRUCTURE values, for the purpose of determining the AREASIZE value.

When a file with a FILESTRUCTURE value of STREAM is created, the AREASIZE attribute is meaningful only if the BLOCKSTRUCTURE value is FIXED. If the AREASIZE value is less than 1, the value is changed to 1. For all other values, the AREASIZE value is not changed. The default value of AREASIZE is the AREALENGTH default value divided by the MAXRECSIZE attribute value.

ASSIGNTIME

Kinds	Remote
Interrogate	When open
Modify	Never
Type	Integer
Range	0 through 86400
Default	Not applicable
Stored Permanently	No
Parameters	1 required
FTAM File Access	Not supported
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;
```

ATTERR

Kinds	All
Interrogate	Anytime
Modify	Never
Type	Boolean
Range	TRUE, FALSE
Default	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Supported
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.

ATTVALUE

Kinds	All
Interrogate	Anytime
Modify	Never
Type	Real
Range	See following text
Default	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Supported
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 in this section.

ATTYPE

Kinds	All
Interrogate	Anytime
Modify	Never
Type	Mnemonic
Range	See following text
Default	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Supported
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
Stored Permanently	No
Parameters	None
FTAM File Access	Not Supported
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.

† 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
Stored Permanently	No
Parameters	None
FTAM File Access	Supported
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. However, the operator is required to resolve duplicate file situations when they arise. 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
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
Host Services Logical I/O	Not supported
Port Services	All

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.

The AVAILABLEONLY attribute requires a subfile index as a parameter if the value of the MAXSUBFILES attribute is greater than 1.

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 the correspondent host 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	If the port file is establishing a dialogue with a local correspondent endpoint, the following actions occur depending on the value of the AVAILABLEONLY attribute: ● If AVAILABLEONLY is set to FALSE, the OPEN operation is suspended until a dialogue is established, or until the CONNECTTIMELIMIT value is exceeded. ● If AVAILABLEONLY is set to TRUE and no match is found, the OPEN operation fails, your program resumes, and a NOFILEFOUND (4) SUBFILEERROR is returned and the subfile is not available for subsequent matching. When your port file is establishing a dialogue with a remote correspondent endpoint, the AVAILABLEONLY attribute has no effect.
TCP	The AVAILABLEONLY attribute does not affect TCP.

BACKUPKIND

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

Use the BACKUPKIND attribute in conjunction with the KIND attribute to specify the printer and punch backup devices 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.
TAPE7	13	Use a 7-track nonreturn to zero (NRZ) tape.
TAPE9	14	Use a 9-track NRZ tape.
TAPEPE	15	Use a 9-track phase-encoded (PE) or group-coded recording (GCR) tape.

Note: *The mnemonics TAPE7, TAPE9, and TAPEPE will be deimplemented in a future release.*

The BACKUPKIND value is used when the file is opened to select an output device for printer and punch backup files. When no output devices are specified with the BACKUPKIND attribute, other system and task options are used to select an online printer or punch device. 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 or punch file, or if the file is directly attached to a printer or punch device, a value of DONTCARE is returned.

Backup files cannot be unlabeled.

BANNER

Kinds	Printer or punch
Interrogate	Anytime
Modify	Anytime
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Stored Permanently	Backup file
Parameters	None
FTAM File Access	Not supported
Host Services Logical I/O	Not supported
Port Services	Not applicable

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

The content of the banner page or card 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 captured by using the PS BANNER command, 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 and punch files
Interrogate	Anytime
Modify	Anytime
Type	Mnemonic
Range	NULL, ZERO
Default	NULL
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Host Services Logical I/O	Not supported
Port Services	BNANATIVESERVICE

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
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
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	Anytime
Type	Integer
Range	0 through 1440 minutes
Default	0
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
Host Services Logical I/O	Not supported
Port Services	BNANATIVESERVICE, OSINATIVESERVICE, OSISESSIONSERVICE, TCPIPNATIVESERVICE, TCPNATIVESERVICE, TCPPUSHEDMSGSERVICE, BASICSERVICE

If you are using BNANATIVESERVICE, OSINATIVESERVICE, OSISESSIONSERVICE, TCPNATIVESERVICE or TCPPUSHEDMSGSERVICE with the 1.2 or 3.0 TCP/IP release, 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 or TCPPUSHEDMSGSERVICE with the 3.1 or later TCP/IP release, 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.

The BLOCKEDTIMEOUT attribute is not supported by the BNA Version 1 implementation.

If the MAXSUBFILES value is greater than 1, the BLOCKEDTIMEOUT attribute requires a subfile index as a parameter.

BLOCKSIZE

Kinds	All except CD and port
Interrogate	Anytime
Modify	When closed
Type	Integer
Range	0 through 65535
Default	MAXRECSIZE
Stored Permanently	Disk or tape
Parameters	None
FTAM File Access	Not supported
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.

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.

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 in this section.

When an existing disk or tape file is being opened and the DEPENDENTSPECS attribute value is FALSE or the FILETYPE attribute value is not 7 or 8, the logical file BLOCKSIZE value is checked against the BLOCKSIZE value of the physical file for consistency. If an inconsistency is detected, a run-time warning is issued.

The disk file is considered inconsistent if one of the following conditions is true:

- The physical BLOCKSIZE value is not a multiple of the disk sector size, and the logical BLOCKSIZE value is unequal to the physical BLOCKSIZE value.
- The physical BLOCKSIZE is a multiple of the disk sector size, and the value of the logical BLOCKSIZE is not a multiple of the disk sector size.
- 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 has a logical BLOCKSIZE value unequal to the physical BLOCKSIZE value.

Note: *In a future release, opening an existing disk file with a logical BLOCKSIZE value unequal to the physical BLOCKSIZE value will not be allowed unless 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.

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 4, "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 or BACKUPPUNCH.

Note: *In a future release, 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
Stored Permanently	Disk or tape
Parameters	None
FTAM File Access	Restricted values
Host Services Logical I/O	Restricted values
Port Services	BNANATIVESERVICE, NETBIOSSESSIONSERVICE, OSINATIVESERVICE, OSISESSIONSERVICE, TCPNATIVESERVICE, TCPPUSHEDMSGSERVICE, 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.

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.
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.
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.

continued

continued

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.

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.

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 after each READ operation. If the BLOCKSTRUCTURE value is EXTERNAL, only the data received by the service provider is placed into the application buffer. In either case, you can determine the length of the data placed into the application buffer by interrogating either the CURRENTRECORDLENGTH attribute or the STATE attribute.

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
Stored Permanently	No
Parameters	1 optional (BNA Version 2 port files)
FTAM File Access	Not supported
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 or when duplicated disk files are written.

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.

The BUFFERS attribute is not supported by the BNA Version 1 implementation.

If the MAXSUBFILES value is greater than 1, the subfile index parameter is required.

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
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
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, do not specify a value for this attribute. The default value is either a value assigned by the site administrator or a system default value that should provide adequate I/O performance without placing an undue burden on memory resources. For information about how a site administrator can assign a default value for BUFFERSIZE, refer to the discussion of the BUFFERGOAL factor in the description of the SF (Set Factor) system command in the *System Commands Reference Manual*.

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
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
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

<table>
<tr><th>Mnemonic Value</th><th>Integer Value</th><th>Meaning</th></tr>
<tr><td>CTLASA</td><td>1</td><td>
<table>
<tr><th>Character</th><th>Carriage Motion</th></tr>
<tr><td>7</td><td>Skip to channel 7</td></tr>
<tr><td>8</td><td>Skip to channel 8</td></tr>
<tr><td>9</td><td>Skip to channel 9</td></tr>
</table>
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".
</td></tr>
<tr><td>CTL360</td><td>2</td><td>
The first character controls carriage motion using the various fields in the character as follows:
<table>
<tr><th>Field</th><th>Action</th></tr>
<tr><td>[0:1]</td><td>If this field is 1, printing occurs before carriage control is performed.</td></tr>
<tr><td>[1:1]</td><td>If this field is 1, no printing occurs, only carriage control is performed.</td></tr>
<tr><td>[2:1]</td><td>Ignored.</td></tr>
<tr><td>[6:4]</td><td>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].</td></tr>
<tr><td>[7:1]</td><td>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].</td></tr>
</table>
</td></tr>
<tr><td>STANDARD</td><td>0</td><td>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 BEFORE/AFTER ADVANCING syntax of COBOL, COBOL74, and COBOL85.</td></tr>
</table>

CCSVERSION

Kinds	Disk
Interrogate	Anytime
Modify	When unassigned
Type	Mnemonic
Range	See following text
Default	NOTSET
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
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 *A Series MultiLingual System (MLS) Administration, Operations, and Programming Guide.*

The CCSVERSION attribute has the following mnemonic and integer values:

Mnemonic Value	Integer Value	Meaning
ASERIESNATIVE	0	Use the rules of the default A Series ccsversion.
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 Czechoslovakian ccsversion.
FRANCE	35	Use the rules of the French ccsversion.
LATINGREEK	49	Use the rules of the Latin-Greek ccsversion.
NORWAY	71	Use the rules of the Norwegian 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.

CENSUS

Kinds	Remote, port, or port subfile
Interrogate	Anytime
Modify	Never
Type	Integer
Range	0 through 65535
Default	Not applicable
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
Host Services Logical I/O	Supported
Port Services	BNANATIVESERVICE, NETBIOSSESSIONSERVICE, OSINATIVESERVICE, OSISESSIONSERVICE, TCPPUSHEDMSGSERVICE, 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.

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

CHANGEDSUBFILE

Kinds	Port
Interrogate	Anytime
Modify	Never
Type	Integer
Range	0 through the MAXSUBFILES attribute value
Default	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
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.

CHANGEEVENT

Kinds	Port or port subfile
Interrogate	Anytime
Modify	Never
Type	Event
Range	Not applicable
Default	Not applicable
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
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 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 *A Series Task Management Programming Guide* or the *A Series ALGOL Programming Reference Manual, Volume 1: Basic Implementation*.

CHECKPOINT

Kinds	Printer
Interrogate	Anytime
Modify	When closed
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Host Services Logical I/O	Not 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 PRINTDISPOSITION attribute value is DIRECT or if the printer file is a direct I/O file. 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.*

COMPRESSING

Kinds	Port subfile
Interrogate	Anytime
Modify	Never
Type	Boolean
Range	TRUE, FALSE
Default	Not applicable
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
Host Services Logical I/O	Not supported
Port Services	All

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

Compression is supported only by BNANATIVESERVICE and BASICSERVICE. For services that do not support compression, the COMPRESSING attribute always returns the value FALSE.

If the MAXSUBFILES value is greater than 1, the COMPRESSING attribute requires a subfile index as a parameter.

COMPRESSION

Kinds	Port subfile
Interrogate	Anytime
Modify	Anytime
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
Host Services Logical I/O	Not supported
Port Services	BNANATIVESERVICE, BASICSERVICE

For BNA Version 1 port files, set the COMPRESSION attribute to TRUE to specify that data being sent by this subfile is to be compressed during transmission. Compression could improve efficiency by stripping contiguous characters such as blanks, NULs, and ESCs. Support for the compression feature is negotiated when the subfile is opened. Changing the COMPRESSION value to TRUE while the subfile is open has an effect only if both hosts involved in the subfile dialogue support compression and the COMPRESSIONCONTROL attribute value is USER. If compression is not supported on both hosts involved, the COMPRESSION value is changed to FALSE even if you explicitly changed the value to TRUE. If compression is supported, your program can request that specific records be compressed by changing the COMPRESSION value.

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

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.

For BNA Version 2 port files, if you interrogate the COMPRESSION attribute, the COMPRESSING value is returned. When you modify the COMPRESSION value, the COMPRESSIONREQUESTED value is also modified. Use of the COMPRESSING and COMPRESSIONREQUESTED attributes is preferred.

COMPRESSIONCONTROL

Kinds	Port subfile
Interrogate	Anytime
Modify	Anytime
Type	Mnemonic
Range	USER, SYSTEM
Default	USER
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
Host Services Logical I/O	Not supported
Port Services	BNANATIVESERVICE, BASICSERVICE

Use the COMPRESSIONCONTROL attribute to specify whether you 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.

The mnemonic values and integer values for this attribute follow:

Mnemonic Value	Integer Value
SYSTEM	1
USER	0

During dialogue establishment, both hosts negotiate for support of compression. If the COMPRESSIONCONTROL value is SYSTEM and compression can be supported by both hosts, 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 hosts, compression is controlled by the value of the COMPRESSIONREQUESTED attribute. In this case, the COMPRESSING attribute reflects the COMPRESSIONREQUESTED value.

If you are using the BNA Version 1 implementation, modification of the COMPRESSIONCONTROL attribute is not supported and SYSTEM is returned when COMPRESSIONCONTROL is interrogated.

If the MAXSUBFILES value is greater than 1, the COMPRESSIONCONTROL attribute requires a subfile index parameter.

COMPRESSIONREQUESTED

Kinds	Port subfile
Interrogate	Anytime
Modify	Anytime
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
Host Services Logical I/O	Not supported
Port Services	BNANATIVESERVICE, BASICSERVICE

Set the COMPRESSIONREQUESTED attribute to TRUE to specify that contiguous characters such as blanks, NULs, and ESCs are to be stripped from data during transmission on a subfile that has the COMPRESSIONCONTROL attribute value of USER.

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

The COMPRESSIONREQUESTED attribute is not supported by the BNA Version 1 implementation.

If the MAXSUBFILES value is greater than 1, COMPRESSIONREQUESTED requires a subfile index as a parameter.

COPIES

Kinds	Disk
Interrogate	Anytime
Modify	When closed
Type	Integer
Range	1 through 15
Default	2
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Host Services Logical I/O	Supported
Port Services	Not applicable

If the DUPLICATED attribute is TRUE, use the COPIES attribute to specify the number of file copies to be created and maintained.

The COPIES attribute returns the number of physical copies of the file if it is interrogated when the file is assigned.

COPYINERROR

Kinds	Disk
Interrogate	When open
Modify	Never
Type	Integer
Range	0 and any valid COPIES attribute value
Default	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Host Services Logical I/O	Supported
Port Services	Not applicable

If the DUPLICATED attribute is TRUE, interrogate the COPYINERROR attribute to determine if an error occurred during the last I/O operations on any of the copies of the file. If an error occurred, the COPYINERROR attribute returns the copy number of the file copy in error. If no error has occurred, the COPYINERROR attribute returns a 0 (zero). If the COPYINERROR attribute returns a nonzero value, successive interrogations of the COPYINERROR attribute return the copy numbers of additional copies that contained errors on the last I/O operation. A final value of 0 indicates that no other copies were in error.

When the COPYINERROR attribute returns a nonzero value, the corresponding bit in the duplicated disk file error mask field ([27:16]) of the STATE attribute word is turned off. Refer to the description of the STATE attribute for more information about that error mask field.

COPYNAME

Kinds	Disk
Interrogate	When assigned
Modify	Never
Type	Character string
Range	See following text
Default	Not applicable
Stored Permanently	Disk
Parameters	1 required
FTAM File Access	Not supported
Host Services Logical I/O	Supported
Port Services	Not applicable

If the DUPLICATED attribute is TRUE, interrogate the COPYNAME attribute to determine the external file name of the specified copy of a duplicated file. This attribute can be interrogated only when the logical file is assigned to a physical disk file. The COPYNAME attribute requires the copy number as a parameter. A null string is returned if the copy number specified is greater than the number of copies.

If, for example, a duplicated disk file is created with the external file name AAA, the copies are given external file names of AAA/"COPY#01", AAA/"COPY#02", and so on. The copies of a permanent file are assigned indexes each time the file is opened, and the indexes are assigned to the copies in the order in which they appear in the directory. Because the file name of any copy can be changed, or the copy itself can be removed, the index assigned to a copy or the number of copies can change for subsequent openings of the file.

For further information regarding duplicated files, refer to the *I/O Subsystem Programming Guide.*

CREATEPASSWORD

Kinds	Disk
Interrogate	Anytime
Modify	When closed
Type	Character string
Range	Not applicable
Default	Null string
Stored Permanently	No
Parameters	None
FTAM File Access	Supported
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 later in this section.

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
Type	Integer
Range	0 through 99999
Default	Not applicable
Stored Permanently	Disk, CD, or tape
Parameters	None
FTAM File Access	Supported
Host Services Logical I/O	Supported
Port Services	Not applicable

Interrogate the CREATIONDATE attribute to determine the date when the file was first opened for creation. Each value returned by CREATIONDATE is an integer in the form YYDDD, where YY 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 attribute. For more information, refer to the description of the CREATIONTIME attribute.

CREATIONTIME

Kinds	Disk or CD
Interrogate	When assigned
Modify	Never
Type	Integer
Range	0 through 86400000000
Default	Not applicable
Stored Permanently	Disk
Parameters	None
FTAM File Access	Supported
Host Services Logical I/O	Supported
Port Services	Not applicable

Interrogate the CREATIONTIME attribute to determine the time of day, in microseconds since midnight, when the file was first opened for creation. The values returned by the CREATIONTIME attribute are unaffected by library maintenance.

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

CRUNCHED

Kinds	Disk or CD
Interrogate	When assigned
Modify	Never
Type	Boolean
Range	TRUE, FALSE
Default	Not applicable
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
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.

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
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
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
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Supported
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.

When a WRITE operation to a WORM file threatens to exceed the end of the WORM media, the file is truncated to the end of the WORM media. If the file has a MAXRECSIZE value greater than 1, the file ends in the middle of a record.

For files other than port files and 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 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 actual quantity of data copied into the buffer.

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 MAXRECSIZE attribute

Remote Files

The value of CURRENTRECORDLENGTH is the smallest of one 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
Stored Permanently	Disk or tape
Parameters	None
FTAM File Access	Not supported
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, 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 *A Series 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
Stored Permanently	Tape
Parameters	None
FTAM File Access	Not supported
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	KIND Values
BPI200	2	7-track NRZ reel-to-reel tape	TAPE, TAPE7
BPI556	1	7-track NRZ reel-to-reel tape	TAPE, TAPE7
BPI800	0	7- or 9-track NRZ reel-to-reel tape	TAPE, TAPE7, TAPE9
BPI1250	6	Quarter-inch cartridge tape	TAPE
BPI1600	3	9-track PE reel-to-reel tape	TAPE, TAPEPE, PETAPE
BPI6250	4	9-track GCR reel-to-reel tape	TAPE, TAPEPE, PETAPE
BPI11000	7	8mm cartridge tape	TAPE
BPI38000	5	Half-inch cartridge tape	TAPE

The default DENSITY value for output files is the density setting of the tape unit. The default DENSITY value for input files is the density at which the tape was written.

When you create a tape file, specify the density by using one of the density values. The I/O subsystem sets the specified density if both of the following conditions are true:

- The requested density is valid for the selected tape unit.
- The selected tape unit supports software selection of density.

Until a future release, when you create a file, specifying a DENSITY value does not necessarily mean a certain tape drive is selected. The following two cases are possible:

- If the specified value is BPI1250, BPI6250, BPI11000, or BPI38000, and the I/O subsystem cannot find a tape unit that supports that density, the user task is placed in the waiting state, and a request is issued for a tape unit that does support that density.
- If the specified density is a value other than BPI1250, BPI6250, BPI11000, or BPI38000, the DENSITY value is not taken into consideration by the I/O subsystem when it selects a tape unit. If the density of the selected unit cannot be set to the requested value, the tape file is created at the density selected by the tape unit.

In a future release, you should use the DENSITY attribute to select the type of tape unit, because the KIND attribute values of TAPE7, TAPE9, and PETAPE will be deimplemented. Starting with that release, if a valid DENSITY value is used, but no tape unit that supports the specified density is available, the user task will be placed in the waiting state, and a request will be issued for a tape unit that supports that density.

Two conditions cause the specified DENSITY value to be overwritten. During the creation of a multifile tape, the density of the first file is used for all subsequent files. During the creation of a multivolume file, the density setting remains constant from volume to volume as long as the setting is valid for the tape unit. 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) can be used only with the library maintenance commands ADD and COPY, with the FILEDATA *TDIR* command, by the Data Management System II (DMSII) DMUTILITY and COPYAUDIT programs, and by MAPPER® applications. If these tape drives are used for any other purpose, such as with the OU (Output Unit) system command, an error message is issued.

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

Mnemonic	Synonym
BPI200	LOW
BPI556	MEDIUM
BPI800	HIGH
BPI1600	SUPER

MAPPER is a registered trademark of Unisys Corporation.

DEPENDENTINTMODE

Kinds	All except port files
Interrogate	Anytime
Modify	When closed
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Stored Permanently	No
Parameters	None
FTAM File Access	Supported
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
Stored Permanently	No
Parameters	None
FTAM File Access	Supported
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 or punch
Interrogate	Anytime
Modify	Anytime
Type	Character string
Range	Not applicable
Default	Null string
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
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 or punch 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, card punches, 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 *A Series Task Attributes Programming Reference Manual* for information about the DESTNAME task attribute.

If the value of the PRINTDISPOSITION attribute is DIRECT, the DESTINATION attribute does not apply and is not used by the system.

DIALOGCHECKINTERVAL

Kinds	Port subfile
Interrogate	Anytime
Modify	Anytime
Type	Integer
Range	0 through 1440 minutes
Default	0
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
Host Services Logical I/O	Not supported
Port Services	BNANATIVESERVICE, TCPIPNATIVESERVICE, TCPNATIVESERVICE, TCPPUSHEDMSGSERVICE

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.

If you are using TCPIPNATIVESERVICE, or TCPNATIVESERVICE or TCPPUSHEDMSGSERVICE with the 3.1 or later release of TCP/IP, 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.

The DIALOGCHECKINTERVAL attribute is not supported by the BNA Version 1 implementation.

If the MAXSUBFILES value is greater than 1, the DIALOGCHECKINTERVAL attribute requires a subfile index as a parameter.

DIALOGPRIORITY

Kinds	Port subfile
Interrogate	Anytime
Modify	Anytime
Type	Integer
Range	0 through 7
Default	0
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
Host Services Logical I/O	Not supported
Port Services	BNANATIVESERVICE

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 either the value 0 (zero) or 1. If 2 is used, the program competes with the BNA Version 2 system software and network management functions.

Any attempt to set the attribute to a larger value results in a value of 2.

Note that the DIALOGPRIORITY value should not be changed in rapid sequence. Doing so introduces inefficiencies and does not affect the message delivery order. DIALOGPRIORITY does not dictate priority on a message-by-message basis, but rather indicates the long-term priority of transmissions from this specific dialogue.

If you interrogate DIALOGPRIORITY while the subfile is open, the current DIALOGPRIORITY value is returned. The current DIALOGPRIORITY value might not be the one you specified. If you interrogate DIALOGPRIORITY when the subfile is closed, the value of DIALOGPRIORITY that you specified is returned.

The default for this attribute can be changed through the BNA Operations Interface Command PORTDEFAULTDIALOGPRIORITY.

The DIALOGPRIORITY attribute is not supported by the BNA Version 1 implementation.

If the MAXSUBFILES value is greater than 1, the DIALOGPRIORITY attribute requires a subfile index as a parameter.

DIOFILESTRUCTURE

Kinds	Disk or CD
Interrogate	Anytime
Modify	When closed
Type	Mnemonic
Range	See following text
Default	ALIGNED180
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
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, and STREAM	0, 5, 1	Indicates 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.

continued

continued

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 a FILETYPE attribute 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.

If the new file is a WORM file, the DIOFILESTRUCTURE value must be STREAM, SECTORSTREAM, or DEPENDENT. If the SECTORSTREAM or DEPENDENT value is specified, the FILESTRUCTURE value must be STREAM.

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
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
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
Stored Permanently	No
Parameters	1 required
FTAM File Access	Not supported
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
Stored Permanently	Disk
Parameters	None
FTAM File Access	Supported
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 or WORM 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 A Series 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:

A Series File Attribute 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	Nonsymbolic 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
Stored Permanently	No
Parameters	See following text
FTAM File Access	Not supported
Host Services Logical I/O	Not supported
Port Services	BNANATIVESERVICE

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.

The DONOTSEARCHNETWORK attribute is not supported by the BNA Version 1 implementation.

If the MAXSUBFILES value is greater than 1, the DONOTSEARCHNETWORK attribute requires a subfile index as a parameter.

DUMMYFILE

Kinds	All except port and remote
Interrogate	Anytime
Modify	When closed and unassigned
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
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.

DUPLICATED

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

When creating a new file or accessing an existing duplicated file, set the DUPLICATED attribute to TRUE to indicate that an area-by-area redundancy is to be maintained for the file. This area-by-area redundancy must occur on the same unit of the family. Files with a FILESTRUCTURE value of STREAM or BLOCKED cannot be duplicated.

When creating a new file, use the COPIES attribute to specify the number of copies to be maintained. When opening an existing duplicated file, the DUPLICATED attribute must have the value TRUE, so that the copies can be located in the directory.

For information about using duplicated files, refer to the *I/O Subsystem Programming Guide*.

ENABLEINPUT

Kinds	Remote
Interrogate	When open
Modify	Never
Type	Boolean
Range	TRUE, FALSE
Default	Not applicable
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
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
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
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 4, "Direct I/O Buffer Attributes."

ERRORTYPE

Kinds	Disk or CD
Interrogate	When open
Modify	Never
Type	Mnemonic
Range	0 through 3
Default	Not applicable
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
Host Services Logical I/O	Not supported
Port Services	Not applicable

Interrogate the ERRORTYPE attribute to determine the type of error that was encountered when the current buffer was last used to transfer data to or from the program.

The following mnemonic and integer values can be returned when the ERRORTYPE attribute is interrogated:

Mnemonic Value	Integer Value	Meaning
NOERROR	0	No error occurred.
READCHECKFAILURE	3	Read-check failure occurred.
READPARITYERROR	2	Read parity error occurred.
SUNOTREADY	1	Storage unit not ready.

When a file has more than one buffer, the buffers are rotated after each physical I/O.

When you interrogate ERRORTYPE for duplicated files, you must specify the copy number; the value returned indicates the type of error that occurred on the specified copy.

For related physical I/O error information, refer to the description of the IOINERROR attribute in this section.

More detailed result information can be obtained by interrogating the RESULTLIST attribute.

EXCLUSIVE

Kinds	Disk
Interrogate	Anytime
Modify	When unassigned
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Stored Permanently	No
Parameters	None
FTAM File Access	Supported
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.

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 in this section.

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
Stored Permanently	Disk or tape
Parameters	1 optional
FTAM File Access	Restricted values
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
BINARY	6	Card files only—12 bits per column, 20 words per record
CANSUPPLEBCDIC‡	16	Canadian EBCDIC 8-bit code table with general purpose supplementary graphics
CANSUPPLISO‡	17	Canadian ISO 8-bit code table with general purpose supplementary graphics
CODEPAGE850‡	18	MS-DOS 8-bit code page for multilingual use
CODEPAGE851‡	21	MS-DOS 8-bit code page for Latin/Greek use
EBCDIC or ASERIESEBCDIC	4	8-bit data encoded with the A Series EBCDIC character set

continued

† Valid for printer, punch, reader, and 7-track tape under certain conditions. Refer to the *I/O Subsystem Programming Guide* for information about translation.

‡ Valid for disk files only.

continued

Mnemonic Value	Integer Value	Meaning
FRENCHARABICE‡	22	French/Arabic EBCDIC 8-bit code table
FRENCHARABICISO‡	23	French/Arabic ISO 8-bit code table
HEX	2	4-bit, packed decimal
IA5STRING‡	10	A string of 8-bit frames containing data encoded in the ISO646 G0 and C0 character sets
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
LATINGREEKBTOS‡	25	Latin/Greek BTOS 8-bit code table
LATINGREEKEBCDIC‡	19	Latin/Greek EBCDIC 8-bit code table
LATINGREEKISO‡	20	Latin/Greek ISO 8-bit code table
LATIN1EBCDIC‡	12	Latin1 EBCDIC 8-bit code table
LATIN1ISO‡	13	Latin1 ISO 8-bit code table
LATIN2EBCDIC‡	26	Latin 2 EBCDIC 8-bit code table
LATIN2ISO‡	27	Latin2 ISO 8-bit code table
LATIN5EBCDIC‡	14	Latin5 EBCDIC 8-bit code table
LATIN5ISO‡	15	Latin5 ISO 8-bit code table
NORWAYBTOS‡	24	Norwegian BTOS 8-bit code table
OCTETSTRING‡	11	A string of 8-bit frames each containing any binary value from hexadecimal 00 to hexadecimal FF
SINGLE	0	Word mode, binary, or 48-bit data

† Valid for printer, punch, reader, and 7-track tape under certain conditions. Refer to the *I/O Subsystem Programming Guide* for information about translation.

‡ Valid for disk files only.

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 *A Series MultiLingual System (MLS) Administration, Operations, and Programming Guide.*

The following table indicates the Network Services and the distributed systems services (DSSs) that support the various EXTMODE values.

	Distributed Systems Services (DSSs)		Network Services	
Mnemonic Value	Host Services Logical I/O	FTAM File Access	BNA Version 1 Port Files	BNA Version 2 Port Files
ASCII	Yes	No	No	Yes
BINARY	Yes	No	No	No
CANSUPPLEBCDIC	Yes	No	No	No
CANSUPPLISO	Yes	No	No	No
CODEPAGE850	Yes	No	No	No
CODEPAGE851	Yes	No	No	No
EBCDIC	Yes	No	Yes	Yes
FRENCHARABICE	Yes	No	No	No
FRENCHARABICISO	Yes	No	No	No
IA5STRING	Yes	Yes	No	No
ISOGENERALSTRING	Yes	Yes	No	No
ISOGRAPHICSTRING	Yes	Yes	No	No
ISOVISIBLESTRING	Yes	Yes	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
LATIN5EBCDIC	Yes	No	No	No
LATIN5ISO	Yes	No	No	No
NORWAYBTOS	Yes	No	No	No
OCTETSTRING	Yes	Yes	No	No
SINGLE	Yes	No	Yes	Yes
STDEBCDIC	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 an A Series system, the value is always OCTETSTRING.

When a WORM file is created, the EXTMODE attribute value must be set to a value that specifies 8-bit characters.

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, while card reader and punch files can have an EXTMODE value of EBCDIC or BINARY.

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. When EXTMODE is modified, the subfile index cannot be used.

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
Stored Permanently	Disk or CD
Parameters	2 optional
FTAM File Access	Not supported
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. If the file is duplicated, the copy number is also required.

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. Refer to the *A Series Disk Subsystem Administration and 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;
```

FAMILYNAME

Kinds	Disk, CD, printer, or punch
Interrogate	Anytime
Modify	When closed
Type	Character string
Range	Any valid family name
Default	DISK
Stored Permanently	Disk, CD or backup files
Parameters	None
FTAM File Access	Not supported
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.

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.

If you interrogate FAMILYNAME for a CD-ROM or WORM file, the returned family name might have underscores (_) in the name.

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
Stored Permanently	Tape
Parameters	None
FTAM File Access	Not supported
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 *A Series 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 characters 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 value has 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.

FAMILYSIZE

FAMILYSIZE is a nonpreferred synonym for *STATIONCOUNT*. Refer to the description of the STATIONCOUNT attribute.

FILEEQUATED

Kinds	All
Interrogate	Anytime
Modify	Never
Type	Boolean
Range	TRUE, FALSE
Default	Not applicable
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Supported
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. TRUE indicates that one of these conditions exist.

If you do not specify a parameter, TRUE is returned if any of the attributes for the file were file-equated.

To determine if a specific file attribute was file-equated, you can specify that file attribute when you interrogate FILEEQUATED. The following ALGOL statement determines if the TITLE attribute has been file-equated:

```
TITLE_EQUATED := F(TITLE).FILEEQUATED;
```

The following COBOL74 code determines if the TITLE attribute has been file-equated or not:

```
IF ATTRIBUTE FILEEQUATED OF F(VALUE(TITLE)) = VALUE(TRUE) THEN
        MOVE 1 TO TITLE-EQUATED
   ELSE
        MOVE Ø 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* for more information about the FA command.

FILEKIND

Kinds	Disk, CD, or tape
Interrogate	Anytime
Modify	See following text
Type	Mnemonic
Range	See following text
Default	DATA
Stored Permanently	Disk or tape
Parameters	None
FTAM File Access	Restricted values
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. The FILEKIND attribute can be used for tape files, but the value returned is identical to the value returned by the LABELKIND attribute. When using Host Services logical I/O, the LABELKIND attribute must be used for tape files.

If FILEKIND is interrogated when the file is assigned, the value for the physical file is returned.

The FILEKIND attribute can be modified at any time if the file is a disk or tape file. If a value is specified before the file is opened, that value is used to give a newly created file a FILEKIND value. Specifying a value before the file has been opened does not affect the FILEKIND value of an existing file. To change the FILEKIND value of an existing tape or disk file, the value must be modified while the file is assigned.

If the physical FILEKIND value is inconsistent with the privileges of the program, an attribute error is issued. Compilers can change a data file to a code file, but programs that are not compilers can only change the FILEKIND value to noncode file values. The FILEKIND value of a code file cannot be changed.

User programs cannot write to files classed as code files, compilers, or system files. Disk files with a FILEKIND of FTAUDIT cannot be opened or accessed by user programs.

The rules for the modification of the FILEKIND value are dependent on the value assigned to the CLASS security option. That option is set by using the SECOPT (Security Options) system command, which is available with InfoGuard. The SECOPT system command is discussed in the *Security Administration Guide.*

CLASS Value	Possible Modification
S1 or S2	Value can be modified only 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.
U or S0	Value can be modified only if the program has read or write access to the file. If the program does not have write access to the file, a warning is issued but an attribute error is not issued.

Note: *In a future release, modification of the FILEKIND attribute will be allowed only when the program has write access to the file.*

If the CLASS security option is S1 or greater, user programs cannot write on existing files with a FILEKIND value of GUARDFILE, but files can be created when the FILEKIND value is GUARDFILE.

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
13	FTAUDIT	14	RESTRICTEDFILE
15	XDISKFILE	16	KEYSFILE
17	BACKUPPUNCH	18	REMOTEBACKUP
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

continued

continued

42	XFORTRANCODE	43	JOBCODE
44	DMALGOLCODE	45	NEWPCODE
46	SANSCODE	47	PASCALCODE
48	RPGCODE	50	FORTRAN77CODE
53	SORTCODE	54	COBOL74CODE
55	CCCODE	57	COBOL85CODE
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	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
191	BDDATA	192	DATA
193	SEQDATA	194	GUARDFILE

continued

continued

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		

For disk files, certain restrictions on FILEKIND changes have been defined in terms of some of the FILEKIND value ranges shown in Table 2-11.

Table 2–11. FILEKIND Value Ranges for Disk Files

Class	FILEKIND Value Range
System file	NULLFILE (0) through COMPILERCODEFILE (20)
Compiler	COMPILERCODEFILE (20)
Code file	COMPILERCODEFILE (20) through NDLIICODE (25), and INFOFILE (28) through CODEFILE (63)
Data file	ALGOLSYMBOL (64) through PDUMPFILE (204)
Symbolic file	ALGOLSYMBOL (64) through NDLSYMBOL (98)

For example, CATALOG, DIRECTORY, and XDISKFILE are system FILEKIND values; COMPILERCODEFILE is a compiler FILEKIND value; ALGOLCODE, JOBCODE, and COBOL74CODE are code FILEKIND values; DCALGOLSYMBOL, FORTRANSYMBOL, FIRMWARE, and TEXTDATA are data FILEKIND values.

A user program can change one data FILEKIND value to another and can change anything but a system file or an in-use code file to a data file. In addition, a compiler 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 using Host Services Logical I/O. Only the operating system can change or assign a system FILEKIND value or assign a compiler FILEKIND value.

Only programs with a message control system (MCS) status can create files with a FILEKIND value of RECOVERYFILE or SCHEDULEFILE.

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.

If the file is a WORM file, the default FILEKIND attribute value of DATA can be used. If a value is specified, that value must be ALGOLSYMBOL (64) or higher. The system file and code file values are not supported for WORM files.

Files that have certain FILEKIND values are generally assumed to have particular record formats. Table 2-12 lists the FILEKIND values and their assumed formats.

Table 2–12. Record 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	Not applicable	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	Not applicable	Not applicable
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	Not applicable	80
FORTRAN77SYMBOL	ALGOL	1-72	73-80	81-90	80
JOBSYMBOL	JOB	1-80	83-90	Not applicable	90
LCOBOLSYMBOL	COBOL	7-72	1-6	73-80	72
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

† CSEQDATA files have an arbitrary length; the text field extends through the end of the record.

continued

Table 2–12. Record Formats (cont.)

FILEKIND Value	Format Model	Text Field	Sequence Field	ID Field	Minimum Record Length in Characters
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	Not applicable	80

Files with a FILEKIND value of FORM and FONT are used by the image printer subsystem to hold form and font data.

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 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 *A Series 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
Stored Permanently	Disk or CD
Parameters	None
FTAM File Access	Supported
Host Services Logical I/O	Not supported
Port Services	Not applicable

Interrogate the FILELENGTH attribute to determine the length of the file in FRAMESIZE units. The length is counted from the first record position in the file to the end of the file, whether or not the intervening records have been written or have had space allocated for them on the physical device.

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
Stored Permanently	Disk, CD, tape, or punch
Parameters	None
FTAM File Access	Supported
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.

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.

The FILENAME value can be changed 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. The FILENAME value can be changed only by the owner of the file or a privileged task.

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 should be assigned to the remote file. If a file with the FILENAME value is not found in the DATCOMINFO 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.

WORM Files

When you create a WORM file, observe the following restrictions:

- A file that is not associated with a usercode can have a maximum of 8 nodes.
- A file that is associated with a usercode can have a maximum of 6 nodes, not including the usercode.
- The identifiers must be any combination of EBCDIC letters A through Z, digits 0 through 9, and an underscore (_). The final identifier can contain a single period (.). An identifier that contains a period must be enclosed in quotation marks (" ").

Note: *Once a WORM file has been opened for output, the file name can never be changed.*

When you interrogate FILENAME, the condition of the file affects whether the logical or physical value is returned. The following table lists the various conditions and identifies the value that is returned in that situation:

Condition	Value Returned
Assigned files that are not backup files	Physical file value
Opened and assigned backup files	Physical file value
Closed backup files	Logical file value

Valid A Series File Names

An A Series file name contains up to 12 nodes separated by slashes (/). The file name can be preceded by a usercode enclosed in parentheses or an asterisk (*). A node can contain a 1- to 17-character 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.

The following are examples of valid A Series file names:

- ACCOUNTING/RECEIPTS/021689
- PERSONNEL/ " EMPLOY.LIST "
- (KELLY)DOCUMENTS/CLIENT/6
- *SYSTEM/LOGANALYZER

Valid A Series Tape File Names

An A Series tape file name has the same characteristics as an A Series file name, except that the file name written to the tape label only contains the first node and the last node of the name. 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 A Series tape file names:

- 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.
- 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 a EBCDIC character quotation mark.

The following are examples of valid port file names:

- RESERVATIONS
- " RESERVATION_REQ "

Valid Names for Disk Files Not on A Series Systems

When you are accessing a system that is not an A Series system through a network and the file name is not a valid A Series file name, enclose the name in apostrophes ('). Before the A Series system 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
Stored Permanently	Disk
Parameters	None
FTAM File Access	Restricted values
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 *A Series 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 *A Series 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 *A Series KEYEDIOII Programming Reference Manual.*
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.*

continued

continued

Mnemonic Value	Integer Value	Meaning
NOTRESTRICTED	0	No restrictions are applied.
PLIISAM	4	The file is created by means of the PLISUPPORT index sequential-access method (ISAM) intrinsics.
RELATIVE	1	The file is created with a relative file organization. This organization is derived from COBOL74.

The NOTRESTRICTED mnemonic is the only value that can be used if a file has a FILESTRUCTURE value of STREAM or when the file is to be accessed or created using Host Services Logical I/O or File Transfer, Access, and Management (FTAM).

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
Stored Permanently	Tape
Parameters	None
FTAM File Access	Not supported
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.

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.

The FILESECTION value is used in permanent tape file assignment along with the KIND and FILENAME attribute values and, when appropriate, the CYCLE and VERSION attribute values. The FILESECTION attribute is also used in automatic input-volume switching.

FILESTATE

Kinds	All
Interrogate	Anytime
Modify	Never
Type	Mnemonic
Range	See following text
Default	Not applicable
Stored Permanently	No
Parameters	1 required (remote files) 1 optional (port files)
FTAM File Access	Supported
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 CHANGESUBFILE 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.*

Tables 2-13 through 2-15 list the FILESTATE values and their meanings.

Table 2–13. FILESTATE Values for All Files

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.
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.

Table 2–14. FILESTATE Values for Remote Files

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.
POSTPONED	10	Indicates that the file assignment for the specified remote file has been postponed.

Table 2–15. FILESTATE Values for Port Files

Mnemonic Value	Integer Value	Meaning
AWAITINGHOST	1	Indicates that the host specified by the YOURHOST subfile attribute cannot be reached at this time. Once communication with the specified host is possible, the FILESTATE value is changed to OFFERED 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.

continued

Table 2–15. FILESTATE Values for Port Files (cont.)

Mnemonic Value	Integer Value	Meaning
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.
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.

continued

Table 2–15. FILESTATE Values for Port Files (cont.)

Mnemonic Value	Integer Value	Meaning
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.
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.

continued

Table 2–15. FILESTATE Values for Port Files (cont.)

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.

continued

Table 2–15. FILESTATE Values for Port Files (cont.)

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.

continued

Table 2–15. **FILESTATE Values for Port Files** (cont.)

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	Disk or CD
Interrogate	Anytime
Modify	When closed
Type	Mnemonic
Range	See following text
Default	ALIGNED180 (disk)
Stored Permanently	Disk
Parameters	None
FTAM File Access	Restricted values
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. Since this dependence is not always detectable by the operating system, 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.

A file that has a KIND attribute value of CD must have a FILESTRUCTURE value of STREAM.

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
- A FILEORGANIZATION value other than the NOTRESTRICTED value
- Checkpoint or restart capabilities
- Duplicated files
- Interchange packs

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. Table 2-16 summarizes when a file can be assigned.

Table 2–16. File Assignment Matrix for the FILESTRUCTURE Attribute

Logical File	Permanent File		
	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 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 A Series host is the initiator, File Transfer, Access, and Management (FTAM) file access requires a FILESTRUCTURE value of STREAM. When an A Series host is the responder, FTAM file access supports all FILESTRUCTURE values.

Host Services logical I/O does not support a FILESTRUCTURE value of BLOCKED.

In a future release, the default value of the FILESTRUCTURE attribute will be STREAM instead of ALIGNED180. Because a program that explicitly specifies a FILESTRUCTURE of ALIGNED180 or BLOCKED cannot open a permanent file with a FILESTRUCTURE value of STREAM, a warning is issued when a program that is not using direct I/O explicitly specifies a FILESTRUCTURE value of ALIGNED180 or BLOCKED opens a permanent file with the following characteristics:

- The file was created since the introduction of the FILESTRUCTURE attribute.
- The file was not created using direct I/O.
- The file has a default FILESTRUCTURE value of ALIGNED180.

If the program is to continue to be able to open the file, the program must be modified or the file recreated with an explicit FILESTRUCTURE value of ALIGNED180 or BLOCKED.

Additionally, because the default is changing, a program that does not explicitly specify a FILESTRUCTURE value and does not specify the DEPENDENTSPECS attribute value as TRUE, or the FILETYPE attribute value as 7 or 8, receives a warning whenever it interrogates file attributes that are invalid for a file with a FILESTRUCTURE value of STREAM. This warning is not issued if the program has interrogated the FILESTRUCTURE attribute since the file was assigned.

A program that does specify 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.

To avoid these warnings, do not declare a FILESTRUCTURE value in your programs that access existing files until a FILESTRUCTURE value has been declared in the programs that create the existing files.

FILETYPE

Kinds	All
Interrogate	When unassigned (Host Services logical I/O)
	Anytime (local host)
Modify	When closed
Type	Integer
Range	0 through 4; 6 through 8
Default	0
Stored Permanently	Disk or tape
Parameters	None
FTAM File Access	Restricted values
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.

When a file is written to the WORM media, FILETYPE must have a value of 0.

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.

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
Stored Permanently	No
Parameters	None
FTAM File Access	Supported
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
Interrogate	Anytime
Modify	Anytime
Type	Boolean
Range	TRUE, FALSE
Default	TRUE
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
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 value of FLEXIBLE is TRUE and more areas are needed, space is added to the disk file header to accommodate additional areas and the AREAS attribute value is increased by 10. 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 or is a duplicated file.

FORMID

Kinds	Printer or punch
Interrogate	Anytime
Modify	When closed
Type	Character string
Range	Not applicable
Default	Null string
Stored Permanently	Backup file
Parameters	None
FTAM File Access	Not supported
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 or cards that must be used to print or punch 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 a direct I/O file or any other file with a PRINTDISPOSITION attribute value of DIRECT and a device with a matching form identification is not available, the string assigned to the attribute is displayed as part of the message requesting operator action. 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.

The default value is a null string.

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, 48
Default	Value corresponding to UNITS
Stored Permanently	Disk or tape
Parameters	None
FTAM File Access	Restricted values
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 possible 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 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, or SINGLE. Unisys recommends that the value be 8 for TCPNATIVESERVICE and TCPPUSHEDMSGSERVICE. The value must be 8 for TCPIPNATIVESERVICE.

continued

continued

Value	Meaning
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.

When a WORM file is created, FRAMESIZE must have a value of 8.

FRAMESIZECENSUS

Kinds	Port or port subfile
Interrogate	Anytime
Modify	Never
Type	Integer
Range	0 through 65535
Default	Not applicable
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
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.

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
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Host Services Logical I/O	Supported
Port Services	Not applicable

If cataloging is enabled, use the GENERATION attribute 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.

HOSTNAME

Kinds	All
Interrogate	Anytime
Modify	When closed
Type	Character string
Range	Not applicable
Default	Name of local host
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Supported
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 or lowercase alphanumeric characters, hyphens (-), or underscores (_). 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.

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*.

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
Stored Permanently	No
Parameters	1 optional (port or port subfile)
FTAM File Access	Not supported
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
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
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.*

INTERCHANGE

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

Set the INTERCHANGE attribute to TRUE if a file is to be stored on an interchange disk pack. An interchange disk pack is a disk pack that has been initialized to a multisector format that ensures disk pack compatibility among Unisys systems other than A Series systems.

Files with a FILESTRUCTURE value of STREAM or BLOCKED cannot be written to an interchange disk pack.

Note: *The INTERCHANGE attribute will be deimplemented in a future release.*

INTMODE

Kinds	All
Interrogate	Anytime
Modify	When closed
Type	Mnemonic
Range	See following text
Default	See following text
Stored Permanently	No
Parameters	None
FTAM File Access	Restricted values
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
CANSUPPLEBCDIC†	16	Canadian EBCDIC 8-bit code table with general-purpose supplementary graphics
CANSUPPLISO†	17	Canadian ISO 8-bit code table with general-purpose supplementary graphics
CODEPAGE850†	18	MS-DOS 8-bit code page for multilingual use
CODEPAGE851†	21	MS-DOS 8-bit code page for Latin/Greek use
EBCDIC or ASERIESEBCDIC	4	8-bit data encoded with the A Series EBCDIC character set
FRENCHARABICE†	22	French/Arabic EBCDIC 8-bit code table
FRENCHARABICISO†	23	French/Arabic ISO 8-bit code table
HEX	2	4-bit, packed decimal
IA5STRING†	10	A string of 8-bit frames containing data encoded in the ISO646 G0 and C0 character sets

† Valid for disk files only.

continued

continued

Mnemonic Value	Integer Value	Meaning
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
LATINGREEKBTOS†	25	Latin/Greek BTOS 8-character set
LATINGREEKEBCDIC†	19	Latin/Greek EBCDIC 8-bit code table
LATINGREEKISO†	20	Latin/Greek ISO 8-bit code table
LATIN1EBCDIC†	12	Latin1 EBCDIC 8-bit code table
LATIN1ISO†	13	Latin1 ISO 8-bit code table
LATIN2EBCDIC†	26	Latin2 EBCDIC 8-bit code table
LATIN2ISO†	27	Latin2 ISO 8-bit code table
LATIN5EBCDIC†	14	Latin5 EBCDIC 8-bit code table
LATIN5ISO†	15	Latin5 ISO 8-bit code table
NORWAYBTOS†	24	Norwegian BTOS 8-bit code table
OCTETSTRING†	11	A string of 8-bit frames each containing any binary value from hexadecimal 00 to hexadecimal FF
SINGLE	0	Word mode, binary, or 48-bit data

† Valid for disk files only.

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 Network Systems and the distributed systems services (DSSs) that support the various INTMODE values.

	Distributed Systems Services (DSSs)		**Network Services**	
Mnemonic Value	**Host Services Logical I/O**	**FTAM File Access**	**BNA Version 1 Port Files**	**BNA Version 2 Port Files**
ASCII	Yes	Yes	No	Yes
CANSUPPLEBCDIC	Yes	No	No	No
CANSUPPLISO	Yes	No	No	No
CODEPAGE850	Yes	No	No	No
CODEPAGE851	Yes	No	No	No
EBCDIC	Yes	Yes	Yes	Yes
FRENCHARABICE	Yes	No	No	No
FRENCHARABICISO	Yes	No	No	No
IA5STRING	Yes	Yes	No	No
ISOGENERALSTRING	Yes	Yes	No	No
ISOGRAPHICSTRING	Yes	Yes	No	No
ISOVISIBLESTRING	Yes	Yes	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
LATIN5EBCDIC	Yes	No	No	No
LATIN5ISO	Yes	No	No	No
NORWAYBTOS	Yes	No	No	No
OCTETSTRING	Yes	Yes	No	No
SINGLE	Yes	No	Yes	Yes
STDEBCDIC	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 and COBOL85	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	
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 and BNA Version 2, the INTMODE attribute is used during matching. If the INTMODE attributes of the two subfiles do not match, at

Port Files

For BNANATIVESERVICE and BNA Version 2, 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.

If you specify a value of SINGLE, the INTMODE value is changed to EBCDIC.

INTNAME

Kinds	All
Interrogate	Anytime
Modify	When unassigned
Type	Character string
Range	Not applicable
Default	As declared in program
Stored Permanently	No
Parameters	None
FTAM File Access	Supported
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
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
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.

IOINERROR

Kinds	Disk, CD, or tape
Interrogate	When open
Modify	Never
Type	Boolean
Range	TRUE, FALSE
Default	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
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 ERRORTYPE, RECORDINERROR, and STATE attributes can be consulted for more information.

For duplicated disk files, IOINERROR is TRUE if any copy of the file encountered an error associated with the buffer.

KIND

Kinds	All
Interrogate	Anytime
Modify	When closed
Type	Mnemonic
Range	See following text
Default	DONTCARE
Stored Permanently	No
Parameters	None
FTAM File Access	Restricted values
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 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 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
ODT or SPO	2	No	No
PACK or DISKPACK	17	Yes	Yes
PORT	19	No	No
PRINTER	7	Yes	No
PUNCH or CP	11	Yes	No
READER	9	Yes	No
REMOTE or DC	3	Yes	No
TAPE	45	Yes	No
TAPE7	13	Yes	No
TAPE9	14	Yes	No
TAPEPE or PETAPE (for PE or GCR tapes)	15	Yes	No
VC	40	No	No

If the KIND value is DONTCARE and a new file is to be created when the file is opened, the value is converted to TAPE7. 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. If more than one file is found, a "DUP FILE" message is displayed. Operator action is required to resolve the duplicate file condition.

The value CD is used for both CD-ROM files and WORM 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, although DISK is the preferred value.

The use of voice channel (VC) files is limited to the Network Application Platform (NAP) software. NAP declares various direct files with a KIND attribute value of VC to interface with VC units. VC files can be used only with direct I/O. The use of VC files outside of the NAP software is not supported at this time.

Note: *The values TAPE7, TAPE9, TAPEPE, and PETAPE will be deimplemented in a future release. For this reason, the DENSITY attribute should be used to select the type of tape on which a file is to be created.*

LABEL

Kinds	All except port
Interrogate	Anytime
Modify	Never (remote)
	When unassigned (others)
Type	Mnemonic
Range	See following text
Default	OMITTED (remote)
	STANDARD (others)
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
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

While you can interrogate or specify the LABEL attribute for all devices except port files without receiving an error, you should be aware that tape, printer, and punch files have specific characteristics.

Tape Files

If the value of LABEL is STANDARD when a tape file is created, the I/O subsystem writes ANSI (USASI) labels as the beginning and ending records of the file. The beginning records of the file are succeeded by tape marks and the ending records are preceded by tape marks. 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 DIRECT 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.

The following table summarizes when a top-of-page (skip-to-channel 1) action is performed based on the value of the LABEL attribute:

LABEL Attribute Value	Action Taken
STANDARD	A top-of-page action is performed after the file is printed. This action is controlled by the I/O control words (IOCWs) contained in the backup file.
OMITTED or OMITTEDEOF	A top-of-page action is not performed before or after each backup file 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.*

Punch Files

Labeled punch files use the format described in the discussion of card files in the *I/O Subsystem Programming Guide.* Note, however, that only direct I/O files can create an unlabeled punch file.

LABELKIND

Kinds	Tape
Interrogate	When assigned
Modify	Never
Type	Integer
Range	0 through 14
Default	See following text
Stored Permanently	Tape
Parameters	None
FTAM File Access	Not supported
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	B6500USASI
OMITTED	UNLABELLED
OMITTEDEOF	UNLABELLED

The values and meanings associated with LABELKIND are as follows:

Mnemonic Value	Integer Value	Meaning
B6500USASI	0	B 6500 A Series USASI
UNLABELLED	1	Unlabeled
B5500STANDARD	2	B 5500 standard
SYSTEMSTANDARD	3	System standard
B3500USASI	4	B 3500 USASI
B3500STANDARD	5	B 3500 standard
B6500BACKUPTAPE	6	B 6500 A Series backup tape
B6500LIBRARYTAPE	7	B 6500 A Series library tape
B6500LOADCONTROLTAPE	8	B 6500 A Series load control tape
ASERIESLIBRARYTAPE	9	A Series library tape
B5500BACKUPTAPE	10	B 5500 backup tape
B5500LIBRARYTAPE	11	B 5500 library tape

continued

continued

Mnemonic Value	Integer Value	Meaning
NONBURROUGHSUSASI	12	USASI for systems other than Unisys A Series systems
USERDEFINEDLABEL	13	User-defined label
FIXEDBLOCKLIBRARYTAPE	14	Fixed-block library tape

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.

LABELTYPE

LABELTYPE is a nonpreferred synonym for *LABEL*. Refer to the description of the LABEL attribute.

LASTRECORD

Kinds	Disk or CD
Interrogate	When assigned
Modify	When assigned (see following text)
Type	Integer
Range	–1 through 549755813887
Default	–1
Stored Permanently	Disk or CD
Parameters	None
FTAM File Access	Not supported
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.

If you interrogate the LASTRECORD attribute for a file that has a FILESTRUCTURE value of ALIGNED180 or BLOCKED and the BLOCKSTRUCTURE attribute value is not FIXED or the FILETYPE attribute value is not 0 (zero), the number of the block rather than the number of the last record is returned.

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 PROTECTION attribute value is not PROTECTED.
- The DUPLICATED and CRUNCHED attributes are 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.

If the InfoGuard security-enhancement software is being used, modification of the LASTRECORD value is dependent on the CLASS option value. The CLASS option value is assigned by using the SECOPT (System Options) system command.

CLASS Value	Possible Modification
S1 or S2	The value can be modified only if the program has write access to the file. If the program does not have write access to the file, a security error is issued and the program is terminated.
U or S0	The value can be modified if the program has read or write access to the file. If the program does not have write access to the file, an attribute error is not issued, but a warning is issued.

Note: *In a future release, the LASTRECORD attribute will be allowed to be modified only when the program has write access to the file.*

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
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
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).

Note: *When multiple stacks are operating on the same port file, there is no way to determine which stack initiated the I/O operation that the LASTSUBFILE value is reporting.*

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 in this section 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 Ø],8Ø,A);   %To broadcast

WRITE (<file id>[STATION 3],...);   %To station 3 only
```

LICENSEKEY

Kinds	Disk
Interrogate	When open
Modify	When opened (with restrictions)
Type	Character string
Range	Not applicable
Default	Null string
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
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 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 nonconditioned 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 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
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
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 Table F-1.

The LINENUM value cannot be altered by a file equation.

LOCKEDFILE

Kinds	Disk, tape, printer, and punch
Interrogate	Anytime
Modify	Disk anytime
	Tape when unassigned
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Stored Permanently	Yes
Parameters	None
FTAM File Access	Not Supported
Host Services Logical I/O	Supported
Port Services	All

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.

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>
```

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

The decision by the system to remove printer backup files once the files are printed is based on the value of the SAVEBACKUPFILE file attribute, the LOCKEDFILE file attribute, and the REMOVELOCKEDFILES option of the Print System. If the SAVEBACKUPFILE value is TRUE, the backup file is not removed after it is printed.

If the SAVEBACKUPFILE value is FALSE, the default value, and the LOCKEDFILE value is FALSE, the default value, the backup file is removed after printing.

If the SAVEBACKUPFILE value is FALSE, and the LOCKEDFILE value is TRUE, the removal of the backup file is determined by the value of the REMOVELOCKEDFILES option that is assigned with the PS DEFAULT system command. The initial value is NEVER, which means that the backup file is not removed after printing if the LOCKEDFILE value of the file is TRUE. The other value is AUTOMATIC, which means one of the following two actions occurs if the LOCKEDFILE value of the file is TRUE:

- If the print request is automatically generated, the file is protected until it is printed and is removed after the printing is completed.
- If the print request is generated by the WFL *PRINT* statement, the file is protected until it is printed, but the file is not removed after the printing is completed.

MAXCENSUS

Kinds	Port file or subfile
Interrogate	Anytime
Modify	Anytime
Type	Integer
Range	–1 through 63
Default	63
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
Host Services Logical I/O	Not supported
Port Services	BNANATIVESERVICE, NETBIOSSESSIONSERVICE, OSINATIVESERVICE, OSISESSIONSERVICE, TCPPUSHEDMSGSERVICE, BASICSERVICE

Use the MAXCENSUS attribute to specify the number of input messages that can be queued for a subfile or a file before the service provider requests that the correspondent dialogue endpoint stop sending messages. If the correspondent endpoint is an A Series subfile, it is suspended when a WRITE operation is attempted.

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 the current number of input messages queued. 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 I/O subsystem 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.

The MAXCENSUS attribute requires a parameter if the MAXSUBFILES value is greater than 1.

MAXFRAMESIZECENSUS

Kinds	Port file or subfile
Interrogate	Anytime
Modify	Anytime for TCPNATIVESERVICE When closed for TCPIPNATIVESERVICE
Type	Integer
Range	512 through 65536 for TCPIPNATIVESERVICE 0 through 16777215 for TCPNATIVESERVICE when FRAMESIZE is equal to 8 0 through 2796203 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
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
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 the MAXCENSUS and CENSUS attribute descriptions in this section for information about the functionality of these attributes.

The MAXFRAMESIZECENSUS attribute requires a subfile index as a parameter if the MAXSUBFILES value is greater than 1.

MAXRECSIZE

Kinds	All
Interrogate	Anytime
Modify	When closed
Type	Integer
Range	See following text
Default	See following text
Stored Permanently	Disk or tape
Parameters	1 optional
FTAM File Access	Restricted values
Host Services Logical I/O	Restricted values
Port Services	All

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. If the file is a WORM file, MAXRECSIZE must have a value of 1 or an even value within the range 2 through 65534.

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 in this section for more details.

The legal MAXRECSIZE range for 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 Table 2-17.

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 Table 2-17 for detailed information.

Table 2–17. Valid MAXRECSIZE Range for FTAM Files

DOCUMENTTYPE	BLOCKSTRUCTURE	EXTMODE	Maximum Record Length
FTAM1	FIXED EXTERNAL	IA5STRING	64000
		ISOVISIBLESTRING	
		ISOGRAPHICSTRING	9995
		ISOGENERALSTRING	
	VARIABLE	IA5STRING	9995
		ISOVISIBLESTRING	
		ISOGRAPHICSTRING	
		ISOGENERALSTRING	
FTAM2	VARIABLE	IA5STRING	9995
		ISOVISIBLESTRING	
		ISOGRAPHICSTRING	
		ISOGENERALSTRING	
FTAM3	FIXED, EXTERNAL	OCTETSTRING	64000
	VARIABLE	OCTETSTRING	9995
INTAP1	FIXED	OCTETSTRING	64000
	VARIABLE	OCTETSTRING	9995

For printer or punch 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
CARDREADER or CARDPUNCH	If EXTMODE is BINARY, 20 words; if EXTMODE is EBCDIC, 14 words; if EXTMODE is a value other than BINARY or EBCDIC, 10 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. Reader or punch devices have a maximum value of 80 characters.

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 or 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 1023 (BNA Version 1)
	1 through 65535 (all others)
Default	1
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
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 and if it is not a BNA Version 1 port file. Decreasing the value of the MAXSUBFILES attribute decreases the amount of memory used by the task.

MINRECSIZE

Kinds	All except port
Interrogate	Anytime
Modify	When closed
Type	Integer
Range	0 through MAXRECSIZE
Default	Depends on BLOCKSTRUCTURE value
Stored Permanently	Disk or tape
Parameters	None
FTAM File Access	Supported
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 discussions of disk files and magnetic tape files in the *I/O Subsystem Programming Guide.*

MYDOMAINNAME

Kinds	Port
Interrogate	Anytime
Modify	Never
Type	Character string
Range	MYHOST value or value of TCPIPHN
Default	MYHOST value or value of TCPIPHN
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Host Services Logical I/O	Not supported
Port Services	TCPIPNATIVESERVICE, TCPNATIVESERVICE, TCPPUSHEDMSGSERVICE

If you are using the 3.1 or later release of TCP/IP, 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
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Host Services Logical I/O	Not supported
Port Services	BNANATIVESERVICE, OSINATIVESERVICE, OSISESSIONSERVICE, TCPIPNATIVESERVICE, TCPNATIVESERVICE, TCPPUSHEDMSGSERVICE, 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
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
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.

The MYHOSTGROUP attribute is not supported by the BNA Version 1 implementation. If MYHOSTGROUP is interrogated, the MYHOST attribute value is returned.

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
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
Host Services Logical I/O	Not supported
Port Services	TCPIPNATIVESERVICE, TCPNATIVESERVICE, TCPPUSHEDMSGSERVICE

If you are using the 3.1 or later release of TCP/IP, 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.

The MYIPADDRESS attribute requires a subfile index as a parameter if MAXSUBFILES is greater than 1.

MYNAME

Kinds	Port
Interrogate	Anytime
Modify	When closed
Type	Character string
Range	Not applicable
Default	Null string
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
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 in this section 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, TCPNATIVESERVICE, and TCPPUSHEDMSGSERVICE 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 (Host Services logical I/O)
	Anytime (local host)
Modify	When unassigned (Host Services logical I/O)
	When closed (local host)
Type	Mnemonic
Range	See following text
Default	See following text
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
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, Unisys recommends 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 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 in this section 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 in this section 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.*

Host Services logical I/O does not support modification or interrogation of the MYUSE attribute when the file is assigned.

NETACCESSPOINT

Kinds	Port subfile
Interrogate	Anytime
Modify	When closed
Type	Character string
Range	Not applicable
Default	Null string
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
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 *NW HLCN ADD NETACCESSPOINT <identifier> (NP = <np#>)* system command. For a discussion of this command, refer to the *A Series 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 A Series 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 (_).

If the MAXSUBFILES value is greater than 1, the NETACCESSPOINT attribute requires a subfile index as a parameter.

NEWFILE

Kinds	All except port
Interrogate	Anytime
Modify	When closed
Type	Boolean
Range	Unspecified, TRUE, FALSE
Default	Unspecified
Stored Permanently	No
Parameters	None
FTAM File Access	Supported
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 Value	NEWFILE Value	MYUSE Value	Existing File Found	Action Taken
Output-only devices such as a PRINTER or PUNCH value	Not considered	Not considered	Not applicable	New file created
Input-only devices such as a READER value	Not considered	Not considered	Yes	Existing file opened
			No	NO FILE message issued
Input/output devices such as a TAPE or DISK value	TRUE	Not considered	Not applicable	New file created
	FALSE	Not considered	Yes	Existing file opened
			No	NO FILE message issued
Input/output devices such as a TAPE or DISK value	Unspecified	IN or IO	Yes	Existing file opened

continued

continued

KIND Value	NEWFILE Value	MYUSE Value	Existing File Found	Action Taken
Input/output devices such as a TAPE or DISK value	Unspecified	IN or IO	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 did 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
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
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 in this section.

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.

NORECOVERY

Kinds	Disk
Interrogate	Anytime
Modify	When unassigned
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
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
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
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.

- The file is on an interchange pack, and the pack becomes unready after the first area on that pack has been activated but before all areas on that pack have been activated.
- The file is on an interchange pack, and sectors are required for the first area of a file header on a given continuation pack.

NOTE

Kinds	Printer, punch, or disk
Interrogate	Anytime
Modify	See following text
Type	Character string
Range	Not applicable
Default	Null string
Stored Permanently	Disk and backup file
Parameters	None
FTAM File Access	Not supported
Host Services Logical I/O	Not supported
Port Services	Not applicable

For a printer file or a punch file, use the NOTE attribute to store a message of up to 250 characters that is to be printed or punched on the banner page or card 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), the file title is printed in block characters on the banner page.

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 PRINTDISPOSITION attribute has a value of DIRECT, the NOTE attribute is ignored.

For a printer file or punch 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.

OFNOTIFICATION

Kinds	All except port, printer, and punch files
Interrogate	Anytime
Modify	When closed or unassigned
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Stored Permanently	No
Parameters	None
FTAM File Access	Supported
Host Services Logical I/O	Supported
Port Services	Not applicable

If the OPTIONAL attribute is TRUE, you can set the OFNOTIFICATION attribute to TRUE to enable your program to determine one of the following conditions:

- If a physical file was assigned to the logical file.
- If 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.

When the OFNOTIFICATION value is TRUE, one of the following OPEN I/O result enumerated values is returned when the file is opened:

- OFRSLT (3) if the OF command was entered.
- OKRSLT (1) if a physical file was assigned.

If the OFNOTIFICATION value is FALSE, the OPEN I/O result enumerated value is always OKRSLT (1).

The OFNOTIFICATION attribute can be of significant use to system software such as compilers. The COBOL85 compiler sets OFNOTIFICATION to TRUE.

OLDYOURUSERCODE

Kinds	Port
Interrogate	Anytime
Modify	When closed
Type	Boolean
Range	TRUE, FALSE
Default	TRUE
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Host Services Logical I/O	Not supported
Port Services	BNANATIVESERVICE

Use the OLDYOURUSERCODE attribute to determine or specify how the YOURUSERCODE attribute is handled. Refer to the YOURUSERCODE attribute description for more information.

Note: *In a future release, the value TRUE will no longer be valid. If TRUE is specified, an attribute error will be issued and the value of OLDYOURUSERCODE will not be changed.*

OPEN

Kinds	All
Interrogate	Anytime
Modify	Anytime
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Stored Permanently	No
Parameters	None
FTAM File Access	Supported
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.

The OPEN attribute cannot be specified in a file declaration or used in a file equation.

OPTIONAL

Kinds	All except port, printer, and punch files
Interrogate	Anytime
Modify	When closed
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Stored Permanently	No
Parameters	None
FTAM File Access	Supported
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 an optional file. When a file is identified as an optional file and is not 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.

Regardless of the value of the OPTIONAL attribute, if the appropriate file is not present, 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
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
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
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
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 is this section.

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
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
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	Printer
Interrogate	Anytime
Modify	When closed
Type	Character string
Range	Not applicable
Default	Null string
Stored Permanently	Backup file
Parameters	None
FTAM File Access	Not supported
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=6Ø CLP=8Ø PORTRAIT PAPERSIZE=LETTER
```

PAGESIZE

Kinds	Printer
Interrogate	Anytime
Modify	Anytime
Type	Integer
Range	0 through 255
Default	0
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
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	ODD, EVEN
Default	ODD
Stored Permanently	Tape
Parameters	None
FTAM File Access	Not supported
Host Services Logical I/O	Supported
Port Services	Not applicable

Use the PARITY attribute to determine or specify the parity used on the file.

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

STANDARD and *NONSTANDARD* are nonpreferred synonyms for *ODD* and *EVEN*, respectively.

PARITY can be modified to EVEN (NONSTANDARD) only when the value of the KIND attribute is TAPE7 or the value of the DENSITY attribute is BPI556. Otherwise, a file attribute error is generated as the tape file is opened.

PASSIVEOPEN

Kinds	Port subfile
Interrogate	Anytime
Modify	When closed
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
Host Services Logical I/O	Not supported
Port Services	NETBIOSSESSIONSERVICE, TCPIPNATIVESERVICE, TCPNATIVESERVICE, TCPPUSHEDMSGSERVICE

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 both of the following conditions are met:

- The TCP/IP 3.1 or later release is used.
- 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 if you are using the TCP/IP 3.1 or later release. 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.

PENDINGPROTECTEDFILE

Kinds	CD
Interrogate	Anytime
Modify	Never
Type	Character string
Default	Null string
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Host Services Logical I/O:	Not supported
Port Services	Not applicable

If you are using WORM files, interrogate the PENDINGPROTECTEDFILE attribute to determine the file name of the pending protected file for the WORM media specified by the FAMILYNAME attribute. If there is no pending protected file for the specified WORM media, a null string is returned.

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
Stored Permanently	Disk
Parameters	None
FTAM File Access	Supported
Host Services Logical I/O	Not 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 A Series 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.	FTAM-2
	Read a whole file.	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

continued

continued

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
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Host Services Logical I/O	Not supported
Port Services	Not applicable

Interrogate the POPULATION attribute to determine the number of logical files assigned to a disk file or the number of stations assigned to a remote file.

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.*

PRESENT

Kinds	All except port
Interrogate	Anytime
Modify	Never
Type	Boolean
Range	TRUE, FALSE
Default	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
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 or punch
Interrogate	Anytime
Modify	Anytime
Type	Character string
Range	Valid charge identifier
Default	See following text
Stored Permanently	Backup file
Parameters	None
FTAM File Access	Not supported
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 or punch backup file. The Print System uses this 1- to 17-character name. 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 chargecode validation are applied by the Print System through the USERDATA interface. If chargecode validation is necessary, the value of the PRINTCHARGE attribute is validated when the file is opened.

PRINTCOPIES

Kinds	Printer or punch
Interrogate	Anytime
Modify	Anytime
Type	Integer
Range	1 through 1000
Default	1
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
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 PRINTDISPOSITION attribute value is DIRECT, the PRINTCOPIES attribute does not apply and is not used by the system.

The default value is one copy for each destination.

PRINTDISPOSITION

Kinds	Printer or punch
Interrogate	Anytime
Modify	When closed
Type	Mnemonic
Range	See following text
Default	DIRECT (direct files) EOJ (other files)
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Host Services Logical I/O	Supported
Port Services	Not applicable

Use the PRINTDISPOSITION attribute to determine or specify whether a printer or punch backup file should be created, and if it is created, when the system is to issue an automatic print request for the file. The mnemonic and integer values are defined as follows:

Mnemonic Value	Integer Value	Meaning
CLOSE	2	A print request is automatically issued by the system when the file is closed.
DIRECT	1	The file is to be printed directly: no backup file is created. The value DIRECT is effective only when the file is opened. Each file is printed on a separate page. Standard banner pages are printed if the LABEL attribute value is STANDARD. For direct I/O files, DIRECT is the only valid value.
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 and have private security.
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.

If you use the PRINTDISPOSITION file attribute in a WFL *PRINT* statement, the values CLOSE, DIRECT, and DONTPRINT are ignored, and a print request is generated when the PRINT statement terminates.

PRINTERCONTROL

Kinds	Printer
Interrogate	Anytime
Modify	Anytime
Type	Character string
Range	Valid file title
Default	Null string
Stored Permanently	Backup file
Parameters	None
FTAM File Access	Not supported
Host Services Logical I/O	Supported
Port Services	Not applicable

The PRINTERCONTROL attribute has no functionality after the Mark 3.8 release.

Note: *The PRINTERCONTROL attribute will be deimplemented in a future release.*

PRINTERKIND

Kinds	Printer
Interrogate	Anytime
Modify	Anytime
Type	Mnemonic
Range	See following text
Default	DONTCARE
Stored Permanently	Backup file
Parameters	None
FTAM File Access	Not supported
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.

The PRINTERKIND attribute returns the value corresponding to the type of printer that is assigned. When no printer is assigned, 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
B9275	10	The Unisys B9275 non-impact printer PDL
DONTCARE	0	Any type of printer
D630	8	The Diablo® 630 PDL

continued

Diablo is a registered trademark of Xerox Corporation.

continued

Mnemonic Value	Integer Value	Meaning
D630E	9	The Extended Diablo 630 PDL (recognized by the Unisys AP9208, AP9215, and AP9230 printers)
EPSONFX	13	The Epson® FX PDL (recognized by the Unisys AP1329, AP9215-1, and Model 37 printers)
EPSONLQ	14	The Epson LQ PDL (recognized by the Unisys AP1324 printer)
EXPRESS	15	The Express PDL (recognized by the Unisys AP9206, AP9230, and Model 37 printers)
HPGL	17	The Hewlett-Packard Graphics Language PDL (recognized by the Unisys AP9415 printer and the HP® 7475A plotter)
IMAGEPRINTER or IP	1	A class of printer that generates the image of an entire page at a time
KANJIPRINTER or KP	16	A class of printer that is capable of printing Japanese Kanji characters
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 (recognized by the HP DeskJet and the Unisys AP9210, AP9215-1, and AP9415 printers)
PCL4	5	Level IV of the Hewlett-Packard Printer Command Language PDL (recognized by the HP LaserJet® and the Unisys AP9210, AP9215-1, and AP9415 printers)
PCL5	6	Level V of the Hewlett-Packard Printer Command Language PDL (recognized by the HP LaserJet III printer)
PLOTTER	12	A class of device that generates output drawn with a pen
POSTSCRIPT or PS	7	The PostScript® PDL (recognized by the Unisys AP9210 and AP9415 printers)
PROPRINTER	11	The IBM Proprinter® PDL (recognized by the Unisys AP9215-1 printer)
PROPRINTERXL	19	The IBM Proprinter XL PDL (recognized by the Unisys AP9210 printer)
TTY	18	The Teletype (TTY) PDL (consisting of carriage return, line feed, and form feed)

EPSON is a registered trademark of the Seiko Epson Corporation.
HP is a registered trademark of Hewlett-Packard Company.
LASERJET is a registered trademark of Hewlett-Packard Company.
PostScript is a registered trademark of Adobe Systems, Inc.
PROPRINTER is a registered trademark of International Business Machines Corporation.

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
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
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, 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
```

For detailed information about using this attribute, refer to the *Print System Guide*.

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
Stored Permanently	Disk, tape, or CD
Parameters	None
FTAM File Access	Not supported
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.

Disk Files

If the PROTECTION value 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 PROTECTED value 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 PROTECTION value 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 PROTECTION value 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 PROTECTION value 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 PROTECTION value 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 Host Services logical I/O for tape files.

WORM Files

If the PROTECTION value is PROTECTED, a directory entry for the file is created when the file is opened. After the first physical I/O to the file has been completed, the file can never be removed. If a system interruption occurs before the file is closed, the file is recovered at the end of the last written physical sector and has a status of pending. Before another file can be opened for output, the pending protected file must be opened. Refer to the *I/O Subsystem Programming Guide* for information about creating a pending protected file and handling such a file after a system interruption.

If the file has a PROTECTION value of SAVE, it is made permanent when it is closed. However, if a system interruption occurs before the file is closed, the file is lost.

Note *If a file with a PROTECTION value of either SAVE or TEMPORARY is not made a permanent file, the WORM media space that contained that file is lost.*

PROVIDERGROUP

Kinds	Port subfile
Interrogate	Anytime
Modify	When closed
Type	Character string
Range	Not applicable
Default	Null string
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
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 1, BNA Version 2, OSI
BNANATIVESERVICE	BNA Version 1, BNA Version 2
NETBIOSSESSIONSERVICE	HLCN
OSINATIVESERVICE	OSI
OSISESSIONSERVICE	OSI
TCPNATIVESERVICE	TCP
TCPPUSHEDMSGSERVICE	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:

- *BNAV1 for BNA Version 1
- *BNAV2 for BNA Version 2
- *BNAOSI for OSI
- *HLCN for NETBIOS

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 Unisys predeclared provider groups.

This attribute requires a subfile index as a parameter if the MAXSUBFILES value is greater than 1.

Note: *Subports with the values of *BNAV2 and *BNAOSI 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.*

RECEPTIONS

Kinds	Remote
Interrogate	When open
Modify	Never
Type	Integer
Range	Not applicable
Default	Not applicable
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
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
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
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
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
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
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
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 in this section for information about KEYEDIOII files.

REEL

REEL is a nonpreferred synonym for *FILESECTION*. Refer to the description of the FILESECTION attribute.

RELEASEID

Kinds	Disk
Interrogate	Anytime
Modify	See following text
Type	Character string
Range	Not applicable
Default	Null string
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
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. Unisys uses the RELEASEID attribute 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 space, and 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 RELEASEID, the returned value is terminated by a period or space.

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.

The previous description also generally applies to duplicated files, except that all copies of duplicated files are set to the same value of RELEASEID. For assigned duplicated files, an attribute error occurs if a RELEASEID value is specified and the first copy of the file contains a RELEASEID value. If the first copy does not contain a RELEASEID value, the value specified is set in all copies of the physical file.

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
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Host Services Logical I/O	Not supported
Port Services	BNANATIVESERVICE, NETBIOSSESSIONSERVICE, OSINATIVESERVICE, OSISESSIONSERVICE, TCPNATIVESERVICE, TCPPUSHEDMSGSERVICE, BASICSERVICE

Use the REQUESTEDMAXRECSIZE attribute to request the maximum length, in FRAMESIZE units, to be used in READ and WRITE operations for this file. The REQUESTEDMAXRECSIZE attribute is valid only for TCPNATIVESERVICE and TCPPUSHEDMSGSERVICE when the 1.2 or 3.0 release of TCP/IP is being used.

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.

Refer to the discussion of the ACTUALMAXRECSIZE attribute for more information.

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
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
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
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
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 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
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Supported
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.

For duplicated files, a copy number must be specified. The value returned by the RESULTLIST attribute shows the result that occurred on the specified copy. An attribute error occurs if a copy number is not specified.

If the file is not a port or duplicated file and a parameter is specified, an attribute error occurs.

ROWADDRESS

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

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

The value returned by the ROWADDRESS 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 row is not assigned. To find out only whether or not the area is allocated, use the AREAALLOCATED attribute.

The ROWADDRESS attribute requires the row number parameter. If the disk file is duplicated, the copy number parameter is required also.

Row numbers begin at 0; copy numbers begin at 1.

The following ALGOL examples show how to interrogate the ROWADDRESS attribute:

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

ADDRESS := DUPLDSK(ROWNBR,COPYNBR).ROWADDRESS;
```

ROWSINUSE

Kinds	Disk or CD
Interrogate	When assigned
Modify	Never
Type	Integer
Range	0 through 1000
Default	Not applicable
Stored Permanently	Disk or CD
Parameters	None
FTAM File Access	Not supported
Host Services Logical I/O	Not supported
Port Services	Not applicable

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

SAVEBACKUPFILE

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

Set the SAVEBACKUPFILE attribute to TRUE to prevent the printer backup file from being removed after it is printed as part of a print request. The Backup Processor utility can remove a backup file even when SAVEBACKUPFILE is TRUE.

When the SAVEBACKUPFILE value is FALSE, the backup file is removed after it is processed by the Print System unless the LOCKEDFILE attribute has the value TRUE. In this situation, the removal of the backup file is determined by the value of the REMOVELOCKEDFILES option of the Print System, which is assigned with the PS DEFAULT system command. The initial value is NEVER, which means that the backup file is not removed after printing if the LOCKEDFILE value of the file is TRUE. The other value is AUTOMATIC, which means one of two actions occurs if the LOCKEDFILE value of the file is TRUE;

- If the print request is automatically generated, the file is protected until it is printed and is removed after the printing is completed.
- If the print request is generated by the WFL *PRINT* statement, the file is protected until it is printed, but the file is not removed after the printing is completed.

If the value of the PRINTCOPIES attribute or the number of copies derived from the DESTINATION attribute is greater than 1, the file is not removed until the last copy of it has been printed.

This attribute does not apply to backup files on tape or to files that are not backup files.

SAVEFACTOR

Kinds	Disk or tape
Interrogate	Anytime
Modify	See following text
Type	Integer
Range	0 through 999
Default	0
Stored Permanently	Disk or tape
Parameters	None
FTAM File Access	Not supported
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. The SAVEFACTOR attribute returns the value for the physical file if the file is interrogated when the file is assigned.

When the first file on a tape volume with a LABELKIND value of B6500USASI expires, the system treats the tape as a scratch tape if 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 a scratch 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.

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 following information is true for the identified utilities:

- 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 *A Series System Software Utilities Manual* for information about the FILECOPY utility.

SCREEN

Kinds	Remote
Interrogate	When open
Modify	Never
Type	Boolean
Range	TRUE, FALSE
Default	Not applicable
Stored Permanently	No
Parameters	1 required
FTAM File Access	Not supported
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
Stored Permanently	No
Parameters	1 required
FTAM File Access	Not supported
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;
```

SECTORSIZE

Kinds	Disk or CD
Interrogate	When assigned
Modify	Never
Type	Integer
Range	See following text
Default	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
Host Services Logical I/O	Supported
Port Services	Not applicable

Interrogate the SECTORSIZE attribute to determine the size in bytes of a physical sector on the family where the file resides.

SECURITYGUARD

Kinds	Disk, CD, and tape
Interrogate	Anytime
Modify	Anytime (disk and CD) When closed (tape)
Type	Character string
Range	Not applicable
Default	Null string
Stored Permanently	Disk or tape
Parameters	None
FTAM File Access	Not supported
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 *A Series 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.

If the SECURITYGUARD attribute is not specified for the permanent file, or if the specified guard file is not on the specified pack, a SECURITYTYPE attribute value of PRIVATE is used instead of GUARDED or CONTROLLED. If the specified pack for the guard file is not present, the “REQUIRES PK” message is displayed, unless the opening program is a message control system (MCS), in which case the OPEN operation fails with a “FILE NOT AVAILABLE” message. The operator is required to either continue or terminate the process of accessing the permanent file.

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.

When the logical file is assigned to a physical file and the file is a CD-ROM or WORM file, a period (.) is always returned. For all other types of files when the logical file is assigned to a physical file, 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, a null string is returned.

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 value is changed while the file is open, the value of the physical file is changed immediately. If the value is changed when the file is closed, the value of the physical file is changed when the physical file is opened.

Tape File Security

The SECURITYGUARD attribute can be used for tape files if the InfoGuard security enhancements system 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 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 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.

Host Services Logical I/O Security

For 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;
```

SECURITYTYPE

Kinds	Disk, CD, port, or tape
Interrogate	Anytime
Modify	Anytime (disk or CD) When closed (port or tape)
Type	Mnemonic
Range	See following text
Default	See following text
Stored Permanently	Disk, CD, or tape
Parameters	None
FTAM File Access	Not supported
Host Services Logical I/O	Restricted values
Port Services	BNANATIVESERVICE, NETBIOSSESSIONSERVICE, OSINATIVESERVICE, OSISESSIONSERVICE, TCPNATIVESERVICE, TCPPUSHEDMSGSERVICE, 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. 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 physical file is created or the port file is opened by a task running with a usercode. If the InfoGuard security enhancements system 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 value is changed while the file is open, the SECURITYTYPE value of the physical file is changed immediately. If the value is changed when the file is closed, the SECURITYTYPE value of the physical file is changed when the file is opened.

WORM File Security

When you create a WORM file, which is a file with a KIND value of CD, you must assign a SECURITYTYPE value of either PUBLIC or PRIVATE. Regardless of the SECURITYTYPE value, a WORM file that is being created cannot be accessed by any other logical file until it is closed.

Note: *Once a WORM file has been opened for output, its SECURITYTYPE value can never be changed. The WORM media is similar to tape media in that a SECURITYTYPE of PRIVATE will not protect the data stored on the WORM media from unauthorized access if the physical security of the WORM media is not maintained.*

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, TCPPUSHEDMSGSERVICE, and BASICSERVICE port files must have the SECURITYTYPE value equal to PUBLIC. The fixed default setting of SECURITYTYPE can be changed by the *NW SET DEFAULT SECURITYTYPE* system command.

SECURITYTYPE is valid only for TCPNATIVESERVICE and TCPPUSHEDMSGSERVICE if you are using the 1.2 or 3.0 TCP/IP release.

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 InfoGuard security enhancements system 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 a tape file is accessed through 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 Host Services logical I/O in the *I/O Subsystem Programming Guide.*

Host Services Logical I/O Security

GUARDED, PRIVATE, and PUBLIC are supported by Host Services logical I/O; CONTROLLED is not supported by Host Services logical I/O.

SECURITYUSE

Kinds	Disk, CD, or tape
Interrogate	Anytime
Modify	Anytime (disk) Never (CD) When closed (tape)
Type	Mnemonic
Range	See following text
Default	IO
Stored Permanently	Disk or tape
Parameters	None
FTAM File Access	Not supported
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 value is changed while the file is open, the SECURITYUSE value of the physical file is changed immediately. If the value is changed when the file is closed, the SECURITYUSE value of the physical file is changed when it is opened.

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 and WORM File Security

When a WORM file is opened for output, the SECURITYUSE value returns IO. When an existing WORM or 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 InfoGuard security enhancements system 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 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 Host Services logical I/O in the *I/O Subsystem Programming Guide.*

SENSITIVEDATA

Kinds	Disk or CD
Interrogate	Anytime
Modify	When closed
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
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 InfoGuard 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
Stored Permanently	Disk or tape
Parameters	1 optional
FTAM File Access	Not supported
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) inhibits the assignment of scratch tapes unless the serial numbers match. If the SERIALNUMBER system option has been set and a serial number has not been specified, the "<file name> REQUIRES" message is displayed and the operator can respond with a DS (Discontinue), OU (Output Unit), or SN (Serial Number) system command. If the SERIALNUMBER system option is not set, the system uses any available scratch tape. For further information about using the OP(Options), DS, OU, and SN 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 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 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 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 tollowing text
Default	BNANATIVESERVICE (for port files)
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Supported
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 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 DEFAULT SERVICE* 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 can be provided across a BNA or OSI network environment. This service 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 allows A Series applications to communicate with PC applications on Novell® NetWare® local area networks (LANs).

continued

Novell is a registered trademark of Novell, Inc.
NetWare is a registered trademark of Novell, Inc.

continued

Mnemonic Value	Integer Value	Description
OSINATIVESERVICE	3	Offers a port service that is native to an OSI network environment.
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 is available only with the TCP/IP 3.1 or later release. This service offers byte-stream data transfer. **Note:** *This service will replace TCPNATIVESERVICE and TCPPUSHEDMSGSERVICE 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. **Note:** *This service will be deimplemented in a future release.*
TCPPUSHEDMSGSERVICE	1	Offers a message-oriented port service over a TCP/IP network environment. **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.

FTAM and Host Services Logical I/O Files

Use the SERVICE attribute to specify whether 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 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
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
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 unit, the following system message is displayed:

```
PK uu (FAMILYINDEX nnn) sss SECTORS REQUIRED
```

When space is unavailable on the original family member, the system displays the following message:

```
PK uu (FAMILYINDEX nnn) sss SECTORS REQUIRED
```

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 *A Series Disk Subsystem Administration and Programming 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
Stored Permanently	Disk or tape
Parameters	None
FTAM File Access	Not supported
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
Stored Permanently	Disk or tape
Parameters	None
FTAM File Access	Not supported
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
Stored Permanently	No
Parameters	None
FTAM File Access	Supported
Host Services Logical I/O	Not supported
Port Services	All

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
Stored Permanently	Disk or tape
Parameters	None
FTAM File Access	Not supported
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
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Supported
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 for information in Section 4 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**

[1:1] The attention bit for duplicated files. If this bit is on, the duplicated disk file error mask is found in field [27:16]. Refer to the topic of duplicated files in the *I/O Subsystem Programming Guide.*

[2:1] The I/O operation was canceled.

continued

continued

Field	**Contents**
[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.
[5:1]	A deleted or duplicated record was referenced in a file that has a FILEORGANIZATION value that is not NOTRESTRICTED.
[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.

Field **Row**

[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.

[27:16] If the attention bit [1:1] is on, these bits are the duplicated disk file error mask. If the bit is on in this field, the corresponding copy had an error. Note that bit 12 is copy number 1, bit 13 is copy number 2, and so forth. Bit 27 is unused. For further information, refer to the description of the COPYINERROR attribute.

[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 operation. This field corresponds to the value of the CURRENTRECORDLENGTH attribute. When the value of DUMMYFILE is TRUE, this field is 0 (zero).

Normally, error results for WRITE operations are reported when the buffer is reused. However, when Host Services logical I/O is used, the error result does not become available to the system until the buffer is reused. Because of this delay, the error is reported one buffer later than normal.

STATIONCOUNT

Kinds	Remote
Interrogate	When open
Modify	Never
Type	Integer
Range	Not applicable
Default	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
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
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
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 * - "L1Ø11.";
```

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
Stored Permanently	No
Parameters	1 required
FTAM File Access	Not supported
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
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
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
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
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
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
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.

This attribute requires a subfile index as a parameter if the MAXSUBFILES value is greater than 1.

The mnemonic values and meanings associated with the SUBFILEERROR attribute are as follows:

Mnemonic Value	Integer Value	Meaning
NOERROR	0	No error occurred.
DISCONNECTED	1	Communication with the corresponding port subfile was severed due to network failure.
DATALOST	2	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.
NOBUFFER	3	A WRITE operation with the DONTWAIT option failed because no buffer space was available.
NOFILEFOUND	4	Dialogue establishment failed because a matching dialogue endpoint was not found.
UNREACHABLEHOST	5	The foreign host became unreachable during the dialogue establishment.

continued

continued

Mnemonic Value	Integer Value	Meaning
UNSUPPORTEDFUNCTION	6	An attempted OPEN or AWAITOPEN operation on this port subfile resulted in a request for an unsupported function.
UNSUPPORTEDPROTOCOLTYPE	7	An attempt to establish a dialogue on this port subfile required an unsupported protocol type.
PROTOCOLERROR	8	A protocol error occurred.
LACKOFRESOURCES	9	An attempt to establish a dialogue on this port subfile failed due to a lack of resources.
HOSTNOTINHOSTGROUP	10	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 Version 2 port files only.
UNAUTHORIZEDFOR-APPLICATIONGROUP	11	An attempt to establish a dialogue on this port subfile failed because the user is not an authorized user of the specified application group.
IPCUNAVAILABLE	12	An attempt to establish a dialogue on this port subfile failed because the system software that supports ports is presently unavailable.
BADATTRIBUTESFOROPEN	13	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.
UNAVAILABLEFUNCTION	14	An attempt to establish a dialogue on this port subfile failed because the service requested is unavailable.
UNSUPPORTEDINTMODE	15	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.
NETWORKINGNOTSUPPORTED	16	An attempted OPEN operation on this port subfile failed because networking is not supported on the local host.
DISCONNECTEDDURINGOPEN	17	The matching process was aborted and the subfile is closed. This error can be returned anytime during dialogue establishment.
UNSUPPORTEDPARAMETER	18	The attempted OPEN, AWAITOPEN, CLOSE, or RESPOND statement was invoked with parameters or options not supported by the service of the subfile.

continued

continued

Mnemonic Value	Integer Value	Meaning
LOCALIPCNOTSUPPORTED	19	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.
UNSUPPORTEDTRANSLATION	20	An attempt to establish a dialogue failed because a system translation that the service could not provide was requested.
OPENREJECTED	21	An attempted OPEN operation failed because the correspondent endpoint rejected the request made by the port subfile.
ENDPOINTUNKNOWN	23	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.
INVALIDRESPOND	24	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.
INVALIDPROVIDER	25	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.
PROVIDERRESTRICTED	26	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.
TIMELIMITEXCEEDED	27	The attempted OPEN or AWAITOPEN operation failed because the value of CONNECTTIMELIMIT was exceeded.
UNSUPPORTEDAPPLCONTEXT	28	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.
BADASSOCIATEDDATA	31	The ASSOCIATEDDATA was unreadable. For a description of ASSOCIATEDDATA, refer to the *I/O Subsystem Programming Guide.*
ASSOCIATEDDATATOOLONG	32	The ASSOCIATEDDATA was too long. For a description of ASSOCIATEDDATA, refer to the *I/O Subsystem Programming Guide.*

continued

continued

Mnemonic Value	Integer Value	Meaning
BADATTRIBUTESFORRESPOND	33	The attempted RESPOND operation failed because one or more attributes were set to invalid values.
BADATTRIBUTESFORAWAITOPEN	34	The attempted AWAITOPEN operation failed because one or more attributes were set to invalid values.
WARNPORTATTIGNORED	36	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.
WARNCONTEXTCHANGED	37	The APPLICATIONCONTEXT attribute changed this negotiation. This warning does not cause the OPEN operation to fail.
WARNCLOSECOLLINITIATOR	38	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.
WARNCLOSECOLLRESPONDER	39	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.
ENDPOINTINCOMPWITHSERVICE	40	The dialogue endpoint cannot be accessed by this service.
UNSUPPORTEDPRIMITIVE	41	A statement not supported by the service of the subfile was invoked.
BASICSERVICEVIOLATION	42	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.
INVALIDFILESTATE	43	The RESPOND operation was invoked while the subfile was in an invalid file state.
TRANSPARENTNOTSUPPORTED	44	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.
OPENABORTBYCORRESPONDENT	45	The dialogue establishment failed because an abort was sent by the correspondent endpoint.
OPENREJECTEDTRANSIENT	46	The OPEN operation failed because the correspondent endpoint rejected the request for transient reasons.

continued

continued

Mnemonic Value	Integer Value	Meaning
WARNCONTEXTIGNORED	47	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.
WARNABORTDATAIGNORED	48	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.
PROVIDERGROUPUNDEFINED	49	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.
WARNDEFINEPROVIDERGROUP	50	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.
WARNINITIATINGPROVIDER	51	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.

continued

continued

Mnemonic Value	Integer Value	Meaning
NETBIOSNAMEINUSE	54	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.*
RESOLVERNOTAVAILABLE	56	The A Series Resolver is not running. The Resolver must be running to resolve the YOURDOMAINNAME file attribute value.
NAMESERVICENOTAVAILABLE	57	The A Series Resolver has not been configured with any known name server.
NAMESERVICEUNREACHABLE	58	The A Series Resolver cannot contact any name server to process the open request.
CONNECTIONINUSE	59	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 that uses the 3.1 or later release of TCP/IP must have a unique combination of these four attributes on the local host.

Refer to Appendix E, "OPEN, CLOSE, and RESPOND Results" and to the discussion of ports in the *I/O Subsystem Programming Guide.*

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)
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
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.

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.

If the KIND attribute value is CD, physical I/O must be done in integral multiples of the physical sector size of the WORM media. Refer to the *I/O Subsystem Programming Guide* for information about accomplishing this task.

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.

continued

continued

Mnemonic Value	Integer Value	Meaning
OUT	1	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], 3Ø, 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.

TANKING

Kinds	Remote
Interrogate	Anytime
Modify	When closed
Type	Mnemonic
Range	See following text
Default	UNSPECIFIED
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
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
Stored Permanently	Tape
Parameters	None
FTAM File Access	Not supported
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
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
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
Type	Word
Range	Not applicable
Default	Not applicable
Stored Permanently	Disk or CD
Parameters	None
FTAM File Access	Not supported
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*.

TITLE

Kinds	All
Interrogate	Anytime
Modify	Anytime (disk) When closed (other devices)
Type	Character string
Range	Not applicable
Default	FILENAME
Stored Permanently	Disk, CD, or tape
Parameters	1 optional
FTAM File Access	Restricted Use
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.

If the TITLE attribute is interrogated for a disk, CD-ROM, WORM, 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.

When you create a WORM file, observe the following restrictions for a file name:

- A file that is not associated with a usercode can have a maximum of 8 nodes.
- A file that is associated with a usercode can have a maximum of 6 nodes not including the usercode.
- The identifiers must be any combination of EBCDIC letters A through Z, digits 0 through 9, and an underscore (_). The final identifier can contain a single period (.). An identifier that contains a period must be enclosed in quotation marks (" ").

Note: *Once a WORM file has been opened for output, the file name can never be changed.*

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.

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.

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 or punch 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[Ø:44];

REPLACE POINTER(A) BY LINE.TITLE;
```

When the file is open and assigned to backup on pack, the fragment above responds in the following form:

```
*BD/ØØØ<job #>/ØØØ<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 in this section.

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 in this section.

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 should be assigned to the remote file. If a file with the TITLE value is not found in the DATCOMINFO 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.

Interchange Packs

For the special naming conventions used for files on interchange packs, refer to the discussion of interchange packs in the *I/O Subsystem Programming Guide.*

TOTALSECTORS

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

Interrogate the TOTALSECTORS attribute to determine the number of physical 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	Printer
Interrogate	Anytime
Modify	When closed
Type	Mnemonic
Range	See following text
Default	NOID
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
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
ASCII72	15	ASCII72
EBCDIC96	16	EBCDIC96
KATAKANA	17	KATAKANA

continued

continued

Mnemonic Value	Integer Value	Print Train
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

TRANSFORM

Kinds	Printer
Interrogate	Anytime
Modify	When closed
Type	Character string
Range	Not applicable
Default	Null string
Stored Permanently	Backup file
Parameters	None
FTAM File Access	Not supported
Host Services Logical I/O	Not 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 DIRECT, 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>────┘
```

If the library title is specified, the transform name must be an object within the specified library.

If 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)
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
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 A Series system in the *I/O Subsystem Programming Guide.* The DEFAULTTRANS value is not supported by 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.

continued

continued

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

The TRANSLATE attribute is not supported by the BNA Version 1 implementation. BNA Version 2 does support translation, but 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
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
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 occurring on the local host for the subfile dialogue. FALSE indicates that translation for the dialogue is occurring on the host specified by the YOURHOST attribute of the dialogue, or no translation is occurring at either host. Translation is supported only by BNANATIVESERVICE (BNA Version 2). All other services return a value of FALSE when TRANSLATING is interrogated.

The TRANSLATING attribute has a subfile index as a required parameter if the MAXSUBFILES value is greater than 1.

TRANSMISSIONNO

Kinds	Remote
Interrogate	When open
Modify	Never
Type	Integer
Range	Not applicable
Default	Not applicable
Stored Permanently	No
Parameters	1 required
FTAM File Access	Not supported
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
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
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 or punch backup, or remote
Interrogate	Anytime
Modify	When closed
Type	Boolean
Range	TRUE, FALSE
Default	TRUE for printer or punch backup, FALSE for remote
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
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 or punch 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 and Punch 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 or punch 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.

UNITNO

Kinds	All except port and remote
Interrogate	Anytime
Modify	When unassigned
Type	Integer
Range	Not applicable
Default	Not applicable
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
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 opened disk file or a printer or punch 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 in this section. 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 InfoGuard security-enhancement software is installed and the CLASS option value is S2, assignment to the unit specified by the UNITNO value requires operator intervention if the task is not privileged.

UNITS

Kinds	All
Interrogate	When unassigned (Host Services logical I/O)
	Anytime (local host)
Modify	When closed
Type	Mnemonic
Range	See following text
Default	WORDS
Stored Permanently	Disk or tape
Parameters	None
FTAM File Access	Supported
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.

Host Services logical I/O does not support interrogation of the UNITS attribute when the file is assigned. Use the FRAMESIZE attribute instead.

If the file is to be written to the WORM media—that is, a file with a KIND attribute value of CD—either the FRAMESIZE attribute must have a value of 8 or the UNITS attribute must have a value of CHARACTERS and the INTMODE attribute must have a value that specifies 8-bit-character encoding.

UPDATEFILE

Kinds	Disk
Interrogate	Anytime
Modify	When closed
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
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.

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
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
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.

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 *A Series System Administration Guide.*

USEDATE

Kinds	Disk or CD
Interrogate	When assigned
Modify	Never
Type	Integer
Range	0 through 99999
Default	0
Stored Permanently	Disk or CD
Parameters	None
FTAM File Access	Not supported
Host Services Logical I/O	Supported
Port Services	Not applicable

Interrogate the USEDATE attribute to determine when the file was last read by a user program or written to a user program. You can also find out the date when a code file was last executed. The value returned by USEDATE is an integer in the form YYDDD, where YY and DDD represent the year and day, respectively, in Julian form.

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

For code files, the USEDATE value is changed upon execution, at the beginning of a job or task. For a permanent disk file, the USEDATE value is changed when the file is closed if a READ, WRITE, SEEK, or SPACE operation has been performed on the file. For control files, guard files, the USERDATAFILE file, and other system files, the USEDATE value is updated when the file is closed. For the various disk files referenced by the operating system, the USEDATE 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.

For CD-ROM or WORM files, the USEDATE value never changes.

USERBACKUPNAME

Kinds	Printer or punch
Interrogate	Anytime
Modify	When closed
Type	Boolean
Range	TRUE, FALSE
Default	FALSE
Stored Permanently	No
Parameters	None
FTAM File Access	Not supported
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 or punch 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
Stored Permanently	No
Parameters	None
FTAM File Access	Supported
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 (disk)
Type	Word
Range	Not applicable
Default	0
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
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.

If the InfoGuard security-enhancement software is installed, the CLASS option value controls when the USERINFO attribute can be modified. The SECOPT (Security Options) system command is used to specify a CLASS option value. The following table describes when the CLASS option can be modified:

CLASS Value	Possible Modifications
S1 or S2	The value can be modified only 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.
U or S0	The value can be modified if the program has read or write access to the file. If the program does not have write access to the file, an attribute error and a warning are issued.

Note: *In a future release, modification of the USERINFO attribute will be unconditionally handled in the manner described for CLASS values S1 and S2.*

The values that the USERINFO attribute returns are unaffected by library maintenance.

USETIME

Kinds	Disk or CD
Interrogate	When assigned
Modify	Never
Type	Integer
Range	0 through 86400000000
Default	0
Stored Permanently	Disk or CD
Parameters	None
FTAM File Access	Not supported
Host Services Logical I/O	Not supported
Port Services	Not applicable

Interrogate the USETIME attribute to determine the time of day, in microseconds since midnight, associated with the USEDATE attribute. The USETIME attribute is changed only when the USEDATE attribute is changed.

For CD-ROM or WORM files, the USETIME value never changes.

VERSION

Kinds	Disk or tape
Interrogate	Anytime
Modify	When not in directory (disk)
	When closed (tape)
Type	Integer
Range	0 through 99
Default	0
Stored Permanently	Disk or tape
Parameters	None
FTAM File Access	Not supported
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 in this section.

WARNINGS

Kinds	Disk
Interrogate	When assigned
Modify	When assigned
Type	Character string
Range	Not applicable
Default	Null string
Stored Permanently	Disk
Parameters	None
FTAM File Access	Not supported
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
Stored Permanently	No
Parameters	1 required
FTAM File Access	Not supported
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	Valid domain name
Range	Not applicable
Default	Null string
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
Host Services Logical I/O	Not supported
Port Services	TCPIPNATIVESERVICE, TCPNATIVESERVICE, TCPPUSHEDMSGSERVICE

If you are using the 3.1 or later release of TCP/IP, 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.

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
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Supported
Host Services Logical I/O	Supported
Port Services	BNANATIVESERVICE, OSINATIVESERVICE, OSISESSIONSERVICE, TCPIPNATIVESERVICE, TCPNATIVESERVICE, TCPPUSHEDMSGSERVICE, BASICSERVICE

Use the YOURHOST attribute to specify the host name of the physical file or the correspondent host of the port subfile. The host name is 1 to 17 characters long.

This attribute requires a subfile index as a parameter if the MAXSUBFILES value is greater than 1.

Host Services Logical I/O Files

YOURHOST is a nonpreferred synonym for *HOSTNAME*. 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.

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.
TCPNATIVESERVICE and TCPPUSHEDMSGSERVICE using TCP/IP 1.2 or 3.0 releases	Local dialogue is not supported. That is, if the value of YOURHOST for the subfile equals the value of MYHOST for the subfile, the OPEN operation returns a LOCALIPCNOTSUPPORTEDRSLT (130) open error and a LOCALIPCNOTSUPPORTED (19) SUBFILEERROR.
TCPNATIVESERVICE and TCPPUSHEDMSGSERVICE using TCP/IP 3.1 or later releases	Local dialogue is supported.
TCPIPNATIVESERVICE	Local dialogue is supported.

YOURHOSTGROUP

Kinds	Port subfile
Interrogate	Anytime
Modify	When closed
Type	Character string
Range	Not applicable
Default	Null string
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
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.

The YOURHOSTGROUP attribute is not supported by the BNA Version 1 implementation.

This attribute requires a subfile index as a parameter if the MAXSUBFILES value is greater than 1.

YOURIPADDRESS

Kinds	Port subfile
Interrogate	Anytime
Modify	When closed
Type	Character string
Range	Valid IP address
Default	Null string
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
Host Services Logical I/O	Not supported
Port Services	TCPIPNATIVESERVICE, TCPNATIVESERVICE, TCPPUSHEDMSGSERVICE

If you are using the 3.1 or later release of TCP/IP, 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.

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
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
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.

For information about using the YOURNAME attribute with TCP files, refer to the PASSIVEOPEN attribute in this manual.

NETBIOSSESSIONSERVICE, TCPNATIVESERVICE, and TCPPUSHEDMSGSERVICE 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*.

YOURUSERCODE

Kinds	Port subfile
Interrogate	Anytime
Modify	When closed
Type	Character string
Range	Not applicable
Default	MYSELF.USERCODE
Stored Permanently	No
Parameters	1 optional
FTAM File Access	Not supported
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 and the value of OLDYOURUSERCODE is FALSE, use the YOURUSERCODE attribute to discover the usercode of the corresponding task. This process enables you to develop a more flexible security mechanism.

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 in this section and to the discussion of ports in the *I/O Subsystem Programming Guide.*

If the OLDYOURUSERCODE attribute value is the default value TRUE, assigning the null string to YOURUSERCODE causes its value to revert to the system-supplied default value. However, if OLDYOURUSERCODE is FALSE, assigning the null string to YOURUSERCODE causes its value to be the null string. Also, when the subfile is opened, YOURUSERCODE contains the usercode of the task that declares the complementary subfile. YOURUSERCODE reverts to its previous value when the subfile is closed.

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.

For files that are not port files, *YOURUSERCODE* is a nonpreferred synonym for *USERCODE.*

This attribute requires a subfile index as a parameter if the MAXSUBFILES value is greater than 1.

Section 3
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 (except duplicated files)
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
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 four entry points for UDDFAs. Their functions are to 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 *A Series 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 following table describes the library entry points:

Library Entry Point	Description
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.
NAMEDATTRIBUTESET	Changes the value of a UDDFA. 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 words "operation not allowed" are returned.
NAMEDATTRIBUTEDESTROY	Deletes a UDDFA. 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 follows:

```
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[*];
```

The parameters are described as follows:

Parameter	Description
FYLE	An open 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.

continued

continued

Parameter	Description
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

continued

continued

Value	Meaning
6	Array of 48-bit values (words)

Parameter	Description

Value	Meaning
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 3-1 for more information about the relationship of the TYPE, VALR, and VALA parameters.

Parameter	Description
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.

Table 3-1 summarizes what the VALR and VALA parameters contain when a specific TYPE parameter is selected and gives the size of information that is stored in the header.

Table 3–1. VALR and VALA Information by TYPE

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

continued

continued

Value	Meaning
8	Value out of range (for example, > 255 for a type-1 attribute)
9	VALA too small
10 through 31	Reserved

Field	Name	Description
[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 ALGOL example 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[Ø:4], VALA[Ø: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;
```

Example 3–1. Using UDDFAs with ALGOL

```
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 =        Ø #,
     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 =              Ø #,
     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)  !!
```

Example 3–1. Using UDDFAs with ALGOL (cont.)

```
                             " expected "             !!
                             STRING(EXPECTED,*)       !!
                             ", found "               !!
                             STRING(VALR,*);
                       DISPLAY(MSG);
                       END
                  ELSE;
             END   #;

     %=========================
OPEN(F);

REPLACE POINTER(NAME) BY "BLUE.";
TYPE := Ø &  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 := Ø &  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 := Ø &  BOOLEAN_TYPE  DATA_TYPEF;
VALR := Ø;
RESULT := NAMEDATTRIBUTESET (F,NAME,TYPE,VALR,VALA);
CHECK_RSLT (RESULT,FALSE,VALR,EXPECTED,NAME);

REPLACE POINTER(NAME) BY "HHHH4.";
TYPE := Ø &  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;
```

Example 3–1. Using UDDFAs with ALGOL (cont.)

```
   RESULT := NAMEDATTRIBUTEINTERROGATE (F,NAME,TYPE,VALR,VALA);
   CHECK_RSLT (RESULT,TRUE,VALR,EXPECTED,NAME);
   TYPE := Ø &  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 := Ø &  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 Ø 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 := Ø &  BYTE_ARRAY_TYPE  DATA_TYPEF;
   FOR A := 44 STEP -3 UNTIL Ø 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.
```

Example 3–1. Using UDDFAs with ALGOL (cont.)

Section 4
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*.

Tables 4-1 through 4-3 describe the values that are used with the preceding characteristics.

Table 4–1. Possible Values for Interrogate

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.

continued

Table 4–1. **Possible Values for Interrogate** (cont.)

Value	Meaning
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.

Table 4–2. Possible Values for Modify

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.

Table 4–3. Possible Values for Type

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.*

HYPERchannel is a registered trademark of Network Systems Corporation.

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.

IOCANCEL has no effect for disk and remote files.

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.

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.

It should be noted that the value returned from a 7-track tape drive is the number of 6-bit frames required to store the record on tape and not, if the data is EBCDIC or ASCII, the number of 8-bit characters in the record.

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: *In a future release, you will no longer be able to set field [35:12] in the IOCW.*

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 Large Systems (LS), 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 Entry and Medium Systems (EMS), 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 specify the I/O Error Mask field of the I/O control block (IOCB). The IOMASK attribute allows the user to specify the error conditions that the system is to overlook by not initiating recovery. The I/O subsystem assumes that you will handle these errors programmatically. Once the IOMASK attribute is modified, it is used instead of the system-supplied default. The IOMASK value reverts to the system-supplied default when the attribute is modified to the value 0 (zero).

On all systems, a value of 1 in bit 9 of IOMASK causes the normal system end-of-file (EOF) action to be suppressed. This action is of a particular utility for tape files because suppression of the EOF action of the system circumvents automatic volume switching.

On Large Systems (LS), the IOMASK attribute can cause certain conditions that would otherwise be suppressed to be reported through the IORESULT attribute. For example, a value of 1 in bit 8 of IOMASK causes end-of-tape conditions to be reported when they are encountered on tape READ operations.

The IOMASK attribute is ignored for host control files.

This buffer attribute cannot be used with a HYPERchannel file.

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.
[12:01]	HYPERchannel command rejected.
	Image printer page printed.
[11:01]	Card reader control card READ operation.
	HYPERchannel channel check.

continued

continued

Field	Description
	HYPERchannel A223 deadman timer (RUNTIMER).
[10:01]	Short block READ operation.
	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 Large Systems (LS), 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 Entry and Medium Systems (EMS), 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).

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.

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 Table A-1 to identify the general file attributes that can be used with each device type.

Table A–1. Device Types and Associated File Attributes

Attribute	Disk	CD	Printer	Punch	Remote	Tape	Other	Port
ACTUALMAXRECSIZE								X
AFTER			X	X				
ALIGNFILE			X					
ALIGNMENT			X					
ALTERDATE	X	X						
ALTERTIME	X	X						
ANYSIZEIO	X	X						
APL	X	X						
APPLICATIONCONTEXT								X
APPLICATIONGROUP								X
AREAALLOCATED	X	X						
AREALENGTH	X	X						
AREAS	X	X						
AREASECTORS	X	X						
AREASIZE	X	X						
ASSIGNTIME					X			
ATTERR	X	X	X	X	X	X	X	X
ATTVALUE	X	X	X	X	X	X	X	X
ATTYPE	X	X	X	X	X	X	X	X
AUTOUNLOAD						X		
AVAILABLE	X	X	X	X	X	X	X	X
AVAILABLEONLY								X

continued

Table A–1. **Device Types and Associated File Attributes** (cont.)

Attribute	Disk	CD	Printer	Punch	Remote	Tape	Other	Port
BACKUPKIND			X	X				
BANNER			X	X				
BLANK	X	X			X	X	X	X
BLOCK	X		X	X		X	X	
BLOCKEDTIMEOUT								X
BLOCKSIZE	X		X	X	X	X	X	
BLOCKSTRUCTURE	X	X	X	X	X	X	X	X
BUFFERS	X	X	X	X	X	X	X	X
BUFFERSIZE	X	X	X	X		X	X	
CARRIAGECONTROL			X					
CCSVERSION	X							
CENSUS					X			X
CHANGEDSUBFILE								X
CHANGEEVENT								X
CHECKPOINT			X					
COMPRESSING								X
COMPRESSION								X
COMPRESSIONCONTROL								X
COMPRESSIONREQUESTED								X
COPIES	X							
COPYINERROR	X							
COPYNAME	X							
CREATEPASSWORD	X							
CREATIONDATE	X	X				X		
CREATIONTIME	X	X						
CRUNCHED	X	X						
CURRENTBLOCKLENGTH	X		X	X	X	X	X	
CURRENTRECORDLENGTH	X	X	X	X	X	X	X	X
CYCLE	X					X		
DENSITY						X		
DEPENDENTINTMODE	X	X	X	X	X	X	X	

continued

Table A–1. Device Types and Associated File Attributes (cont.)

Attribute	Disk	CD	Printer	Punch	Remote	Tape	Other	Port
DEPENDENTSPECS	X	X	X	X	X	X	X	
DESTINATION			X	X				
DIALOGCHECKINTERVAL								X
DIALOGPRIORITY								X
DIOFILESTRUCTURE	X	X						
DIRECTION						X		
DISPOSITION					X			
DOCUMENTTYPE	X	X						
DONOTSEARCHNETWORK								X
DUMMYFILE	X	X	X	X		X	X	
DUPLICATED	X							
ENABLEINPUT					X			
EOF						X		
ERRORTYPE	X	X						
EXCLUSIVE	X							
EXTMODE	X	X	X	X	X	X	X	X
FAMILYINDEX	X	X						
FAMILYNAME	X	X	X	X				
FAMILYOWNER						X		
FILEEQUATED	X	X	X	X	X	X	X	X
FILEKIND	X	X				X		
FILELENGTH	X	X						
FILENAME	X	X	X	X	X	X	X	X
FILEORGANIZATION	X	X						
FILESECTION						X		
FILESTATE	X	X	X	X	X	X	X	X
FILESTRUCTURE	X	X						
FILETYPE	X	X	X	X	X	X	X	X
FILEUSE	X	X	X	X	X	X	X	
FLEXIBLE	X							
FORMID			X	X				

continued

Table A–1. **Device Types and Associated File Attributes** (cont.)

Attribute	Disk	CD	Printer	Punch	Remote	Tape	Other	Port
FRAMESIZE	X	X	X	X	X	X	X	X
FRAMESIZECENSUS								X
GENERATION	X					X		
HOSTNAME	X	X	X	X	X	X	X	X
INPUTEVENT					X			X
INPUTTABLE	X	X	X	X	X	X	X	
INTERCHANGE	X							
INTMODE	X	X	X	X	X	X	X	X
INTNAME	X	X	X	X	X	X	X	X
IOCLOCKS	X	X	X	X	X	X	X	X
IOINERROR	X	X				X		
KIND	X	X	X	X	X	X	X	X
LABEL	X	X	X	X	X	X	X	
LABELKIND						X		
LASTRECORD	X	X						
LASTSUBFILE								X
LICENSEKEY	X							
LINENUM			X					
LOCKEDFILE	X		X	X		X		
MAXCENSUS								X
MAXFRAMESIZECENSUS								X
MAXRECSIZE	X	X	X	X	X	X	X	X
MAXSUBFILES								X
MINRECSIZE	X	X	X	X	X	X	X	
MYDOMAINNAME								X
MYHOST								X
MYHOSTGROUP								X
MYIPADDRESS								X
MYNAME								X
MYUSE	X	X	X	X	X	X	X	
NETACCESSPOINT								X

continued

Table A–1. Device Types and Associated File Attributes (cont.)

Attribute	Disk	CD	Printer	Punch	Remote	Tape	Other	Port
NEWFILE	X	X	X	X	X	X	X	
NEXTRECORD	X	X	X	X		X	X	
NORECOVERY	X							
NORESOURCEWAIT	X							
NOTE	X		X	X				
OFNOTIFICATION	X	X			X	X	X	
OLDYOURUSERCODE								X
OPEN	X	X	X	X	X	X	X	X
OPTIONAL	X	X			X	X	X	
OUTPUTEVENT								X
OUTPUTTABLE	X	X	X	X	X	X	X	
PAGE			X					
PAGECOMP			X					
PAGESIZE			X					
PARITY						X		
PASSIVEOPEN								X
PENDINGPROTECTEDFILE		X						
PERMITTEDACTIONS	X	X						
POPULATION	X				X			
PRESENT	X	X	X	X	X	X	X	
PRINTCHARGE			X	X				
PRINTCOPIES			X	X				
PRINTDISPOSITION			X	X				
PRINTERCONTROL			X					
PRINTERKIND			X					
PRINTPARTIAL			X					
PROTECTION	X	X				X		
PROVIDERGROUP								X
RECEPTIONS					X			
RECORD	X		X	X		X	X	
RECORDINERROR	X	X				X		

continued

Table A–1. Device Types and Associated File Attributes (cont.)

Attribute	Disk	CD	Printer	Punch	Remote	Tape	Other	Port
RECORDLEVELLOCK	X							
RELEASEID	X							
REQUESTEDMAXRECSIZE								X
RESIDENT	X	X	X	X	X	X	X	
RESTRICTED	X	X						
RESULTLIST	X	X	X	X	X	X	X	X
ROWADDRESS	X	X						
ROWSINUSE	X	X						
SAVEBACKUPFILE			X	X				
SAVEFACTOR	X					X		
SCREEN					X			
SCREENSIZE					X			
SECTORSIZE	X	X						
SECURITYGUARD	X					X		
SECURITYTYPE	X	X				X		X
SECURITYUSE	X	X				X		
SENSITIVEDATA	X	X						
SERIALNO	X					X		
SERVICE	X							X
SINGLEUNIT	X							
SIZEMODE	X	X	X	X	X	X	X	
SIZEOFFSET	X	X	X	X	X	X	X	
SIZEVISIBLE	X	X	X	X	X	X	X	
SIZE2	X	X	X	X	X	X	X	
STATE	X	X	X	X	X	X	X	X
STATIONCOUNT					X			
STATIONLIST					X			
STATIONNAME					X			
STATIONSALLOWED					X			
STATIONSDENIED					X			
SUBFILEERROR								X

continued

Table A–1. Device Types and Associated File Attributes (cont.)

Attribute	Disk	CD	Printer	Punch	Remote	Tape	Other	Port
SYNCHRONIZE	X	X				X		
TANKING					X			
TAPEREELRECORD						X		
TIMELIMIT					X			
TIMESTAMP	X	X						
TITLE	X	X	X	X	X	X	X	X
TOTALSECTORS	X	X						
TRAINID			X					
TRANSFORM			X					
TRANSLATE	X	X	X	X	X	X	X	X
TRANSLATING	X	X	X	X	X	X	X	X
TRANSMISSIONNO					X			
TRANSMISSIONS					X			
TRIMBLANKS			X	X	X			
UNITNO	X	X	X	X		X	X	
UNITS	X	X	X	X	X	X	X	X
UPDATEFILE	X							
USECATALOG	X					X		
USEDATE	X	X						
USERBACKUPNAME			X	X				
USERCODE	X							
USERINFO	X	X						
USETIME	X	X						
VERSION	X					X		
WARNINGS	X							
WIDTH					X			
YOURDOMAINNAME								X
YOURHOST	X	X	X	X	X	X	X	X
YOURHOSTGROUP								X
YOURIPADDRESS								X
YOURNAME								X

continued

Table A–1. Device Types and Associated File Attributes (cont.)

Attribute	Disk	CD	Printer	Punch	Remote	Tape	Other	Port
YOURUSERCODE								X

Appendix B
Port File Attributes and Associated Services

Use Table B–1 to determine the port file attributes that are supported by specific port services. The table includes attributes that can be used by port files.

Table B–1. Port File Attributes and Associated Services

Attribute	BASIC	BNA	NETBIOS	OSI-N	OSI-S	TCPIP-N	TCP-N	TCP-P
ACTUALMAXRECSIZE	X	X	X	X	X	X	X	X
APPLICATIONCONTEXT				X				
APPLICATIONGROUP	X	X†		X	X			
ATTERR	X	X	X	X	X	X	X	X
ATTVALUE	X	X	X	X	X	X	X	X
ATTYPE	X	X	X	X	X	X	X	X
AVAILABLE	X	X	X	X	X	X	X	X
AVAILABLEONLY	X	X	X	X	X			
BLANK		X						
BLOCKEDTIMEOUT	X†	X†		X	X	X	X	X
BLOCKSTRUCTURE	X	X	X	X	X		X	X
BUFFERS		X†	X					
CENSUS	X	X	X	X	X			X
CHANGEDSUBFILE	X	X	X	X	X	X	X	X
CHANGEEVENT	X	X	X	X	X	X	X	X
COMPRESSING	X	X	X	X	X	X	X	X
COMPRESSION	X	X						

† For BNA Version 2 only.

continued

Legend

BASIC	BASICSERVICE
BNA	BNANATIVESERVICE
NETBIOS	NETBIOSSESSIONSERVICE
OSI-N	OSINATIVESERVICE
OSI-S	OSISESSIONSERVICE
TCPIP-N	TCPIPNATIVESERVICE
TCP-N	TCPNATIVESERVICE
TCP-P	TCPPUSHEDMSGSERVICE

Table B–1. Port File Attributes and Associated Services (cont.)

Attribute	BASIC	BNA	NETBIOS	OSI-N	OSI-S	TCPIP-N	TCP-N	TCP-P
COMPRESSIONCONTROL	X	X						
COMPRESSIONREQUESTED	X	X†						
CURRENTRECORDLENGTH	X	X	X	X	X	X	X	X
DIALOGCHECKINTERVAL		X†				X	X‡	X‡
DIALOGPRIORITY		X†						
DONOTSEARCHNETWORK		X†						
EXTMODE		X						
FILEEQUATED	X	X	X	X	X	X	X	X
FILENAME	X	X		X	X			
FILESTATE	X	X	X	X	X	X	X	X
FRAMESIZE	X	X	X	X	X	X	X	X
FRAMESIZECENSUS						X	X	
INPUTEVENT	X	X	X	X	X	X	X	X
INTMODE		X						
INTNAME	X	X	X	X	X	X	X	X
KIND	X	X	X	X	X	X	X	X
LASTSUBFILE	X	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	X
MYDOMAINNAME‡						X	X	X
MYHOST	X	X		X	X	X	X	X
MYHOSTGROUP		X						
MYIPADDRESS‡						X	X	X

† For BNA Version 2 only.

‡ For TCP/IP 3.1 and later releases only.

continued

Legend

BASIC	BASICSERVICE
BNA	BNANATIVESERVICE
NETBIOS	NETBIOSSESSIONSERVICE
OSI-N	OSINATIVESERVICE
OSI-S	OSISESSIONSERVICE
TCPIP-N	TCPIPNATIVESERVICE
TCP-N	TCPNATIVESERVICE
TCP-P	TCPPUSHEDMSGSERVICE

Table B–1. Port File Attributes and Associated Services (cont.)

Attribute	BASIC	BNA	NETBIOS	OSI-N	OSI-S	TCPIP-N	TCP-N	TCP-P
MYNAME	X	X	X	X	X	X	X	X
NETACCESSPOINT			X					
OLDYOURUSERCODE		X						
OUTPUTEVENT	X	X	X	X	X	X	X	X
PASSIVEOPEN			X			X	X	X
PROVIDERGROUP	X	X	X	X	X	X	X	X
REQUESTEDMAXRECSIZE	X	X	X	X	X		X§	X¶
RESULTLIST	X	X	X	X	X	X	X	X
SECURITYTYPE	X	X	X	X	X		X¶	X¶
SERVICE	X	X	X	X	X	X	X	X
STATE	X	X	X	X	X	X	X	X
SUBFILEERROR	X	X	X	X	X	X	X	X
TRANSLATE		X†						
TRANSLATING		X						
YOURDOMAINNAME‡						X	X	X
YOURHOST	X	X		X	X	X	X	X
YOURHOSTGROUP		X†						
YOURIPADDRESS‡						X	X	X
YOURNAME	X	X	X	X	X	X	X	X
YOURUSERCODE		X						

† For BNA Version 2 only.

‡ For TCP/IP 3.1 and later releases only.

§ For TCP/IP 1.2 and 3.0 releases only.

Legend

BASIC	BASICSERVICE
BNA	BNANATIVESERVICE
NETBIOS	NETBIOSSESSIONSERVICE
OSI-N	OSINATIVESERVICE
OSI-S	OSISESSIONSERVICE
TCPIP-N	TCPIPNATIVESERVICE
TCP-N	TCPNATIVESERVICE
TCP-P	TCPPUSHEDMSGSERVICE

Table B–2 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.

Table B–2. Additional Port File Attributes

Attribute	BASIC	BNA	NETBIOS	OSI-N	OSI-S	TCPIP-N	TCP-N	TCP-P
FILETYPE	X	X	X	X	X		X	X
HOSTNAME	X	X		X	X	X	X	X
IOCLOCKS	X	X	X	X	X	X	X	X
MAXRECSIZE	X	X	X	X	X	X	X	X
OPEN	X	X	X	X	X	X	X	X
TITLE	X	X	X	X	X	X	X	X
UNITS	X	X	X	X	X	?	X	X

Legend

BASIC	BASICSERVICE
BNA	BNANATIVESERVICE
NETBIOS	NETBIOSSESSIONSERVICE
OSI-N	OSINATIVESERVICE
OSI-S	OSISESSIONSERVICE
TCPIP-N	TCPIPNATIVESERVICE
TCP-N	TCPNATIVESERVICE
TCP-P	TCPPUSHEDMSGSERVICE

Appendix C
Host Services Logical I/O and FTAM File Attributes

Use Table C-1 to determine the file attributes that are valid for Host Services logical I/O files and FTAM files.

Table C–1. Host Services Logical I/O and FTAM File Attributes

Attribute	Host Services	FTAM
ALTERDATE	Supported	Supported
ALTERTIME	Supported	Supported
AREAALLOCATED	Supported	Not supported
AREALENGTH	Supported	Not supported
AREAS	Supported	Not supported
AREASECTORS	Supported	Not supported
AREASIZE	Restricted usage	Not supported
ATTERR	Supported	Supported
ATTVALUE	Supported	Supported
ATTYPE	Not supported	Supported
AUTOUNLOAD	Supported	Not supported
AVAILABLE	Supported	Not supported
BACKUPKIND	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

continued

Table C–1. Host Services Logical I/O and FTAM File Attributes (cont.)

Attribute	Host Services	FTAM
COPIES	Supported	Not supported
COPYINERROR	Supported	Not supported
COPYNAME	Supported	Not supported
CREATEPASSWORD	Not supported	Supported
CREATIONDATE	Supported	Supported
CREATIONTIME	Supported	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
DUPLICATED	Supported	Not supported
EXCLUSIVE	Not supported	Supported
EXTMODE	Restricted values	Restricted values
FAMILYINDEX	Supported	Not supported
FAMILYNAME	Supported	Not supported
FAMILYOWNER	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

continued

Table C–1. Host Services Logical I/O and FTAM File Attributes (cont.)

Attribute	Host Services	FTAM
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
HOSTNAME	Supported	Supported
INPUTTABLE	Supported	Not supported
INTMODE	Restricted values	Restricted values
INTNAME	Supported	Supported
KIND	Restricted values	Restricted values
LABEL	Supported	Not supported
LABELKIND	Supported	Not supported
LASTRECORD	Supported	Not supported
LASTSUBFILE	Supported	Not supported
LICENSEKEY	Restricted usage/values	Not supported
LINENUM	Supported	Not supported
LOCKEDFILE	Supported	Not supported
MAXRECSIZE	Restricted values	Restricted values
MINRECSIZE	Supported	Supported
MYUSE	Restricted usage	Not supported
NEWFILE	Supported	Supported
NEXTRECORD	Supported	Not supported
OFNOTIFICATION	Supported	Supported
OPEN	Supported	Supported
OPTIONAL	Supported	Supported
OUTPUTTABLE	Supported	Not supported

continued

Table C–1. Host Services Logical I/O and FTAM File Attributes (cont.)

Attribute	Host Services	FTAM
PAGE	Supported	Not supported
PAGECOMP	Supported	Not supported
PAGESIZE	Restricted usage	Not supported
PARITY	Supported	Not supported
PERMITTEDACTIONS	Not supported	Supported
PRINTDISPOSITION	Supported	Not supported
PRINTERCONTROL	Supported	Not supported
PRINTERKIND	Supported	Not supported
PRINTPARTIAL	Supported	Not supported
PROTECTION	Restricted usage	Not supported
PROVIDERGROUP	Not supported	Not supported
RELEASEID	Supported	Not supported
RESTRICTED	Supported	Not supported
RESULTLIST	Supported	Supported
SAVEFACTOR	Supported	Not supported
SCREEN	Supported	Not supported
SCREENSIZE	Supported	Not supported
SECTORSIZE	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

continued

Table C–1. **Host Services Logical I/O and FTAM File Attributes** (cont.)

Attribute	Host Services	FTAM
STATIONLIST	Supported	Not supported
STATIONNAME	Supported	Not supported
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
TRANSLATE	Restricted values	Not supported
TRANSLATING	Supported	Not supported
TRANSMISSIONNO	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 attributes by their attribute number and the other lists the attributes alphabetically and identifies the attribute number.

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	56	NETACCESSPOINT
58	FILEKIND	59	CRUNCHED
60	ROWSINUSE	61	USEDATE
62	An internal representation for the SERIALNO attribute	64	ROWADDRESS

continued

continued

66	INTERCHANGE	67	DUPLICATED
69	COPIES	72	INTNAME
73	UNITS	74	ATTERR
75	ATTYPE	76	ATTVALUE
77	IOCLOCKS	78	UNITNO
79	SECURITYGUARD	80	SECURITYTYPE
81	SECURITYUSE	84	RECORDINERROR
85	ERRORTYPE	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
100	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	135	BACKUPKIND
136	NEWFILE	140	MYNAME
141	YOURNAME	142	FILEUSE
143	HOSTNAME YOURHOST	144	FILEORGANIZATION
146	TIMESTAMP	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
163	COPYNAME	164	STATIONNAME
165	BLOCKSTRUCTURE	166	DEPENDENTSPECS

continued

continued

167	SCREENSIZE	168	CURRENTRECORDLENGTH CURRENTRECORD
169	SUBFILEERROR	170	FILESTATE
171	NEXTRECORD	172	LABELKIND
173	STATIONSALLOWED	174	COPYINERROR
175	USETIME	176	USERINFO
177	ALTERDATE	178	ALTERTIME
179	CREATIONTIME	180	BLANK
181	PRINTERKIND	182	PRINTERCONTROL
183	SIZEVISIBLE	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
200	OLDYOURUSERCODE	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

continued

continued

245	FRAMESIZECENSUS	246	MAXFRAMESIZECENSUS
249	AUTOUNLOAD	256	AREASECTORS
257	SECTORSIZE	258	TOTALSECTORS
264	DEPENDENTINTMODE	265	CCSVERSION
267	PAGECOMP	268	PRINTPARTIAL
269	LOCKEDFILE	281	PENDINGPROTECTEDFILE
283	YOURDOMAINNAME	284	MYDOMAINNAME
285	YOURIPADDRESS	286	MYIPADDRESS

The following list organizes the attributes alphabetically and identifies the attribute number:

ACTUALMAXRECSIZE (201)	AFTER (184)
ALIGNFILE (185)	ALIGNMENT (186)
ALTERDATE (177)	ALTERTIME (178)
ANYSIZEIO (230)	APL (19)
APPLICATIONCONTEXT (237)	APPLICATIONGROUP (202)
AREAALLOCATED (160)	AREALENGTH (159)
AREAS (18)	AREASECTORS (256)
AREASIZE (17)	ASSIGNTIME (113)
ATTERR (74)	ATTVALUE (76)
ATTYPE (75)	AUTOUNLOAD (249)
AVAILABLE (48)	AVAILABLEONLY (203)
BACKUPKIND (135)	BANNER (187)
BLANK (180)	BLOCK (44)
BLOCKEDTIMEOUT (204)	BLOCKSIZE (14)
BLOCKSTRUCTURE (165)	BUFFERS (26)
BUFFERSIZE (231)	CARRIAGECONTROL (45)
CCSVERSION (265)	CENSUS (108)
CHANGEDSUBFILE (158)	CHANGEEVENT (154)
CHECKPOINT (190)	COMPRESSING (205)
COMPRESSION (156)	COMPRESSIONCONTROL (206)
COMPRESSIONREQUESTED (207)	COPIES (69)
COPYINERROR (174)	COPYNAME (163)
CREATEPASSWORD (43)	CREATIONDATE (2)
CREATIONTIME (179)	CRUNCHED (59)

continued

continued

CURRENTBLOCK (28)	CURRENTBLOCKLENGTH (28)
CURRENTRECORD (168)	CURRENTRECORDLENGTH (168)
CYCLE (3)	DATE (2)
DENSITY (6)	DEPENDENTINTMODE (264)
DEPENDENTSPECS (166)	DESTINATION (191)
DIALOGCHECKINTERVAL (208)	DIALOGPRIORITY (209)
DIOFILESTRUCTURE (236)	DIRECTION (27)
DISPOSITION (111)	DOCUMENTTYPE (241)
DONOTSEARCHNETWORK (210)	DUMMYFILE (198)
DUPLICATED (67)	ENABLEINPUT (102)
EOF (36)	ERRORTYPE (85)
EXCLUSIVE (90)	EXTMODE (10)
FAMILY (101)	FAMILYINDEX (98)
FAMILYNAME (39)	FAMILYOWNER (218)
FAMILYSIZE (99)	FILEEQUATED (215)
FILEKIND (58)	FILELENGTH (232)
FILENAME (162)	FILEORGANIZATION (144)
FILESECTION (1)	FILESTATE (170)
FILESTRUCTURE (233)	FILETYPE (13)
FILEUSE (142)	FLEXIBLE (22)
FORMID (54)	FORMMESSAGE (54)
FRAMESIZE (161)	FRAMESIZECENSUS (245)
GENERATION (96)	HOSTNAME (143)
INPUTEVENT (152)	INPUTTABLE (93)
INTERCHANGE (66)	INTMODE (29)
INTNAME (72)	IOCLOCKS (77)
IOINERROR (87)	KIND (8)
LABEL (9)	LABELKIND (172)
LASTRECORD (49)	LASTSTATION (106)
LASTSUBFILE (106)	LICENSEKEY (220)
LINENUM (52)	LOCKEDFILE (269)
MAXCENSUS (157)	MAXFRAMESIZECENSUS (246)
MAXRECSIZE (15)	MAXSUBFILES (150)
MINRECSIZE (16)	MYDOMAINNAME (284)

continued

continued

MYHOST (151)
MYHOSTGROUP (211)
MYHOSTNAME (151)
MYIPADDRESS (286)
MYNAME (140)
MYUSE (20)
NETACCESSPOINT (56)
NEWFILE (136)
NEXTRECORD (171)
NORECOVERY (228)
NORESOURCEWAIT (109)
NOTE (192)
OFNOTIFICATION (244)
OLDYOURUSERCODE (200)
OPEN (30)
OPTIONAL (11)
OUTPUTEVENT (153)
OUTPUTTABLE (94)
PACKNAME (39)
PAGE (51)
PAGECOMP (267)
PAGESIZE (50)
PARITY (7)
PASSIVEOPEN (234)
PENDINGPROTECTEDFILE (281)
PERMITTEDACTIONS (242)
POPULATION (100)
PRESENT (31)
PRINTCHARGE (188)
PRINTCOPIES (193)
PRINTDISPOSITION (194)
PRINTERCONTROL (182)
PRINTERKIND (181)
PRINTPARTIAL (268)
PROTECTION (12)
PROVIDERGROUP (240)
RECEPTIONS (107)
RECORD (46)
RECORDINERROR (84)
RECORDLEVELLOCK (227)
REEL (1)
RELEASEID (221)
REQUESTEDMAXRECSIZE (212)
RESIDENT (42)
RESTRICTED (216)
RESULTLIST (243)
ROWADDRESS (64)
ROWSINUSE (60)
SAVEBACKUPFILE (195)
SAVEFACTOR (5)
SCREEN (116)
SCREENSIZE (167)
SECTORSIZE (257)
SECURITYGUARD (79)
SECURITYTYPE (80)
SECURITYUSE (81)
SENSITIVEDATA (97)
SERIALNO (119)
SERVICE (235)
SINGLEPACK (40)
SINGLEUNIT (40)
SIZEMODE (32)
SIZEOFFSET (33)
SIZEVISIBLE (183)
SIZE2 (34)
STATE (35)
STATIONCOUNT (99)
STATIONLIST (101)

continued

continued

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)

UNITNO (78)

UNITS (73)

UPDATEFILE (47)

USECATALOG (95)

USEDATE (61)

USERBACKUPNAME (197)

USERCODE (155)

USERINFO (176)

USETIME (175)

VERSION (4)

WARNINGS (214)

WIDTH (118)

YOURDOMAINNAME (283)

YOURHOST (143)

YOURIPADDRESS (285)

YOURNAME (141)

YOURUSERCODE (155)

Appendix E
OPEN, CLOSE, and RESPOND Results

This appendix contains two tables. Table E-1 contains the OPEN, CLOSE, and RESPOND results that might be returned during an OPEN, CLOSE, or RESPOND operation and Table E-2 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 or PUNCH, and the BACKUPKIND value is TAPE.

Table E–1. OPEN, CLOSE, and RESPOND Results

Mnemonic Value	Integer Value	Meaning
NOTAVAILABLERSLT	0	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.

continued

Table E–1. OPEN, CLOSE, and RESPOND Results (cont.)

Mnemonic Value	Integer Value	Meaning
OKRSLT	1	The operation was successful. If the control option specified is DONTWAIT or OFFER, this result indicates only that the process was successfully started.
NOFILEFOUNDRSLT	2	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.
OFRSLT	3	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.
GENEALOGYMISMATCHRSLT	4	The genealogy of the file does not match.
SERIALNOMISMATCHRSLT	6	The serial number does not match.
WARNCONTEXTCHANGEDRSLT	7	See Table E-2.
FILEOFFLINERSLT	8	The file is not on disk, but there is a backup copy of the file listed in the catalog or archive directory.
WARNPORTATTINGNOREDRSLT	9	See Table E-2.
NOTAPERESOURCESRSLT	10	No resources are available for a tape file.
WARNCLOSECOLLINITIATORRSLT	11	See Table E-2.
NOFAMILYFOUNDRSLT	12	A required disk is not online.
WARNCLOSECOLLRESPONDERRSLT	13	See Table E-2.
ACCESSCODEMISMATCHRSLT	14	The file is not available due to ACCESSCODE protection in the directory of the user.

continued

Table E–1. OPEN, CLOSE, and RESPOND Results (cont.)

Mnemonic Value	Integer Value	Meaning
WARNCONTEXTIGNOREDRSLT	15	See Table E-2.
WARNABORTDATAIGNOREDRSLT	17	See Table E-2.
INVALIDKINDLISTRSLT	18	An invalid KIND list or no KIND list is specified on a foreign host.
WARNDEFINEPROVIDERGROUPRSLT	19	See Table E-2.
DIALOGFAILURERSLT	20	An error occurred during initiation of a Host Services Logical I/O dialogue when the host was specified by the HOSTNAME attribute and the OPEN statement with the AVAILABLE option was specified. This error can occur when the HOSTNAME attribute specifies an unavailable or unknown host.
WARNINITIATINGPROVIDERRSLT	21	See Table E-2.
WARNFTAMFILEUSERSLT	23	See Table E-2.
WARNPURGELOCKEDFILETAPERSLT	25	See Table E-2.
WARNPURGEWRITEPROTECTEDTAPERSLT	27	See Table E-2.
FILENOTOPENRSLT	30	An attempt was made to close a closed subfile.
IOERRORRSLT	32	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.
RECORDCOUNTMISMATCHRSLT	34	The record counts in the EOF1 and EOV1 labels for a tape file do not match the number of records read.
BLOCKCOUNTMISMATCHRSLT	36	The block counts in the EOF1 and EOV1 labels for a tape file do not match the number of blocks read.

continued

Table E–1. OPEN, CLOSE, and RESPOND Results (cont.)

Mnemonic Value	Integer Value	Meaning
UNAVAILABLEHOSTRSLT	38	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.
FILENOTCLOSEDRSLT	40	The value of the FILESTATE attribute for the specified subfile is not CLOSED.
BADSUBFILEINDEXRSLT	42	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.
ERRORINOPENRSLT	44	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.

continued

Table E-1. OPEN, CLOSE, and RESPOND Results (cont.)

Mnemonic Value	Integer Value	Meaning
ERRORINCLOSERSLT	46	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.
UNSUPPORTEDFUNCTIONRSLT	48	An attempted OPEN or AWAITOPEN operation on this port subfile resulted in a request for an unsupported function.
BADFILEORGANIZATIONRSLT	52	The logical file cannot be assigned to a physical file; the FILEORGANIZATION value is incompatible.
UNSUPPORTEDPROTOCOLTYPERSLT	54	A request was made to use an unsupported protocol type.
PROTOCOLERRORRSLT	56	A protocol error was detected.
LACKOFRESOURCESRSLT	58	The OPEN or AWAITOPEN operation failed due to a lack of resources.
HOSTNOTINHOSTGROUPRSLT	60	The OPEN request on this subfile failed because the YOURHOST value is not a legal member of the specified YOURHOSTGROUP.
UNAUTHORIZEDFORAGRSLT	62	The OPEN or AWAITOPEN operation failed because the user is not authorized to use the specified application group.
SUPPORTLIBUNAVAILABLERSLT	64	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.

continued

Table E–1. OPEN, CLOSE, and RESPOND Results (cont.)

Mnemonic Value	Integer Value	Meaning
BADATTRIBUTESRSLT	66	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.
UNAVAILABLEFUNCTIONRSLT	68	The OPEN or AWAITOPEN operation on this subfile failed because the function sought was not available for this port service.
UNSUPPORTEDINTMODERSLT	70	The OPEN request on this subfile failed because the specified INTMODE value is not supported on the host specified by YOURHOST.
NETWORKINGNOTSUPPORTEDRSLT	72	The OPEN or AWAITOPEN operation on this subfile failed because the networking feature is not supported on this host.
DISCONNECTEDDURINGOPENRSLT	74	The matching process for an OPEN or AWAITOPEN operation was aborted by the provider.
ACCESSRESTRICTEDRSLT	76	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.*
CANNOTCREATERESTRICTEDFILERSLT	78	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.
UNSUPPORTEDDEVICETYPERSLT	80	A device type was specified that logical I/O does not support.

continued

Table E–1. OPEN, CLOSE, and RESPOND Results (cont.)

Mnemonic Value	Integer Value	Meaning
INCOMPATIBLEWITHMCPRSLT	82	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.
SETMODEFAILEDRSLT	86	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.
UNSUPPORTABLEBLOCKSIZERSLT	88	The file cannot be opened because the tape drive cannot support the BLOCKSIZE value of the physical or logical file.
FAMILYLIMITRSLT	90	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 *A Series Disk Subsystem Administration and Operations Guide* for information on the disk resource control system.

continued

Table E–1. OPEN, CLOSE, and RESPOND Results (cont.)

Mnemonic Value	Integer Value	Meaning
INTEGRALLIMITRSLT	92	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 *Disk Subsystem Guide* for information on the disk resource control system.
ILLEGALMYUSERSLT	94	The MYUSE attribute value is incompatible with the device.
ILLEGALNEWFILERSLT	96	The NEWFILE attribute value is incompatible with the device.
OPENAFTERLOCKRSLT	98	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.
INCOMPATIBLEBLOCKINGRSLT	100	The BLOCKSIZE attribute value is incompatible with other attributes of the file.
SHORTTAPEBLOCKINGRSLT	102	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*.
LOGICALPHYSICALMISMATCHRSLT	104	No physical file can match the attribute values of the logical file.
FRAMESIZEINCOMPATIBLERSLT	106	The FRAMESIZE and INTMODE attribute values are incompatible. Refer to the description of the FRAMESIZE attribute.

continued

Table E-1. OPEN, CLOSE, and RESPOND Results (cont.)

Mnemonic Value	Integer Value	Meaning
SIZEVISIBLECONFLICTRSLT	108	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.
INVALIDANYSIZEIOSPECRSLT	110	The specified value of the ANYSIZEIO attribute is incompatible with other attributes of the file.
AREALENGTHTOOBIGRSLT	112	The size requested for an area requires more sectors than the maximum number of sectors allowed in an area.
INTMODEFRAMESIZECONFLICTRSLT	114	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.
INCOMPATIBLEFILESTRUCTERRSLT	116	The value of FILESTRUCTURE for the logical file is incompatible with the value of FILESTRUCTURE for the permanent file.
MUSTBESTREAMRSLT	118	The value of FILESTRUCTURE for the permanent file is not STREAM.
INCONSISTENTMAXRECSIZERSLT	120	The MAXRECSIZE values for the permanent and logical files are not equal.
INCONSISTENTBLOCKSIZERSLT	122	The BLOCKSIZE value for the logical file is inconsistent with the BLOCKSIZE value of the permanent file.
MAXRECSIZEGTRAREALENGTHRSLT	124	The MAXRECSIZE value of a file with a FILESTRUCTURE value of STREAM exceeds the value of the AREALENGTH attribute.
FRAMESIZEAREALENGTHCONFLICTRSLT	126	The length of an area of a file with a FILESTRUCTURE value of STREAM is not an integral number of the FRAMESIZE units.

continued

Table E–1. **OPEN, CLOSE, and RESPOND Results** (cont.)

Mnemonic Value	Integer Value	Meaning
UNSUPPORTEDPARAMETERRSLT	128	The statement was invoked with parameters or parameter values not supported by the service of the port subfile.
LOCALIPCNOTSUPPORTEDRSLT	130	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.
UNSUPPORTEDTRANSLATIONRSLT	132	The service of the subfile does not provide the translation requested.
OPENREJECTEDRSLT	134	The OPEN request failed because the correspondent application rejected the request made by the subfile.
ENDPOINTUNKNOWNRSLT	138	The OPEN request failed because the local system operator failed to register the endpoint or the endpoint is unknown at the correspondent host.
INVALIDRESPONDRSLT	140	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.
INVALIDPROVIDERRSLT	142	The provider group specified does not provide the port service requested for the subfile.
PROVIDERRESTRICTEDRSLT	144	Providers currently support subfiles of the port file that cannot be mixed with the provider of the service of this subfile.
TIMELIMITEXCEEDEDRSLT	146	The OPEN or AWAITOPEN operation failed because the value of CONNECTTIMELIMIT was exceeded.
UNSUPPORTEDAPPLCONTEXTRSLT	148	The correspondent rejected the request because it did not support the application context.

continued

Table E–1. OPEN, CLOSE, and RESPOND Results (cont.)

Mnemonic Value	Integer Value	Meaning
BADASSOCIATEDDATARSLT	154	The ASSOCIATEDDATA was unreadable.
ASSOCIATEDDATATOOLONGRSLT	156	The ASSOCIATEDDATA was too long.
BADATTRIBUTESFORRESPONDRSLT	158	One or more attributes were set to invalid values on a RESPOND operation.
BADATTRIBUTESFORAWAITOPENRSLT	160	One or more attributes were set to invalid values on an AWAITOPEN operation.
SECTORSTREAMREQDEPSPECSRSLT	162	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.
FILESTRUCTURENOTSETRSLT	164	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.
ENDPTINCOMPATIBLESERVICESLT	166	The dialogue endpoint cannot be accessed by this service.
UNSUPPORTEDPRIMITIVERSLT	168	A statement not supported by the service of the port file was invoked.
BADCONNECTTIMELIMITRSLT	170	The OPEN or AWAITOPEN operation failed because the value of the CONNECTTIMELIMIT attribute is invalid.
BASICSERVICEVIOLATIONRSLT	172	A protocol message from the correspondent system or endpoint was received that used a feature that BASICSERVICE does not support. The dialogue is closed.
INVALIDFILESTATERSLT	174	The RESPOND operation was invoked while the subfile was in an invalid file state.

continued

Table E–1. OPEN, CLOSE, and RESPOND Results (cont.)

Mnemonic Value	Integer Value	Meaning
TRANSPARENTNOTSUPPORTEDRSLT	176	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.
CANNOTACCESSFILEKINDRSLT	178	The permanent file cannot be opened because of the value of its FILEKIND attribute.
KEYEDOPENFAILURERSLT	180	The KEYEDIO or KEYEDIOII library indicated an error condition, which caused the OPEN request to fail.
UNSUPDMSIIFSRSLT	182	This value of FILESTRUCTURE is not supported when DMIO is TRUE. Information about the DMIO attribute is found in the *A Series DMALGOL Programming Reference Manual.*
INSUFFICIENTSPACERSLT	184	While the buffers were being flushed, a no-space condition occurred, and the NORESOURCEWAIT attribute is TRUE.
UNMATCHEDDIOFILESTRUCTURERSLT	186	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.
OPENREJECTEDTRANSIENTRSLT	190	The OPEN operation failed because the correspondent endpoint rejected the request for transient reasons.
PROVIDERGROUPUNDEFINEDRSLT	192	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.

continued

Table E–1. OPEN, CLOSE, and RESPOND Results (cont.)

Mnemonic Value	Integer Value	Meaning
OPENABORTBYCORRESPONDENTRSLT	194	The OPEN operation failed because an abort was sent by the correspondent endpoint.
NOUSERCODERSLT	196	The task is running without a USERCODE value, and an attempt was made to open a file through Host Services Logical I/O.
ASSIGNEDTAPECHGDRSLT	198	An operator physically replaced a tape on a drive that was in use.
NOTRANSLATETABLERSLT	200	One or both translate tables are missing. If only one table is missing, it is the required table.
WORMBEINGWRITTENRSLT	208	An OPEN operation was attempted on a WORM file that is currently open for output. Only the logical file that is creating output for the file can access the file until the file is closed.
WORMBADATTRIBUTESRSLT	210	One of the following attributes has a value that is not supported for a WORM file: BLOCKSTRUCTURE, FILETYPE, FRAMESIZE, UNITS, MAXRECSIZE, FILESTRUCTURE, DIOFILESTRUCTURE, INTMODE, EXTMODE, FILEKIND, or SECURITYTYPE.
WORMBADTITLERSLT	212	The specified FILENAME value does not conform to restrictions imposed by ISO 9660 standards.
WORMBADRECSZRSLT	214	A WORM file with a PROTECTION attribute value of PROTECTED did not have a MAXRECSIZE attribute value that can be evenly divided into the SECTORSIZE attribute value.
WORMFILEDIRCONFLICTRSLT	216	A WORM file has a FILENAME attribute value that is the same as a directory name. Adding a file with the specified FILENAME value would conflicts with an existing file.
WORMEXCESSDIRSRSLT	218	The specified FILENAME attribute value has more nodes than the ISO 9660 standards permit.

continued

Table E-1. OPEN, CLOSE, and RESPOND Results (cont.)

Mnemonic Value	Integer Value	Meaning
WORMDIRTOOBIGRSLT	220	The specified FILENAME attribute value would cause a node in the directory hierarchy to become too large to be stored in memory in an A Series array row.
WORMDUPFILENAMERSLT	222	A file already exists on the WORM media with the specified FILENAME attribute value.
WORMCREATESECRSLT	224	The user does not have privileges that allow for the creation of a WORM file with the specified FILENAME WORM media is owned by a different user.
WORMNOSPACERSLT	226	There is no more space on the specified WORM media.
NETBIOSNAMEINUSERSLT	228	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.*
RESOLVERNOTAVAILABLERSLT	240	The A Series Resolver is not running. The Resolver must be running to resolve the YOURDOMAINNAME file attribute value.
NAMESERVICENOTAVAILABLERSLT	242	The A Series Resolver has been configured with an unknown name server.
NAMESERVICEUNREACHABLERSLT	244	The A Series Resolver cannot contact any name server to process the open request.
CONNECTIONINUSERSLT	246	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 that uses the 3.1 or later release of TCP/IP must have a unique combination of these four attributes on the local host.

Table E-2 identifies OPEN and CLOSE results that are warnings. The OPEN or CLOSE operation does not fail.

Table E–2. OPEN and CLOSE Warnings

Mnemonic Value	Integer Value	Meaning
WARNCONTEXTCHANGEDRSLT	7	The APPLICATIONCONTEXT attribute changed this negotiation.
WARNPORTATTIGNOREDRSLT	9	One or more port attributes were set to values invalid for the service of the subfile and are being ignored.
WARNCLOSECOLLINITIATORRSLT	11	The ORDERLY CLOSE operation of the dialogue cannot complete because a collision of CLOSE requests occurred. This endpoint is the initiator of the dialogue.
WARNCLOSECOLLRESPONDERRSLT	13	The ORDERLY CLOSE operation cannot complete because a collision of CLOSE requests occurred. This endpoint is the responder of the dialogue.
WARNCONTEXTIGNOREDRSLT	15	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.
WARNABORTDATAIGNOREDRSLT	17	The associated data parameter specified with the CLOSE ABORT statement was not sent with the abort because it is too long.
WARNDEFINEPROVIDERGROUPRSLT	19	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.

continued

Table E–2. OPEN and CLOSE Warnings (cont.)

Mnemonic Value	Integer Value	Meaning
WARNINITIATINGPROVIDERRSLT	21	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.
WARNFTAMFILEUSERSLT	23	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.
WARNPURGELOCKEDFILETAPERSLT	25	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.
WARNPURGEWRITEPROTECTEDTAPERSLT	27	A CLOSE operation with a disposition of purge was unable to purge the specified tape because the tape was write-protected.

Appendix F
I/O Result Enumerated Values

Table F-1 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 table shows the mnemonic value of the result, the integer value of the result, and the meaning of the result.

Table F-1. List of I/O Result Enumerated Values

Mnemonic Value	Integer Value	Meaning
No mnemonic	< 0	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.
NOERROR	0	No error occurred.
NOTREADYTYPE	1	Valid only for IOERRORTYPE. Reported as NOTREADY by other attributes. This result indicates that a peripheral unit is not ready.
ASSIGNMENTDENIED	1	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.*

continued

Table F–1. List of I/O Result Enumerated Values (cont.)

Mnemonic Value	Integer Value	Meaning
PARITYERRORTYPE	2	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.
ASSIGNMENTPOSTPONED	2	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.*
REWINDINGTYPE	3	Valid only for IOERRORTYPE. Reported as REWINDING by other attributes. This result indicates the tape drive is rewinding.
ILLEGALFILEUSE	3	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.*
DESCRIPTORERROR	4	A descriptor error occurred (direct I/O).
DATAERROR	4	A data error occurred (normal I/O).
BEGINNINGORENDOFTAPE	5	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.

continued

Table F–1. List of I/O Result Enumerated Values (cont.)

Mnemonic Value	Integer Value	Meaning
WLOOREOF	6	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.
WRITELOCKOUT	6	Not valid for IOERRORTYPE. Reported as WLOOREOF by IOERRORTYPE. This result indicates a write lockout occurred.
READPASTROW	7	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.
CANCELED	8	I/O was canceled. If returned by IOERRORTYPE, this result includes the conditions reported as IMPLICITOPENFAILED and UNSPECIFIEDEXCEPTION by other attributes.
SHORTRECORD	9	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.
BREAKONOUTPUT	10	This result indicates a break on output occurred (remote files only).
PACKPROGRAMOROPERATORERROR	11	Valid only for IOERRORTYPE. This result indicates a disk pack program or operator error occurred.

continued

Table F–1. List of I/O Result Enumerated Values (cont.)

Mnemonic Value	Integer Value	Meaning
PACKHARDWAREERROR	12	Valid only for IOERRORTYPE. This result indicates a disk pack hardware error occurred.
PACKHARDWAREORDATAERROR	13	Valid only for IOERRORTYPE. This result indicates a disk pack hardware or data error occurred.
PACKDATAERROR	14	Valid only for IOERRORTYPE. This result indicates a disk pack data error occurred.
TIMEOUT	15	A disk storage unit timeout, blank tape timeout, or time limit was exceeded for remote files.
DMSROWLOCKOUT	16	Valid only for IOERRORTYPE. This result indicates an area WRITE operation locked out by Data Management System II (DMSII). The program has read an area that cannot be written to.
HCWRITETIMEOUT	18	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.
BADIOPARTNER	19	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.
NOWRITEPARTNER	20	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.
WRONGHUBINDEX	21	Valid only for IOERRORTYPE. This result indicates that a host control file READ operation with the READPARTNER specified received input from a different hubindex.

continued

Table F-1. List of I/O Result Enumerated Values (cont.)

Mnemonic Value	Integer Value	Meaning
OPERATIONABORT	22	Valid only for IOERRORTYPE. Refer to the HY file IOERRORTYPE values for direct I/O HY files in the *I/O Subsystem Programming Guide.*
REMOTEOPERATIONABORT	23	Valid only for IOERRORTYPE. Refer to the HY file IOERRORTYPE values for direct I/O HY files in the *I/O Subsystem Programming Guide.*
DEADMANTIMEOUT	24	Valid only for IOERRORTYPE. Refer to the HY file IOERRORTYPE values for direct I/O HY files in the *I/O Subsystem Programming Guide.*
DATACONFLICT	25	Valid only for IOERRORTYPE. Refer to the HY file IOERRORTYPE values for direct I/O HY files in the *I/O Subsystem Programming Guide.*
INPROGRESS	26	Valid only for IOERRORTYPE. An I/O operation is not complete.
NOTREADY	27	Not valid for IOERRORTYPE. Reported as NOTREADYTYPE by IOERRORTYPE. This result indicates that the peripheral unit was no ready.
PARITYERROR	28	Not valid for IOERRORTYPE. Reported as PARITYERRORTYPE by IOERRORTYPE. This result indicates that a parity error has occurred.
REWINDING	29	Not valid for IOERRORTYPE. Reported as REWINDINGTYPE by IOERRORTYPE. This result indicates that the tape drive is rewinding.
ADDNONDELCAPABLE	30	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.
DELETENONDELCAPABLE	31	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.

continued

Table F–1. List of I/O Result Enumerated Values (cont.)

Mnemonic Value	Integer Value	Meaning
UPDATENOREAD	32	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.
BEYONDEXTENDAREA	33	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.
UPDATENOTLOCKED	34	For KEYEDIOII files created with the RPG extended file system (EFS), an attempt was made to update an unlocked record.
POSITIONOCCUPIED	35	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.
SEQWRITETODIRECT	36	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.
FAMILYLIMITERR	39	This result indicates that an attempt was made to exceed the family limit.
INTEGRALLIMITERR	40	This result indicates that the family integral limit was exceeded.
TEMPFILEEXCEEDED	41	This result indicates that an attempt was made to exceed the temporary file limit.
URGENTFLAG	42	For port files, this result indicates that the message read contains data marked URGENT.
BADWRITEOPTION	43	For port files, a WRITE option that is inappropriate to the value of the SERVICE attribute was specified.
UNDEFINEDERROR	44	This result indicates that an undefined error occurred.
ENDOFTAPE	45	Not valid for IOERRORTYPE. Reported as BEGINNINGORENDOFTAPE by IOERRORTYPE. This result indicates that an end-of-tape condition occurred.

continued

Table F–1. List of I/O Result Enumerated Values (cont.)

Mnemonic Value	Integer Value	Meaning
DELETEDDISKAREA	45	Not valid for IOERRORTYPE. Reported as BEGINNINGORENDOFTAPE by IOERRORTYPE. The I/O operation referenced a deleted disk area.
ENDOFFILE	46	Not valid for IOERRORTYPE. Reported as WLOOREOF by IOERRORTYPE. An end-of-file (EOF) condition was encountered.
SHORTBLOCK	47	A short block was read, or if the FILESTRUCTURE attribute value is STREAM, a short record at the end of the file was read.
CONTROLCARD	48	A control card was read.
ENDOFPAGE	49	An end-of-page condition was encountered.
SECURITYERROR	50	A security violation has been attempted.
DSSABORT	51	DSS Management detected an error while communicating with the remote host.
INVALIDSUBFILE	52	For port files, an invalid subfile index was specified.
NODATAFORREAD	53	For port files, a READ operation with the DONTWAIT option failed because no data was available.
BROADCASTWRITEFAILED	54	For port files, a broadcast WRITE operation failed for one or more subfiles.
NOBUFFERFORWRITE	55	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.

continued

Table F–1. **List of I/O Result Enumerated Values** (cont.)

Mnemonic Value	Integer Value	Meaning
DELETEDORDUPLICATERECORD	56	For files with a FILEORGANIZATION value of RELATIVE, an attempt was made to read a deleted record or to write a duplicate record.
ENDOFMEDIUM	57	No space is available on the medium. For disk files, an area could not be allocated.
INSUFFICIENTSPACE	71	Not valid for IOERRORTYPE. Reported as BEGINNINGORENDOFTAPE by IOERRORTYPE. A no-space condition occurred and the NORESOURCEWAIT attribute is TRUE.
OPENDATA	72	Indicates that the message read was user data on an incoming OPEN request.
OPENRESPONSEDATA	73	Indicates that the message read was user data on an incoming OPEN response.
CLOSEREQUESTDATA	74	Indicates that the data read was user data on an incoming ORDERLY CLOSE request.
CLOSEABORTDATA	75	Indicates that the data read was user data on an incoming ABORT indication.
CLOSERESPONSEDATA	76	Indicates that the message read was user data on an incoming CLOSE response.
ENCODEDDATAERROR	77	Indicates that the READ or WRITE operation failed because encoded data was truncated.
MESSAGETRUNCATEDWARNING	79	Indicates that the message read or written was truncated.
IMPLICITOPENFAILED	91	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.
UNSPECIFIEDEXCEPTION	92	Not valid for IOERRORTYPE. Reported as CANCELED by IOERRORTYPE. An exception occurred, but there is no information about its nature. This can occur in 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).

continued

Table F-1. List of I/O Result Enumerated Values (cont.)

Mnemonic Value	Integer Value	Meaning
SETMODEFAILED	93	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.
DISKALLOCATIONFAILED	94	An unspecified failure occurred during an attempt to allocate disk space for a file.
RECORDNOTFOUND	95	For KEYEDIO and KEYEDIOII files, no record with a matching key was found.
PRIMARYKEYSNOTEQUAL PRIMARYKEYNOTEQUAL	96	For KEYEDIO and KEYEDIOII files, an attempt was made to change the key of a record when that record was rewritten.
DUPLICATEKEYS DUPLICATEKEY	97	For KEYEDIO and KEYEDIOII files, an attempt was made to write a record with a duplicate key when duplicate keys are not allowed.
KEYISINVALID	98	For KEYEDIO and KEYEDIOII files, the key information supplied is not valid.
LASTIOMUSTBEREAD	99	For KEYEDIO and KEYEDIOII files, an attempt was made to use a sequential operation to rewrite or delete the current record without first reading it.
DIFFERENTLENGTHRECORDS DIFFERENTLENGTHRECORD	100	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.
PRIMARYKEYSOUTOFORDER PRIMARYKEYOUTOFORDER	101	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.
KIOIIWRITEERROR	106	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.
NOTCLOSERETAINED	107	The file must be in a closed retained state—CLOSE (<FILE>, *)—to perform the requested operation.

continued

Table F-1. List of I/O Result Enumerated Values (cont.)

Mnemonic Value	Integer Value	Meaning
VALIDONLYFORDISK	108	ERASE is valid only for disk type files.
DUPLICATEDFILENOTALLOWED	109	ERASE is not valid for duplicated files.
NODISKHEADER	110	There is no disk file for the passed file to erase.
NOTONLYUSEROFFILE	111	ERASE requires that the file not be in use by any user but the caller.
INTERCHANGEFILENOTALLOWED	112	ERASE is not valid for interchange files.
DATAMAYBECORRUPTED	114	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.
RESPSEEKFAIL	115	The foreign system returned an error when a SEEK operation was performed through FTAM.
RESPREADFAIL	116	The foreign system returned an error when a READ operation was performed through FTAM.
RESPWRITEFAIL	117	The foreign system returned an error when a WRITE operation was performed through FTAM.
WRITEONINPUT	118	A WRITE operation to a file failed because the FILEUSE attribute value of the file was IN.
READONOUTPUT	119	A WRITE operation to a file failed because the FILEUSE attribute value of the file was OUT.
HYDATALOST	120	Valid only for IOERRORTYPE. Refer to the HY file IOERRORTYPE values for direct I/O HY files in the *I/O Subsystem Programming Guide.*
HYMESSAGEOVERFLOW	121	Valid only for IOERRORTYPE. Refer to the HY file IOERRORTYPE values for direct I/O HY files in the *I/O Subsystem Programming Guide.*

continued

Table F-1. **List of I/O Result Enumerated Values** (cont.)

Mnemonic Value	Integer Value	Meaning
NOTATEND	122	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.
NONAPPENDWORMWRITE	123	The WRITE operation did not append the record to the current logical end of file.
NONBOUNDARYWORMWRITE	124	A WORM file with a SYNCHRONIZE attribute value of OUT issued a WRITE operation to a file after the previous WRITE operation did not leave the end of file at a physical sector boundary. When a WRITE operation does not leave the end of file at a physical sector boundary, no further WRITE operations can be made to the file.
INSUFFICIENTBUFFERS	132	For port files, a WRITE operation with the DONTWAIT option requested more data be sent than the provider could send at the moment.
DSEDLOGICALFILEINUSE	1023	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 Table F-1 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 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.

Table F-2 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).

Table F–2. Primary and Secondary I/O Result Meanings

Bit	Primary I/O Result Meaning	Secondary I/O Result Meaning
[16:1]	SECURITYERROR (50)	
[15:1]	TIMEOUT (15)	
[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)
		RECORDNOTFOUND (95)
[8:1]	ENDOFTAPE (45)	NODATAFORREAD (53)
		BROADCASTWRITEFAILED (54)
		NOBUFFERFORWRITE (55)
		INSUFFICIENTSPACE (71)
[7:1]	PARITYERROR (28)	DSSABORT (51)
[6:1]	WRITELOCKOUT (6)	PRIMARYKEYSNOTEQUAL (96)

continued

Table F–2. Primary and Secondary I/O Result Meanings (cont.)

Bit	Primary I/O Result Meaning	Secondary I/O Result Meaning
		PRIMARYKEYSOUTOFORDER (101)
[5:1]	UNDEFINEDERROR (44)	DELETEDORDUPLICATERECORD (56)
		DUPLICATEKEY (97)
[4:1]	DESCRIPTORERROR (direct I/O) (4)	
	DATAERROR (nondirect I/O) (4)	
[3:1]	NOTREADY (27)	INVALIDSUBFILE (52)
[2:1]	CANCELED (8)	

Glossary

In this glossary, definitions taken from outside sources are preceded by an abbreviation enclosed in parentheses. Definitions from *Vocabulary for Data Processing, Telecommunications, and Office Systems* are preceded by VDP.

A

A Series Resolver

See Resolver.

ALGOL

Algorithmic language. A structured, high-level programming language that provides the basis for the stack architecture of the Unisys A Series systems. ALGOL was the first block-structured language developed in the 1960s and served as a basis for such languages as Pascal and Ada. It is still used extensively on A Series systems, primarily for systems programming.

ANSI

American National Standards Institute. A nongovernmental, nonprofit organization that is the central body for coordinating voluntary standards in the United States. ANSI also serves as the United States member of the International Standards Organization (ISO).

area

The amount of contiguous disk space that is allocated at one time to a disk file as it is being created or expanded.

ASCII

American Standard Code for Information Interchange. A standard 7-bit or 8-bit information code used to represent alphanumeric characters, control characters, and graphic characters on a computer system.

attribute

(1) A characteristic or property. (2) The information that describes a characteristic of an entity.

B

backup file

(1) A printer or punch file assigned to a backup peripheral for subsequent output. The default backup peripheral is a disk. (2) A copy of a file that is stored offline so that it can be copied back onto the system if the original file becomes corrupted or inaccessible.

BNA

The network architecture used on A Series, B 1000, and V Series systems as well as CP9500 and CP 2000 communications processors to connect multiple, independent, compatible computer systems into a network for distributed processing and resource sharing.

buffer

An area in which data is stored temporarily.

C

CANDE

See Command and Edit.

CD-ROM

A high-density read-only storage medium. The data is read by a laser beam.

COBOL

Common Business-Oriented Language. A widely used, procedure-oriented language intended for use in solving problems in business data processing. The main characteristics of COBOL are the easy readability of programs and a considerable degree of machine independence. COBOL is the most widely used procedure-oriented language.

COBOL74

A version of the COBOL language that is compatible with the ANSI X3.23-1974 standard.

COBOL85

The latest version of the COBOL language. This version is compatible with the American National Standard, X3.23-1985.

Command and Edit (CANDE)

A time-sharing message control system (MCS) that enables a user to create and edit files, and develop, test, and execute programs, interactively.

correspondent application

An application that runs in conjunction with another application, and information is passed between the two applications.

correspondent dialogue

A connection used between two correspondent applications.

correspondent endpoint

The receiving or sending point for a correspondent dialogue.

crunch

To return the unused portion of the last row of disk space of a disk file (beyond the end-of-file indicator) to the system. Only disk files can be crunched.

D

Data Management System II (DMSII)

A specialized system software package used to describe a database and maintain the relationships among the data elements in the database.

dialogue

A single instance of a two-way communication between two processes. A subfile supports one dialogue.

direct I/O file

A file that allows a program to control the physical I/O more closely than when the logical I/O subsystem is used.

disk file

A file stored on a disk or disk pack.

disk file header

A data structure that contains information about a disk file, such as the physical location of the file on the disk and various file attributes. A disk file header is also referred to as a header.

distributed systems service (DSS)

One of a collection of services provided on Unisys hosts to support communications across multihost networks. DSSs can be services such as file handling, station transfer, and mail transfer.

DMSII

See Data Management System II.

domain name

An identifier for directing queries relating to addressing in an Internet Protocol (IP) network. A domain name uniquely identifies a host system or host-dependent entity in terms of the logical tree structure that defines the domain name space. The name is written in the form of a series of nodes, progressing from local to general. For example, HOSTA.SITENY.COINC.COM. is a hypothetical domain name that uniquely identifies an A Series host HOSTA within the Internet Protocol (IP) network. To be a complete domain name, the domain name must include the terminal period. *See also* domain name space.

domain name space

A tree-structured logical space consisting of all the zones in an Internet Protocol (IP) network. *See also* zone.

domain name system

A system consisting of the domain name space in an Internet Protocol (IP) network and all the name servers and resolvers that exist in that domain name space. Gateway hosts providing access to other networks, such as BNA or OSI networks, can be part of the domain name system, but the networks to which they provide access cannot be.

DSS

See distributed systems service.

E

EBCDIC

Extended Binary Coded Decimal Interchange Code. An 8-bit code representing 256 graphic and control characters that are the native character set of most mainframe systems.

EFS

See extended file system.

EMS

See Entry and Medium Systems.

end of file (EOF)

A code at the end of a data file that signals that the last record in the file has been processed.

Entry and Medium Systems (EMS)

A designation referring to the Micro A and A 1 through A 10 systems.

EOF

See end of file.

extended file system (EFS)

A file system controlled by KEYEDIOII. This file system supports sequential, direct, and indexed organization; alternate index files; record-level locking; and delete-capability for RPG files.

F

family name

The name, consisting of up to 17 alphanumeric characters, assigned by an installation to identify a family of disks.

file name

A unique identifier for a file, consisting of name constants separated by slashes. Each name constant consists of letters, digits, and selected special characters. A file name can be optionally preceded by an asterisk (*) or usercode, and optionally followed by ON and a family name.

file title

The complete identifier for a file that consists of the file name, the word ON, and the family name.

File Transfer, Access, and Management (FTAM)

The standard developed by the International Standards Organization (ISO) for file exchange and management across an Open Systems Interconnection (OSI) network. FTAM systems can access file attributes (for example, password information) and the contents of files (including individual records, as well as entire files). *See also* OSI File Transfer, Access, and Management.

File Transfer Protocol (FTP)

A Unisys distributed systems service (DSS) that enables A Series systems to transfer files by means of the FTP protocol across an IP network.

FORTRAN

Formula Translation. A high-level, structured programming language intended primarily for scientific use.

FORTRAN77

A version of the FORTRAN language that is compatible with the ANSI X3.9-1978 standard.

FTAM

See File Transfer, Access, and Management.

FTP

See File Transfer Protocol.

G

gateway

A conceptual or logical network station that serves to interconnect two otherwise incompatible networks, network nodes, subnetworks, or devices.

GCR

See group-coded recording.

group-coded recording (GCR)

A scheme for recording data on a magnetic tape (MT).

H

halt/load

A system-initialization procedure that temporarily halts the system and loads the master control program (MCP) from disk or pack to main memory.

HC unit

See host control (HC) unit.

host control (HC) unit

A specialized data link processor (DLP) that enables host systems to communicate through an intersystem control (ISC) hub on a channel-to-channel communications interface between A Series and CP9500 systems.

HYPERchannel network

A network that allows communication between computer systems from different manufacturers by using hardware provided by the Network Systems Corporation.

I

I/O

Input/output. An operation in which the system reads data from or writes data to a file on a peripheral device such as a disk drive.

I/O control block (IOCB)

A data structure used for communication between the host system and the I/O subsystem.

I/O control word (IOCW)

The area in the I/O control block (IOCB) where information about the I/O operation to be performed resides.

I/O processor (IOP)

A specialized processor for moving data between system memory and the I/O subsystem.

ICP

See integrated communications processor.

indexed sequential-access method (ISAM)

A method that provides efficient, flexible random access to records identified by keys stored in an index.

InfoGuard

The Unisys security-enhancement software for A Series systems. InfoGuard provides such features as password management, selective logging and auditing, tape volume security, and simplified system-security configuration.

integrated communications processor (ICP)

In BNA Version 2, a data link processor (DLP) that handles a significant portion of communications overhead. The ICP also provides a connection to the communications processor local area network (CPLAN). At least one ICP for each A Series host is needed to connect the host to a BNA Version 2 network.

International Standards Organization (ISO)

A division of the United Nations under which the Consultative Committee on International Telegraphy and Telephony (CCITT) operates. The ISO was established in 1947 to promote the development of standards to facilitate international trade, and to develop mutual cooperation in areas of intellectual, scientific, technological, and economic activity.

Internet Protocol (IP)

The fundamental protocol in the Transmission Control Protocol/Internet Protocol (TCP/IP) data communications protocols. IP interconnects networks and routes information packets between networks thus connected. IP provides the basis for the Transfer Control Protocol (TCP) session management service.

intersystem control (ISC)

A direct hardware connection that enables data transfer between independent systems. The components that make up an ISC are a hub and its attached host control (HC) units. The HC unit type used to connect an I/O channel to a hub depends on the type of machine, specifically the I/O subsystem protocol.

IOCB

See I/O control block.

IOCW

See I/O control word.

IOP

See I/O processor.

IP

See Internet Protocol.

IP address

A numerical designator that uniquely identifies a host within a specific local area network (LAN) or wide area network (WAN) that forms a part of an Internet Protocol (IP) network. For example, 192.59.12.1 might be a hypothetical IP address for an A Series host HOSTA.

ISAM

See indexed sequential-access method.

ISC

See intersystem control.

ISO

See International Standards Organization.

J

job

An independent process. The job of a particular task is the independent process that is the eldest ancestor of that task.

K

KEYEDIO

A Unisys product that provides random access to the records of files by indexing the records based on keys.

KEYEDIOII

An extended version of KEYEDIO. *See also* KEYEDIO.

L

label

The first 28 sectors on a disk, on which information about the disk is stored. This information includes the family name and serial number, the master available table (MAT), the family index number, information about the family base pack, and a pointer to the system directory if the disk contains a directory.

LAN

See local area network.

Large Systems (LS)

Refers to systems that interface with a host data unit (HDU) or a resource management module (RMM).

local area network (LAN)

A network that enables high-speed communications among various devices within a relatively small area.

logical file

A file variable declared in a program, which represents the file and its structure to the program. A logical file has no properties of its own until it is described by file attributes or associated with a physical file.

logical result descriptor (LRD)

A word of information that describes the result of an I/O operation.

LRD

See logical result descriptor.

LS

See Large Systems.

M

MARC

See Menu-Assisted Resource Control.

master available table (MAT)

A table stored on each disk that lists the valid sectors on the disk that were successfully processed by the initialization, verification, and relocation (IVR) procedure. Pointers to defective sectors are deleted from the MAT so that these sectors cannot be accessed. Normally, the MAT shows the entire disk as being available, minus any defective sectors.

master control program (MCP)

The MCP controls the operational environment of the system by performing job controls the operational environment of the system by performing job management, dynamic subroutine linkage, and logging of errors and system utilization.

MAT

See master available table.

MCP

See master control program.

MCS

See message control system.

Menu-Assisted Resource Control (MARC)

A menu-driven interface to A Series systems that also enables direct entry of commands.

message control system (MCS)

A program that controls the flow of messages between terminals, application programs, and the operating system. MCS functions can include message routing, access control, audit and recovery, system management, and message formatting.

message proper

A control message used with a HYPERchannel network.

N

name server

A server process on a host in an Internet Protocol (IP) network. A name server provides message-routing information, including IP addresses, to other processes that request such information. *See also* IP address, resolver.

NAP

See Network Application Platform.

National Institute for Standards and Technology (NIST)

The division of the U.S. Department of Commerce that ensures standardization within government agencies. NIST develops implementors agreements for Open Systems Interconnection (OSI) protocols. NIST was formerly known as the National Bureau of Standards (NBS).

Network Application Platform (NAP)

A combination of software and hardware that facilitates the development and deployment of network applications on Unisys A Series computers.

NIST

See National Institute for Standards and Technology.

nonreturn to zero (NRZ)

A scheme for recording data on a magnetic tape (MT). (VDP) The magnetic recording of binary digits such that the patterns of magnetization used to represent zeros and ones occupy the whole storage cell. No part of the cell is magnetized to the reference condition.

NRZ

See nonreturn to zero.

O

ODT

See operator display terminal.

Open Systems Interconnection (OSI)

A set of data communications standards defined by the International Organization for Standardization (ISO) that provide for communications between different types of computer systems. The application services defined under OSI include File Transfer, Access, and Management (FTAM) and the Message Handling System (MHS).

operator display terminal (ODT)

A terminal or other device that is connected to the system in such a way that it can communicate directly with the operating system. The ODT allows operations personnel to accomplish system operations functions through either of two operating modes: system command mode or data comm mode.

OSI

See Open Systems Interconnection.

OSI File Transfer, Access, and Management (OSI FTAM)

An A Series distributed systems service (DSS) that supports Open Systems Interconnection (OSI) standards for file management functions such as reading, writing, and copying files on remote hosts.

OSI FTAM

See OSI File Transfer, Access, and Management.

P

page description language (PDL)

A language for describing the composition of a printed page.

PC

See personal computer.

PDL

See page description language.

PE

See phase-encoded.

peripheral

A device used for input, output, or file storage. Examples are magnetic tape drives, disk drives, printers, or operator display terminals (ODTs).

personal computer (PC)

A microcomputer that is compatible with the Unisys Personal Workstation ^{2}, the IBM PC®, the PC/XT™, the Personal Computer AT®, or the IBM Personal System/2® line of computers.

phase-encoded (PE)

A scheme for recording data on a magnetic tape (MT) in which the phase of the carrier shifts when the data switches from 0 to 1 or from 1 to 0.

physical file

A file as it is stored on a particular recording medium such as a disk or a tape.

port file

A type of file for which file operations occur between a local user process and another process on the same host or on a remote host that is reachable through a network. A port file is made up of one or more subfiles, each of which supports exactly one dialogue.

R

relative station number (RSN)

A number that indicates the relative position of a station in a list of stations.

remote device

An I/O unit or other piece of equipment that is physically removed from the computer but connected by a communications line.

resolver

A client process on a host in an Internet Protocol (IP) network. A resolver can send standard queries to one or more name servers and receive information in return. The resolver enables users and appropriate software programs to initiate queries and receive the responses. *See also* name server, Resolver.

Resolver

The Unisys A Series implementation of a resolver. The Resolver provides an interface that gives A Series users and user programs access to the domain name system. *See also* resolver.

RPG

Report Program Generator. A high-level, commercially oriented programming language used most frequently to produce reports based on information derived from data files.

RSN

See relative station number.

IBM PC is a registered trademark of International Business Machines Corporation.
PC/XT is a trademark of International Business Machines Corporation.
Personal Computer AT is a registered trademark of International Business Machines Corporation.
IBM Personal System/2 is a registered trademark of International Business Machines Corporation.

S

scratch tape

A labeled magnetic tape (MT) whose label indicates that there are no files on the tape. Old data might remain on the tape, but this old data cannot be read unless the tape is read as an unlabeled tape. The old data present on a scratch tape is written over when new data is written to the tape.

system command

Any of a set of commands used to communicate with the operating system. System commands can be entered at an operator display terminal (ODT), in a Menu-Assisted Resource Control (MARC) session, or by way of the DCKEYIN function in a privileged Data Communications ALGOL (DCALGOL) program.

T

task

A single, complete unit of work performed by the system, such as compiling or executing a program, or copying a file from one disk to another. Tasks are initiated by a job, by another task, or directly by a user.

TCP

See Transmission Control Protocol.

TCP/IP

See Transmission Control Protocol/Internet Protocol.

Transmission Control Protocol (TCP)

A data communications protocol that establishes sessions between hosts in a Transmission Control Protocol/Internet Protocol (TCP/IP) network and ensures reliable interhost communications by implementing error recovery procedures.

Transmission Control Protocol/Internet Protocol (TCP/IP)

(1) A family of protocols that were originally developed for use in a Department of Defense network, and which have been widely adopted as standard protocols for multivendor networks. The applications protocols typically supported by TCP/IP are File Transfer Protocol (FTP), Simple Mail Transfer Protocol (SMTP), and Telnet.
(2) Unisys A Series product that provides telecommunications capabilities between A Series, 1100 Series, 5000 Series, and other computing systems. The A Series TCP/IP product is designed to interoperate with other TCP/IP systems in a multivendor network using industry and military standard protocols. The A Series TCP/IP product provides file transfer, station transfer, and (with the Mail System) mail transfer.

U

UDDFAs

See user-defined disk file attributes.

user-defined disk file attributes (UDDFAs)

A class of file attributes in which both the attribute name and its value or values are defined by the user.

usercode

An identification code used to establish user identity and control security, and to provide for segregation of files. Usercodes can be applied to every task, job, session, and file on the system. A valid usercode is identified by an entry in the USERDATAFILE.

V

VC

See voice channel.

voice channel

A logical or physical, or both, communication path allowing audio signals to be transmitted in both directions simultaneously.

W

WAN

See wide area network.

WFL

See Work Flow Language.

Work Flow Language (WFL)

A Unisys language used for constructing jobs that compile or run programs on A Series systems. WFL includes variables, expressions, and flow-of-control statements that offer the programmer a wide range of capabilities with regard to task control.

wide area network (WAN)

A network that enables communications among various devices spread over a large area (for example, devices located in different cities).

WORM

Write-once, read-many. *See also* WORM device, WORM media.

WORM device

A random-access optical disk drive that can write to the WORM media only once, but can access the files on the WORM media many times.

WORM media

A high-density storage medium that can be written to only once and can be read from many times.

Z

zone

A logical space in an Internet Protocol (IP) network. The zone is controlled or "seen" by one master name server.

Bibliography

A Series ALGOL Programming Reference Manual, Volume 1: Basic Implementation (8600 0098). Unisys Corporation.

A Series Disk Subsystem Administration and Operations Guide (8600 0668). Unisys Corporation.

A Series Distributed Systems Service (DSS) Operations Guide (8600 0122). Unisys Corporation.

A Series DMALGOL Programming Reference Manual (8600 0874). Unisys Corporation.

A Series GETSTATUS/SETSTATUS Programming Reference Manual (8600 0346). Unisys Corporation.

A Series Host LAN Connection (HLCN) Installation and Configuration Guide (8600 0932). Unisys Corporation.

A Series I/O Subsystem Programming Guide (8600 0056). Unisys Corporation.

A Series KEYEDIOII Programming Reference Manual (8600 0684). Unisys Corporation.

A Series Mark 4.0 Software Release Capabilities Overview (8600 1617). Unisys Corporation.

A Series MultiLingual System (MLS) Administration, Operations, and Programming Guide (8600 0288). Unisys Corporation.

A Series Print System (PrintS/ReprintS) Administration, Operations, and Programming Guide (8600 1039). Unisys Corporation.

A Series Security Administration Guide (8600 0973). Unisys Corporation.

A Series Security Features Operations and Programming Guide (8600 0528). Unisys Corporation.

A Series System Administration Guide (8600 0437). Unisys Corporation.

A Series System Commands Operations Reference Manual (8600 0395). Unisys Corporation.

A Series System Operations Guide (8600 0387). Unisys Corporation.

A Series System Software Utilities Operations Reference Manual (8600 0460). Unisys Corporation.

A Series Task Attributes Programming Reference Manual (8600 0502). Unisys Corporation.

A Series Task Management Programming Guide (8600 0494). Unisys Corporation.

A Series Work Flow Language (WFL) Programming Reference Manual (8600 1047). Unisys Corporation.

Vocabulary for Data Processing, Telecommunications, and Office Systems (GC10-1699-6). Poughkeepsie, New York: International Business Machines Corporation, 1981.

Index

Help Us To Help You

Publication Title

Document Number　　　　　　　　　　Date

Unisys Corporation is interested in your comments and suggestions regarding this manual. We will use them to improve the quality of your Product Information. Please check type of suggestion:

○ Addition　　　○ Deletion　　　○ Revision

Recommended Change (Please identify affected pages)

Name　　　　　　　　　　Telephone Number

Title　　　　　　　　　　Company

Address (Street, City, State, Zip)

Help Us To Help You

Publication Title

Document Number　　　　　　　　　　Date

Unisys Corporation is interested in your comments and suggestions regarding this manual. We will use them to improve the quality of your Product Information. Please check type of suggestion:

○ Addition　　　○ Deletion　　　○ Revision

Recommended Change (Please identify affected pages)

Name　　　　　　　　　　Telephone Number

Title　　　　　　　　　　Company

Address (Street, City, State, Zip)

Help Us To Help You

Publication Title

Document Number　　　　　　　　　　Date

Unisys Corporation is interested in your comments and suggestions regarding this manual. We will use them to improve the quality of your Product Information. Please check type of suggestion:

○ Addition　　　○ Deletion　　　○ Revision

Recommended Change (Please identify affected pages)

Name　　　　　　　　　　Telephone Number

Title　　　　　　　　　　Company

Address (Street, City, State, Zip)

No Postage
necessary
if mailed in the
United States

BUSINESS REPLY MAIL

First Class Permit No. 817 Detroit, MI 48232

Postage Will Be Paid By Addressee

Unisys Corporation
ATTN: Technical Publications
25725 Jeronimo Road
Mission Viejo, CA 92691-2792 USA

No Postage
necessary
if mailed in the
United States

BUSINESS REPLY MAIL

First Class Permit No. 817 Detroit, MI 48232

Postage Will Be Paid By Addressee

Unisys Corporation
ATTN: Technical Publications
25725 Jeronimo Road
Mission Viejo, CA 92691-2792 USA

No Postage
necessary
if mailed in the
United States

BUSINESS REPLY MAIL

First Class Permit No. 817 Detroit, MI 48232

Postage Will Be Paid By Addressee

Unisys Corporation
ATTN: Technical Publications
25725 Jeronimo Road
Mission Viejo, CA 92691-2792 USA

86000064-100